



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERIA

ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

**SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL
ÁREA DE SISTEMAS EN LA COMPAÑÍA NAVIERA NATALIA
S.A.C.**

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE
INGENIERIA DE SISTEMAS**

AUTOR:

AIQUIPA TELLO ALEX ALDO

ASESOR:

MG. HUAROTE ZEGARRA RAÚL EDUARDO

LÍNEA DE INVESTIGACIÓN:

SISTEMAS DE INFORMACIÓN Y COMUNICACIONES

LIMA – PERÚ

2018

El Jurado encargado de evaluar la tesis presentada por don (a):

AIQUIPA TELLO, ALEX ALDO

cuyo título es:

**SISTEMA WEB PARA EL CONTROL DE SERVICIOS DEL AREA DE SISTEMAS EN LA
COMPAÑÍA NAVIERA NATALIA S.A.C.**

Reunido en la fecha, escuchó la sustentación y la resolución de preguntas por el estudiante, otorgándole el calificativo de: **13** (números) **TRECE** (letras).

Lima, Jueves 20 de Diciembre del 2018



.....
PRESIDENTE
Dra. ROMERO VALENCIA MONICA
PATRICIA



.....
SECRETARIO
Mgtr. SAENZ APARI ABRAHAM RAFAEL



.....
VOCAL
Mgtr. HUAROTE ZEGARRA RAUL
EDUARDO

Elaboró	Dirección de Investigación	Revisó	Representante de la Dirección / Vicerrectorado de Investigación y Calidad	Aprobó	Rectorado
---------	----------------------------	--------	---	--------	-----------

PÁGINAS PRELIMINARES

Dedicatoria

A mis padres por haber inculcado en mí el valor del buen trabajo y la persistencia ante las adversidades.

A Dios por su gran amor hacia nosotros, que me ha permitido conocer buenas personas y aprender de ellos.

A mis hermanos por su sincero y valioso apoyo.

Agradecimiento

Al padre celestial por brindarme, bienestar, salud, vida y paciencia, al personal de la empresa, al que hasta la fecha laboro, por su comprensión, paciencia y apoyo, a mi querida universidad, docentes y compañeros de estudio, quienes, con sus conocimientos y experiencias compartidas aportaron buenos momentos durante mi carrera, enriqueciendo mi aprendizaje con alegría y motivación

.

Declaratoria de autenticidad

Yo, Alex Aldo Aiquipa Tello, estudiante del Programa de Ingeniería de Sistemas de la Escuela de Pregrado de la Universidad César Vallejo, identificado con DNI 10627303, con la tesis titulada "Sistema web para el control de los servicios del área de sistemas en la Compañía Naviera Natalia S.A.C. "

Declaro bajo juramento que:

- 1) La tesis es de mi autoría
- 2) He respetado las normas internacionales de citas y referencias para las fuentes consultadas. Por tanto, la tesis no ha sido plagiada ni total ni parcialmente.
- 3) La tesis no ha sido auto plagiada; es decir, no ha sido publicada ni presentada anteriormente para obtener algún grado académico previo o título profesional.
- 4) Los datos presentados en los resultados son reales, no han sido falseados, ni duplicados, ni copiados y por tanto los resultados que se presenten en la tesis se constituirán en aportes a la realidad investigada.

De identificarse la falta de fraude (datos falsos), plagio (información sin citar a autores), auto plagio (presentar como nuevo algún trabajo de investigación propio que ya ha sido publicado), piratería (uso ilegal de información ajena) o falsificación (representar falsamente las ideas de otros), asumo las consecuencias y sanciones que de mi acción se deriven, sometiéndome a la normatividad vigente de la Universidad César Vallejo.

Los olivos, 04 de diciembre del 2018.



.....
Alex Aldo Aiquipa Tello

DNI: 10627303

Presentación

Señores miembros del Jurado:

Dando cumplimiento a las normas establecidas en el Reglamento de Grados y Títulos sección de Pregrado de la Universidad César Vallejo para optar el grado de Ingeniera de Sistemas, presento el trabajo de investigación pre-experimental denominado: "Sistema web para el control de servicios del área de sistemas en la Compañía Naviera Natalia S.A.C.

La investigación, tiene como propósito fundamental: determinar cómo influye un sistema web en la Compañía Naviera Natalia S.A.C., 2018.

La presente investigación está dividida en siete capítulos: En el primer capítulo se expone el planteamiento del problema: incluye formulación del problema, los objetivos, la hipótesis, la justificación, los antecedentes y la fundamentación científica. En el segundo capítulo, que contiene el marco metodológico sobre la investigación en la que se desarrolla el trabajo de campo de la variable de estudio, diseño, población y muestra, las técnicas e instrumentos de recolección de datos y los métodos de análisis. En el tercer capítulo corresponde a la interpretación de los resultados. En el cuarto capítulo trata de la discusión del trabajo de estudio. En el quinto capítulo se construye las conclusiones, en el sexto capítulo las recomendaciones y finalmente en el séptimo capítulo están las referencias bibliográficas.

Señores miembros del jurado espero que esta investigación sea evaluada y merezca su aprobación.

Índice

I.	INTRODUCCIÓN	15
1.1.	REALIDAD PROBLEMÁTICA	15
1.2.	TRABAJOS PREVIOS	20
1.3.	TEORÍAS RELACIONADAS A TEMA	23
1.4.	FORMULACIÓN DE PROBLEMA	52
1.5.	JUSTIFICACIÓN DE ESTUDIO	52
1.6.	HIPÓTESIS	54
1.7.	OBJETIVOS	54
II.	MÉTODO	57
2.1.	DISEÑO DE INVESTIGACIÓN	57
2.2.	VARIABLES, OPERACIONALIZACIÓN	57
2.3.	POBLACIÓN Y MUESTRA DE ESTUDIO	59
2.4.	TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS, VALIDEZ Y CONFIABILIDAD	61
2.5.	MÉTODOS DE ANÁLISIS DE DATOS	61
2.6.	ASPECTOS ÉTICOS	65
III.	RESULTADOS	67
IV.	DISCUSIÓN	79
V.	CONCLUSIÓN	80
VI.	RECOMENDACIONES	80
VII.	PROPUESTA	80
VIII.	REFERENCIA	83

Índice de figuras

Figura 1: Índice de Cobertura de Mantenimiento (Abril del 2018)	18
Figura 2: Grado de Cumplimiento (Abril del 2018)	19
Figura 3: Área de estudio	19
Figura 4: Memex	24
Figura 5: Interfaz ARPANET para el PDP-10 de Xerox	25
Figura 6: Comunicación Cliente / Servidor Web	26
Figura 7: Funcionamiento de páginas ASP	27
Figura 8: Flujo típico MVC ASP.NET	28
Figura 9: Diagrama E-R	31
Figura 10: Modelo de datos Orientado a Objetos	32
Figura 11: Esquema XML	33
Figura 12: Procedimiento de mantenimiento preventivo	39
Figura 13: Diagrama del Proceso de mantenimiento correctivo de la Compañía Naviera Natalia S.A.C.	41
Figura 14: Marco Scrum	46
Figura 15: Fases y disciplinas de la metodología RUP	49
Figura 16: Fases de la metodología XP	50
Figura 17: Gráfica de la distribución normal	65
Figura 18: Índice de cobertura de mantenimiento antes y después de la implementación del sistema.	68
Figura 19: Grado de cumplimiento antes y después de la implementación del sistema web	69
Figura 20: Test de normalidad del Índice de cobertura de mantenimiento antes de la implementación	71
Figura 21: Test de normalidad del índice de cobertura de mantenimiento después de la implementación	72
Figura 22: Prueba de normalidad del grado de cumplimiento antes de la implementación del Sistema Web	73
Figura 23: Prueba de normalidad del grado de cumplimiento después de la implementación del Sistema Web	74

Figura 24: Índice de Cobertura- Comparativa general	75
Figura 25: Prueba T-Student – Índice de Cobertura	76
Figura 26: Grado de cumplimiento - Comparativa general	78
Figura 27: Prueba T-Student – Grado de cumplimiento	79

ÍNDICE DE TABLAS

Tabla 1: Ejemplo modelo relacional (Tabla Empresa)	30
Tabla 2: Importancia del mantenimiento	38
Tabla 3: Evaluación de Metodologías	51
Tabla 4: Validación de instrumentos	59
Tabla 5: Medidas descriptivas del índice de cobertura de mantenimientos realizados antes y después de la implementación del sistema web	67
Tabla 6: Medidas descriptivas del Grado de cumplimiento antes y después de la implementación del sistema web	68
Tabla 7: Test de normalidad del índice de cobertura de mantenimiento antes y después de implementado el sistema web	70
Tabla 8: Prueba de normalidad del grado de cumplimiento	72
Tabla 9: Prueba T. Student de Índice de cobertura de mantenimiento antes y después de implementado el sistema web	76
Tabla 10: Prueba de T. Student del grado de cumplimiento antes y después de la implementación el sistema web	78

ÍNDICE DE ANEXOS

Anexo 1: Instrumento Pre Test Índice De Cobertura De Mantenimiento	88
Anexo 2: Instrumento Pre Test Grado De Cumplimiento	89
Anexo 3: Instrumento Post Test Índice De Cobertura De Mantenimiento	90
Anexo 4: Instrumento Post Test Grado De Cumplimiento	91
Anexo 5: Validación de Metodología de desarrollo de Software Experto 1	92
Anexo 6: Validación de Metodología de desarrollo de Software Experto 2	93
Anexo 7: Validación de Metodología de desarrollo de Software Experto 3	94
Anexo 8: Validación de experto 1 para indicador de cumplimiento	95
Anexo 9: Validación de experto 2 para indicador de cumplimiento	96
Anexo 10: Validación de experto 3 para indicador de cumplimiento	97
Anexo 11: Evaluación de Experto 1 para Indicador Índice de Cobertura	98
Anexo 12: Evaluación de Experto 2 para Indicador Índice de Cobertura	99
Anexo 13: Evaluación de Experto 3 para Indicador Índice de Cobertura	100
Anexo 14: Carta de aceptación del proyecto	101
Anexo 15: Entrevista al Jefe Corporativo de la Compañía Naviera Natalia S.A.C.	102
Anexo 16: Diagrama de procesos de nivel 0 de la Compañía Naviera Natalia S.A.C.	103
Anexo 17: Matriz de consistencia	104

RESUMEN

La investigación se basa en la influencia de un sistema web para el control de servicios del área de Sistemas en la Compañía Naviera Natalia SAC., se presentó deficiencias en cuanto a los cumplimientos de los mantenimientos preventivos y atenciones realizadas por el departamento de sistemas hacia sus clientes internos, el bajo índice de cobertura de máquinas con mantenimiento preventivo y bajo grado de cumplimiento de las atenciones prestadas conllevó la necesidad de crear un sistema web que permita monitorizar el cumplimiento de estos servicios, es por ello que el objetivo del proyecto de investigación fue desarrollar e implementar un sistema web para el control de servicios del área de sistemas, que permita contrarrestar los principales problemas álgidos del proceso.

La investigación está dividida en dos etapas investigativas. En la primera etapa, se realizó el levantamiento de información en el área de sistemas, detectando irregularidades en el servicio de soporte técnico frente a las incidencias reportadas. Se midieron los indicadores de índice de cobertura de máquinas con mantenimiento preventivo y grado de cumplimiento de las atenciones solicitadas, el tipo de investigación fue de tipo Aplicada – Experimental, con un diseño pre -experimental debido a que la investigación se basa en una prueba pre-test y una prueba post-test. Para el proceso de recolección de datos, se usó la técnica de Fichaje con el instrumento de la ficha de registro; es decir, la muestra se registró en fichas de registro. El objetivo principal del estudio es determinar la influencia del sistema web para el control de servicios del área de sistemas, específicamente evaluando los indicadores de cobertura de mantenimiento y grado de cumplimiento. Con la implementación del sistema web, se observó un aumento del índice de cobertura y el grado de cumplimiento.

ABSTRACT

The research is based on the influence of a web system for the control of services in the area of Systems in the naval company Natalia SAC. There were deficiencies in the compliance with preventive maintenance and the attention that the systems department made for its internal operations. Customers, the low rate of coverage of machines with preventive maintenance and low degree of compliance with the services provided, led to the need to create a web system to monitor the compliance of these services, so the objective of the project research was to develop and implement a web system for the control of services in the area of systems, which allows to counteract the main problems of the process.

The investigation is divided into two investigative stages. In the first stage, information was collected in the systems area, detecting irregularities in the technical support service against the incidents reported. The coverage indicators of machines with preventive maintenance and degree of compliance of the care were measured, the type of research was of the Applied - Experimental type, with a pre-experimental design because the research is based on a previous test. -test and a subsequent test. For the data collection process, the Transfer technique was used with the instrument of the registration form; That is, the sample was recorded on registration cards. The main objective of the study is to determine the influence of the web system for the control of services in the systems area, specifically evaluating the indicators of maintenance coverage and the degree of compliance. With the implementation of the web system, there was an increase in the coverage index and the degree of compliance.

INTRODUCCIÓN

I. INTRODUCCIÓN

1.1. Realidad problemática

La calidad de servicio a nivel mundial, se ha considerado como uno de los asuntos más importantes para los negocios, su objetivo principal es el cumplimiento de los requerimientos del cliente y verificar que todos los procesos concernientes a la organización estén enfocados a la satisfacción del cliente, si ellos están satisfechos con los servicios obtenidos, recomendarán y retornarán con el mismo proveedor cada vez que requiera de sus servicios, es por ello que para ser competitivos en el mundo actual hay que brindar bienes y servicios de alta calidad, ya que es una característica muy apreciada por el cliente, pues espera obtener la rapidez, confiabilidad, eficiencia y garantía del bien o servicio.

AENOR ISO 9001 (2015), menciona que “La organización debe establecer, implementar, mantener y mejorar de manera continua un sistema de gestión de calidad, incluidos los procesos necesarios y sus interacciones [...], para aumentar la satisfacción del cliente mediante el cumplimiento de los requisitos del cliente”.

A nivel nacional "solo el 1% de las empresas formales, cuenta con sistemas de gestión de calidad". El Perú es uno de los países de la región con menos número de empresas certificadas en la gestión de calidad y solo Bolivia está por debajo de nosotros, “Actualmente se tiene un total de 1329 empresas con certificación ISO 9001, de un total de empresas formales activas en el Perú que llega a 1 382 899”. Esto expresa la realidad problemática de las empresas en nuestro país, son muy pocas empresas que se preocupan en asegurar y certificar la calidad de bienes y/o servicios que brindan a sus clientes, lo que les permite generar una ventaja competitiva frente a 99% de las demás empresas en Perú.

La Compañía Naviera Natalia S.A.C. es una de estas empresas que cuenta con certificación ISO 9001-2008 y actualmente está en proceso de re-categorización a

la versión ISO 2001-2015, es una empresa dedicado al transporte marítimo es parte de un grupo empresarial denominado MigivaGroup, cuya sede administrativa está ubicada en San Isidro, en esta sede, se encuentra el área de sistemas, el cual dispone de seis procedimientos para la atención de los usuarios internos de la empresa, el problema que se viene presentando en la actualidad, es que carece de un sistema centralizado que controle los servicios del área de sistemas, que permita controlar, monitorizar, realizar seguimientos y notificar al personal de sistemas, para su oportuna intervención proactiva en el cumplimiento y mejora de los servicios correspondientes a los procedimientos que tiene el área de sistemas, puesto que actualmente estos procedimientos son requeridos y validados su atención por el usuario interno, a través del correo electrónico, la búsqueda de estos correos ya sean de casos sin atender (para su seguimiento de atención) como de conformidad de atención (para el llenado de registros de atención), resulta un poco tedioso para el personal de sistemas, demandándole más tiempo, algunos de esos requerimientos dependen de la atención de proveedores externos y su falta de control y seguimiento trae como consecuencia una falsa percepción del área.

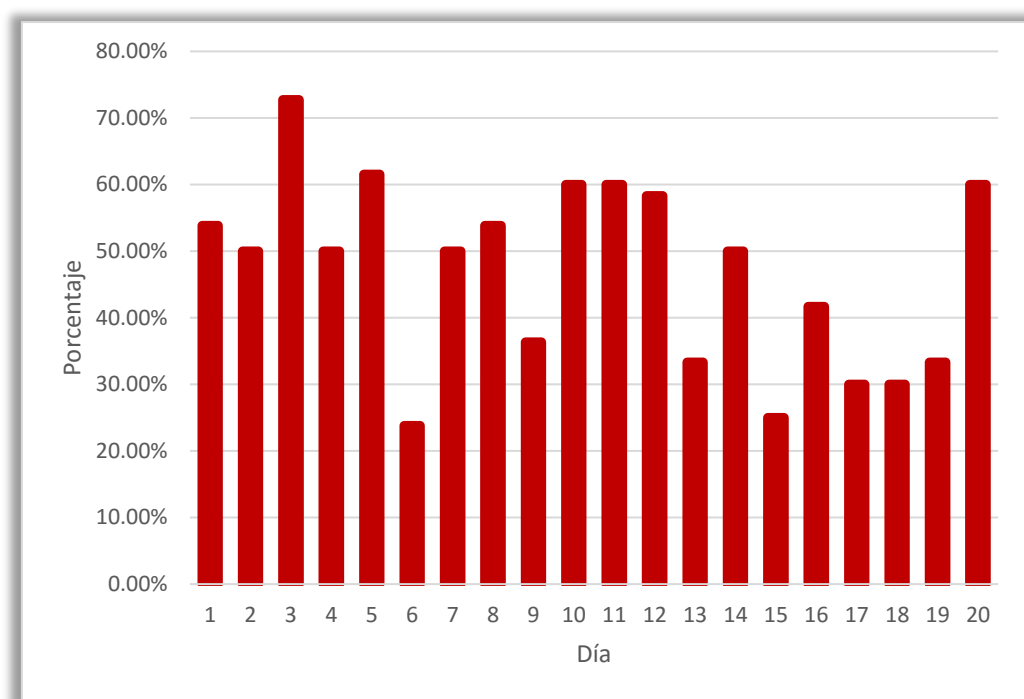
Se debe de tener en claro que hay dos tipos de casos, los cuales son cuando el usuario solicita un requerimiento y cuando el usuario reporta una incidencia, un requerimiento es algo nuevo que el usuario necesita, pero una incidencia es algo que está fallando y está deteniendo sus labores, normalmente las mesas de ayuda, solamente controlan las incidencias y no controlan los requerimientos, pero hay muchas cosas, que hoy en día, vemos necesario controlar, como por ejemplo el aprovisionamiento de equipos a los usuarios, La solución para ello es que debe de comenzar a realizar una perfecta recolección de información, debe tener un ciclo de aprobación por parte del jefe del área, entregar equipos de cómputos, asignarle permisos para esto se debe tener en claro cuáles son los roles, quienes son las personas que van a interactuar con el sistema, debemos contemplar que hay que actuar con todo tipo de solicitudes, definir qué es lo que se va a atender, entregar el servicio en el momento adecuado, evitar las respuestas lentas o los tiempos de solución indeterminable, esto ocurre porque a veces no tenemos ni idea, de qué fue lo que pasó, porque no sabemos si esto ha pasado antes o no, es por ello la importancia del sistema web en la compañía su implementación permitirá dar la

continuidad y seguimiento al servicio para asegurar que la solución de la incidencia quede registrado en el sistema, de tal manera que si esta misma incidencia es reportado por otro usuario, tengamos ya la solución definitiva en nuestro sistema aplicarlo y evitar de esta forma realizar las tareas repetitivas.

El área de sistemas está liderada por el Jefe Corporativo de Tecnología de Información quien tiene a su cargo dos equipos de trabajo, el equipo de Analistas de Sistemas y el equipo de Soporte. Cada uno de estos equipos tiene un personal a cargo; en total son 7 personas que atienden las incidencias, requerimientos, asesoría de compras de equipos informáticos tanto en la sede principal como en las sedes operativas a través de asistencias remotas, los tiempos de atención al usuario se ha ido incrementando debido la cantidad de usuarios que maneja la empresa, si bien este área cuenta con políticas y procedimientos para su atención, a veces con el apuro de resolver una incidencia de alta prioridad no se registra correctamente esta atención, puesto que el usuario va directamente al analista o soporte a reportarlo de manera verbal que se resuelva en el momento, debido a ello algunas atenciones quedan pendiente y esto genera una mala percepción del área, es por ello que se desarrollara un sistema web para el control de los servicios que brinda el área de sistemas, para tener un mayor seguimiento de las atenciones reportadas, asignación de responsabilidades, generaciones de registros en el formato que estipula sus procedimientos y estadísticas de atención. “Uno de los problemas principales que viene afrontando el área de sistemas es la falta de un sistema de control, que dé seguimiento a estas solicitudes, que permitiría en cierta forma tener un control sobre quien está a cargo de tal solicitud, si se le está atendiendo en el tiempo adecuado, a tal punto que me permita tener un historial en línea de las atenciones realizados con éxito” (Entrevista al Jefe Corporativo de T.I. Anexo p.93). Se pudo identificar que hay algunos equipos de oficina que no se ha brindado el mantenimiento preventivo en su fecha de programación, y esto se dio, porque no se cuenta con un inventario actualizado del hardware, ni de los programas instalados, es imprescindible que cada máquina este correctamente inventariada antes de realizar el mantenimiento, puesto que, de no ser así ,no habría forma de validar si las piezas que forman parte del ordenador, son las mismas o han sido cambiados o extraídas, de igual manera es importante conocer

que licencia software tiene asignado cada computadora, para evitar su desinstalación o formateo, puesto que si tiene una licencia por dispositivo este se podría perderse al realizar una reinstalación del sistema operativo, tampoco se puede saber el estado de los componentes asociados al equipo. En la siguiente figura, se puede observar la gráfica de las mediciones realizadas para el Índice de cobertura de mantenimiento correspondiente a una muestra de 20 fichas de registro de órdenes de mantenimiento preventivo de un total de 240 programados desde el 2 al 30 de abril del 2018 (ver anexo, p.79).

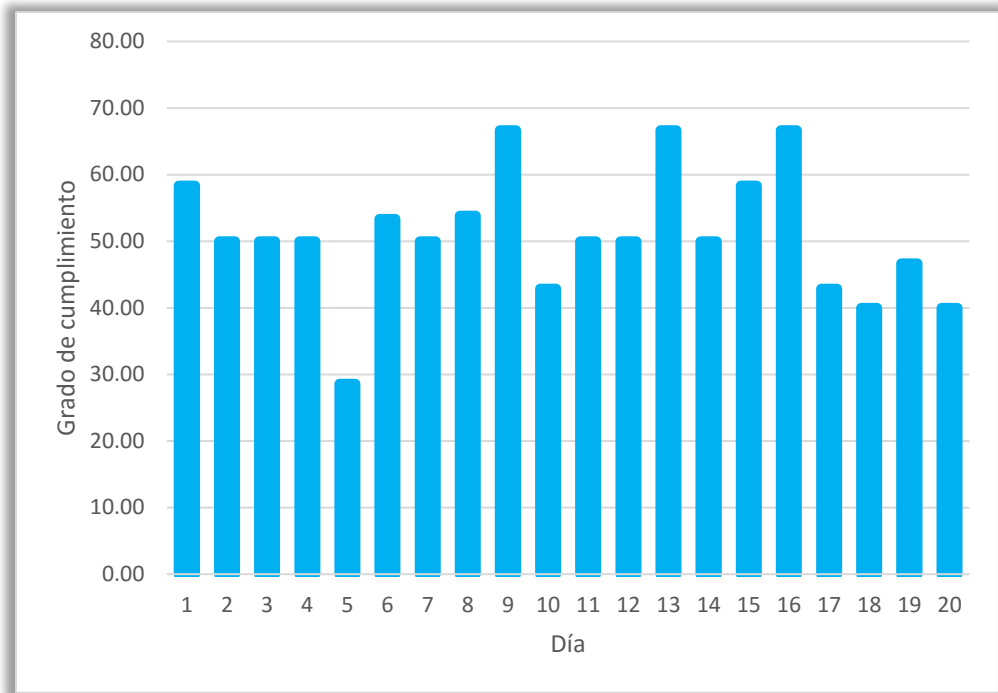
Figura 1: Índice de Cobertura de Mantenimiento (Abril del 2018)



Fuente: Elaboración propia

El grado de cumplimiento de las solicitudes registradas por el área, es otro de los problemas identificados, según la información obtenida de la entrevista al Jefe corporativo de TI, no se está logrando la variable del grado del cumplimiento de las atenciones dadas por su personal, de acuerdo a los resultados arrojados en el pre test, esta variable solo alcanzó un 50,74% lo que indica un bajo flujo de trabajo. A continuación, se muestra la gráfica representativa del grado de cumplimiento de las atenciones efectuadas por el área de sistemas correspondiente al mes de abril del 2018 (ver anexo, p.84).

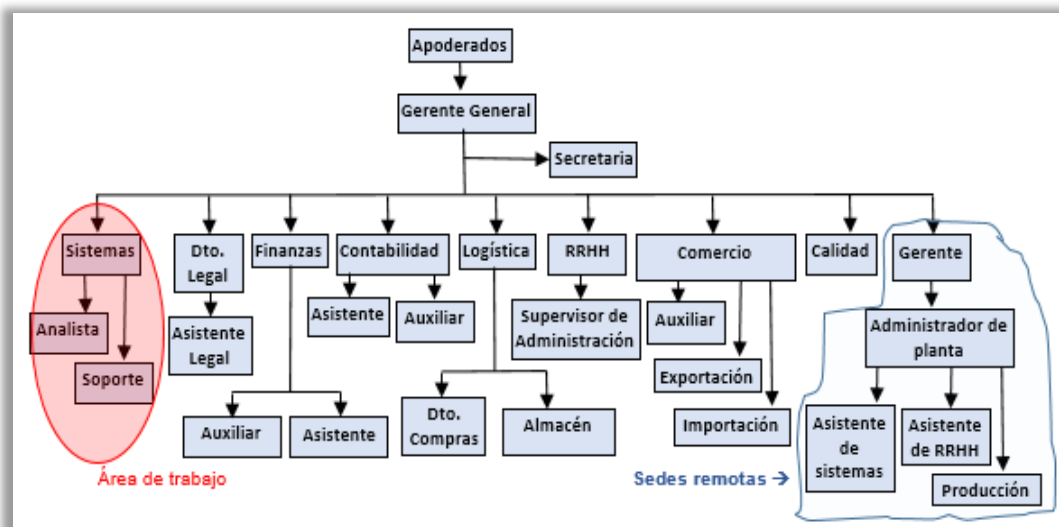
Figura 2: Grado de Cumplimiento (Abril del 2018)



Fuente: Elaboración propia

El área de estudio donde se efectuará el proyecto, es el de Sistemas de la Compañía Naviera Natalia S.A.C. y está localizado en el distrito de San Isidro de la ciudad de Lima en Perú, al igual que las demás áreas administrativas, brindan servicio tanto a la compañía como a todo el grupo empresarial pertenecientes al mismo propietario, en la siguiente figura se observa marcado en rojo el área donde se efectuará el proyecto:

Figura 3: Área de estudio



Fuente: Elaboración propia

1.2. Trabajos previos

FLORES (2017), en su tesis titulado “Desarrollo de una aplicación web para mejorar la gestión del mantenimiento preventivo y correctivo de equipos informáticos en el hospital la Caleta-Chimbote” realizado en la Universidad Cesar Vallejo de Nuevo Chimbote, tuvo como objetivo principal mejorar la gestión de mantenimiento correctivo que se le brindaba de manera irregular a los equipos informáticos del hospital de la misma manera que buscaba elevar la percepción de los usuarios, reducir el tiempo del registro del mantenimiento preventivo y correctivo, reducir el tiempo de asignación del técnico y por último, reducir el tiempo de búsqueda de equipos sin mantenimiento, desarrollo una aplicación web con una arquitectura de 3 capas hizo de java como lenguaje de programación , alojó su aplicación web en un servidor ApacheTomcat y usó PostgreSQL como gestor de base de datos, los resultados obtenidos después de la implementación del aplicativo web fueron: reducción del tiempo de búsqueda de un 87% , incremento en un 31.2% el nivel de satisfacción de usuarios, el aporte para este proyecto es que brinda una base de conocimiento para el desarrollo del marco teórico.

DOMINGUEZ (2016), Realizó una tesis denominado “Aplicación de métricas de calidad en uso utilizando la ISO 9126 para determinar el grado de satisfacción del Sistema Único de Matrícula” en la Universidad Nacional Mayor de San Marcos, el objetivo principal del proyecto, fue hacer uso de la norma ISO/IEC 9126 de calidad de software y aplicarlo al sistema SUM de la UNMSM para la evaluación de su calidad en uso. Su población abarcó a 16 usuarios con perfiles administrativos de la universidad nacional Mayor de San Marcos que usaban el sistema único de matrícula, logrando como resultado obtener las métricas del uso del sistema, obtuvieron el 89.11% que corresponde como valor de métrica equivalente a “Satisfactoriamente” lo cual quiere decir que usuarios administrativos muestran un conocimiento referente a las tareas que puede finalizar con éxito el sistema SUM, entre otras métricas también se establecieron lo relacionado al concepto de satisfacción del usuario con respecto al producto, cuyo resultado obtenido fue del 79.36% lo cual indica que, para los usuarios administrativos, el sistema SUM no se encuentra satisfaciendo las necesidades de los usuarios administrativo referentes

que no cumple todas las funcionalidades del usuario como la necesidad de realizar filtros específicos en sus reportes. El aporte que entrega a este proyecto es el uso de estas métricas como antecedente para la autoevaluación del sistema web a implementar.

CORONADO (2015), Realizo una investigación denominado Implementación de un Sistema Help Desk para mejorar la Calidad de Servicio del usuario interno del Hospital San Juan de Lurigancho”, proyecto realizado en la universidad Cesar Vallejo, que tuvo como objetivo, mejorar la calidad de servicio brindado al usuario interno del hospital, ofreciendo una solución eficaz y rápida a los incidentes concernientes a los equipos tecnológicos que disponía el hospital. De una población de 883 miembros de la organización, obtuvieron una muestra de 67 usuarios a los cuales después de la implementación, aplicaron una ficha de encuesta y lograron obtener como resultado la reducción y simplificación de los procesos, gracias a la tecnología de información, se redujo el tiempo empleado en brindar la solución a la incidencia, mejorando sustancialmente el número de atenciones por parte del servicio técnico y el nivel de satisfacción del usuario. El aporte brindado a este proyecto se basa en la construcción del marco teórico

CUESTAS (2017), En su investigación denominado “Diseño e implementación de una mesa de servicios, utilizando ITIL v3.0, para el servicio técnico en el departamento tecnológico del Ministerio del Ambiente”, tesis realizado en la Escuela Politécnica Nacional en Quito – Ecuador y tuvo como objetivo el mejorar, corregir y optimizar el servicio técnico a través de un software libre, que les permita alinear sus procesos a las mejores prácticas para el ciclo de vida del servicio proporcionadas por ITIL, los resultados obtenidos fueron una mejora en el desempeño del personal técnico en la atención oportuna de las atenciones de incidencias reportadas, puesto que después de aplicar el programa, se atendieron 1188 incidencias dentro del tiempo establecido, de un total de 1219 incidencias ingresadas, lo que demuestra un porcentaje de cumplimiento promedio de 96,45 % de incidencias atendidas a tiempo esto demuestra una notable mejora en el tiempo de respuesta lo que contribuye a la resolución de manera ordenada de estas solicitudes y generar una base de conocimientos para brindar mayor

eficiencia en la atención de los servicios, permitiéndoles además, a los directivos del departamento, la evaluación del desempeño de cada personal técnico. La aportación hacia el proyecto de investigación, radica en el conocimiento de los requerimientos funcionales que el programa debe de contemplar.

LINKE y SILVEIRA (2015) En su investigación denominado “Implantação de um Sistema Help-Desk: um estudo de caso na Exatus Soluções Estratégicas” que fue desarrollado en Brasil, en el cual desarrollan e implementan un sistema de mesa de ayuda para una empresa que brinda servicio de asesoría y consultoría de soluciones estratégicas correspondientes al área de salud, su objetivo principal, fue el de mejorar la atención hacia sus clientes, para lograr ello, re-definieron su proceso de atención, en base a una propuesta el cual agrega un sector responsable de la atención hacia sus clientes, el uso de un sistema informático que les permite registrar sucesos, tomar acciones proactivas, medidas preventivas y correctivas, reduciendo así los costos operativos y centralizando una variedad de información y asignando de manera ordenada las tareas de atención. Para el desarrollo del sistema hicieron uso de herramientas licencia publica general (Open Source, GPL), utilizaron PHP como lenguaje de programación e hicieron uso de MySQL como base de datos, en lo que corresponde al nivel de usabilidad del programa por parte del usuario, se demostró un nivel de calificación de 4.09 puntos de una escala de 5. La contribución hacia este proyecto es el concepto de cómo deben estar definidos los procesos correspondientes a las atenciones de sus clientes, establecimientos de responsabilidades y el nivel de usabilidad que debería tener todo sistema.

PALACIOS (2017) realizo una tesis en la universidad Cesar Vallejo denominado “Aplicación de operación de T.I. en el proceso de gestión de incidencias en la empresa Babymodas S.A.C.” Tuvo como principal objetivo establecer el efecto del uso de la operación de servicio de la tecnología de información en el proceso de gestión de incidencias en la sede central de la empresa Babymodas S.A.C. Su población estuvo integrada por setecientos quince registros de incidencias tomadas entre julio y agosto del 2017, se determinó las mejoras en cuanto a sus indicadores, con un aumento de 7,73 atenciones de incidencias realizados por día y un aumento

de 1,5 atenciones de incidencias críticas por día, disminuyendo significativamente el tiempo promedio utilizado en la resolución de la incidencia de 137,80 a 23 minutos con la aplicación de operación de T.I. El aporte al proyecto es la obtención de conceptos básicos informativos para la elaboración del marco teórico.

1.3. Teorías relacionadas a tema

1.3.1. Sistema Web

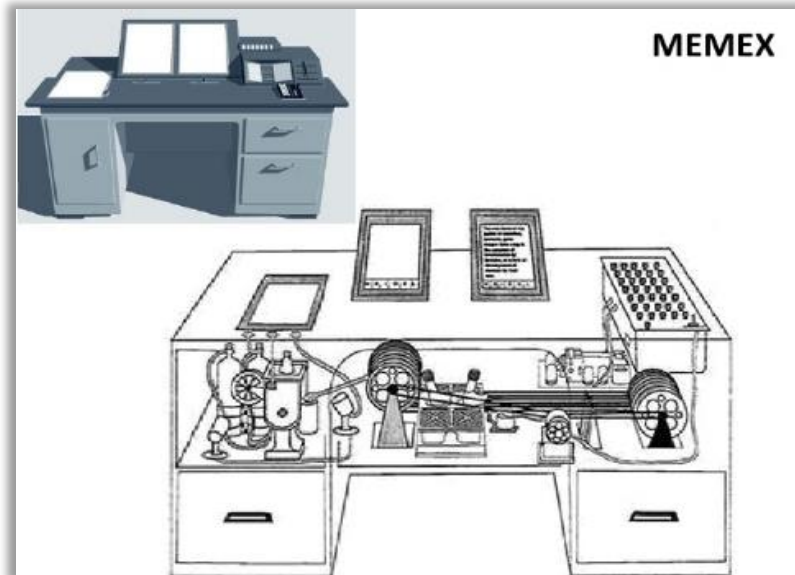
Para definir un sistema web Berzal, Cortijo y Cubero sostienen al respecto:

[...]Son las aplicaciones cuya interfaz se elabora a partir de páginas web, las cuales están constituidos de ficheros de texto en formato estandarizado conocido como lenguaje de marcado de hiper texto o HTML, estos ficheros se encuentran alojados dentro de un servidor web, para su acceso desde cualquier ordenador solo hace falta que este tenga instalado cualquier tipo de navegador ya sea de Microsoft, Google o Mozilla y haga uso del protocolo de transferencia de Híper Texto o HTTP para la comunicación entre ellos. (2005, p. 1).

Para entender cómo funcionan las aplicaciones web, hay que tener en cuenta que es una página web y como ha ido evolucionando a través de la historia, todo surge en 1945 bajo la necesidad de un sistema para el tratamiento de los conocimientos acumulados por la humanidad y de poder compartir esta información entre los científicos de una manera rápida y centralizada para que puedan estar actualizados con toda la información correspondientes a sus trabajos, en una época donde todavía no se disponía de la tecnología de información que hoy día disponemos, un ingeniero eléctrico de nombre Vannevar Bursh no solo plasma esta necesidad, sino que también idealiza una solución tecnológica al que nombro Memex, en un artículo Titulado “As We May Think” lo que traducido al español sería “Como Pudiera Pensarse” publicado en 1945 por la revista Atlantic Monthly. En su artículo Vannevar, definió a Memex como dispositivo en el que la persona guarda sus registros, libros y comunicaciones, mecanizado de tal manera que permite hacer las consultas de manera rápida y flexible, este dispositivo lo ilustro como una mesa que disponía de pantallas translucidas, teclado, juegos de botones y palancas que permitían la navegación, selección de los datos almacenados y proyección de estas imágenes hacia sus pantallas traslucidas, el uso de esta indexación

asociativa para la vinculación de la información requerida, tomo mayor relevancia y su artículo sirvió de inspiración a muchos desarrolladores de hardware y software para computadoras personales.

Figura 4: Memex



Fuente: Jiménez, 2010

1.3.1.1. Nacimiento del Internet

En 1960, en plena guerra, EEUU a través de su ministerio de defensa, crea una red cuyo objetivo principal era la de crear comunicaciones directas entre ordenadores, para que puedan comunicarse entre las diferentes bases de investigación de manera rápida y segura de tal manera que permita brindar el acceso a la información militar desde cualquier lugar del país, a esta red se le denominó ARPA (Agencia de proyectos de investigación avanzada) y estaba constituido por unos 200 científicos de alto nivel, en 1967 el ARPA publica un plan para crear una red de ordenadores al cual nombraron ARPANET, esta red fue creciendo y para el año de 1971 ya se contaba con 23 puntos conectados brindando el acceso a la red a cualquier persona con fines académicos o de información con lo que a principios de los 80 se comenzaron a desarrollar los ordenadores de forma exponencial.

Figura 5: Interfaz ARPANET para el PDP-10 de Xerox



Fuente: PARC, 1970

1.3.1.2. Hipertexto

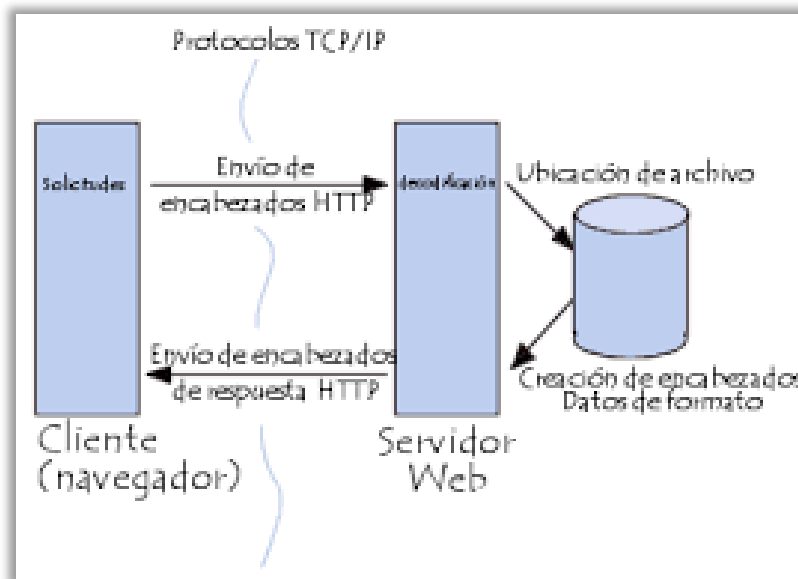
En 1965 Ted Nelson, quien defendía las ideas de Bush, define el término Hipertexto como una serie de fragmentos de texto que permite al lector, mediante enlaces, definir su propio recorrido a través de la información, la cual podría lograrse con ayuda de un programa interactivo que le permite escribir y leer en una pantalla la información que necesitaba, facilitando así una gran disposición de variedad de bibliotecas de información, al cual pueden acceder de manera inmediata.

1.3.1.3. WWW (World Wide Web)

En 1989, Tim Berners-Lee conocido el padre de la web, quien laboraba como científico en el CERN (centro europeo para la investigación nuclear), presenta la primera propuesta de la www mediante la unión del internet y del hipertexto, una iniciativa de hipermedia con base en la internet cuyo propósito era compartir el conocimiento de manera global, en 1990 escribió su primer servidor y cliente web, creó su primer explorador y lo almacenó en su primer sitio web en una máquina, corrigiéndose en el tiempo en base a la idea que utilizó para su creación, refinando sus especificaciones de URI, HTTP y HTML a medida que se propagaba su tecnología web y es así como, en agosto de 1991, se produce el lanzamiento oficial en línea de la www, posteriormente en 1994, cuando Tim laboraba en el laboratorio de ciencias de la computación e inteligencia artificial de instituto de tecnología de

Massachusetts, en EEUU, se crea la W3C el cual es una organización cuya función es de aplicar y regularizar los estándares de la tecnología web. Un servidor web utiliza el protocolo HTTP (Protocolo de Transferencia de HyperTexto) para comunicarse con los clientes, los clientes a través de un navegador (Chrome, Firefox, IExplorer) realiza un tipo de petición (imágenes, textos, videos) al servidor, esta toma la petición y entrega una respuesta.

Figura 6: Comunicación Cliente / Servidor Web



Fuente: <http://temarioweb.blogspot.com>

1.3.2. Herramientas para su desarrollo

1.3.2.1. Visual Studio.net

Para definir visual Studio.net, Del Sole sostiene al respecto:

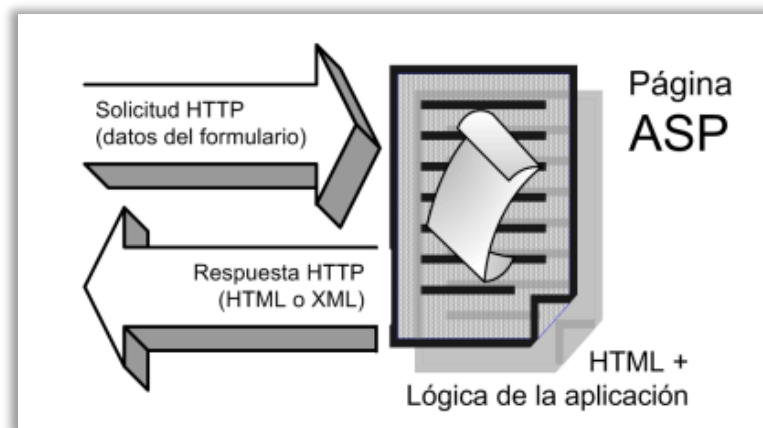
[...]Es un completo set de herramientas para el desarrollo de Servicios Web XML, aplicaciones web, de escritorio y móviles. Visual C++, Visual Basic y Visual C# hacen uso de este entorno de desarrollo integrado, puesto que les permite utilizar las herramientas compartidas facilitando la elaboración de soluciones en diversos lenguajes, utilizan las funciones de .NET Framework, esto le permite tener el acceso a tecnologías que simplifica el desarrollo de Servicios Web XML y aplicaciones web ASP, cabe mencionar que hay dos tipos de páginas web las cuáles son las estáticas y las dinámicas, en la actualidad hay una mayor preferencia a los sitios web o páginas web hechas para que funcionen dinámicamente. (2017, p. 37).

1.3.2.2. ASP (Paginas Activas del Servidor)

Para definir ASP, Berzal et al. sostiene al respecto:

[...]Es de propiedad de Microsoft, permite elaborar aplicaciones web que se ejecutan en el servidor de información de internet (IIS). Una página ASP no es más que un archivo HTML cuya extensión es asp o aspx, en el que agregamos códigos, que, por lo general, es del tipo de VBScript (Visual Basic Script), pero que también puede hacer uso de otros intérpretes de lenguaje, esto hace que cuando un cliente accede a la página, el IIS interprete el código que incluye la página ASP y envíe una respuesta a través del navegador instalado en el equipo del cliente. (2007, p.26).

Figura 7: Funcionamiento de páginas ASP



Fuente: Berzal, 2007.

1.3.2.3. El Patrón MVC.

Para definir el patrón Modelo Vista Controlador, Eslava sostiene:

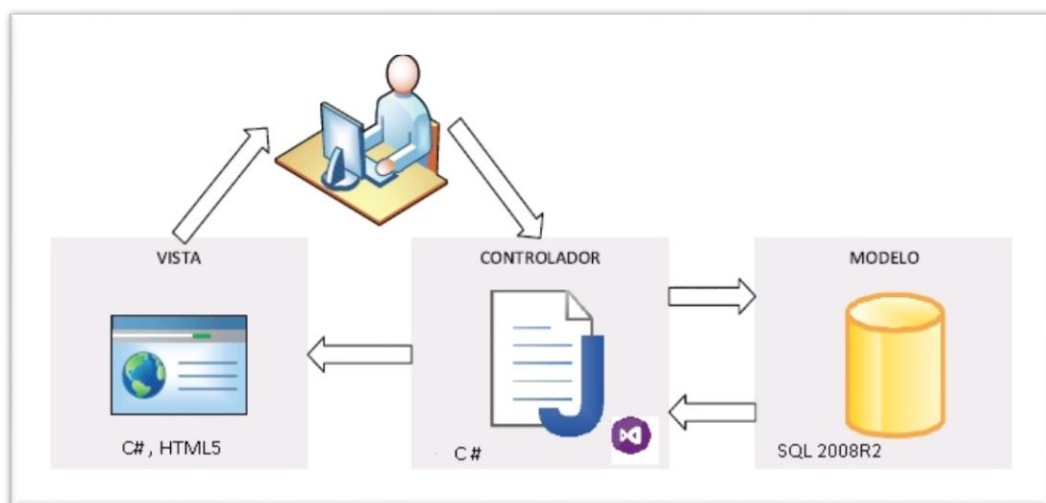
[...]Es un patrón de arquitectura de software cuyo objetivo es separar la lógica del negocio de la interfaz del usuario por lo que es el más utilizado en aplicaciones Web, ya que facilita su mantenibilidad, escalabilidad y funcionalidad del sistema, de manera sencilla y simple, lo que a su vez permite no mezclar los lenguajes de la programación en el mismo código. (2008, p.109).

Es un tipo de arquitectura de desarrollo de software el cual se divide en tres componentes los cuales son; modelo, vista y controlador. Entre las ventajas de trabajar con este tipo de arquitectura destaca su adaptabilidad a los diferentes tipos de frameworks existentes, su programación se realiza de manera ordenada, facilitando tanto su mantenimiento en equipo, como su escalamiento, su diseño nos permite facilitar la programación de aplicaciones empresariales complejas, ya que cada componente trabaja de manera independiente, de tal forma que la

modificación en uno de estos no afecta en gran medida a las otros componentes, para lograr esto, cada componente tiene asignado una función específica, por hacer mención del modelo, cuya función es la de conectarse e interactuar con la base datos haciendo uso de colecciones de clases que le permite enviar o recibir los datos hacia el con el controlador, el controlador tiene la función de recibir las órdenes del usuario y solicitar los datos solicitados al modelo para posteriormente comunicar los datos devueltos a la vista, es aquí donde se desarrollan todas las acciones de los diferentes tipos de eventos que en toda programación suele requerir, como por ejemplo, para una solicitud de crear un nuevo registro de usuario, el controlador recibe de la vista las variables que representan los datos que conforma esta entidad y se las envía al modelo, si los parámetros son correctos, guarda todos los cambios realizados y redirige a una acción específica que retorna la vista en el cual listara todos los registros de la entidad afectada, con referente a la vista su función es la de presentar los datos del sistema de manera ordenada en una interfaz agradable a la vista del usuario y de fácil usabilidad, para lograr ello hace uso de diferentes tipos de elementos HTML con estilos propios, personalizados o referenciados, es aquí donde se diseña la presentación de las páginas, su encabezado, pie de página, paneles laterales, vistas parciales, contenidos, etc.

A continuación, se muestra una representación del patrón modelo vista controlador:

Figura 8: Flujo típico MVC ASP.NET



Fuente: Mera, 2015

1.3.2.4. Base de Datos

Para su definición, Kendal sostiene:

[...]Es una fuente centralizada de datos cuya finalidad es que los usuarios hagan uso de ella de manera compartida durante el desarrollo e implementación de sus aplicaciones. El DBMS es el corazón de una base de datos, puesto que permite al administrador de bases de datos, dar mantenimiento a la base de datos, resguardar y respaldar los datos, generar informes, notificaciones, asignar permisos y otras muchas funciones importantes que facilitara al administrador cumplir con los objetivos estipulados por la organización. (2011, p. 403).

Para Silberchatz y Sudarshan (2006, p.1), “consiste en una colección de datos, que contiene información relevante para la mayoría de las organizaciones”.

1.3.2.4.1. SGBD (Sistema gestor de base de datos)

Con respecto a la definición del SGBD, Silberchatz y Sudarshan sostienen:

[...]Está compuesto de un conjunto de datos interrelacionados y un conjunto de herramientas para su acceso y modificación, incluye la definición de la estructura que tendrá el almacenamiento de la información y los mecanismos para su procesamiento, su principal objetivo es almacenar y recuperar la información de la base de datos, de forma eficiente y práctica. (2006, p.1).

1.3.2.4.2. Lenguaje de base de datos

Por su funcionalidad está constituido por 2 tipos de lenguaje que son:

1.3.2.4.2.1. Lenguaje de almacenamiento y definición de datos (LDD)

Conjunto de instrucciones utilizados para definir la estructura de una base de datos y de todas sus propiedades, (que usuarios tendrán el acceso a la base de datos y cuáles serán sus permisos), utilizados también para la creación de tablas, definición de las propiedades de los campos de una tabla.

1.3.2.4.2.2. Lenguaje de manipulación de datos (LMD)

Conjunto de instrucciones utilizados para el mantenimiento de la información de una base de datos (insertar, borrar, modificar, o seleccionar datos de una tabla).

1.3.2.4.3. Modelo de datos

Son un conjunto de herramientas conceptuales que nos permite plasmar el diseño de una base de datos a nivel físico, lógico y de vistas, esto nos permite tener un mayor conocimiento de la base de datos (Tipos de datos de una tabla, como están constituidos y relacionados, nombres de tablas, restricciones ...etc.). Existen 4 categorías de modelos de datos los cuales son:

1.3.2.4.3.1. Modelo relacional

Constituido por un conjunto de tablas relacionadas, ordenados y organizados, cada tabla tiene varios dominios (campos o columnas) y tuplas (filas o registros). Cada campo representa un área en el que se almacena datos de un mismo tipo por ejemplo en la siguiente tabla se puede observar que la tabla “empresas” tiene 6 campos los cuales son: "Id", que es el campo que almacena datos del tipo entero con propiedad de ser la llave primaria y único; “RazonSocial”, en el que se almacenan datos del tipo texto; “RUC”, que es un campo en él que se almacena datos del tipo texto, con propiedades de 11 caracteres; “TipoEmpresa”, en el que se almacena datos del tipo entero y tiene la propiedad de ser una llave foránea; “Telefono”, en el que se almacenan datos del tipo texto y por último el campo “Correo” en donde se almacena datos del tipo texto. El registro es una colección de datos de iguales o de diferentes tipos que están relacionados, por ejemplo, el décimo registro de la tabla empresa contendría los datos: 10, BMP SAC, 20602371931, 3, 3345689 y info@constructoraroble.com, esto representa un conjunto de datos de diferentes tipos que están relacionados, pues pertenecen a una empresa. La tabla es una colección de registros relacionados, que continuando que con nuestro ejemplo sería la tabla “Empresas”.

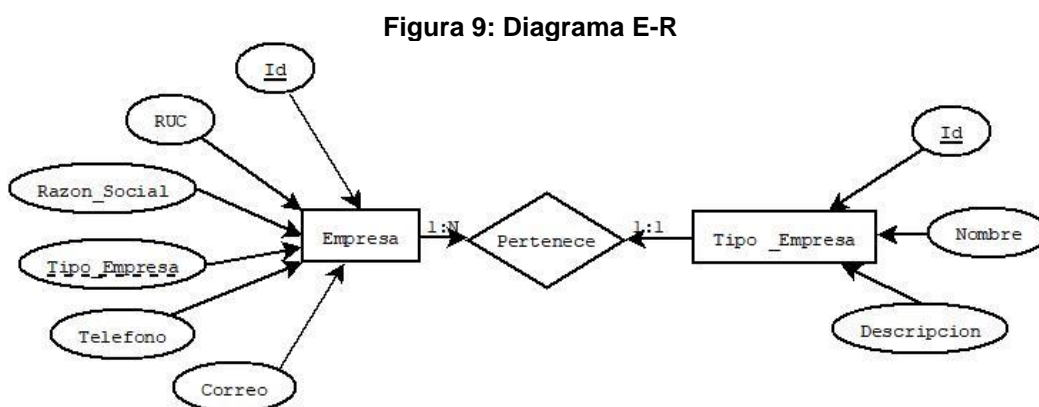
Tabla 1: Ejemplo modelo relacional (Tabla Empresa)

Id	RazonSocial	RUC	TipoEmpresa	Telefono	Correo
1	Constructora El Roble SAC	12345678789	1	2020899	facturacion.electronica@constructoraroble.com
2	Pesquera Natalia SAC	20513452153	3	2020890	facturacion.electronica@pesquanatalia.com
3	Compañía Naviera Natalia SAC	20537243709	3	2070899	facturacion.electronica@navieranatalia.com
4	Sobitec	20513511770	3	3020899	facturacion.electronica@sobitecperu.com
5	Migiva	20143296254	3	2010899	facturacion.electronica@migivagroup.com
6	American Quality AquaCulture	20472086457	3	4020899	info@constructoraroble.com
7	Cultimarine	20519330874	3	5020899	facturacion.electronica@cultimarine.com
8	CultiPerú SAC	20600970055	3	2055301	facturacion.electronica@cultiperu.com
9	Agrícola Andrea SAC	20505688903	3	4267830	facturacion.electronica@agricolaandrea.com
10	BMP SAC	20602371931	3	3345689	info@constructoraroble.com

Fuente: Elaboración propia

1.3.2.4.3.2. Modelo entidad-relación

Este modelo representa los casos reales en como una entidad se relaciona con las demás o consigo misma, consta de elementos básicos representados por símbolos los cuales son usados para esquematizar la base de datos a través de un diagrama entidad relación, estos símbolos son: un rectángulo para representar la entidad que es un objeto que va ser gestionado en la base de datos; un rombo representa el tipo de asociación entre dos identidades o entre sí misma; el elipse representa cada atributo que una entidad puede tener, que por lo general son las propiedades de una entidad; una línea se utiliza para unir la entidad con otra entidad, con sus atributos o con sus relaciones. Cabe destacar que cada entidad dispone de una única clave primaria que lo identifica y se presenta por medio un atributo con su nombre subrayado; otro elemento son las cardinalidades que es el mínimo y máximo número de identidades con se puede relacionar una entidad y se representa como 1:1(relación de 1 a 1); 1: N (relación de 1 a varios); 1:1(relación de varios a 1); N: N (varios a 1 a varios); en la siguiente figura se muestra un ejemplo de un diagrama entidad relación. Es necesario enfatizar que la información de la base de datos tiene que estar normalizada.



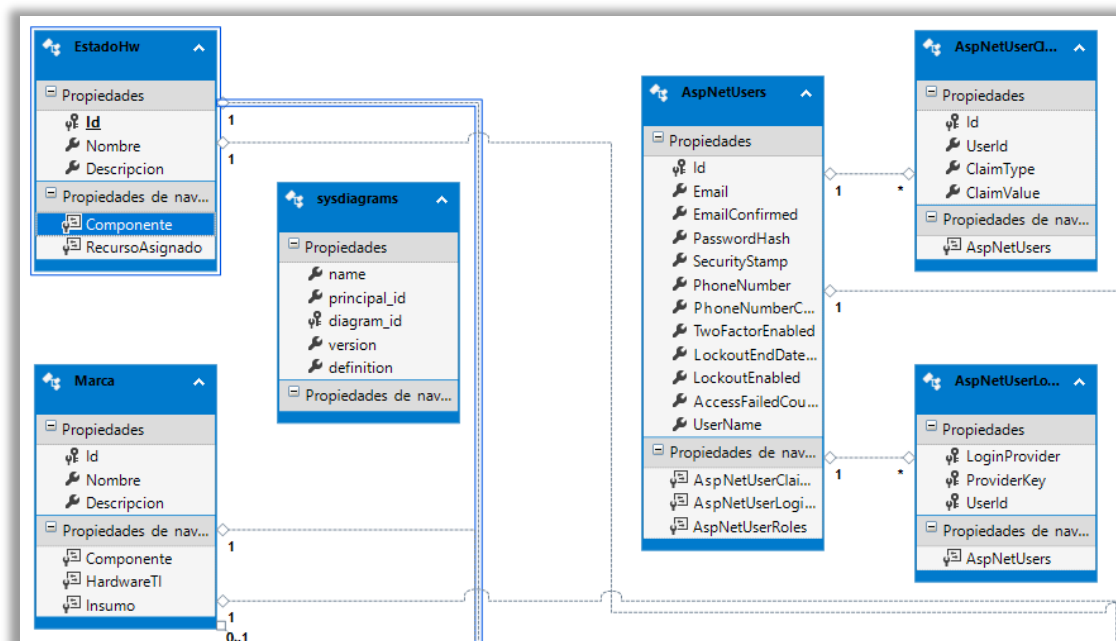
Fuente: Elaboración propia

1.3.2.4.3.3. Modelo de datos orientado a objetos

Extiende el modelo de datos relacional ofreciendo un sistema de tipos más ricos que incluye tipos de datos complejos y orientados a objetos. Los sistemas de datos basados en este modelo brindan a los usuarios del modelo relacional, la facilidad de migrar a este modelo para que puedan utilizar todas sus características. Un sistema de base de datos orientado a objeto debe satisfacer 2 criterios: debe tener

un gestor de base de datos y debe ser un sistema orientado a objetos, en lo referente a lo primero este se traduce en 5 características que son: persistencia, concurrencia, recuperación, gestión de almacén y facilidad de consultas y el segundo se traduce en 8 características los cuales son: abstracción, encapsulación, tipos o clases, modularidad, jerarquía, generalidad, concurrencia y persistencia, las ventajas de utilizar este tipo de modelo son: la flexibilidad, manejo de datos complejos, se ajusta al espacio necesario y elimina espacio desperdiciado, manipulación de objetos complejos de manera rápida y ágil.

Figura 10: Modelo de datos Orientado a Objetos



Fuente: Elaboración propia

1.3.2.4.3.4. Modelo de datos semiestructurados

En esta categoría se encuentran los datos que no residen de una base de datos relacionales sin embargo presentan una organización interna que facilita el tratamiento de sus datos, se caracteriza por no tener una estructura rígida no preferida, se encuentran en los documentos XML, no respetan un esquema particular, pueden ser estructuras incompletas, pueden no respetar la estructura de los datos pre-existente, el manejo de los datos se realiza mediante una colección de nodos, cada uno conteniendo datos con diferentes esquemas, el nodo contiene información sobre la estructura de su contenido.

Figura 11: Esquema XML

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE peliculas SYSTEM "Peliculas.dtd">

<peliculas>
  <pelicula tipo="comedia" sistema="PG-13" ejemplares="5" año="1987">
    <titulo>Raising Arizona</titulo>
    <guionista>Ethan Coen</guionista>
    <guionista>Joel Coen</guionista>
    <productor>Ethan Coen</productor>
    <director>Joel Coen</director>
    <actor>Nicolas Cage</actor>
    <actor>Holly Hunter</actor>
    <actor>John Goodman</actor>
    <comentarios>Una pelicula clasica de Comedia.</comentarios>
  </pelicula>

  <pelicula tipo="ciencia-ficcion" sistema="PG-13" ejemplares="4" año="1989">
    <titulo>The Abyss</titulo>
    <guionista>James Cameron</guionista>
    <productor>Gale Anne Hurd</productor>
    <director>James Cameron</director>
    <actor>Ed Harris</actor>
    <actor>Mary Elizabeth Mastrantonio</actor>
    <comentarios>Una buena pelicula.</comentarios>
  </pelicula>
</peliculas>
```

Fuente: www.mundolinux.info

1.3.2.5. Oracle.

Fundado en 1977 por Ed Oates, Larry Ellison y Bob Miner, bajo el nombre de laboratorio de desarrollo de software que más adelante se cambiaría a Oracle, fue uno de los primeros en lanzarse al mercado, además de su sistema gestor de base de datos, dispone de herramientas de consultas, inteligencia de negocios, análisis, minería de datos, entre otras interesantes herramientas que permiten la integración con el servidor a continuación describiremos alguna herramientas disponibles de Oracle: como para el diseño de datos que están incluidas en Oracle Developer Suite, contiene herramientas para desarrollar formularios, realizar modelados de dato, de clases actividades, flujo de control, diagramas de entidad – relación, entre otros, también dispone de herramientas para consulta útiles como para la generación de informes, consultas desde la web, dispone de características como la recuperación y el control de concurrencias, Oracle utiliza una estructura de memoria dividida en 3 áreas: una destinada para reservado para el código de software, otra para el área global del programa y otra para el área global de sistema (Silberchatz, Korth y Sudarshan, 2006, p. 833).

1.3.2.6. SQL Server

“Es de propiedad de Microsoft, está definido como un sistema que analiza y administra la base de datos relacionales, actualiza sus versiones críticas para proporcionar mayor rendimiento, seguridad, facilidad de uso y disponibilidad” (Del Sole, 2017).

SQL Server es uno de los más fuertes competidores de Oracle, Estos dos sistemas mencionados son los que más usan los desarrolladores y tienen una mayor acogida en el mercado en el sector de las bases de datos. SQL Server tiene muchas características similares a Oracle, aunque también tiene diferencias evidentes, por ejemplo; SQL Server dispone de Transact – SQL, el cual es un conjunto de programas que permite tener un control en el tratamiento de excepciones y errores; permiten además el procesamiento y la extracción de datos directos de la Web; permiten el uso de varios lenguajes de programación y por mencionar otra de sus características que lo hace especial es el gran carácter administrativo que dispone, en la seguridad, las funciones y gran flexibilidad en sus bases de datos (Acero Linares, 2018).

En el presente trabajo de investigación se dio preferencia a SQL Server, debido a que la empresa ya dispone de esa infraestructura de la base de datos licenciada sobre el cual se está almacenando los datos de los sistemas de la empresa y que además se busca integrar y centralizar la información para disponer de ella en futuros desarrollos, es decisión del cliente, trabajar sobre su misma infraestructura, para que dispongan del mayor control por políticas de seguridad.

1.3.2.7. MYSQL

Es la base de datos en el que todos pueden acceder, y por ser de código abierto, se da libre disponibilidad para que puedan modificar, arreglar, incluir elementos o mejorar el código fuente, es por ello que, gracias la dedicación del personal de MySQL y al valor colaborativo de la comunidad MySql se ha convertido en una completa herramienta, superando a un gran número de bases de datos de similares características, hay en día este gestor se ha hecho muy popular, dispone de las herramientas necesarias para su instalación , generar niveles de accesos al

usuario, proteger, respaldar y restaurar la base de datos, utiliza SQL (lenguaje de consulta estructurado) para la creación de la base de datos, generación de consultas, manteniendo de tablas, entre otro (Gilfillan, 2003, p.39).

1.3.3. Control de Servicios Informáticos

ITIL V3 (2009, p.31) “El control de servicio consiste en un ciclo continuo de monitorización, comunicación y acción. Esto es muy importante para la provisión, soporte y mejora de servicios”.

“Todo técnico o persona responsable de la tecnología de información tiene como principal deseo, poder tener todo bajo control, es por ello que la industria ha tomado conciencia de la importancia de 2 procesos que le permitirá tener el control de la tecnología de información y de los servicios asociados, los cuales son: el proceso de la gestión de cambios y el proceso de la gestión de la configuración” (AENOR, 2011 p.621).

1.3.3.1. Proceso de la gestión de cambios. Describe una buena práctica para controlar y efectuar los cambios ya sea hardware, software, de infraestructura o de servicios TI. De manera planificada, evaluada y documentada, se asigna roles y responsabilidades, el gestor de cambios es quien agenda estas acciones, como buena práctica ITIL recomienda usar herramientas y metodología de proyectos para administrar estos cambios, seguir este proceso de gestión de cambios mitigara el impacto que pueda producir un cambio no planificado. Existen 3 tipos de cambio los cuales son: Cambio estándar los cuales son aquellos que ya han sido previamente planificados y por ello se rigen bajo un procedimiento definido; El cambio Normal, el cual al igual que el estándar a diferencia que para efectuar dicho cambio se necesita de una autorización y, por último; el cambio de emergencia, el cual se realiza para dar solución inmediata a un problema grave de un elemento TI que puede producir un impacto negativo en la organización. (AENOR, 2010, p.662).

1.3.3.2. Proceso de la gestión de la configuración. Describe una buena práctica para registrar y actualizar la información correspondiente a todos los elementos de TI. (hardware, software) tiene como principal objetivo validar que el cambio

realizado no afecte de manera negativa a cualquier otro elemento TI, esta gestión aporta una detallada base de conocimiento, actualizada y centralizada que será de mucha ayuda para el personal que brinda servicio TI en la organización. (AENOR, 2010, p. 625).

La Compañía Naviera Natalia S.AC. Dispone de 6 procedimientos para el área de sistema, los cuales mencionare a continuación:

1.3.3.1. Creación de Código o Asignación de Nivel de Acceso dentro de un Sistema de Gestión de Usuario

Es el proceso para brindar acceso dentro sistema informáticos o de gestión a los usuarios que necesiten utilizarlo y asignar alcances de operaciones dentro del mismo.

El manual P-SIST-01-V03 denominado “Creación de Código y Asignación de Nivel de Acceso dentro de un Sistema de Gestión de la Compañía Naviera Natalia S.A.C.” Indica que el proceso inicia con la solicitud de la creación de código de usuario y asignación de nivel de acceso a los sistemas, esta solicitud es realizada por el jefe del área y lo realiza mediante un correo electrónico dirigido a sistemas donde se especifica; el sistema de gestión a acceder, los procesos del sistema que accederá y los límites de acceso. El Jefe de Sistemas recibe el correo y asigna de manera oral o telefónica la creación de código de usuario y asignación de Nivel de Acceso a un Analista, quien antes de crear el código de usuario debe considerar los siguientes parámetros: Sistema a acceder, Nombre de Usuario, clave, accesos a opciones del sistema y el alcance de los accesos, luego de ello realizará las pruebas de acceso al sistema con el nuevo código de usuario creado, concluido las pruebas, el Analista enviará un correo al usuario brindándoles los datos correspondientes para su acceso y capacitara al usuario sobre el uso básico del sistemas, posteriormente registrara esta labor en el listado de “Personal Activo en Sistemas de la Empresa”.

1.3.3.2. Control de Resguardo de Información Digital

Arjen, Jan, Axel y Mike sostienen:

[...]El resguardo de la información y su restauración son 2 de los factores más relevantes para mantener la continuidad de un servicio TI, mediante el diseño de servicio se debe explayar las mejores estrategias para el resguardo de la información de cada servicio y a través de la transición del servicio se garantizará su ejecución y validación de los datos respaldados o resguardados, El único propósito de realizar estas acciones es garantizar que la información se pueda restaurar sin problemas ante cualquier tipo de fallo. (2008).

El manual P-SIST-02-V03 denominado “Control de Backup de Información digital” Especifica dos tipos de resguardo, una para los datos del sistemas de gestión y otra para el resguardo de información de los usuarios administrativos, los resguardos hacia otro medio de almacenamiento se realizan de manera automática a través de scripts y Job, el copiado por duplicado de este resguardo se realiza cada mes por el personal de soporte y sistemas una copia es entregado para Sistemas, la que se guardará en un ambiente seguro distinto al de los servidores y otra para el Apoderado o Gerente General, quien la mantendrá a buen recaudo fuera del local de la empresa. El Jefe de Sistemas le entrega a Gerente General o Apoderado, para su resguardo fuera de las instalaciones de la empresa, la copia de información referida cada mes, adjuntando el formato Registro de Entrega de Resguardo de Información Digital debiendo firmar ambos el documento, dejando una copia en poder de Gerente General o Apoderado y el original para el Jefe de Sistemas.

1.3.3.3. Dar de Baja a Usuarios

Arjen et al. (2008),” Los dispositivos de registro y la información de los registros se deben proteger contra manipulaciones indebidas y accesos no autorizados”.

Este procedimiento se realiza con el fin de impedir el acceso a Sistemas de Información al personal que ya no labora en la empresa y se inicia con la solicitud del Gerente de RRHH al Jefe de Sistemas la baja del usuario mediante correo, el Jefe de Sistemas supervisara la baja del usuario que incluye el cambio de la contraseña del correo usando, configuración de reenvío de los correos del Usuario al correo del Jefe de Área. El analista y el personal de soporte revisaran el registro

del personal activo para saber a qué sistemas tiene accesos y así poder generar una clave aleatoria a la cuenta para impedir su acceso a los sistemas, si el Jefe de Área solicita un resguardo actualizado de información, Soporte Técnico procederá con el resguardo.

1.3.3.4. Mantenimiento Preventivo de Equipos de Computo

Para definir el mantenimiento preventivo, ISO 20000 sostiene al respecto:

De forma general, todo servicio requiere en algún momento de su prestación la realización de tareas de mantenimiento. El mantenimiento de los servicios debe llevarse a cabo de forma que se minimice su impacto sobre el servicio y su disponibilidad, utilizándose cuando es posible las denominadas ventanas de mantenimiento. (2007).

“El mantenimiento preventivo tiene como meta asegurarse que los equipos estén operativos el mayor tiempo posible, funcionando sin problemas. Resalta la importancia de los componentes que conforman el equipo, del software instalado y de los datos que esta contiene” (Gallego, 2010, p.6).

Tabla 2: Importancia del mantenimiento

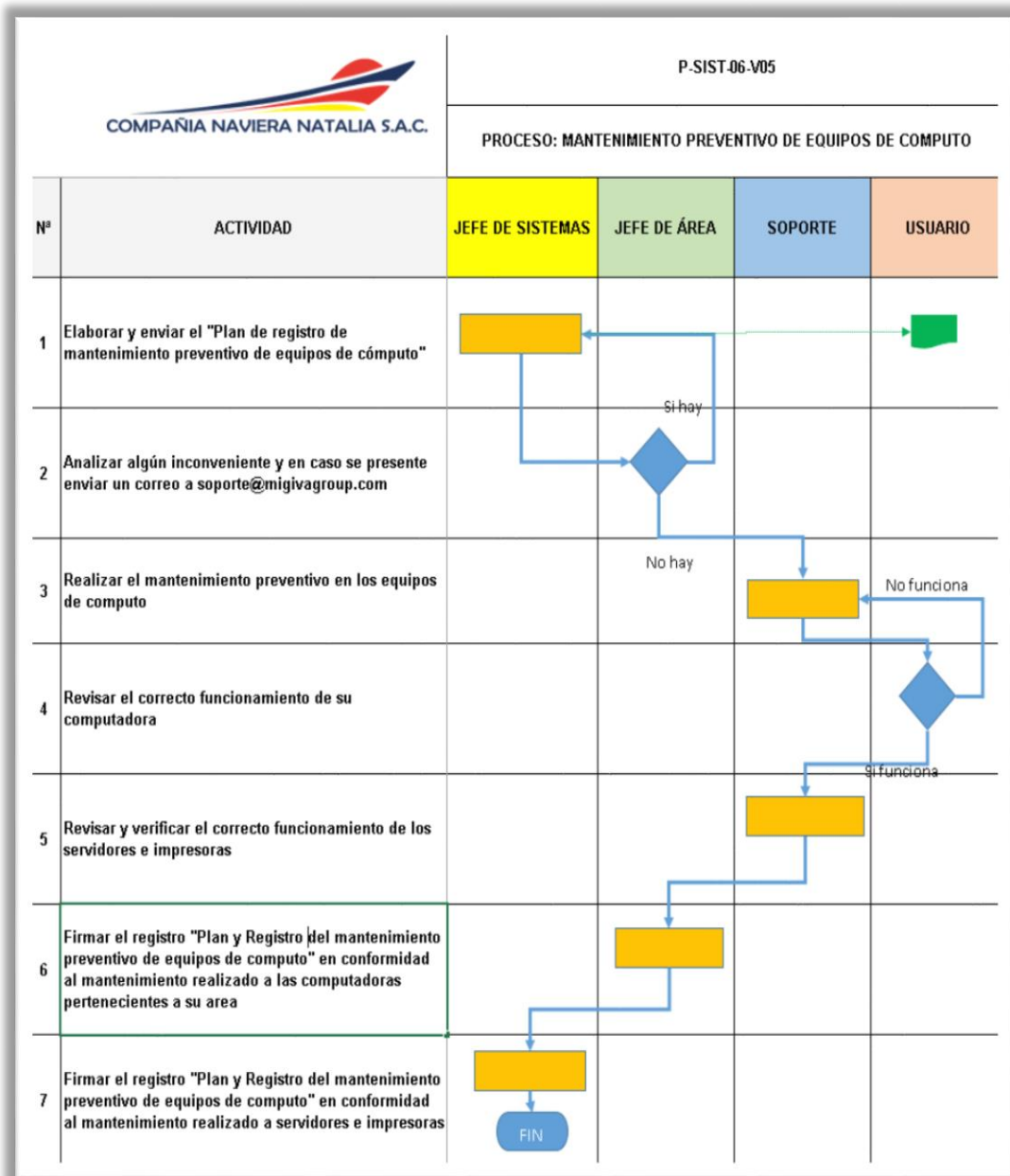
Número de razón	Descripción de razón
R-01	A media y largo plazo el coste del sistema es menor, ya que siempre será más económico mantener un equipo para evitar averías que reemplazarlo por estar averiado.
R-02	Un fallo del equipo de cómputo puede provocar una pérdida de información de costes incalculables.
R-03	Problemas intermitentes en determinados equipos del sistema retrasan el trabajo, lo que influye en el rendimiento y en la productividad

Fuente: PCPI- Mantenimiento de sistemas microinformáticos

Procedimiento para brindar el soporte adecuado en cuanto al mantenimiento preventivo y correctivo de los Equipos informáticos, con los cuales trabaja la empresa, en la Central y Unidades Operativas, este procedimiento se realiza una vez al año y se inicia con la elaboración y publicación mediante correo del “plan de mantenimiento preventivo de equipos de cómputo” por parte del jefe de área, previa coordinación con los jefes de áreas, culminada la labor, el jefe de área firmara el

registro "Plan y registro de mantenimiento preventivo de equipos de cómputo" en conformidad al servicio realizado a los equipos de sus áreas.

Figura 12: Procedimiento de mantenimiento preventivo



Fuente: Elaboración propia.

1.3.3.6. Creación e Instalación de Correos

Procedimiento para brindar acceso a una cuenta de correo electrónico con dominio de la empresa a usuarios autorizados por la Gerencia de Recursos Humanos, se inicia cuando el jefe de área envía un correo con la solicitud

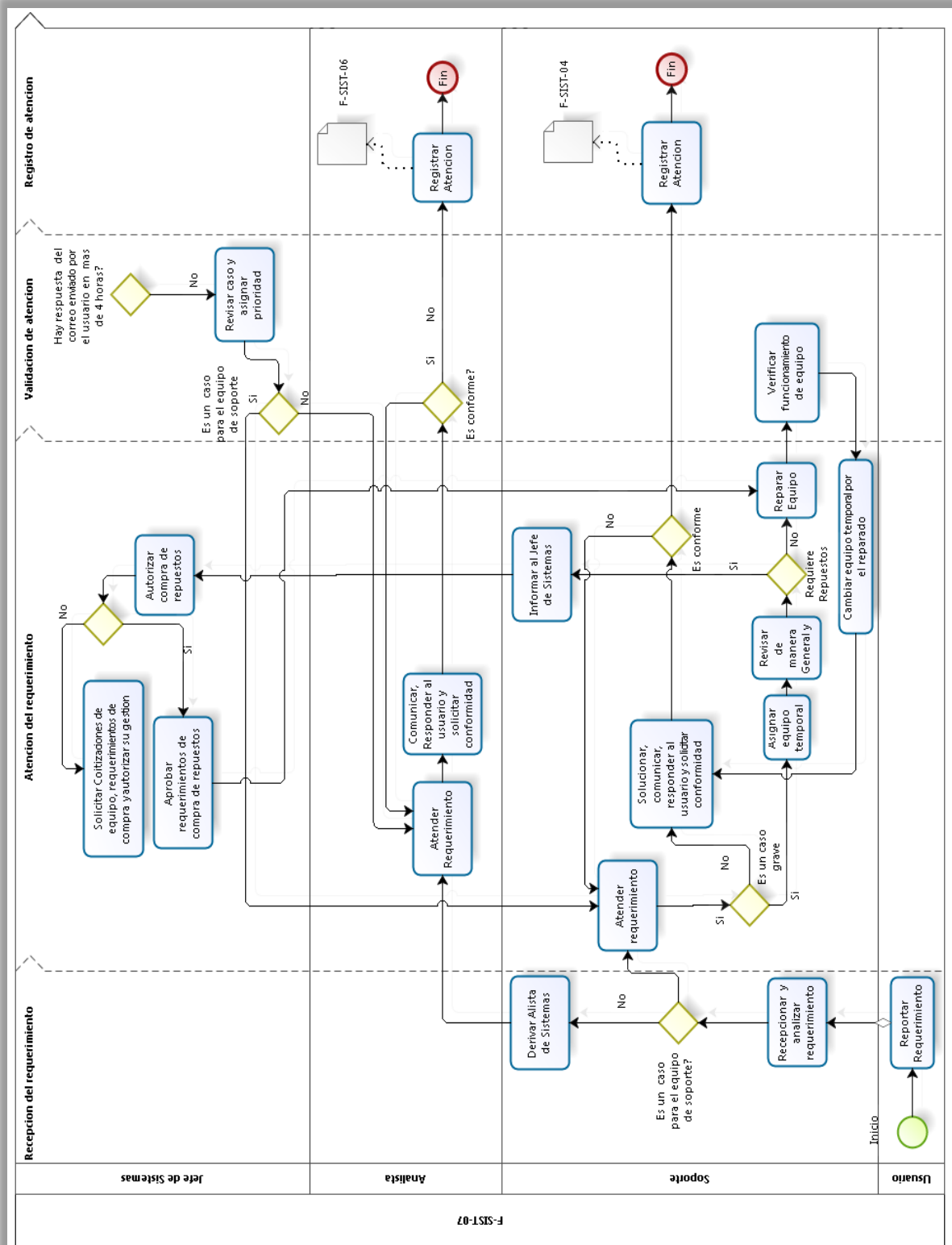
correspondiente al Gerente de RRHH el cual la recibe, para su aprobación, de no aprobarla, entonces responde el correo informando el porqué de su negativa pero de aprobarlo entonces envía el detalle del correo a crear a sistemas el cual procederá a crear la cuenta para que posteriormente, el personal de Soporte configure la cuenta de correo electrónico en el programa Outlook del cliente, culminado la labor el personal de Soporte actualizará el Listado de Correos de la empresa.

1.3.3.5. Mantenimiento Correctivo de Equipos de Cómputo y Software de Gestión

Arjen, Jan, Axel, Mike (2008),” La gestión del problema parte de un incidente o de un conjunto de ellos, para determinar que lo provocó y resolverlo provisional o definitivamente”.

Este procedimiento permite brindar el soporte adecuado en cuanto al mantenimiento correctivo de los equipos de cómputo y del sistema de gestión con los que trabaja la empresa, inicia cuando el Equipo de Cómputo o el Sistema de Gestión presenta algún inconveniente, el usuario solicita al área de sistemas su atención, si el requerimiento corresponde a un problema con el equipo de cómputo entonces soporte procederá a dar la solución al problema, si el tiempo para dar la solución es prolongado entonces se derivara a una atención personalizada para la cual soporte asignara un equipo de respaldo con toda la información que el usuario pueda necesitar y así poder internar el equipo defectuoso a taller sin perjudicar al cliente, ya en taller, soporte procederá a realizar una revisión general del equipo y evaluará si la falla acredita o no el cambio de repuestos para su gestión de compra, una vez culminará la reparación del equipo soporte comunicará al usuario por correo que el problema fue resuelto y coordinara el momento ideal para efectuar cambio de equipo, finalizada la labor Soporte actualizará el formato “Reporte de Mantenimiento Correctivo”. En caso de tratarse de un software de gestión, el analista de sistemas solucionará el problema y enviará un correo al usuario informándole la causa del inconveniente y un resumen de la labor efectuada, para posteriormente llenar el formato “Reporte de Atención de Requerimientos”.

Figura 13: Diagrama del Proceso de mantenimiento correctivo de la Compañía Naviera Natalia S.A.C.



Fuente: Elaboración propia

1.3.4. Metodologías de Desarrollo

1.3.4.1. Scrum

Scrum es una metodología ágil que permite una gestión evolutiva de avances en lugar de la tradicional, formulado bajo la necesidad de la adaptación continua, cumplimiento de fechas y costos, busca la colaboración constante del cliente e interacción de los individuos para dar una respuesta inmediata ante la solicitud de un cambio por parte del cliente por encima de un seguimiento de plan o documentación exhaustiva se basa en 12 principios de manifiesto ágil los cuales comprende la satisfacción del cliente generado por entrega oportuna y continua de software, esto representa la principal prioridad; el cambio en los requisitos del cliente, aunque se halla sido solicitado en pleno desarrollo; la entrega periódica de tiempos breves de software funcional; la colaboración cotidiana entre el personal del negocio y los desarrolladores; la generación de confianza y motivación a partir de oportunidad y respaldo que se le brinda a cada persona para que realicen su labor; la efectividad y eficiencia en la comunicación a través de una conversación directa en entre los miembros del equipo; la concepción de tomar la entrega de software funcional como principal indicador de progreso; el desarrollo sostenido entre todos los participantes del proyecto; la búsqueda de la excelencia técnica a través de la atención continua; la simplicidad para reducir la carga laboral como esencia; el surgimiento de mejoras en el desarrollo consecuente a la auto organización del equipo; la concientización del equipo de la forma en cómo se están realizando las tareas asignadas para realizan los ajustes posteriores y mejorar su efectividad de manera iterativa (Menzinsky, López y Palacio 2016, p.11).

El scrum técnico está conformado por los roles, artefactos y eventos, dentro de los roles tenemos al equipo scrum, al dueño del producto, al scrum master y a los interesados; por otro lado, los artefactos están conformados por la pila del producto, pila del sprint y el incremento; mientras que los eventos están conformados por el sprint; la reunión para la planificación del sprint; el scrum diario, las revisiones del sprint y su retrospectiva, más adelante se describirá cada uno de los artefactos mencionados.

1.3.4.1.1. Roles

1.3.4.1.1.1. Los interesados

Es el rol que incluye al resto de personas tales como usuarios, comerciales, proveedores o cualquier otro, que de una forma u otra estén implicado en el producto, se encargan de asesorar y observar el producto.

1.3.4.1.1.2. El propietario del producto

Es la persona designada por el cliente para su representación, tiene como principal responsabilidad recoger las necesidades y requerimientos de la organización, conocer el negocio del cliente y dar a conocer tanto al scrum master como al equipo de trabajo esta información. El propietario del producto establecerá las prioridades de la pila del producto y las funcionalidad que debe cubrir, participara en las reuniones en donde dará su conformidad a la finalización de un sprint o en caso contrario genera una retrospectiva para la mejora del producto, es responsable de dar valor al producto pues conoce los alcances del producto, la inversión que se está realizando y el retorno de inversión, tiene la capacidad de tomar decisiones alineadas a los requerimientos del cliente.

1.3.4.1.1.3. Scrum Master

Su principal función es facilitar las tareas del equipo de trabajo, resuelve cualquier problema que pueda retrasar al equipo de trabajo, como por ejemplo las dudas sobre las tareas que tienen que llevar a cabo los desarrolladores o el ambiente de trabajo, tiene participación en cada una de las fases del scrum está implicado en todas las fases de la metodología scrum, lidera de manera servicial al equipo de trabajo, trabaja directamente con el propietario del producto, organiza las tareas, coordina y realiza una reunión antes de iniciar cada sprint, es la persona intermedia, verifica y valida la pila de producto, mantiene una dinámica positiva y cooperativa en el grupo, brinda la asesoría necesaria tanto al equipo de trabajo como al propietario del producto.

1.3.4.1.1.4. El equipo Scrum

Lo conforman un grupo de personas profesionales que trabajan en conjunto de manera colaborativa, respetando las ideas y los aportes de cada miembro,

multifuncionales en sus labores, responsables del incremento de cada sprint asignado al equipo; son los que desarrollan el producto y tienen la capacidad de auto organizarse para lograr un propósito común, que es el superar la expectativa del cliente; responden en conjunto; su jerarquía organizacional es lineal, todos toman decisiones y contribuyen en las mejoras, puesto que cada miembro comprende y conoce lo que el propietario del producto requiere. Como una buena práctica es necesario que los miembros del equipo estén ubicados en una sola oficina puesto que esto contribuye en la comunicación directa o en persona de cada miembro, y facilita el apoyo y soporte entre colaboradores.

1.3.4.1.2. Artefactos

1.3.4.1.2.1. Pila de producto

Representa el registro de los requerimientos del producto que el cliente representado por el propietario del producto da a conocer al scrum master y al equipo de trabajo para su elaboración, esta lista evoluciona en el desarrollo según la retrospectiva del propietario del producto. Si se tratase de un producto de software, esta lista la conformarían los requerimientos funcionales del software, su evolución comprendería la lista de mejoras y corrección de errores del sprint, el propietario del producto es la persona designada para ordenar y priorizar esta lista en función a los que aportan mayor valor al producto.

1.3.4.1.2.2 Pila del Sprint

Representa una lista de tareas que se realizarían para la construcción de las historias de usuario, un grupo de estas tareas conformaran un sprint, esta descomposición se realiza para asignar tareas a cada miembro del equipo de trabajo y así poder monitorizar el avance del desarrollo.

1.3.4.1.2.3. Incremento

Es una parte del producto que ya ha sido terminada, puede ser usada y documentada.

1.3.4.1.3. Eventos

Los eventos realizados durante el desarrollo del producto son:

1.3.4.1.3.1. Sprint

Es un evento que en el que se desarrolla, culmina, despliega y distribuye una parte del producto, la cual debe realizarse en un tiempo no mayor a 4 semanas.

1.3.4.1.3.2. Planificación del sprint

Es una reunión conducida por el scrum master, en el cual, asisten el propietario del producto y el equipo de trabajo, tiene una duración máxima de un día, se lleva a cabo al iniciar un nuevo sprint, se determinan el objetivo y los detalles de las funcionalidades, que este deberá incorporar en base a la pila de producto presentada por el propietario del producto, los resultados obtenidos de esta reunión son la pila del sprint, tiempo estimado de realización del siguiente sprint, si se tratase del primer sprint, el scrum master podría presentar el avance del producto con el último incremento y su dato de velocidad de desarrollo del equipo de trabajo, para poder estimar de manera razonable la cantidad de trabajos a realizar en el siguiente sprint.

1.3.4.1.3.3. Scrum diario

Es una reunión que se realiza todos los días entre los miembros del equipo y el scrum master, siendo responsabilidad el equipo, establecer esta reunión, tiene una duración máxima de 15 minutos, en esta reunión se muestra el grafico de avance del incremento, explica las labores realizadas, las tareas que se tienen que realizar hasta la próxima reunión y los obstáculos que encontraron cada miembro del equipo, como resultado de esta reunión, se obtendrá la pila del sprint, grafico burn-down actualizados y se identificara los posibles impedimentos y necesidades.

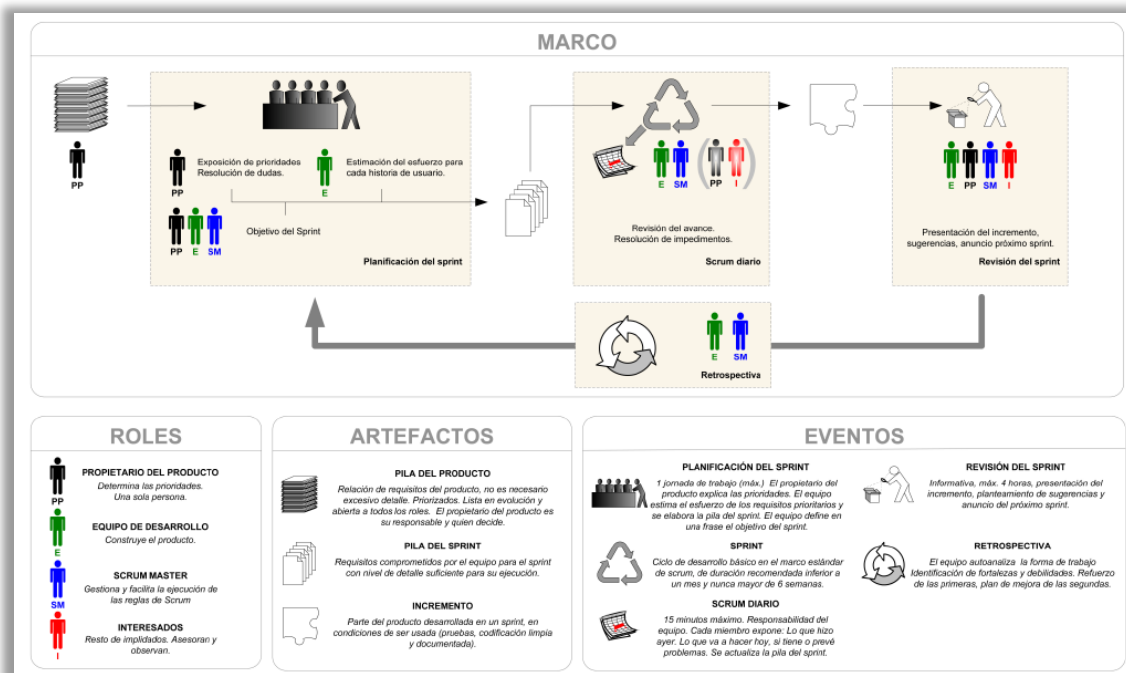
1.3.4.1.3.4. Revisión del Sprint

Es una reunión que se realiza al culminar un sprint para validar el incremento, tiene un tiempo de duración no mayor a 4 horas, asisten el dueño del producto, el scrum master y el equipo de trabajo, es el propietario del producto quien prueba el avance del sistema que ante su conformidad el equipo recibe un feedback para revisar la pila de producto y se establece la siguiente reunión.

1.3.4.1.3.5. Retrospectiva

Es una reunión que se realiza después de revisar del sprint y antes de la planificación del próximo sprint, en el que participan el equipo de trabajo y el scrum master, tiene un tiempo de duración de 1 a 3 horas, y se trata temas de como lo estamos trabajando concientizando en el equipo la forma en que se está realizando las tareas, de tal manera que sea capaz de reconocer sus fortalezas y debilidades en pos de una mejora continua.

Figura 14: Marco Scrum



Fuente: 2005-2015 – ScrumManager - <http://www.scrummanager.net>

Es la fase inicial del Scrum, corresponde a la lista de los requerimientos funcionales y/o no funcionales del producto, conocido también como historias de usuario que prácticamente son conjunto de requisitos del cliente, ya sean funcionales, de mejora, de tecnología o corrección de errores todo ello se incorpora al producto y se prioriza de acuerdo al valor que aportan al negocio, en la segunda fase es la planificación del sprint, el cual por medio de una reunión guiada por el responsable del funcionamiento del marco scrum a la que pueden asistir todo personal implicado en el proyecto, se determinan en base a las necesidades y prioridades del cliente, cuáles van a ser las funcionalidades que se incorporan al producto de tal manera que se obtendrá como resultado de esta fase una pila de sprint con la duración,

fecha próxima reunión y objetivos establecidos, seguidamente en la tercera fase se realiza diariamente reuniones de un tiempo no mayor a los 15 minutos, en el que se sincroniza el trabajo y se establece un plan para las próximas 24 horas, la cuarta fase abarca la revisión del sprint que es un reunión al finalizar el sprint realizado para comprobar el incremento no debe tener un tiempo mayor a 4 horas de tratarse de spint largos y 2 horas para sprint de 1 a 2 semanas y como quinta fase la retrospectiva que es una reunión realizada después de revisar cada spint y antes de la reunión de planificación del próximo día con la finalidad de conocer cómo lo estamos construyendo para que de esta manera podamos analizar los problemas y definir aspectos de mejora.

1.3.4.2. RUP (Proceso Unificado de Rational)

Kruchten sostiene:

Es un proceso de desarrollo utilizado en la ingeniera de software, que permite definir, implementar y distribuir aplicaciones de software, hace uso del Lenguaje de Modelamiento de Datos UML (Lenguaje Unificado de Modelado) como guía para la diagramación de los procesos y correcta implementación, su finalidad es integrar un producto de software con la estructura todos los procesos, asigna de forma disciplinada las tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo) y mide en la eficiencia de la organización. (2004).

1.3.4.2.1. Características de RUP

- Utiliza UML.
- Gramática bien definida.
- Terminología bien usada para definición de los procesos.
- Está dirigido por casos y usos.
- Centrado en su arquitectura.
- Iteración incremental.
- Asignar las tareas y responsabilidades de manera disciplinada.
- El desarrollo del software es iterativo.
- Control de cambios.
- Presenta vistas del modelo de software.
- Revisión de calidad del software.

1.3.4.2.2. Definición de UML

El Lenguaje Unificado de Modelado es una herramienta que permite plasmar la idea de un sistema, documentarlo y presentarlo a los implicados en el proceso de desarrollo, esto se realiza a través de un conjunto de símbolos representativos y diagramas tales como, los orientados a objetos, clases, casos de uso, estado, secuencias, actividades, componentes, distribución, colaboraciones, entre otros, para facilitar la comprensión del negocio, requerimientos y sistema a desarrollar a los programadores.

Para la gestión del proyecto, la metodología RUP proporciona una solución disciplinada, como tareas y responsabilidades señaladas dentro de una organización de desarrollo de software.

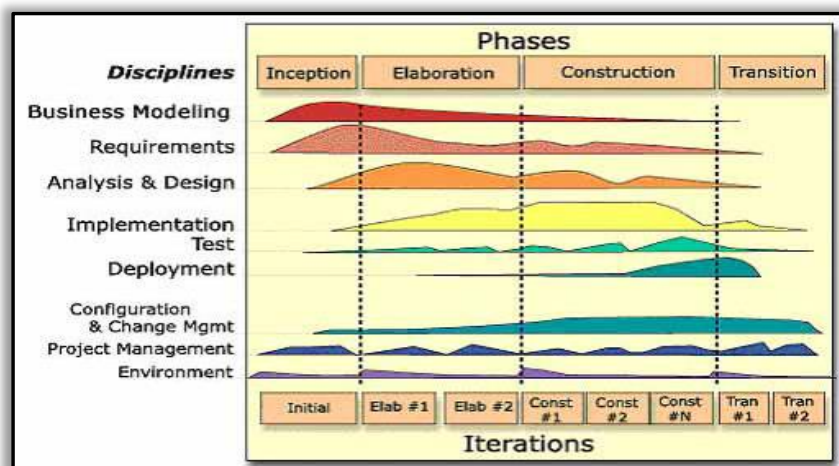
1.3.4.2.3. Disciplinas del RUP

- **Modelado del negocio:** Analizar y entender cuál son las necesidades del negocio para el desarrollo de software.
- **Análisis de requisitos:** Proveer una base para los costos y estimación de los tiempos para el sistema a desarrollarse.
- **Análisis y diseño:** Todo lo que en su momento se analizado anteriormente a un sistema automatizado y desarrollar las arquitecturas necesarias para el sistema.
- **Implementación:** Crea el software alineado a la arquitectura del diseño, cumpliendo las necesidades del cliente.
- **Test:** Asegurarse que tenga el comportamiento que se desea el sistema y a las exigencias del cliente.
- **Distribución:** Producir distribuciones para que sean distribuidos a los usuarios.
- **Gestión de configuración y cambios:** Guardar todas las versiones que se tienen del proyecto.
- **Gestión del proyecto:** Administración de tiempos y de recursos para el proyecto.
- **Gestión del entorno:** Administración de los ambientes de desarrollo.

1.3.4.2.4. Fases del RUP

- **Fase de Incepción:** Definir y acordar los alcances del Proyecto y los participantes.
- **Fase de Elaboración:** Definir la base en la cual va estar el sistema y los casos de uso, y el primer análisis acerca del problema.
- **Fase de Construcción:** En esta etapa se clarifican los requerimientos pendientes y se administran los acuerdos de los cambios para el sistema.
- **Plan de Transición:** Que el software que s diseño esté disponible para los usuarios con los ajustes de los errores que se han encontrado.
- **Evolución:** Contiene los procesos anteriores, para las nuevas mejoras del producto.

Figura 15: Fases y disciplinas de la metodología RUP



Fuente: IBM Developerworks

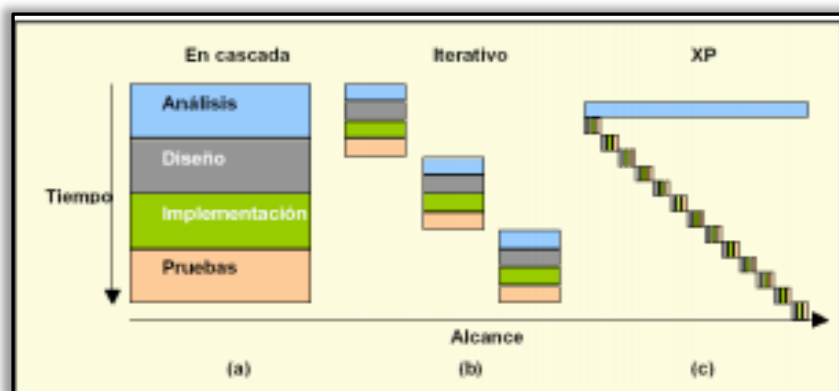
1.3.4.3. Metodología XP

“La programación extrema se apoya en cinco valores, los cuales enfatizan la esencia colaborativa del equipo, estos valores son la comunicación, simplicidad, retroalimentación, respeto y coraje” (Bahit, 2012, p.70).

Es una metodología de desarrollo de software formulado por Kent Beck , da mayor énfasis a la adaptabilidad a los cambios de los requisitos durante el desarrollo del software, busca la simplicidad tanto en la documentación como en el diseño, pero agiliza su desarrollo y facilita su mantenimiento, mantienen una comunicación fluida con el cliente puesto que forma parte del equipo desarrolladores el cliente quien

establece las prioridades de sus requerimientos, al estar integrado el cliente en el proyecto será capaz de realizar las pruebas unitarias de una parte del sistema terminado y su opinión permitirá la retroalimentación en la codificación para corregir algunas falencias. Este método consta de fases las cuales son: Exploración en la que los clientes planifican sus requerimientos que son de mayor interés para la primera entrega del producto, aquí también los desarrolladores se familiarizan con las herramientas que van a desarrollar para el desarrollo del software, se prueba la tecnología de tal manera que se va construyendo un prototipo, la siguiente fase es la Planificación de la Entrega es donde el cliente asigna la prioridad de cada requerimiento del usuario y los programadores estipulan el costo del esfuerzo, en la siguiente fase que es la de Iteraciones se elabora varias iteraciones con una duración de tres semanas como máximo, al finalizar la última iteración el sistema estará listo para la fase de la producción, en la fase de la Producción, se realiza pruebas adicionales tanto de funcionamiento como de rendimiento antes que el sistema sea entregado al entorno cliente, de acuerdo a los resultados de estas pruebas se determinará si será necesario realizar ajustes, en la fase de Mantenimiento se da importancia a la mantenibilidad del sistema de tal manera que este en óptimo funcionamiento, esta fase puede requerir contratar nuevo personal; dentro del equipo y cambios en su estructura en la última fase que es la fase de Muerte del proyecto es cuando se cumplen con todas las historias del cliente y este satisfecho con el sistema, pero la muerte del proyecto puede ocurrir también cuando no alcanza el presupuesto o no cumple con la expectativa del cliente.

Figura 16: Fases de la metodología XP



Fuente: Borja, Evolución de los largos ciclos de desarrollo en cascada (a) a ciclos iterativos más cortos (b) y a la mezcla que hace XP.

A continuación, se mostrará la tabla de los resultados de la evaluación de los expertos de la metodología a utilizar.

Tabla 3: Evaluación de Metodologías

Experto	Metodología		
	Scrum	XP	RUP
Mg. Sáenz Apari Abrahán Rafael	36	24	31
Mg. Chumpe Agesto, Juan Brues	39	35	32
Mg. Pacheco Pumaleque, Alex Alberto	40	30	30

Fuente: Evacuación propia

1.3.5. Indicador Índice de Cobertura de Mantenimiento

Chávez (2010, p.12). “Es una inspección que se programa cada cierto tiempo para asegurarse que los equipos y las máquinas estén operativas y en buenas condiciones”.

1.3.5.1. Cobertura de Mantenimiento

Díaz y Ruiz (2012, p.207), “El índice de cobertura es el porcentaje del total de equipos con mantenimiento sobre el total de equipos a realizar el mantenimiento preventivo y correctiva”.

Chávez (2010, p.108) La ecuación a usar es la siguiente:

$$ICM = \frac{EPR}{TEP} * 100\%$$

Dónde:

ICM = Índice de cobertura de mantenimiento

EPR = Equipos con mantenimiento preventivo

TEP = Total de equipos programados

1.3.6. Grado de Cumplimiento

ITIL V3 (2008), Define que tiene que ver con la conclusión de una tarea. Los indicadores de cumplimiento están relacionados con las razones que indican el

grado de consecución de tareas o trabajos, Ejemplo: cumplimiento del programa de pedidos.

ITIL V3 (2008) Se emplea la formula siguiente:

$$GC = (RET/ RE) * 100 \%$$

Dónde:

GC: Grado de Cumplimiento

RET: Entregados a tiempo

RE: Esperados

1.4. Formulación de problema

1.4.1. Problema Principal

P.G: ¿De qué manera un sistema web influye en el control de servicios del área de Sistemas en la Compañía Naviera Natalia S.A.C.?

1.4.2. Problemas Secundarios

PE1: ¿De qué manera un sistema web influye el índice de cobertura de los mantenimientos preventivos realizados por el área de sistemas en la Compañía Naviera Natalia S.A.C.?

PE2: ¿De qué manera un sistema web influye en grado de cumplimiento de las atenciones realizadas por el área de sistemas en la Compañía Naviera Natalia S.A.C.?

1.5. Justificación de estudio

Garrido sostiene:

Hoy en día las organizaciones están tomando conciencia de la importancia del uso de las tecnologías de información y su capacidad para dar seguimiento a través del ciclo de vida de cada elemento. Sin un adecuado sistema que gestione estos dispositivos y servicios TI, las empresas se afrontan a riesgos que podrían afectar su operatividad (2013, p.1).

Bon, Jong, Kolthof, Pieper, Tjassing, Der, Verheijen (2008, p.60), “La automatización del servicio puede tener un efecto considerable sobre el rendimiento

de Activos del Servicio como la gestión, la organización, las personas, los procesos, el conocimiento y la información”.

Con la implementación de un sistema web, se podrá gestionar de manera centralizada estos elementos TI. y a la vez que aportará una base de conocimiento que contendrá registros de los servicios realizados y los recursos asignados.

1.5.1. Justificación Económica

Para la justificación económica, AENOR sostiene:

Una efectiva administración de nivel de servicio TI, mantendrá la disponibilidad de los elementos TI funcionando de manera correcta, lo que beneficia a la organización con ahorros económicos y un rápido retorno de inversión. Los usuarios colaboradores podrán realizar sus labores sin interrupciones, minimizando así los riesgos. Y en lo que se refiere al área TI, serán más asertivos en la resolución de incidencias y problemas gracias al catálogo de conocimiento registrado por cada atención. (2009, p.235).

A implementar el sistema y hacer uso de ella se dispondrá de un catálogo de conocimiento que permita al personal de sistemas solucionar un problema con mayor fluidez y en el menor tiempo posible, de tal manera que se evitará un gasto en horas hombre por la inoperatividad extendida de sus equipos informáticos, se optimizaría el tiempo empleado en el llenado de registros para las auditorías, este tipo de regularización por lo general tomaba más de una semana en realizarlo y demandaba más personal de sistemas.

1.5.2. Justificación Institucional

El sistema web para el control de servicios del área de sistemas permitirá a la organización, mejorar la relación existente entre los usuarios internos y su área de sistemas, permitiéndoles integrar esta aplicación en las otras sedes institucionales pertenecientes al grupo empresarial, mejorando y enriqueciendo su catálogo de conocimientos de soluciones efectivas de problemas reportados para una rápida solución que permita mantener continuidad de la productividad administrativa de la organización y su buena imagen constitucional.

Bestratén Belloví sostiene:

Hoy en día, tanto las tecnologías TI, como los nuevos sistemas de gestión están constante cambio, las organizaciones toman conciencia de ello y planifican la forma de adaptarse a ello para mantenerse eficientes y competitivos (2017, p.1).

1.5.3. Justificación Operativa

La gestión de incidencias describe procedimientos estandarizados y fáciles de entender, su implementación contribuye a una rápida atención y monitorización del cumplimiento de los objetivos corporativos. (Gómez, 2012).

Es por ello que el siguiente proyecto permitió realizar un estudio en el control de servicios del área de sistemas para dar soporte y la solución oportuna a las fallas que se presentan en los equipos informáticos en pleno desarrollo de las actividades de la empresa.

1.6. Hipótesis

1.6.1. Hipótesis general

HG: El sistema web influye en el control de servicios del área de sistemas en la Compañía Naviera Natalia S.A.C.

1.6.2. Hipótesis específicas

HE1: El sistema web incrementa el índice de cobertura del servicio de mantenimiento preventivo en la Compañía Naviera Natalia S.A.C.

HE2: El sistema web incrementa el grado de cumplimiento de las atenciones realizadas por el área de sistemas en la Compañía Naviera Natalia S.A.C

1.7. Objetivos

1.7.1. Objetivo General

OG: Determinar la influencia de un sistema web en el control de servicios del área de sistemas en la Compañía Naviera Natalia S.A.C.

1.7.2. Objetivos Específicos

OE1: Determinar la influencia de un sistema web en el índice de cobertura del servicio de mantenimiento preventivo realizado por el área de sistemas en la Compañía Naviera Natalia S.A.C.

OE2: Determinar la influencia de un sistema web en el grado de cumplimiento de las atenciones realizadas por el área de sistemas en la Compañía Naviera Natalia S.A.C

Método

II. MÉTODO

2.1. Diseño de investigación

SAMPIERI, Hernández (2014), “El término diseño se refiere al plan o estrategia concebida para obtener la información que se desea con el fin de responder al planteamiento del problema”.

La siguiente investigación es de tipo aplicada, pues utiliza los conocimientos teóricos para resolver un problema concreto. Asimismo, se usa un diseño pre experimental, con medida antes y después de la intervención de grupo único, pues se manipuló intencionalmente una variable (variable independiente), en este caso, el sistema web, para analizar el efecto en otra variable (variables dependientes), asumiendo este rol la variable control de calidad de sistemas. Se realizó un pre-test para evaluar el estado inicial de la variable dependiente, luego se aplicará el sistema web (tratamiento o variable independiente) a un solo grupo (usuarios internos de la compañía) y posteriormente se aplicó el post-test para medir el efecto que tuvo la variable independiente. El diagrama de este diseño es el siguiente:

G 01 X 02

G: Grupo

O1: Pre-prueba

X: Tratamiento o variable independiente (Sistema web)

O2: Post-prueba

2.2. Variables, operacionalización

2.2.1. Definición Conceptual

2.2.1.2. Variable Independiente (VI): Sistema Web

Kendal, Kendal (2011), define un Sistema web como un conglomerado de etapas y procesos que intervienen sobre un grupo de datos estructurados según los requerimientos de una organización, con el fin de seleccionar, construir, transformar y disponer la información requerida para las operaciones de dicha

organización, brindando beneficios como la posibilidad de acceder las 24 horas del día, mejora la utilidad y capacidad del uso de la interfaz y poder expandirse globalmente en vez de permanecer en un entorno local

Nuestro sistema web será desarrollado con C# y usará motor de base de datos SQL 2008 R2.

2.2.1.3. Variable Dependiente (VD): Control de Servicio

ITIL V3 (2008, p.31) El control de servicio consiste en un ciclo continuo de monitorización, comunicación y acción. Esto es muy importante para la provisión, soporte y mejora de servicios. El control de servicios mantiene, controla, administra y supervisa la infraestructura de la información, va a incluir actividades como la ejecución y monitorización de los tratamientos por lote, planificación de los trabajos, supervisión de las redes de seguridad, restauraciones y almacenamiento, mantenimiento preventivo y correctivo del equipamiento de la infraestructura.

Es por ello que para esta investigación elegimos 2 indicadores lo cual es el índice de cobertura de mantenimiento y el grado de cumplimiento porque implica la culminación de una actividad. Los indicadores de cumplimiento se vinculan con las razones que expresan el nivel de logro de actividades o tareas.

2.2.2. Definición Operacional

2.2.2.1. Variable Independiente (VI): Sistema Web.

Es un mecanismo que propicia el registro de solicitudes y atenciones requeridos por los clientes internos hacia el área de sistemas, facilitando el seguimiento del estado de las solicitudes de los clientes por medio de alarmas de estado, programadas para la notificación oportuna al personal de sistemas, también permite la creación de una base de conocimiento centralizado, en el que el personal de sistemas podrá consultar las soluciones brindadas a una incidencia o problema ya reportado, permitiéndole cerrar el caso de atención en un tiempo cada vez más reducido, permitirá la diversificación de los datos requeridos por la Compañía Naviera Natalia S.A.C., en busca del control oportuno de los servicios que brinda el área de sistemas.

2.2.2.2. Variable Dependiente (VD): Control de servicios

Es una manera sistematizada de monitorizar los indicadores, filtrar y categorizar los eventos con la finalidad de mejorar la satisfacción de los clientes internos en cuanto al cumplimiento oportuno de los servicios solicitados, para la medición del grado de satisfacción de los usuarios internos respecto al servicio brindado por el personal de sistemas.

A continuación, se mostrará una tabla con los resultados de la evaluación de los expertos referente a los instrumentos utilizados.

Tabla 4: Validación de instrumentos

Experto	Instrumentos	
	Índice de cobertura de mantenimiento	Grado de Cumplimiento
Mg. Sáenz Apari Abrahán Rafael	83%	83%
Mg. Chumpe Agosto, Juan Brues	50%	80%
Mg. Pacheco Pumaleque, Alex Alberto	84%	84%

Fuente: Elaboración propia

2.3. Población y muestra de estudio

2.3.1. Población

SAMPIERI (2006) refiere que la población constituye el conjunto de casos que reúnen una serie de características que son objeto de la investigación.

La población para la presente investigación está conformada por el número de fichas de registro para la evaluación de mantenimiento preventivo de equipos, siendo está 20 para el índice de cobertura de mantenimiento, mientras que el Indicador grado de cumplimiento está representada por 20 fichas.

SAMPIERI (2014), afirma que la muestra representa subconjunto de la población objeto de estudio. Es sobre este subconjunto de donde se obtendrán los datos, por lo cual deberá de especificarse en sus características con anticipación y precisión, de tal forma que representen a la población.

$$n = \frac{NZ^2p(1 - p)}{Ne^2 + Z^2p(1 - p)}$$

Dónde:

n: muestra

N = Número de elementos para la población

Nivel de Confianza = 95% → Z = 1.96

e = (error muestra admisible) = 0.05

p = 0.5

Si $N \leq 30$ entonces

$$n = N$$

Muestra para el indicador índice de cobertura de mantenimiento

$$n = 20$$

La muestra para el indicador índice de cobertura de mantenimiento será 20 casos. Como la muestra es reducida, se tomará como a la población en su totalidad

Muestra para el indicador Grado de Cumplimiento

$$n = 20$$

La muestra para el indicador grado de cumplimiento es de 20 reportes ya que la muestra es pequeña, se tomará como toda la población.

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad

La técnica empleada en esta investigación son documentos. El instrumento utilizado son las fichas de registros propios de la empresa. Estas fichas de registro contienen el número de requerimientos que los usuarios internos envían de forma diaria al personal de sistemas, así como el número de respuestas entregadas por dicho personal, además contiene las fechas en las que se solicitaron los requerimientos y las fechas en las que se entregaron los mismo

2.5. Métodos de análisis de datos

Para el procesamiento de los datos se emplearán métodos estadísticos descriptivos como media aritmética, desviación estándar y varianza. Para comprobar la hipótesis se usó la prueba paramétrica t de student para una misma muestra, dado el nivel de medición de las variables, y se usará un nivel de significancia de 0.05.

Para Hernández, Roberto, Fernández, Carlos y Baptista (1998, p.243) el método de análisis de datos reúne información empírica de fenómenos objetos de cuantificación, y, en congruencia, tiene como producto números.

El método de análisis de datos usado en el estudio es la estadística inferencial, la misma que implica procedimientos para inferir valores numéricos específicos a partir de una muestra pequeña a la población.

Gómez (2006) Indica que la prueba t de student es una prueba estadística para determinar posibles diferencias estadísticamente significativas entre los puntajes promedios de dos grupos.

Por su parte, Gómez (2006) refiere que la prueba de Shapiro Wilk representa una de las pruebas de uso frecuente para corroborar si los puntajes de una determinada variable se ajustan a una distribución teórica en específico, siendo para efectos del estudio, la distribución normal.

Si $n < 50$ -> Pruebas Shapiro Wilk.

Si $n > 50$ -> Pruebas kolmogorov-Smirnov.

En relación con lo mencionado anteriormente, se utilizó la prueba de Shapiro Wilk ya que la muestra en esta investigación es menor a 50. Por otro lado, tal como indica Martínez si se sabe la varianza muestral y la muestra es menor a 30, en una prueba de hipótesis, entonces se utiliza la prueba t de Student .

2.5.2. Definición de Variables

Ia: Representa el valor del Indicador antes de la implementación del sistema web para el control de Servicios del área de sistemas en la compañía Naviera Natalia S.A.C.

Id: Representa el valor del Indicador posterior a la implementación del sistema web para el control de Servicios del área de sistemas en la compañía Naviera Natalia S.A.C.

2.5.3. Hipótesis Estadística

2.5.3.1. Hipótesis Específica 1 (HE1): El sistema web incrementa el índice de cobertura de los mantenimientos preventivos realizados por el área de sistemas en la compañía naviera Natalia S.A.C.

Variables:

- **ICa:** Índice de cobertura medido previo a la implementación del sistema web.
- **ICd:** Índice de cobertura medido posterior a la implementación del sistema web.

2.5.3.1.1. Hipótesis Nula (H0): El sistema web no incrementa el índice de cobertura de los mantenimientos preventivos realizados por el área de sistemas en la compañía naviera Natalia S.A.C.

$$H_0: ICa \geq ICd$$

2.5.3.1.2. Hipótesis Alternativa (HA): El sistema web incrementa el índice de cobertura de los mantenimientos preventivos realizados por el área de sistemas en la compañía naviera Natalia S.A.C.

$$HA: ICa < ICd$$

2.5.3.2. Hipótesis Específica 2 (HE2): El sistema web incrementa el grado de cumplimiento de las atenciones realizadas por el área de sistemas en la compañía Naviera Natalia S.A.C.

Variables:

- **GCa:** Grado de cumplimiento medido previo a la implementación del sistema web
- **GCd:** Grado de cumplimiento medido posterior a la implementación del sistema web

2.5.3.2.1. Hipótesis Nula (H0): El sistema web no incrementa el grado de cumplimiento de las atenciones realizadas por el área de sistemas en la compañía Naviera Natalia S.A.C.

$$HO: GCa \geq GCd$$

2.5.3.2.2. Hipótesis Alternativa (HA): El sistema web aumenta el grado de cumplimiento de atenciones realizadas por el área de sistemas en la compañía Naviera Natalia S.A.C.

$$HA: GCa < GCd$$

Nivel de Significación

Nivel de significación (α): 0.05

Nivel de confianza ($\gamma = 1 - \alpha$): 0.95

2.5.4. Estadístico de Prueba

Pearson (2004), refiere que la prueba estadística es un valor calculado a partir de datos muestrales, que se utiliza para tomar la decisión sobre el rechazo de la hipótesis nula.

La prueba estadística es la prueba de T-Student ya que si las pruebas de hipótesis para el promedio aritmético (μ), cuando se comprende la desviación típica (σ), o el valor de la muestra es reducido (25 a menos); se determina a partir de:

$$t = \frac{\bar{x} - u}{\frac{S_x}{\sqrt{n}}}$$

n = Tamaño de muestra

\bar{x} = Media

u = Valor a analizar

S_x = Desviación Estándar

2.5.4.1. Varianza Muestral (S²)

Pearson (2004) "La varianza muestra se define como la "media" de los cuadrados de las desviaciones a la media muestral. La varianza muestral, denotada por S², se define como:"

$$S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n}$$

\bar{x} = Media muestral

X_i = Valores de la variable.

N =Tamaño de la población.

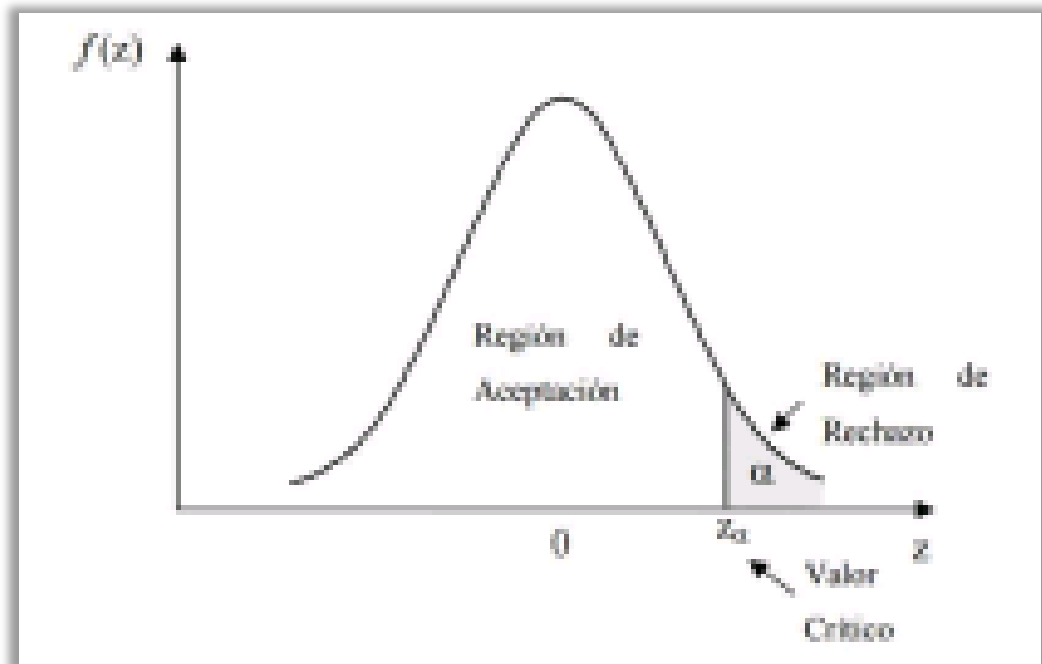
2.5.4.2. Desviación Estándar

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

Análisis de Resultado:

La gráfica de la distribución normal es de la siguiente manera:

Figura 17: Gráfica de la distribución normal



Fuente: León (2010).

Dónde:

RR: Es la región de rechazo.

RA: Es la región de aceptación.

2.6. Aspectos éticos

La siguiente investigación cumplirá con aspectos éticos tales como la participación voluntaria de los colaboradores, el respeto del anonimato, la confidencialidad de los datos propiciados por la empresa, el consentimiento informado verbal de los participantes y la adecuada citación de los aportes de otros investigadores.

RESULTADOS

III. RESULTADOS

3.1. Análisis Descriptivo

En el estudio se aplicó un sistema web para determinar el índice de cobertura de mantenimiento de equipos informáticos y el grado de cumplimiento de las atenciones solicitadas al área de sistemas; para ello se aplicó un Pre Test el cual permitió conocer las condiciones iniciales del indicador; después de la implementación del sistema web se analizaron nuevamente los datos, estos resultados se muestran en la tabla 6 y 7.

3.1.1. Indicador: Índice de cobertura de Mantenimiento Preventivo

Los resultados descriptivos del índice de cobertura se observan en la siguiente tabla:

Tabla 5: Medidas descriptivas del índice de cobertura de mantenimientos realizados antes y después de la implementación del sistema web

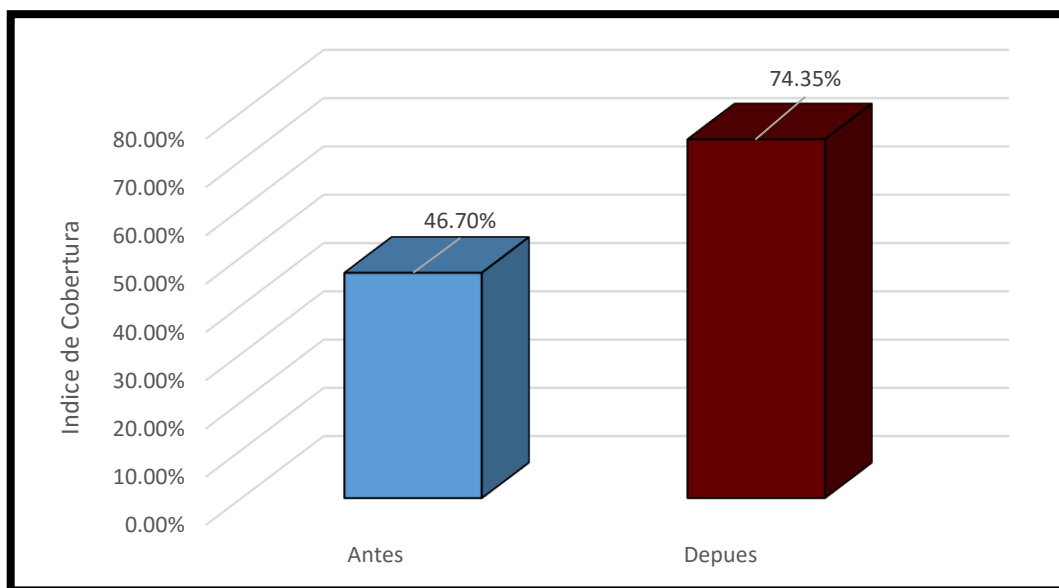
Variables	N	Mínimo	Máximo	Media	Desviación estándar
Índice _Cobertura_Pretest	20	0.24	0.73	0.4670	0.14124
Índice _Cobertura_Posttest	20	0.50	1.00	0.7435	0.14057
N válido (por lista)	20				

Fuente: Elaboración propia

Para el índice de cobertura de mantenimientos preventivos, el valor promedio de este índice en las pruebas realizadas antes de utilizar el sistema fue del 46.70% y después de usar el sistema este valor asciende a 74.35%; lo que da constancia, que hay una influencia positiva al utilizar el sistema, cabe destacar que el mínimo valor registrado en las pruebas realizadas antes de la implementación, fue del 24.00% y 50.00% posterior a la implementación.

En lo correspondiente a la desviación, tuvieron una variabilidad de 14.12% antes de la implementación del sistema web y un 14.05% después de la implementación.

Figura 18: Índice de cobertura de mantenimiento antes y después de la implementación del sistema.



Fuente: Elaboración propia

3.1.2. Indicador: Grado de cumplimiento de las atenciones realizadas

A continuación, se muestran los resultados descriptivos de las medidas del grado de cumplimiento de las atenciones realizadas.

Tabla 6: Medidas descriptivas del Grado de cumplimiento antes y después de la implementación del sistema web

Variables	N	Mínimo	Máximo	Media	Desviación estándar
Grado_Cumplimiento_PreTest	20	0.29	0.67	0.5080	0.09611
Grado_Cumplimiento_PostTest	20	0.64	1.00	0.8355	0.09534
N válido (por lista)	20				

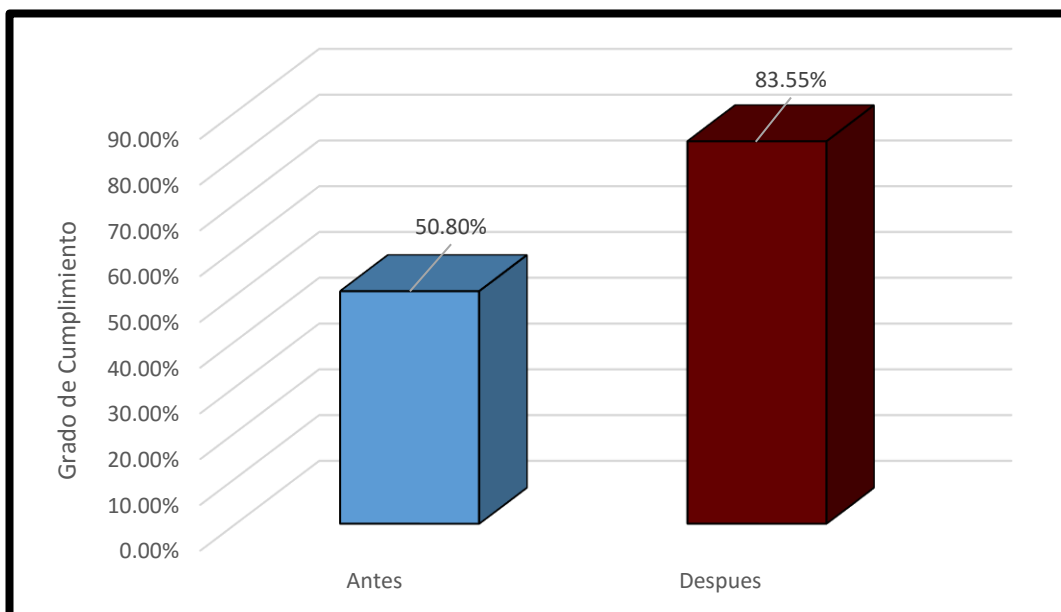
Fuente: Elaboración propia

Para el grado de cumplimiento de las atenciones realizadas, el valor promedio obtenido antes de usar el sistema, fue del 50.80%, mientras que el valor obtenido después de la implementación fue del 83.55%, lo que indica una considerable

diferencia; así mismo, se observa que el grado de cumplimiento mínimo fue del 29.00% antes, y del 64.00% después usar el sistema.

La dispersión del grado de cumplimiento de las atenciones realizadas, en el pre test tuvieron una variabilidad de 9.61%, en cambio en el post test fue de 9.53%.

Figura 19: Grado de cumplimiento antes y después de la implementación del sistema web



Fuente: Elaboración propia

3.2. Análisis Inferencial

3.2.1. Prueba de Normalidad

Considerando que el tamaño de la muestra estratificada está constituido por 20 registros para ambos indicadores (índice de cobertura de mantenimiento y grado de cumplimiento de las atenciones realizadas) y al ser esta cantidad menor que 50, se aplicara el método Shapiro-Wilk para calcular la prueba de normalidad tal como indica Hernández, Fernández y Baptista (2006, p. 376). Para efectuar esta prueba se utilizó un software para la estadística denominado SPSS versión 24.0, se insertaron los datos correspondientes a los registros obtenidos antes y después de la implementación del sistema web, se configuro el software para un nivel de confiabilidad del. 95%, el valor de la significancia determinara según el resultado

obtenido para cada prueba, si es que adopta o no una distribución normal, este resultado sigue las condiciones siguientes:

Si

Sig. < 0.05 adopta una distribución no normal.

Sig. \geq 0.05 adopta una distribución normal.

Dónde: Sig. Representa el nivel de significancia.

Los resultados obtenidos de las pruebas realizadas fueron los siguientes:

3.2.1. Indicador: Índice de cobertura de mantenimiento preventivo

Los datos obtenidos del registro de índice de cobertura, tanto antes como después de la implementación del sistema, fueron sometidos a la prueba de normalidad al 95% de confiabilidad y los resultados obtenidos de esta prueba a través del SSPS se muestran en la siguiente tabla:

Tabla 7: Test de normalidad del índice de cobertura de mantenimiento antes y después de implementado el sistema web

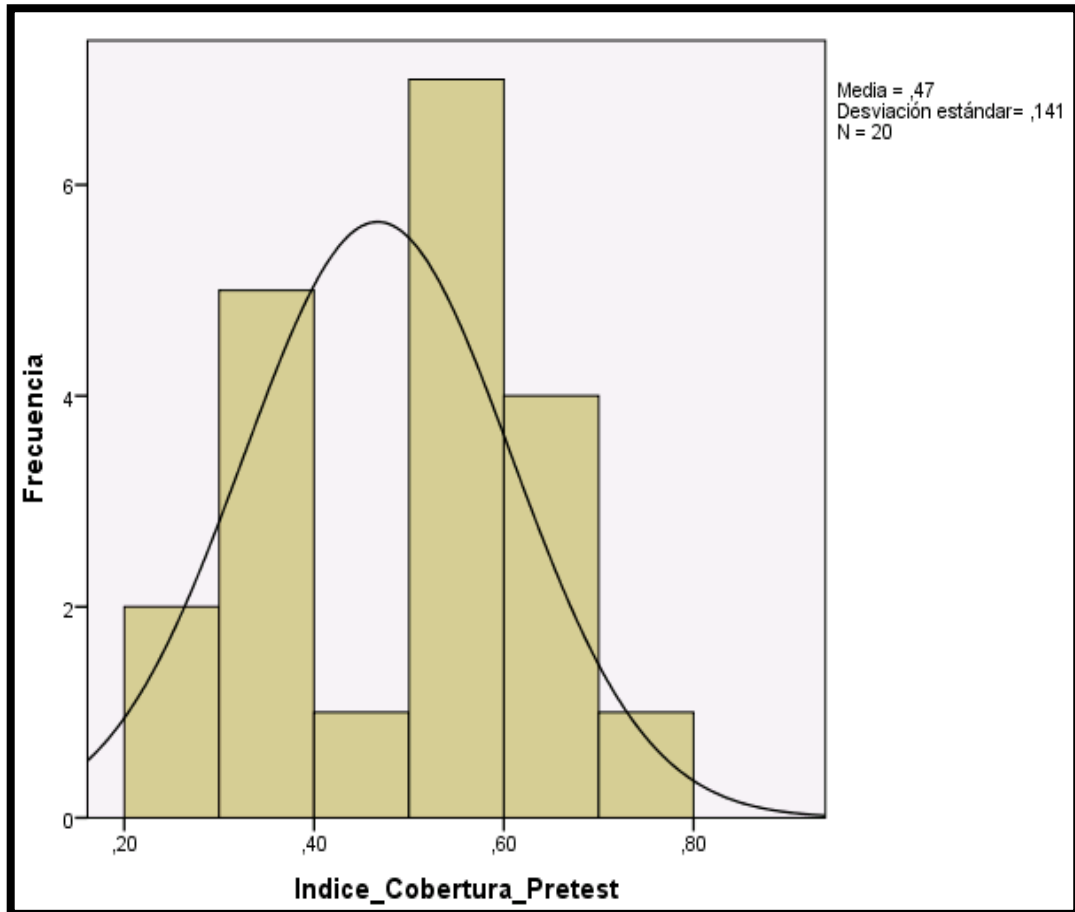
	Shapiro-Wilk		
	Estadístico	gl	Sig.
Índice_Cobertura_Prestest	0.937	20	0.210
Índice_Cobertura_Posttest	0.970	20	0.755

Fuente: Elaboración propia

Como se puede observar en los resultados obtenidos, el nivel de significancia (Sig.) correspondiente al índice de cobertura de mantenimiento preventivo fue 0.210 en las pruebas realizadas antes de la implementación, este resultado es mayor a 0.05, lo que determina, que el índice de cobertura se distribuye normalmente, continuando con la prueba se siguió el mismo procediendo para los datos obtenidos de los registros después de la implementación del sistema y la prueba arrojó un 0.755, como este resultado también es mayor que 0.05 se determina que también sigue la distribución normal. De esa manera se confirma la distribución normal

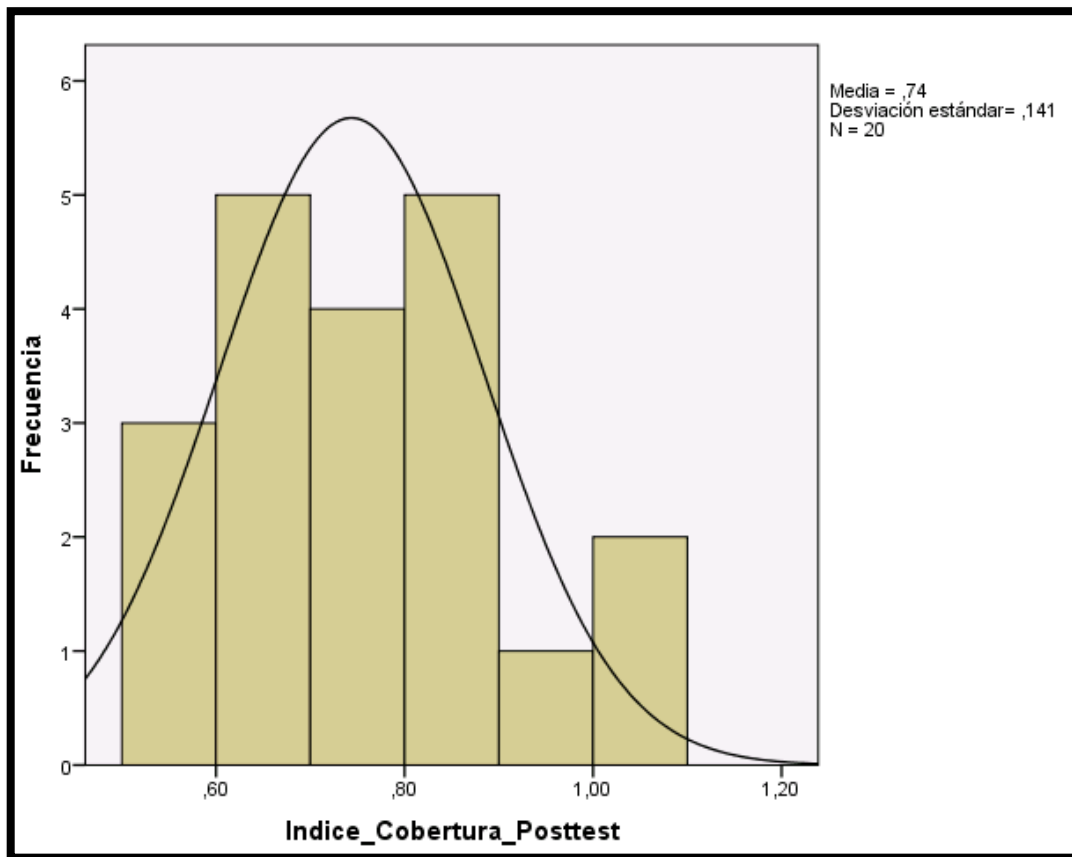
calculada a partir de los datos obtenidos de la muestra, estos resultados nos servirán para determinar el tipo de prueba de hipótesis que se aplicará para este proyecto de investigación, a continuación, se mostraran las gráficas de las pruebas de normalidad aplicadas antes y después de la implementación.

Figura 20: Test de normalidad del Índice de cobertura de mantenimiento antes de la implementación



Fuente: Elaboración propia

Figura 21: Test de normalidad del índice de cobertura de mantenimiento después de la implementación



Fuente: Elaboración propia

3.2.2. Indicador: Grado de cumplimiento

Los datos obtenidos del registro de grado de cumplimiento, tanto antes como después de la implementación del sistema, fueron sometidos a la prueba de normalidad al 95% de confiabilidad y los resultados obtenidos de esta prueba a través del SSPS se muestran en la siguiente tabla:

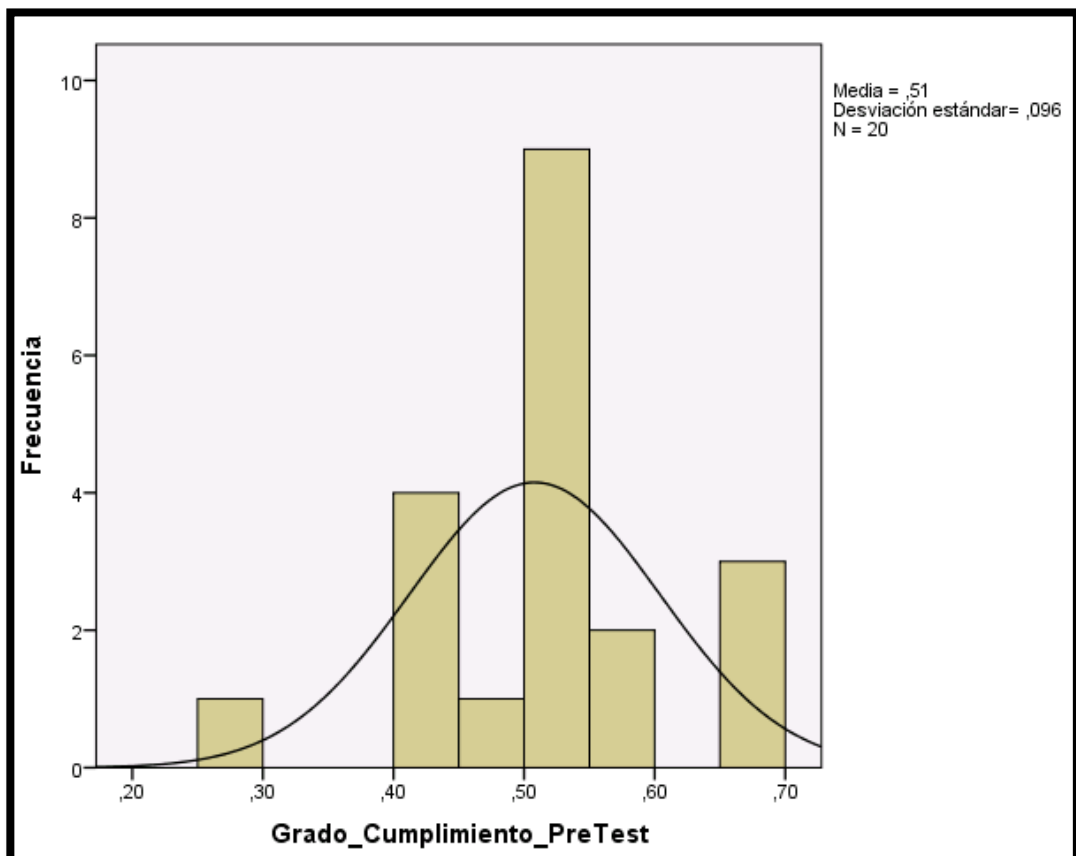
Tabla 8: Prueba de normalidad del grado de cumplimiento

	Shapiro-Wilk		
	Estadístico	Gl	Sig.
GradoCumplimiento_PreTest	0,181	20	0,086
GradoCumplimiento_PostTest	0,151	20	0,20

Fuente: Elaboración propia

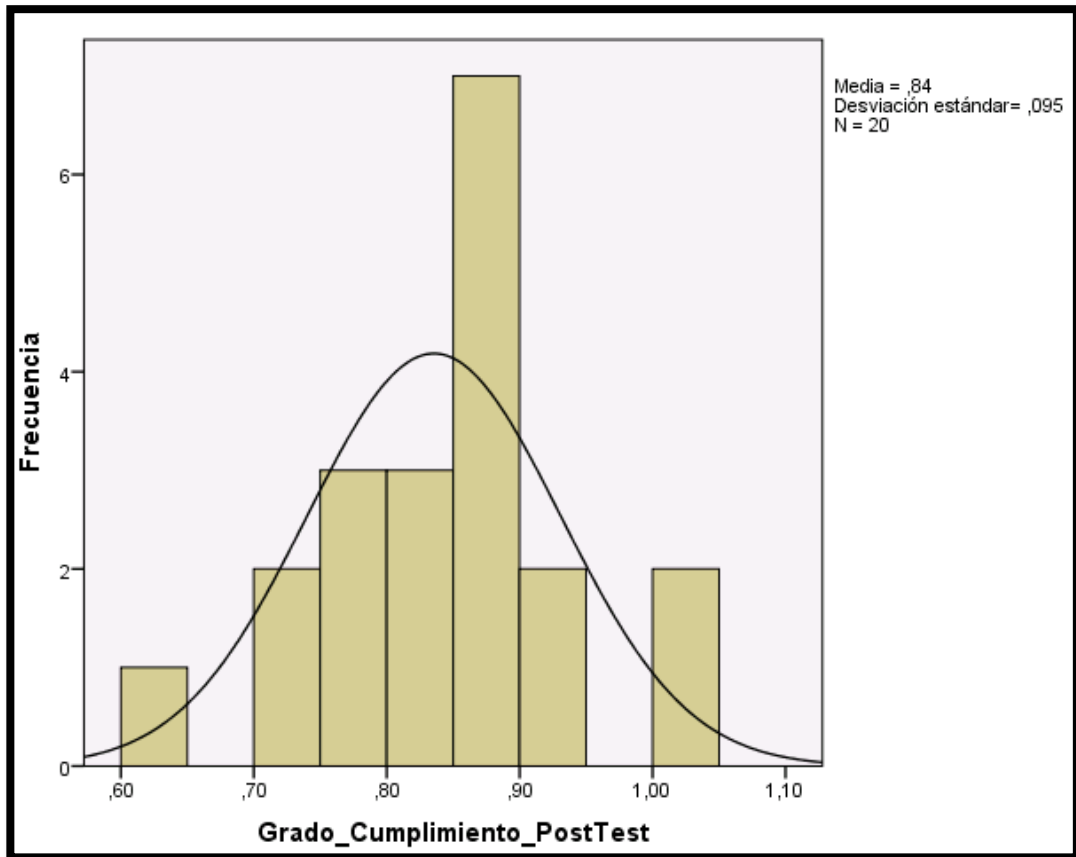
Como se puede observar en los resultados obtenidos, el nivel de significancia (Sig.) correspondiente al grado de cumplimiento de las atenciones realizadas fue de 0.086 en las pruebas realizadas antes de la implementación, este resultado es mayor que 0.05 e indica, que el índice de cobertura se distribuye normalmente, continuando con la prueba se siguió el mismo procediendo para los datos obtenidos de los registros después de la implementación del sistema y la prueba arrojó un 0.20, como este resultado también es mayor que 0.05 se determina que también sigue la distribución normal. De esa manera se confirma la distribución normal calculada a partir de los datos obtenidos de la muestra, estos resultados nos servirán para determinar el tipo de prueba de hipótesis que se aplicará para este proyecto de investigación, a continuación, se mostraran las gráficas de las pruebas de normalidad aplicadas antes y después de la implementación del sistema web.

Figura 22: Prueba de normalidad del grado de cumplimiento antes de la implementación del Sistema Web



Fuente: Elaboración propia

Figura 23: Prueba de normalidad del grado de cumplimiento después de la implementación del Sistema Web



Fuente: Elaboración propia

3.3. Prueba de Hipótesis

Considerando que todas las pruebas de normalidad realizadas a los dos indicadores siguen una distribución normal y que esto contribuye para un análisis paramétrico se utilizara el método T-Student para la prueba de la hipótesis a cada uno de estos indicadores, como sostiene Hernández et al. (2016, p. 453)

3.3.1. Hipótesis de Investigación 1:

- **H1:** El sistema web mejorará el índice de cobertura de mantenimiento preventivo realizado por el área de sistemas en la Compañía Naviera Natalia S.A.C.
- **Indicador:** Índice de cobertura

Definiciones de Variables:

A continuación se definen las dos variables utilizadas para este proyecto:

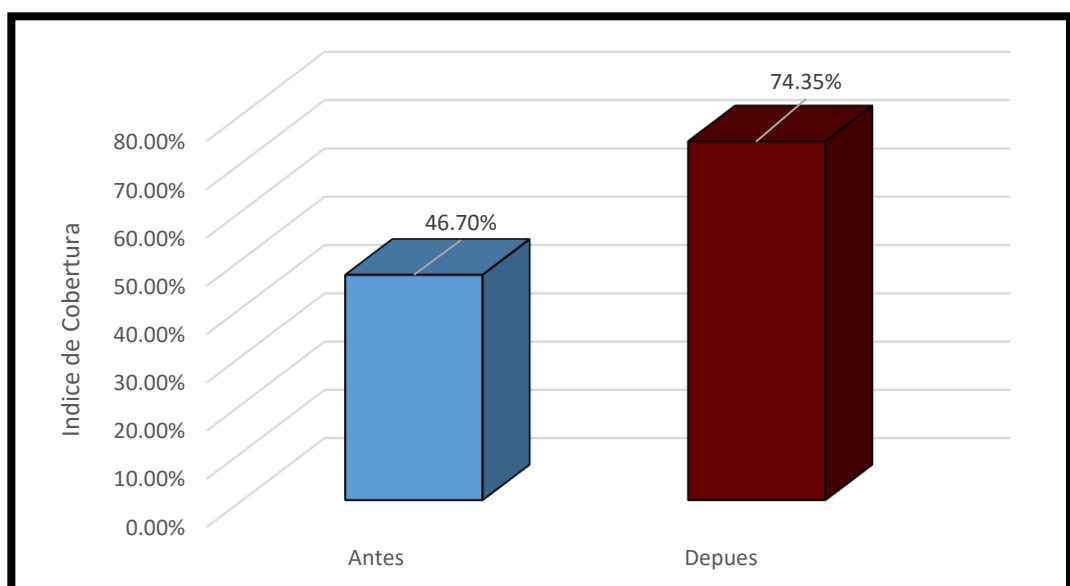
- **ICa** = Índice de cobertura antes de utilizar el sistema web.
- **ICd** = Índice de cobertura después de utilizar el sistema web.
- **H0:** El sistema web no incrementa el índice de cobertura de mantenimiento preventivo realizado por el área de sistemas en la Compañía Naviera Natalia S.A.C, lo que significa que sin el sistema web el indicador es igual o mejor que con el sistema web implementado.

$$H_0: ICa \geq ICd$$

- **HA:** El sistema web incrementa el índice de cobertura de mantenimiento preventivo realizado por el área de sistemas en la Compañía Naviera Natalia S.A.C, lo que significa que con el sistema web implementado el indicador es mejor que con el indicador sin el sistema.

$$H_A: ICa < ICd$$

Figura 24: Índice de Cobertura- Comparativa general



Fuente: Elaboración propia

En la figura anterior se muestra el incremento del índice de cobertura de mantenimiento, el cual se puede validar al realizar la comparativa entre las 2 medidas realizadas, que ha ascendido del 46.70 % al 74.35%.

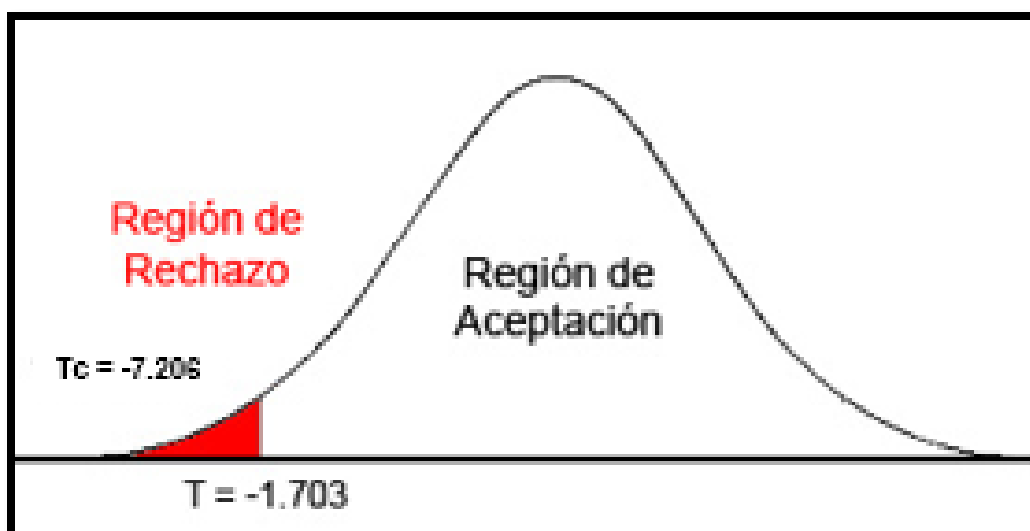
En lo que corresponde a la prueba de T-Student realizado para el índice de cobertura de mantenimiento, el resultado de T contraste fue de -7.206, lo cual es menor que -1.716 y se puede observar en la siguiente tabla.

Tabla 9: Prueba T. Student de Índice de cobertura de mantenimiento antes y después de implementado el sistema web

	Media	Prueba de T-Student		
		T	gl	Sig. (bilateral)
Índice_Cobertura_Prestest	0.46700			
		-7.206	19	0.000
Índice_Cobertura_Posttest	0.74350			

Fuente: Elaboración propia

Figura 25: Prueba T-Student – Índice de Cobertura



Fuente: Elaboración propia

Por lo tanto, se rechaza la hipótesis nula, y se acepta la hipótesis alterna con un 95% de confianza. Además, el valor T obtenido, como se observa en la figura, anterior, se ubica en la zona de rechazo. Por lo tanto, El sistema web incrementa el índice de cobertura de mantenimiento preventivo realizado por el área de sistemas en la Compañía Naviera Natalia S.A.C. en el año 2018.

3.3.2. Hipótesis de Investigación 2:

- **H1:** El sistema web incrementa el grado de cumplimiento de las atenciones realizadas por el área de sistemas en la Compañía Naviera Natalia S.A.C.
- **Indicador:** Grado de cumplimiento

Definiciones de Variables:

- **GCa** = Grado de cumplimiento antes de utilizar el sistema web
- **GCd** = Grado de cumplimiento después de utilizar el sistema web
- **H0:** El sistema web no incrementa el grado de cumplimiento de las atenciones realizadas por el área de sistemas en la Compañía Naviera Natalia S.A.C.

$$H0: GCa \geq GCd$$

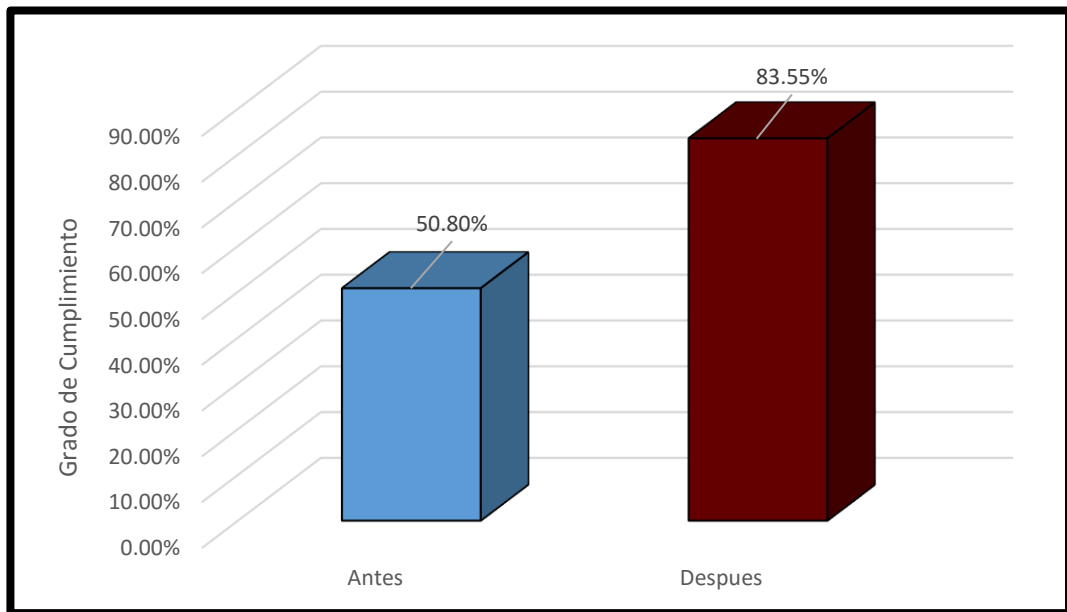
El indicador sin el sistema web es mejor que el indicador con el sistema web.

- **HA:** El sistema web incrementa el grado de cumplimiento de las atenciones realizadas por el área de sistemas en la Compañía Naviera Natalia S.A.C.

$$HA: GCa < GCd$$

El indicador con el sistema web es mejor que el indicador sin el sistema web.

Figura 26: Grado de cumplimiento - Comparativa general



Fuente: Elaboración propia

En la figura anterior se muestra el incremento del grado de cumplimiento, el cual se puede validar al realizar la comparativa entre las 2 medidas realizadas, que ha ascendido del 50.80 % al 83.55%.

En lo que corresponde a la prueba de T-Student realizado para el grado de cumplimiento de las atenciones realizadas, el resultado de T contraste fue de -11.127, lo cual es menor a -1.703 y se puede observar en la siguiente tabla.

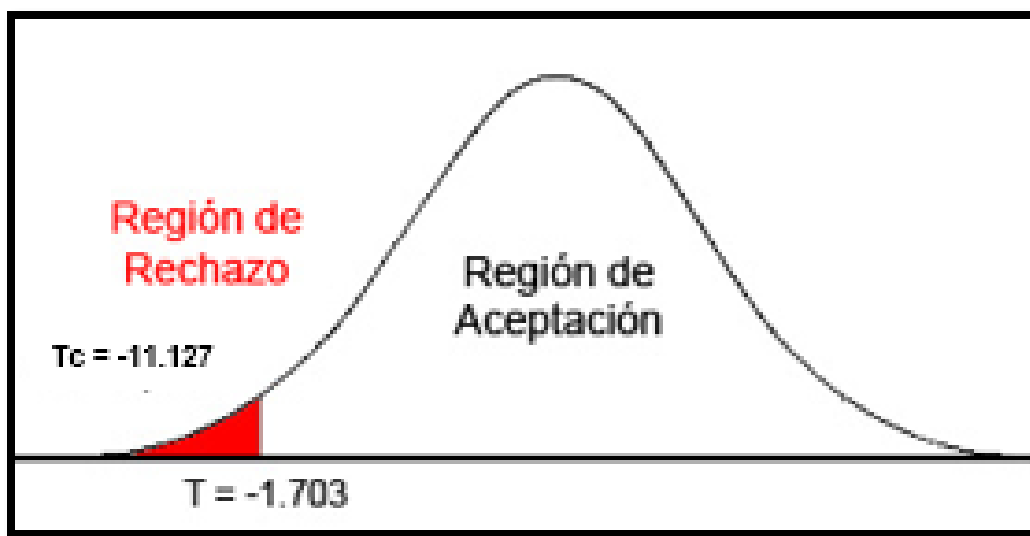
Tabla 10: Prueba de T. Student del grado de cumplimiento antes y después de la implementación el sistema web

	Media	Prueba de T-Student		
		T	gl	Sig. (bilateral)
Grado_Cumplimiento_PreTest	0.5080			
		-11.127	19	0.000
Grado_Cumplimiento_PostTest	0.8355			

Fuente: Elaboración propia

Por lo tanto, se rechaza la hipótesis nula y se acepta la hipótesis alterna con un 95% de confianza. Lo que se ratifica que el sistema web incrementa el grado de cumplimiento de las atenciones realizadas, En la siguiente figura se observa la región de rechazo para la hipótesis notándose un valor T se ubica muy por debajo de la región de rechazo.

Figura 27: Prueba T-Student – Grado de cumplimiento



Fuente: Elaboración propia

IV. DICUSIÓN

En la presente investigación, se tuvo como resultado que, con el sistema web, se incrementó el índice de cobertura de mantenimientos de un 46.69% a un 74.35%, el equivalente a un aumento promedio de 27.66%. De la misma manera que Renzo Flores en su investigación denominada “Desarrollo de una aplicación web para mejorar la gestión del mantenimiento preventivo de correctivo de equipos informáticos en el hospital la Caleta-Chimbote” obtuvo una considerable mejora en la cobertura de mantenimientos realizados a los equipos informáticos después de la implementación de su aplicación web, logrando alcanzar un incremento del 31.2%. También se tuvo como resultado que sistema web incrementó el grado de cumplimiento del control de servicios de un 50.80% a un 83.55%, lo que equivale a un incremento promedio del 32.75 %. De la misma manera que Cuestas, en su investigación denominado “Diseño e implementación de una mesa de servicios,

utilizando ITIL v3.0, para el servicio técnico en el departamento tecnológico del Ministerio del Ambiente”, tesis realizado en la Escuela Politécnica Nacional en Quito Ecuador obtuvo el 96.45%. después de la implementación.

V. CONCLUSIÓN

En conclusión, el sistema web mejora el control de los servicios del área de sistemas en la Compañía Naviera Natalia S.A.C mejora el control de servicio de mantenimiento preventivo, pues mediante su uso, incremento el índice de cobertura de mantenimiento preventivo a un 27.66%., lo que permitió alcanzar los objetivos de esta investigación. Se concluye que el sistema web mejoro el grado de cumplimiento en un 32.75 % Por lo tanto, se afirma que el sistema web aumenta el grado de cumplimiento de los servicios atendidos por el área de sistemas.

VI. RECOMENDACIONES

Se recomienda asignarle un certificado de seguridad a la página para mantener la seguridad al usar el sistema.

Se recomienda la generación de backup automatizados de la base de datos entre servidores de distintas sedes, para minimizar los riesgos de perdida de información por algún desastre natural o siniestro.

Se recomienda la capacitación continua a los usuarios tanto para el buen uso del sistema como para la concientización de los peligros cibernéticos existentes hoy en día, se sugiere hacer uso de este sistema web vía publicación intranet a través de una red privada virtual o publicarlo asignando filtro para fuentes de dirección IP confiable y filtro de protocolos y aplicaciones básicos para su funcionamiento.

VII. PROPUESTA

Se propone implementar este proyecto en cada una de las sedes de la empresa y elevar su nivel de seguridad, publicándolo solo atreves de la intranet de la compañía, vía red privada virtual

VIII. REFERENCIA

ACERO LINARES, Juan Carlos. Las mejores bases de datos. Disponible en:

<https://www.coregistros.com/mejores-bases-de-datos/>

AENOR. Guía completa de aplicación para la gestión de los servicios de tecnologías de la información, 2010, 775 pp.

ISBN:978-84-8143-662-4.

BAHIT, Eugenia. Scrum & Extreme Programming, 2012, 162 pp.

Reg.: 1-20511-563509

BERZAL Fernando, CORTIJO Francisco y CUBERO. Desarrollo Profesional de aplicaciones web con ASP.NET, 2005. 176 pp.

ISBN 84-609-4245-7.

BESTRATÉN, Manuel. Integración de sistemas de gestión: prevención de riesgos laborales, calidad y medio ambiente, Disponible en:

http://www.insht.es/InshtWeb/Contenidos/Documentacion/FichasTecnicas/NTP/Ficheros/501a600/ntp_576.pdf

CHAVEZ GOMEZ, Víctor Hugo. Sistema de información para el control, seguimiento y mantenimiento del Equipamiento hospitalario. Tesis (Ingeniero informático). Perú: Universidad Ricardo Palma, 2010.

CORONADO, Wilber. Implementación de un sistema help desk para mejorar la calidad de servicio al usuario interno del hospital San Juan de Lurigancho, 2015.

CUESTAS, Luis. Diseño e implementación de una mesa de servicios, utilizando ITIL V3.0, para el servicio técnico en el departamento tecnológico del Ministerio del Ambiente, 2017.

DEL SOLE, Alessandro. Visual Studio 2017 Succinctly, 2017, disponible en:

<https://www.syncfusion.com/>

DIALNET. Las Tics en las empresas: evolución de la tecnología y cambio estructural en las organizaciones, 2018.

ISSN: 2477-8818.

DOMINGUEZ, Romell. Aplicación de Métricas De Calidad En Uso Utilizando ISO 9126 Para Determinar El Grado De Satisfacción del Sistema Único De Matricula, 2016.

DIAZ MARCILLA, Jacinto y RUIZ GARCIA. Jesús Enrique. Organización control y mantenimiento instalaciones solares. 1ªed. España: Paraninfo, 2012.

ISBN: 9788428333061

ESLAVA, Muñoz, V. J. El nuevo PHP: conceptos avanzados. 2013, 109 pp.

ISBN: 978-84-686-4433-2.

FERNANDEZ ALARCON, Vicenc. Desarrollo de sistemas de información: Una metodología basada en el modelado. 1ª ed. Barcelona: UPC, 2006.

ISBN 84-8301-862-4

FLORES APARICIO, Renzo Aldair. Desarrollo de una aplicación web para mejorar la gestión de mantenimiento preventivo y correctivo de equipos informáticos en el hospital la Caleta-Chimbote, 2017.

GALLEGO, José. PCPI-Mantenimiento de sistemas microinformáticos. 1ª ed. España: EDITEX, 2010.

ISBN:9788497717670

GARRIDO, Cristián. Revista Gerencia. Por qué la gestión de activos tecnológicos es tan importante.

Disponible en: <http://www.emb.cl/gerencia/articulo.mvc?xid=3271>

GILFILAN, Lan. La biblia de MYSQL, 2003, 873.

ISBN: 9788441515581

GÓMEZ, Marcelo. Introducción a la Metodología de la Investigación Científica. Córdoba, Argentina: Brujas, 2006.

HERNANDEZ Roberto, FERNANDEZ Collado y BAPTISTA Lucio. Metodología de la investigación 3ra edición, 2003, 705 pp.

ISBN: 970-10-3632-8.

JAN VAN BON, ARJEN DE JONG, AXEL KOLTHOF, MIKE PIEPER, RUBY TJASSING, ANNELIES VAN DER VEEN, TIENEKE VERHEIJEN ARJEN; Holanda. Operación de Servicio basado en ITIL. 2008, 196 pp.

ISSN: 9789087531522.

KENDALL Kenneth, KENDALL Julie, Análisis y diseño de sistemas, 2011, 600 p.

ISBN: 978-607-32-0577-1.

KRUCHTEN, Philippe. The rational unified process: an introduction, 2004, 378 pp.

ISBN: 0-321-16609-4.

LINKE, Leonardo; SILVEIRA, Sidnei. Implantação de um Sistema de Help-Desk: um estudo de caso na Exatus Soluções Estratégicas. 2015.

MENZINSKY Alexander, LÓPEZ Gertrudis, PALACIO Juan. Scrum Manager, 2016, 94 pp.

Reg.: 1607208414838.

PALACIOS VELÁSQUEZ, Roosevelt Félix. Aplicación de operación T.I. Para el proceso de gestión de incidencias en la sede central de la empresa Babymodas S.A.C., 2017.

RPP, Conoce cuántas empresas en el Perú producen con calidad,2017.

SAMPIERI, Hernández; COLLADO, Fernández; BAPTISTA, Pilar et al. Metodología de la investigación, 2014.

ISBN: 978-1-4562-2396-0.

SILBERSCHATZ, Abraham, KORTH Henry, SUDARSHAN S. Fundamentos de base de datos, 2006, 953 pp.

ISBN: 0-07-295886-3

VAN BON, Jan. Gestión de Servicios TI basado en ITIL® V3-Guia de Bolsillo. Van Haren, 2008.


VERDU GUERRERO, Wilnel J. y GARCIA DE CECA, Mauricio. Software libre para el control y gestión de los procesos administrativos y académicos de instituciones privadas de educación para los ciclos básico, medio y diversificado. Tesis de grado inédita, Universidad Nueva Esparta, Venezuela, 2012.

ANEXOS

Anexo 1: Instrumento Pre Test Índice De Cobertura De Mantenimiento

N ficha de Registro				1
Observador		Alex Aldo AiquipaTello		
Institución donde se Investiga		Compañía Naviera Natalia S.A.C.		
Ubicación de la Institución		Calle Los Conquistadores 630 San Isidro Lima Perú		
Indicador de la Observación		Índice de Cobertura de Mantenimiento		
Periodo de Observación		2 de Abril de 2018 al 30 de Abril de 2018		
Instrumento			Ficha de Registro	
Item	Fecha Inicial	Número de Equipos con mantenimiento Preventivo	Número total de equipos Programados	Índice de Cobertura de Mantenimiento
				$ICM = \frac{EPR}{TEP} * 100\%$
				ICM = Índice de cobertura de mantenimiento EPR = Equipos con mantenimiento preventivo TEP = Total de equipos programados
1	02/04/2018	7	13	53,85%
2	04/04/2018	6	12	50%
3	05/04/2018	8	11	72,73%
4	06/04/2018	7	14	50%
5	09/04/2018	8	13	61,54%
6	10/04/2018	5	21	23,81%
7	11/04/2018	7	14	50%
8	12/04/2018	7	13	53,85%
9	13/04/2018	4	11	36,36%
10	16/04/2018	9	15	60%
11	17/04/2018	6	10	60%
12	18/04/2018	7	12	58,33%
13	20/04/2018	3	9	33,33%
14	23/04/2018	5	10	50%
15	24/04/2018	2	8	25%
16	25/04/2018	5	12	41,67%
17	26/04/2018	3	10	30%
18	27/04/2018	3	10	30%
19	28/04/2018	4	12	33,33%
20	30/04/2018	6	10	60%

El promedio alcanzado del índice de cobertura de mantenimiento alcanzado en la ficha de registro PRE-TEST es de 46, 69% evidenciando así que no se está alcanzando el porcentaje esperado no cumpliéndose así con el total de equipos programados por día que deben pasar mantenimiento preventivo.


 ING. HUGO CANDELA
 JEFE DE SISTEMAS CORPORATIVO

Anexo 2: Instrumento Pre Test Grado De Cumplimiento

N ficha de Registro		2		
Observador		Alex Aldo Aiquipa Tello		
Institución donde se Investiga		Compañía Naviera Natalia S.A.C.		
Ubicación de la Institución		Calle Los Conquistadores 630 San Isidro Lima Perú		
Indicador de la Observación		Grado de Cumplimiento		
Periodo de Observación		2 de Abril del 2018 al 30 de Abril del 2018		
Instrumento		Ficha de Registro		
Item	Fecha Inicial	Entregados	Esperados	Grado de Cumplimiento
$GC = (RET / RE) * 100 \%$ Dónde: GC: Grado de Cumplimiento RET: Entregados RE: Esperados				
1	02/04/2018	7	12	58.33%
2	04/04/2018	6	12	50.00%
3	05/04/2018	7	14	50.00%
4	06/04/2018	6	12	50.00%
5	09/04/2018	4	14	28.57%
6	10/04/2018	8	15	53.33%
7	11/04/2018	6	12	50.00%
8	12/04/2018	7	13	53.85%
9	13/04/2018	8	12	66.67%
10	16/04/2018	6	14	42.86%
11	17/04/2018	6	12	50.00%
12	18/04/2018	7	14	50.00%
13	20/04/2018	8	12	66.67%
14	23/04/2018	7	14	50.00%
15	24/04/2018	7	12	58.33%
16	25/04/2018	8	12	66.67%
17	26/04/2018	6	14	42.86%
18	27/04/2018	6	15	40.00%
19	28/04/2018	7	15	46.67%
20	30/04/2018	6	15	40.00%

El promedio alcanzado del grado de cumplimiento alcanzado en la ficha de registro PRE-TEST es de 50,74% evidenciando así que no se está alcanzando el porcentaje esperado


 ING. HUGO CANDELA
 JEFE DE SISTEMAS CORPORATIVO

Anexo 3: Instrumento Post Test Índice De Cobertura De Mantenimiento

N ficha de Registro				3
Observador				Alex Aldo Aiquipa Tello
Institución donde se Investiga				Compañía Naviera Natalia S.A.C.
Ubicación de la Institución				Calle Los Conquistadores 630 San Isidro Lima Perú
Indicador de la Observación				Índice de Cobertura de Mantenimiento
Periodo de Observación				1 de Octubre del 2018 al 31 de Octubre del 2018
Instrumento				Ficha de Registro
Item	Fecha Inicial	Número de Equipos con mantenimiento Preventivo	Número total de equipos Programados	Índice de Cobertura de Mantenimiento $ICM = \frac{EPR}{TEP} * 100\%$ ICM = Índice de cobertura de mantenimiento EPR = Equipos con mantenimiento preventivo TEP = Total de equipos programados
1	01/10/2018	10	15	66.67%
2	03/10/2018	7	9	77.78%
3	04/10/2018	8	10	80.00%
4	05/10/2018	6	11	54.55%
5	08/10/2018	10	12	83.33%
6	09/10/2018	11	15	73.33%
7	10/10/2018	15	18	83.33%
8	12/10/2018	9	10	90.00%
9	13/10/2018	15	17	88.24%
10	15/10/2018	12	15	80.00%
11	17/10/2018	5	8	62.50%
12	18/10/2018	11	19	57.89%
13	19/10/2018	7	11	63.64%
14	22/10/2018	7	7	100.00%
15	23/10/2018	10	13	76.92%
16	24/10/2018	4	8	50.00%
17	25/10/2018	10	16	62.50%
18	26/10/2018	7	11	63.64%
19	29/10/2018	5	7	71.43%
20	31/10/2018	8	8	100.00%

El promedio alcanzado del índice de cobertura de mantenimiento alcanzado en la ficha de registro POST-TEST es de 73, 75% evidenciando así un importante incremento del índice de cobertura de mantenimiento.


ING. HUGO CANDELA
 JEFE DE SISTEMAS CORPORATIVO

Anexo 4: Instrumento Post Test Grado De Cumplimiento

X ficha de Registro				
Observador		Alex Aldo Aiquipa Tello		
Institución donde se Investiga		Compañía Naviera Natalia S.A.C.		
Ubicación de la Institución		Calle Los Conquistadores 630 San Isidro Lima Perú		
Indicador de la Observación		Grado de Cumplimiento		
Periodo de Observación		1 de Octubre del 2018 al 31 de Octubre del 2018		
Instrumento		Ficha de Registro		
Item	Fecha Inicial	Entregados	Esperados	Grado de Cumplimiento
				$GC = (RET / RE) * 100$ Dónde: GC: Grado de Cumplimiento RET: Entregados a tiempo RE: Esperados
1	01/10/2018	13	15	86.67
2	02/10/2018	12	12	100
3	03/10/2018	11	13	84.62
4	04/10/2018	13	15	86.67
5	05/10/2018	16	18	88.89
6	09/10/2018	8	10	80
7	10/10/2018	10	14	71.43
8	12/10/2018	7	11	63.64
9	13/10/2018	8	10	80
10	15/10/2018	9	12	75
11	16/10/2018	15	16	93.75
12	17/10/2018	10	14	71.43
13	19/10/2018	13	15	86.67
14	22/10/2018	10	13	76.92
15	23/10/2018	10	12	83.33
16	24/10/2018	10	10	100
17	25/10/2018	9	12	75
18	26/10/2018	12	13	92.31
19	29/10/2018	14	16	87.50
20	31/10/2018	12	14	85.71

El promedio alcanzado del grado de cumplimiento alcanzado en la ficha de registro POST-TEST es de 83.77 % evidenciando así una mejora significativa.


 ING. HUGO CANDELA
 JEFE DE SISTEMAS CORPORATIVO

Anexo 5: Validación de Metodología de desarrollo de Software Experto 1

TABLA DE EVALUACIÓN DE EXPERTOS
(Metodología de desarrollo de Software)

Nombres y Apellidos del experto: Abraham Rafael Sotelo Apurí
 Institución donde Labora: Universidad Cesar Vallejo
 Cargo que ocupa: Docente
 Universidad que labora: Universidad César Vallejo Lima Norte
 Fecha: 15-06-2018
 Autor: Aiquipa Tello Alex Aldo

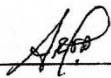
Proyecto:

**SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL ÁREA DE SISTEMAS
EN LA COMPAÑÍA NAVIERA NATALIA S.A.C.**

Mediante la tabla de evaluación de expertos, usted tiene la facultad de evaluar cada una de las preguntas a través de un puntaje. Asimismo, se le solicita la corrección de los ítems indicando sus observaciones o sugerencias, con la finalidad de mejorar la coherencia de las preguntas.

ITEMS	Puntaje a colocar: Muy Bueno = 4, Bueno = 3, Regular = 2, Malo = 1			
	Preguntas - Criterios	Scrum	XP	RUP
1	Su objetivo es reducir el tiempo de las solicitudes	4	2	3
2	Es una metodología de rápida implementación	3	2	3
3	Es una metodología flexible a los cambios funcionales durante el proyecto	3	2	3
4	Divide el grupo en una lista de entregables pequeños, concretos y estima el esfuerzo relativo de cada elemento	4	3	3
5	Las iteraciones de entregas son de 2 a 3 semanas	4	3	4
6	Lo que se termina, funciona bien, se aparta y ya no se toca	4	2	3
7	Involucra el trabajo cotidiano en equipo tanto las personas del negocio como el equipo de desarrollo	3	2	3
8	Interactúan constantemente con el cliente y logaran acuerdos mutuos	4	3	3
9	Permite la entrega de software en un breve periodo de tiempo	3	2	3
10	Adecuado para el desarrollo de proyectos en cortos tiempo sin aumentar el costo del proyecto	4	3	3
TOTAL				

SUGERENCIAS: _____

Firma del Experto: 

Anexo 6: Validación de Metodología de desarrollo de Software Experto 2

TABLA DE EVALUACIÓN DE EXPERTOS
(Metodología de desarrollo de Software)

Nombres y Apellidos del experto: Juan S. Chumpi A.
 Institución donde Labora: Universidad Cesar Vallejo
 Cargo que ocupa: Docente - Investigador
 Universidad que labora: Universidad César Vallejo Lima Norte
 Fecha: 20/05/2018
 Autor: Aiquipa Tello Alex Aldo

Proyecto:

**SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL ÁREA DE SISTEMAS
EN LA COMPAÑÍA NAVIERA NATALIA S.A.C.**

Mediante la tabla de evaluación de expertos, usted tiene la facultad de evaluar cada una de las preguntas a través de un puntaje. Asimismo, se le solicita la corrección de los ítems indicando sus observaciones o sugerencias, con la finalidad de mejorar la coherencia de las preguntas.

ITEMS	Puntaje a colocar: Muy Bueno = 4, Bueno = 3, Regular = 2, Malo = 1			
	Preguntas - Criterios	Scrum	XP	RUP
1	Su objetivo es reducir el tiempo de las solicitudes	4	4	4
2	Es una metodología de rápida implementación	4	4	3
3	Es una metodología flexible a los cambios funcionales durante el proyecto	4	3	3
4	Divide el grupo en una lista de entregables pequeños, concretos y estima el esfuerzo relativo de cada elemento	4	4	3
5	Las iteraciones de entregas son de 2 a 3 semanas	4	3	3
6	Lo que se termina, funciona bien, se aparta y ya no se toca	4	3	3
7	Involucra el trabajo cotidiano en equipo tanto las personas del negocio como el equipo de desarrollo	4	3	4
8	Interactúan constantemente con el cliente y logran acuerdos mutuos	3	3	3
9	Permite la entrega de software en un breve periodo de tiempo	4	4	3
10	Adecuado para el desarrollo de proyectos en cortos tiempo sin aumentar el costo del proyecto	4	4	3
TOTAL				

SUGERENCIAS: _____

Firma del Experto: _____

Anexo 7: Validación de Metodología de desarrollo de Software Experto 3

TABLA DE EVALUACIÓN DE EXPERTOS (Metodología de desarrollo de Software)

Nombres y Apellidos del experto: Alex Abalanda Pacheco Bermalego
 Institución donde Labora: Universidad Cesar Vallejo
 Cargo que ocupa: Docente
 Universidad que labora: Universidad César Vallejo Lima Norte
 Fecha: 15-06-2018
 Autor: Aiquipa Tello Alex Aldo


Proyecto:

SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL ÁREA DE SISTEMAS EN LA COMPAÑÍA NAVIERA NATALIA S.A.C.

Mediante la tabla de evaluación de expertos, usted tiene la facultad de evaluar cada una de las preguntas a través de un puntaje. Asimismo, se le solicita la corrección de los ítems indicando sus observaciones o sugerencias, con la finalidad de mejorar la coherencia de las preguntas.

ITEMS	Puntaje a colocar: Muy Bueno = 4, Bueno = 3, Regular = 2, Malo = 1			
	Preguntas - Criterios	Scrum	XP	RUP
1	Su objetivo es reducir el tiempo de las solicitudes	4	3	3
2	Es una metodología de rápida implementación	4	3	3
3	Es una metodología flexible a los cambios funcionales durante el proyecto	4	3	3
4	Divide el grupo en una lista de entregables pequeños, concretos y estima el esfuerzo relativo de cada elemento	4	3	3
5	Las iteraciones de entregas son de 2 a 3 semanas	4	3	3
6	Lo que se termina, funciona bien, se aparta y ya no se toca	4	3	3
7	Involucra el trabajo cotidiano en equipo tanto las personas del negocio como el equipo de desarrollo	4	3	3
8	Interactúan constantemente con el cliente y logaran acuerdos mutuos	4	3	3
9	Permite la entrega de software en un breve periodo de tiempo	4	3	3
10	Adecuado para el desarrollo de proyectos en cortos tiempo sin aumentar el costo del proyecto	4	3	3
TOTAL				

SUGERENCIAS: _____

Firma del Experto: 

Anexo 8: Validación de experto 1 para indicador de cumplimiento

TABLA DE EVALUACIÓN DE EXPERTOS

Datos del experto:

1. Apellidos y Nombres: Sáenz Ariani Abianam Rafael
2. Cargo que sustenta: Docente
3. Título y/o Grado: Magister
4. Universidad que labora: Universidad César Vallejo Lima Norte
5. Autor: A. Buiça Tello Alex Aldo
6. Fecha: 15/06/2018

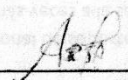
TESIS:

SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL ÁREA DE SISTEMAS EN LA COMPAÑÍA NAVIERA NATALIA S.A.C.

Tabla de Evaluación de Expertos para el Indicador: Grado de cumplimiento

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas llenando con un "%" en las columnas correspondientes. Así mismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias, con la finalidad de mejorar la coherencia de las preguntas.

ITEMS	PREGUNTA	Deficiente 0 – 20 %	Regular 21 – 50 %	Bueno 51 – 70 %	Muy bueno 71 – 80 %	Excelente 81 – 100%
1	¿El instrumento de medición cumple con el diseño adecuado?					83
2	¿El instrumento de recolección de datos tiene relación con el título de la investigación?					83
3	¿En el instrumento de recolección de datos se mencionan las variables de investigación?					83
4	¿El instrumento de recolección de datos facilitará el logro de los objetivos de investigación?					83
5	¿El instrumento de recolección de datos se relaciona con la variable de estudio?					83
6	¿El diseño del instrumento de medición facilitará el análisis y procesamiento de datos?					83
7	¿El instrumento de medición será accesible a la población sujeto de estudio?					83
TOTAL						83


Firma del Experto

Anexo 9: Validación de experto 2 para indicador de cumplimiento

TABLA DE EVALUACIÓN DE EXPERTOS

Datos del experto:

1. Apellidos y Nombres: Plump Agait, J. B.
2. Cargo que sustenta: Cont. Gen.
3. Título y/o Grado: Magister
4. Universidad que labora: Universidad César Vallejo Lima Norte
5. Autor: Ayuppo Tello Alex Aldo
6. Fecha: 16/06/2018

TESIS:

SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL ÁREA DE SISTEMAS
EN LA COMPAÑÍA NAVIERA NATALIA S.A.C.

Tabla de Evaluación de Expertos para el Indicador: Grado de cumplimiento

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas llenando con un "%" en las columnas correspondientes. Así mismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias, con la finalidad de mejorar la coherencia de las preguntas.

ITEMS	PREGUNTA	Deficiente 0 - 20 %	Regular 21 - 50 %	Bueno 51 - 70 %	Muy bueno 71 - 80 %	Excelente 81 - 100%
1	¿El instrumento de medición cumple con el diseño adecuado?				80	
2	¿El instrumento de recolección de datos tiene relación con el título de la investigación?				80	
3	¿En el instrumento de recolección de datos se mencionan las variables de investigación?				80	
4	¿El instrumento de recolección de datos facilitará el logro de los objetivos de investigación?				80	
5	¿El instrumento de recolección de datos se relaciona con la variable de estudio?				80	
6	¿El diseño del instrumento de medición facilitará el análisis y procesamiento de datos?				80	
7	¿El instrumento de medición será accesible a la población sujeto de estudio?				80	
TOTAL					80	


 Firma del Experto

Anexo 10: Validación de experto 3 para indicador de cumplimiento

TABLA DE EVALUACIÓN DE EXPERTOS

Datos del experto:

1. Apellidos y Nombres: Alex Abelardo Pacheco Armalique
2. Cargo que sustenta: Docente
3. Título y/o Grado: Magister
4. Universidad que labora: Universidad César Vallejo Lima Norte
5. Autor: Alicia Tello Alex 180
6. Fecha: 15/06/2018

TESIS:

**SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL ÁREA DE SISTEMAS
EN LA COMPAÑÍA NAVIERA NATALIA S.A.C.**

Tabla de Evaluación de Expertos para el Indicador: Grado de cumplimiento

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas llenando con un "%" en las columnas correspondientes. Así mismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias, con la finalidad de mejorar la coherencia de las preguntas.

ITEMS	PREGUNTA	Deficiente 0 – 20 %	Regular 21 – 50 %	Bueno 51 – 70 %	Muy bueno 71 – 80 %	Excelente 81 – 100%
1	¿El instrumento de medición cumple con el diseño adecuado?					84
2	¿El instrumento de recolección de datos tiene relación con el título de la investigación?					84
3	¿En el instrumento de recolección de datos se mencionan las variables de investigación?					84
4	¿El instrumento de recolección de datos facilitará el logro de los objetivos de investigación?					84
5	¿El instrumento de recolección de datos se relaciona con la variable de estudio?					84
6	¿El diseño del instrumento de medición facilitará el análisis y procesamiento de datos?					84
7	¿El instrumento de medición será accesible a la población sujeto de estudio?					84
TOTAL						84


 Firma del Experto

Anexo 11: Evaluación de Experto 1 para Indicador Índice de Cobertura

TABLA DE EVALUACIÓN DE EXPERTOS

Datos del experto:

1. Apellidos y Nombres: Saenz Apari Abraham Rafael
2. Cargo que sustenta: Docente
3. Título y/o Grado: Mag. Dr.
4. Universidad que labora: Universidad César Vallejo Lima Norte
5. Autor: Alicia Tello Bx A20
6. Fecha: 15/06/2018

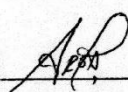
TESIS:

SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL ÁREA DE SISTEMAS EN LA COMPAÑÍA NAVIERA NATALIA S.A.C.

Tabla de Evaluación de Expertos para el Indicador: Índice de Cobertura

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas llenando con un "%" en las columnas correspondientes. Así mismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias, con la finalidad de mejorar la coherencia de las preguntas.

ITEMS	PREGUNTA	Deficiente 0 - 20 %	Regular 21 - 50 %	Bueno 51 - 70 %	Muy bueno 71 - 80 %	Excelente 81 - 100%
1	¿El instrumento de medición cumple con el diseño adecuado?					83
2	¿El instrumento de recolección de datos tiene relación con el título de la investigación?					83
3	¿En el instrumento de recolección de datos se mencionan las variables de investigación?					83
4	¿El instrumento de recolección de datos facilitará el logro de los objetivos de investigación?					83
5	¿El instrumento de recolección de datos se relaciona con la variable de estudio?					83
6	¿El diseño del instrumento de medición facilitará el análisis y procesamiento de datos?					83
7	¿El instrumento de medición será accesible a la población sujeto de estudio?					83
TOTAL						83


 Firma del Experto

Anexo 12: Evaluación de Experto 2 para Indicador Índice de Cobertura

TABLA DE EVALUACIÓN DE EXPERTOS

Datos del experto:

1. Apellidos y Nombres: *Alfonso Aguirre, Juan B.*
2. Cargo que sustenta: *Docente*
3. Título y/o Grado: *Magister*
4. Universidad que labora: Universidad César Vallejo Lima Norte
5. Autor: *Alfonso Aguirre Tello Luis ALDO*
6. Fecha: *16/06/2018*

TESIS:

SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL ÁREA DE SISTEMAS EN LA COMPAÑÍA NAVIERA NATALIA S.A.C.

Tabla de Evaluación de Expertos para el Indicador: Índice de Cobertura

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas llenando con un "%" en las columnas correspondientes. Así mismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias, con la finalidad de mejorar la coherencia de las preguntas.

ITEMS	PREGUNTA	Deficiente 0 - 20 %	Regular 21 - 50 %	Bueno 51 - 70 %	Muy bueno 71 - 80 %	Excelente 81 - 100%
1	¿El instrumento de medición cumple con el diseño adecuado?		50	5		
2	¿El instrumento de recolección de datos tiene relación con el título de la investigación?		50			
3	¿En el instrumento de recolección de datos se mencionan las variables de investigación?		50			
4	¿El instrumento de recolección de datos facilitará el logro de los objetivos de investigación?		50			
5	¿El instrumento de recolección de datos se relaciona con la variable de estudio?		50			
6	¿El diseño del instrumento de medición facilitará el análisis y procesamiento de datos?		50			
7	¿El instrumento de medición será accesible a la población sujeto de estudio?		50			
TOTAL			50			

[Firma manuscrita]
 16/06/2018
 Firma del Experto

Anexo 13: Evaluación de Experto 3 para Indicador Índice de Cobertura

TABLA DE EVALUACIÓN DE EXPERTOS

Datos del experto:

1. Apellidos y Nombres: Alex Abelardo Pacheco Romalique
2. Cargo que sustenta: Docente
3. Título y/o Grado: Mag. I.S.S.
4. Universidad que labora: Universidad César Vallejo Lima Norte
5. Autor: Azupu Tello Alex LINDO
6. Fecha: 15/06/2018

TESIS:

SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL ÁREA DE SISTEMAS EN LA COMPAÑÍA NAVIERA NATALIA S.A.C.

Tabla de Evaluación de Expertos para el Indicador: Índice de Cobertura

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas llenando con un "%" en las columnas correspondientes. Así mismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias, con la finalidad de mejorar la coherencia de las preguntas.

ITEMS	PREGUNTA	Deficiente 0 – 20 %	Regular 21 – 50 %	Bueno 51 – 70 %	Muy bueno 71 – 80 %	Excelente 81 – 100%
1	¿El instrumento de medición cumple con el diseño adecuado?					84
2	¿El instrumento de recolección de datos tiene relación con el título de la investigación?					84
3	¿En el instrumento de recolección de datos se mencionan las variables de investigación?					84
4	¿El instrumento de recolección de datos facilitará el logro de los objetivos de investigación?					84
5	¿El instrumento de recolección de datos se relaciona con la variable de estudio?					84
6	¿El diseño del instrumento de medición facilitará el análisis y procesamiento de datos?					84
7	¿El instrumento de medición será accesible a la población sujeto de estudio?					84
TOTAL						




 Firma del Experto

Anexo 14: Carta de aceptación del proyecto



CARTA DE ACEPTACION

Por medio de la presente se da a conocer el inicio de la implementación de un sistema web para el control de procedimientos de calidad ISO de sistemas para la Compañía Naviera Natalia S.A.C. El cual será diseñado y desarrollado por el Sr. Alex Aldo Aiquipa Tello identificado con documento de identidad 10627303, con el objetivo de tener un mejor control de nuestros procedimientos de calidad ISO y mejora de nuestros servicios



.....
ING. HUGO CANDELA
JEFE DE SISTEMAS CORPORATIVO

Hugo Candela Rios

Av. Los Conquistadores N° 638, Oficina 201, San Isidro, Lima 27, Perú
Tel. (511) 201-5300 / Fax (511) 201-5307
www.navieranatalia.com



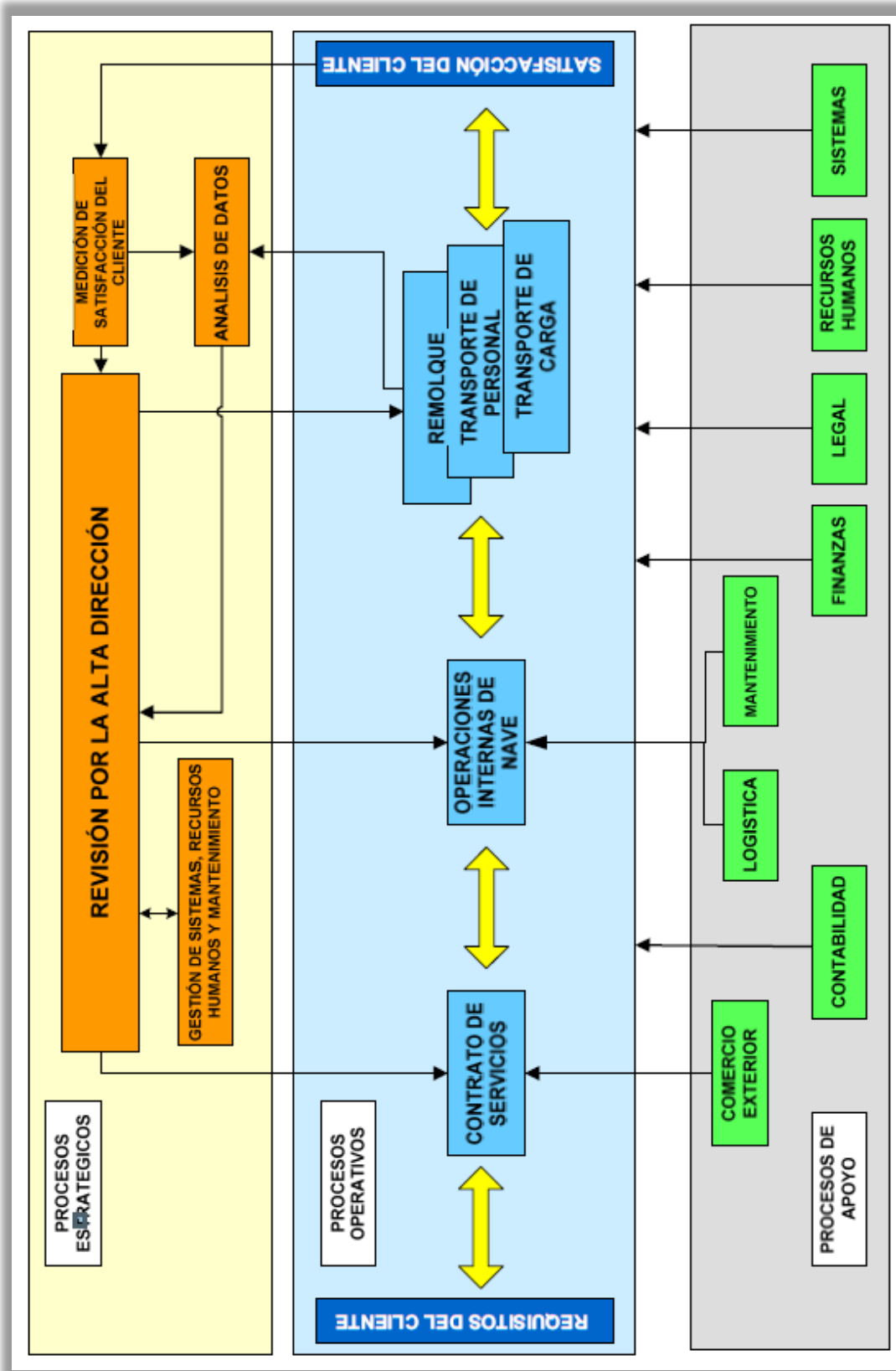
Anexo 15: Entrevista al Jefe Corporativo de la Compañía Naviera Natalia S.A.C.

Entrevista realizada el 16 de abril del 2018 al Sr. Hugo Candela Ríos, Jefe Corporativo de Tecnología de la Información de la Compañía Naviera Natalia S.A.C. establecida en la sede administrativa de San Isidro Lima-Perú

1. ¿A qué se dedica la compañía naviera Natalia S.A.C.?
Compañía Naviera Natalia S.A.C. es una empresa que se dedica a brindar servicio de transporte marítimo y de apoyo logístico para las operaciones fuera de costa para la industria petrolera nacional e internacional, disponemos de naves especializadas que nos permiten brindar apoyo a operaciones de buceo, rescates marítimos, brindamos servicios de remolque y transporte de personas.
2. ¿Cuántas sedes dispone la Compañía Naviera Natalia S.A.C.?
Actualmente disponemos de 3 sedes, las sedes operativas ubicado en Talara y Zorritos y la sede principal administrativa en San Isidro, desde aquí llevamos el control de los procesos contables, financieros, logísticos, de pagos y demás procesos administrativos que se requiera, en coordinación con el personal, que disponemos en las sedes remotas.
3. ¿Cuál es el objetivo a medio plazo que propone el área de sistemas?
Si hablamos a mediano plazo, sería el de mejorar la percepción que tienen los usuarios acerca de nuestra área, en varias ocasiones, durante las reuniones que te tenido con los jefes de área y los usuarios, me han reportado de algunos percances que tuvieron que ver con las demoras de la atención e incidencias repetitivas, que en su momento no se supo tratar.
4. ¿Y porque cree usted que esté ocurriendo este tipo de problema?
Bueno desde mi punto de vista, diría que es por la falta de un sistema de control, que dé seguimiento a estas solicitudes, que nos permitiría en cierta forma tener un control sobre quien está a cargo de tal solicitud, si se le está atendiendo en el tiempo adecuado, a tal punto que me permita tener un historial en línea de las atenciones realizados con éxito.
5. ¿He observado que la compañía tiene certificación ISO 9001-2008, dígame que procesos contempla esta norma para el área que usted está a cargo?
Actualmente ya, nos estamos re categorizando para la versión 2015, el sistema de gestión de control de calidad, para nuestra área dispone de 7 procedimientos de calidad, los cuales tienen que ver con la baja de usuarios, creación de correos, control de accesos, mantenimiento preventivo, correctivo, Backup y control de páginas web y como habrás podido observar el control de estos procedimientos se está realizando de manera manual, se utiliza el correo de soporte para filtrar, atender y solicitar la conformidad del usuario y es más, llenamos estos registros en hojas Excel lo cual es susceptible a errores de digitalización que podría jugarnos en contra al momento de pasar las auditorias.


ING. HUGO CANDELA
JEFE DE SISTEMAS CORPORATIVO

Anexo 16: Diagrama de procesos de nivel 0 de la Compañía Naviera Natalia S.A.C.



Anexo 17: Matriz de consistencia

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES				METODOLOGIA	
			Independiente	Sistema Web	Indicadores	Instrumento		Fórmula
<p>General: P.G: ¿De qué manera un sistema web influye en el control de los servicios del área de Sistemas en la Compañía Naviera Natalia S.A.C.?</p> <p>Específicas: PE1: ¿De qué manera un sistema web influye el índice de cobertura de los mantenimientos preventivos realizados por el área de sistemas en la Compañía Naviera Natalia S.A.C.?</p>	<p>General: OG: Determinar la influencia de un sistema web en el control de servicios del área de sistemas en la Compañía Naviera Natalia S.A.C.</p> <p>Específicas: OE1: Determinar la influencia de un sistema web en el índice de cobertura del servicio de mantenimiento preventivo realizado por el área de sistemas en la Compañía Naviera Natalia S.A.C.</p> <p>OE2: Determinar la influencia de un sistema web en el grado de cumplimiento de las atenciones realizadas por el área de sistemas en la Compañía Naviera Natalia S.A.C.</p>	<p>General: HG: El sistema web influye en el control de los servicios del área de sistemas en la Compañía Naviera Natalia S.A.C.</p> <p>Específicas: HE1: El sistema web incrementa el índice de cobertura del servicio de mantenimiento preventivo en la Compañía Naviera Natalia S.A.C.</p> <p>HE2: El sistema web incrementa el grado de cumplimiento de las atenciones realizadas por el área de sistemas en la Compañía Naviera Natalia S.A.C.</p>	Control de Servicios del área de Ssietemas	Servicio de mantenimiento preventivo	Indice de cobertura de Mantenimiento	Ficha de Registro	$ICM = \frac{EPR}{TRP} * 100\%$	
<p>PE2: ¿De qué manera un sistema web influye en el grado de cumplimiento de las atenciones realizadas por el área de sistemas en la Compañía Naviera Natalia S.A.C.?</p>			Atención oportuna de las atenciones solicitadas	Grado de cumplimiento	Ficha de Registro	$GC = (RET/RE) * 100$		
							<p>Tipo de investigación: Aplicada cuantitativa y experimental.</p> <p>Diseño: Pre-experimental pre-prueba / post-prueba con un solo grupo</p> <p>Población: 20 fichas de registros.</p> <p>Muestra: 20 fichas de registros</p>	

DESARROLLO

Índice del Marco de Trabajo Scrum

PLAN DE METODOLOGÍA DE DESARROLLO	119
INTRODUCCIÓN	119
PROPÓSITO	119
ALCANCE	119
DESCRIPCIÓN DEL PROYECTO	119
DESCRIPCIÓN GENERAL DE LA METODOLOGÍA	120
FUNDAMENTACIÓN	120
DEFINICIÓN DE ROLES DEL PROYECTO	121
SCRUM TEAM (EQUIPO SCRUM)	121
HISTORIAS DE USUARIO	122
MATRIZ DE IMPACTO	125
PRODUCT BACKLOG	125
PILA DEL PRODUCTO (PRODUCT BACKLOG)	126
ENTREGABLES POR SPRINT	127
PLANIFICACIÓN DE LOS SPRINTS	128
PLANIFICACIÓN SPRINT 1	128
PLANIFICACIÓN SPRINT 2	129
PLANIFICACIÓN SPRINT 3	129
SPRINT N° 1: STORY POINT 86	131
SPRINT N° 2: STORY POINT 100	170
SPRINT N° 3: STORY POINT 136	234

Índice de Figuras

Figura N° 1: Historia de Usuario 1	122
Figura N° 2: Historia de Usuario 2	122
Figura N° 3: Historia de Usuario 3	123
Figura N° 4: Historia de Usuario 4	123
Figura N° 5: Historia de Usuario 5	124
Figura N° 6: Historia de Usuario 6	124
Figura N° 7: Historia de Usuario 7	125
Figura N° 8: Caso de uso Sprint 1	131
Figura N° 9. Diagrama Físico de la base de datos	132
Figura N° 10: Prototipo de Inicio De Sesión	133
Figura N° 11. Modificación código Post /Account/Login	133
Figura N° 12. Modificación del código Post /Account/Register	133
Figura N° 13. Modificación del código vista Login	134
Figura N° 14: Modificación del código vista Register	134
Figura N° 15. Acción para el botón ingresar del controlador Account	134
Figura N° 16: Estilos para la vista Login	135
Figura N° 17: Formulario Iniciar Sesión	135
Figura N° 18: Prototipo 1 Index Tipo De Empresa	136
Figura N° 19: Prototipo 2 Crear Tipo De Empresa	136
Figura N° 20: Prototipo 3 Actualizar Tipo De Empresa	136
Figura N° 21: Prototipo 4 Borrar Tipo De Empresa	137
Figura N° 22: Código de creación de la tabla tipo de empresa	137
Figura N° 23: Código del modelo “tipo de empresa”	137
Figura N° 24: Código del controlador Tipo De Empresa	138
Figura N° 25: Código de la vista Index Tipo De Empresa	139
Figura N° 26: Código de la vista Crear Tipo De Empresa	139
Figura N° 27: Código de la vista Actualizar Tipo De Empresa	140
Figura N° 28: Código de la vista Borrar Tipo De Empresa	140
Figura N° 29: Formulario Index Tipo De Empresa	141
Figura N° 30: Formulario Crear Tipo De Empresa	141
Figura N° 31: Formulario Actualizar Tipo De Empresa	141

Figura N° 32: Formulario Eliminar Tipo De Empresa	142
Figura N° 33: Prototipo 1 Listar Empresa	142
Figura N° 34: Prototipo 2 Crear Empresa	142
Figura N° 35: Prototipo 3 Actualizar Empresa	143
Figura N° 36: Prototipo 4 Eliminar Empresa	143
Figura N° 37: Código para la creación de la tabla Empresa	143
Figura N° 38: Código del modelo Empresa	144
Figura N° 39: Código del controlador Empresa	145
Figura N° 40: Código de la vista Index Empresa	146
Figura N° 41: Código de la vista Crear Empresa	146
Figura N° 42: Código de la vista Actualizar Empresa	147
Figura N° 43: Código de la vista Eliminar Empresa	147
Figura N° 44: Formulario Index Empresa	148
Figura N° 45: Formulario Crear Empresa	148
Figura N° 46: Formulario Actualizar Empresa	148
Figura N° 47: Formulario Eliminar Empresa	149
Figura N° 48: Prototipo 1 Index Áreas	149
Figura N° 49: Prototipo 2 Crear Área	149
Figura N° 50: Prototipo 3 Editar Área	150
Figura N° 51: Prototipo 4 Borrar Área	150
Figura N° 52: Código para la creación de la tabla Área	150
Figura N° 53: Código del modelo Área	151
Figura N° 54: Código del controlador Área	151
Figura N° 55: Código de la vista Index Área	152
Figura N° 56: Código de la lista Crear Área	152
Figura N° 57: Código de la vista Actualizar Área	153
Figura N° 58: Código de la vista Eliminar Área	153
Figura N° 59: Formulario Index Área	154
Figura N° 60: Formulario Crear Área	154
Figura N° 61: Formulario Editar Área	154
Figura N° 62: Formulario Eliminar Área	155
Figura N° 63: Prototipo 2 Crear Cargo	155
Figura N° 64: Prototipo 3 Editar Cargo	156

Figura N° 65: Prototipo 3 Eliminar Cargo	156
Figura N° 66: Código para a creación de a taba Cargo	156
Figura N° 67: Código del controlador Cargo	157
Figura N° 68: Código de la vista Index Cargo	158
Figura N° 69: Código de la vista Crear Cargo	158
Figura N° 70: Código de la vista Actualizar Cargo	159
Figura N° 71: Código de la vista Eliminar Cargo	159
Figura N° 72: Formulario Index Cargo	160
Figura N° 73: Formulario Crear Cargo	160
Figura N° 74: Formulario Actualizar Cargo	160
Figura N° 75: Formulario Eliminar Cargo	161
Figura N° 76: Prototipo 1 Index Usuario	161
Figura N° 77: Prototipo 2 Crear Usuario	161
Figura N° 78: Prototipo 3 Actualizar Usuario	162
Figura N° 79: Prototipo 4 Eliminar Usuario	162
Figura N° 80: Código para la creación de la tabla Usuario	163
Figura N° 81: Código del modelo Usuario	163
Figura N° 82: Código del controlador Usuario	164
Figura N° 83: Código de la vista Index Usuario	164
Figura N° 84: Código de la vista Crear Usuario	165
Figura N° 85: Código de la vista Actualizar Usuario	166
Figura N° 86: Código la vista Eliminar Usuario	167
Figura N° 87: Formulario Index Usuarios	167
Figura N° 88: Formulario Crear Usuarios	168
Figura N° 89: Formulario Actualizar Usuarios	169
Figura N° 90: Formulario Eliminar Usuario	169
Figura N° 91: Prototipo 1 Index Cuenta	170
Figura N° 92: Prototipo 2 Crear Cuenta	170
Figura N° 93: Prototipo 3 Editar Cuenta	170
Figura N° 94: Prototipo 4 Eliminar Cuenta	171
Figura N° 95: Código para la creación de la tabla AspnetUser	171
Figura N° 96: Código del controlador Cuenta	172
Figura N° 97: Código de la vista Index Cuenta	172

Figura N° 98: Código de la vista Crear Cuenta	173
Figura N° 99: Código de la vista Editar Cuenta	174
Figura N° 100: Código de la vista Eliminar Cuenta	175
Figura N° 101: Código de la vista Eliminar Cuenta	175
Figura N° 102: Código de la vista Registrar Cuenta	176
Figura N° 103: Código de la vista Editar Cuenta	176
Figura N° 104: Código de la vista Eliminar Cuenta	176
Figura N° 105: Código de la vista Index Acceso	177
Figura N° 106: Código de la vista Agregar Acceso	178
Figura N° 107: Código de la vista Editar Acceso	178
Figura N° 108: Código de la vista Eliminar Acceso	178
Figura N° 109: Código para la creación de la tabla Acceso	179
Figura N° 110: Procedimiento almacenado AccesoUsuarios	179
Figura N° 111: Código del controlador Accesos	180
Figura N° 112: Código de la vista Index Accesos	181
Figura N° 113: Código de la vista Crear Acceso	181
Figura N° 114: Código de la vista Editar Acceso	182
Figura N° 115: Código de la vista Eliminar Acceso	182
Figura N° 116: Formulario Index Acceso	183
Figura N° 117: Formulario Editar Acceso	183
Figura N° 118: Formulario Crear Acceso	183
Figura N° 119: Formulario Eliminar Acceso	184
Figura N° 120: Formulario Completado del módulo de Gestión de Usuario	184
Figura N° 121: Grafica Burndown del Sprint 1	185
Figura N° 122: Caso de uso del sprint 2	188
Figura N° 123: Diseño lógico de la base de datos del sprint 2	189
Figura N° 124: Prototipo 1 Index Hardware	190
Figura N° 125: Prototipo 2 Registrar Hardware	190
Figura N° 126: Prototipo 3 Editar Hardware	191
Figura N° 127: Prototipo 4 Eliminar Hardware	191
Figura N° 128: Código para la creación de la tabla Hardware	191
Figura N° 129: Código del modelo Hardware	192
Figura N° 130: Código del controlador Hardware	192

Figura N° 131: Código de la vista Index Hardware	193
Figura N° 132: Código de la vista Crear Hardware	193
Figura N° 133: Código de la vista Actualizar Hardware	194
Figura N° 134: Código de la vista Eliminar Hardware	194
Figura N° 135: Formulario Index Hardware	195
Figura N° 136: Formulario Crear Hardware	195
Figura N° 137: Formulario Actualizar Hardware	196
Figura N° 138: Formulario Eliminar Hardware	196
Figura N° 139: Prototipo 1 Index Software	197
Figura N° 140: Prototipo 2 Crear Software	197
Figura N° 141: Prototipo 3 Actualizar Software	198
Figura N° 142: Prototipo Eliminar Software	198
Figura N° 143: Código para la creación de la tabla Software	198
Figura N° 144: Código del modelo Software	199
Figura N° 145: Código del controlador Software	199
Figura N° 146: Código de la vista Index Software	200
Figura N° 147: Código de la vista Crear Software	200
Figura N° 148: Código de la vista Editar Software	200
Figura N° 149: Código de la vista Eliminar Software	201
Figura N° 150: Formulario Index Software	201
Figura N° 151: Formulario Crear Software	201
Figura N° 152: Formulario Actualizar Software	202
Figura N° 153: Formulario Eliminar Software	202
Figura N° 154: Prototipo 1 Index Insumos	203
Figura N° 155: Prototipo 2 Crear Insumo	203
Figura N° 156: Prototipo 3: Actualizar Insumo	203
Figura N° 157: Prototipo 4 Borrar Insumo	204
Figura N° 158: Código para la creación de la tabla Insumo	204
Figura N° 159: Código del modelo Insumo	204
Figura N° 160: Código del controlador Insumo	205
Figura N° 161: Código de la vista Index Insumo	205
Figura N° 162: Código de la vista Crear Insumo	206
Figura N° 163: Código de la vista Actuaizar Insumo	206

Figura N° 164: Código de la vista Eliminar Insumo	207
Figura N° 165: Formulario Index Insumo	207
Figura N° 166: Formulario Crear Insumo	208
Figura N° 167: Formulario Actualizar Insumo	208
Figura N° 168: Formulario Eliminar Insumo	209
Figura N° 169: Prototipo 1 Index Componentes	209
Figura N° 170: Prototipo 2 Crear Componente	210
Figura N° 171: Prototipo 3 Editar Componente	210
Figura N° 172: Prototipo Borrar Componentes	211
Figura N° 173: Código para la creación de la tabla Componente	211
Figura N° 174: Código del modelo Componente	211
Figura N° 175: Código del controlador Componente	212
Figura N° 176: Código de la vista Index Componente	213
Figura N° 177: Código de la vista Crear Componente	213
Figura N° 178: Código de la vista Actualizar Componente	214
Figura N° 179: Código de la vista Eliminar Componente	214
Figura N° 180: Formulario Index Componente	215
Figura N° 181: Formulario Crear Componente	215
Figura N° 182: Formulario Editar Componente	216
Figura N° 183: Formulario Eliminar Componente	216
Figura N° 184: Prototipo 1 Index Kits	217
Figura N° 185: Prototipo 2 Crear Kits	218
Figura N° 186: Prototipo 3 Actualizar Kits	218
Figura N° 187: Prototipo 4 Eliminar Kits	218
Figura N° 188. Código para la creación de la tabla Kits	219
Figura N° 189: Código del modelo Kits	219
Figura N° 190: Código del controlador Kits	220
Figura N° 191: Código de la vista Index Kits	220
Figura N° 192: Código de la vista Crear Kits	221
Figura N° 193: Código de la vista Editar Kits	222
Figura N° 194: Código de la vista Borrar Kits	223
Figura N° 195: Formulario Index Kits	223
Figura N° 196: Formulario Crear Kits	224

Figura N° 197: Formulario Editar Kits	224
Figura N° 198: Formulario Eliminar Kits	225
Figura N° 199: Prototipo 1 Index Asignar Insumos	225
Figura N° 200: Prototipo 2 Crear Asignar Insumos	226
Figura N° 201: Prototipo 3 Asignar Actualizar Insumos	226
Figura N° 202: Prototipo 4 Eliminar Asignar Insumos	226
Figura N° 203: Código para la creación de la tabla Asignar Insumo	227
Figura N° 204: Código del controlador Asignar Insumo	227
Figura N° 205: Código de la vista de Asignación de insumos	228
Figura N° 206: Código para la vista Crear Asignación de insumos	228
Figura N° 207: Código para la vista Actualizar Asignar Insumos	229
Figura N° 208: Formulario Index Asignación de Insumo	229
Figura N° 209: Formulario Crear Asignación de Insumos	230
Figura N° 210: Formulario Actualizar Asignación de Insumo	230
Figura N° 211: Formulario Eliminar Asignación de Insumo	231
Figura N° 212: Prototipo 1 Índice Asignación De Equipos	231
Figura N° 213: Prototipo 2 Crear Asignación de Recursos	232
Figura N° 214: Prototipo 3 Editar Asignación de Recursos	232
Figura N° 215: Prototipo 4 Borrar Asignación de Recursos	233
Figura N° 216: Código para la creación de la tabla Asignación de Recursos	233
Figura N° 217: código del modelo Asignación de Recursos	233
Figura N° 218: Código del controlador Asignación de Recursos	234
Figura N° 219: Código de la vista Índice Asignación de Recursos	234
Figura N° 220: Código de la vista Crear Asignación de Recursos	235
Figura N° 221: Código de la vista Editar Asignación de Recursos	236
Figura N° 222: Código de la vista Eliminar Asignación de Recursos	237
Figura N° 223: Formulario Index Asignación de Recursos	237
Figura N° 224: Formulario Crear Asignación de Recursos	238
Figura N° 225: Formulario Actualizar Asignación de Recursos	239
Figura N° 226: Código de la vista Eliminar Asignación de Recursos	240
Figura N° 227: Prototipo 1 Índice Licenciamientos	240
Figura N° 228: Prototipo 2 Crear Licenciamientos	241
Figura N° 229: Prototipo 3 Editar Licenciamientos	241

Figura N° 230: Prototipo 4 Eliminar Licenciamientos	241
Figura N° 231: Código para la creación de la tabla Licenciamiento	242
Figura N° 232: Código del modelo Licenciamiento	242
Figura N° 233: Código del controlador Licenciamiento	243
Figura N° 234: Código de la vista Índice Licenciamiento	243
Figura N° 235: Código de la vista Crear Licenciamiento	244
Figura N° 236: Código de la vista Editar Licenciamiento	245
Figura N° 237: Código de la vista Borrar Licenciamiento	245
Figura N° 238: Formulario Índice Licenciamiento	246
Figura N° 239: Formulario Crear Licenciamiento	246
Figura N° 240: Formulario Actualizar Licenciamiento	247
Figura N° 241: Formulario Borrar Licenciamiento	248
Figura N° 242: Grafico Burn Down Chart del Sprint 2	249
Figura N° 243: Diagrama físico de a base de datos para e Sprint 3	253
Figura N° 244: Prototipo 1 Index Cronograma de Mantenimiento	254
Figura N° 245: Prototipo 2 Crear Cronograma de Mantenimiento	254
Figura N° 246: Prototipo 3 Actualizar Cronograma de Mantenimiento	254
Figura N° 247: Prototipo 4 Eliminar Cronograma de Mantenimiento	254
Figura N° 248: Código para la creación de la tabla Cronograma de Mantenimiento	255
Figura N° 249: Código del modelo Cronograma de Mantenimiento	255
Figura N° 250: Código para el controlador Cronograma de Mantenimiento	255
Figura N° 251: Código de la vista Índice Cronograma de Mantenimiento	256
Figura N° 252: Código de la vista Crear Cronograma de Mantenimiento	256
Figura N° 253: Código de la vista Actualizar Cronograma de Mantenimiento	257
Figura N° 254: Código de la vista Eliminar Cronograma de Mantenimiento	257
Figura N° 255: Formulario Index Cronograma de Mantenimiento	257
Figura N° 256: Formulario Crear Cronograma de Mantenimiento	258
Figura N° 257: Formulario Actualizar Cronograma de Mantenimiento	258
Figura N° 258: Formulario Eliminar Cronograma de Mantenimiento	259
Figura N° 259: Prototipo 1 Index Mantenimiento Preventivo	259
Figura N° 260: Prototipo 2 Registrar Mantenimiento Preventivo	260
Figura N° 261: Prototipo 3 Actualizar Mantenimiento Preventivo	260

Figura N° 262: Prototipo 4 Borrar Mantenimiento Preventivo	260
Figura N° 263: Código para la creación de la tabla Mantenimiento Preventivo	261
Figura N° 264: código del modelo Mantenimiento Preventivo	261
Figura N° 265: Código del controlador Mantenimiento Preventivo	262
Figura N° 266: Código de la vista Index Mantenimiento Preventivo	262
Figura N° 267: Código de la vista Crear Mantenimiento Preventivo	263
Figura N° 268: Código de la vista Actualizar Mantenimiento Preventivo	264
Figura N° 269: Código de la vista Eliminar Mantenimiento Preventivo	264
Figura N° 270: Formulario Index Mantenimiento Preventivo	265
Figura N° 271: Formulario Crear Mantenimiento Preventivo	265
Figura N° 272: Formulario Actualizar Mantenimiento Preventivo	265
Figura N° 273: Formulario Eliminar Mantenimiento Preventivo	266
Figura N° 274: Prototipo 1 Index Atención de Ticket	266
Figura N° 275: Prototipo 2 Agregar Atención de Ticket	267
Figura N° 276: Prototipo 3 Actualizar Atención de Ticket	267
Figura N° 277: Prototipo 4 Eliminar Atención de Ticket	268
Figura N° 278: Código para la creación de la tabla Atención de Ticket	268
Figura N° 279. Código del modelo Atención de Ticket	269
Figura N° 280: Código del controlador Atención Ticket	269
Figura N° 281: Código de la vista Índice Atención de Ticket	270
Figura N° 282: Código de la vista Crear Atención de Ticket	271
Figura N° 283: Código de la vista Editar Atención de Ticket	272
Figura N° 284: Código de la vista Borrar Atención de Ticket	273
Figura N° 285: Formulario Índice Generación de Ticket	274
Figura N° 286: Formulario Crear Atención de Solicitudes	275
Figura N° 287: Formulario Actualizar Atención de Solicitudes	276
Figura N° 288: Formulario Eliminar Atención de Solicitudes	277
Figura N° 289: Prototipo 1 Index Incidencia	278
Figura N° 290: Prototipo 2 Crear Incidencia	278
Figura N° 291: Prototipo 3 Actualizar Incidencia	279
Figura N° 292: Prototipo 4 Borrar Incidencia	279
Figura N° 293: Código para la creación de la tabla Incidencia	280
Figura N° 294: Código del modelo Incidencia	280

Figura N° 295: Código del controlador Incidencia	281
Figura N° 296: Código de la vista Index Incidencia	281
Figura N° 297: Código de la vista Crear Incidencia	282
Figura N° 298: Código de la vista Editar Incidencia	283
Figura N° 299: Código de la vista Eliminar Incidencia	284
Figura N° 300: Formulario Index Incidencia	285
Figura N° 301: Formulario Registrar Incidencia	285
Figura N° 302: Formulario Actualizar Incidencia	286
Figura N° 303: Formulario Eliminar Incidencia	287
Figura N° 304: Prototipo 1 Index Problema	287
Figura N° 305: Prototipo 2 Registrar Problema	288
Figura N° 306: Prototipo 3 Editar Problema	288
Figura N° 307: Prototipo 4 Eliminar Problema	289
Figura N° 308: Código para la creación de la tabla Problema	289
Figura N° 309: Código del modelo Problema	290
Figura N° 310: Código del controlador Problema	290
Figura N° 311: Código de la vista Index Problema	291
Figura N° 312: Código de la vista Crear Problema	292
Figura N° 313: Código de la vista Actualizar Problema	293
Figura N° 314: Código de la vista Eliminar Problema	294
Figura N° 315: Formulario Index Problema	295
Figura N° 316: Formulario Crear Problema	295
Figura N° 317: Formulario Editar Problema	296
Figura N° 318: Formulario Eliminar Problema	297
Figura N° 319: Prototipo 1 Índice Requerimientos	298
Figura N° 320: Prototipo 2 Crear Requerimientos	298
Figura N° 321: Prototipo 3 Editar Requerimientos	299
Figura N° 322: Prototipo 4 Eliminar Requerimientos	299
Figura N° 323: Código para la creación de la tabla Requerimiento	299
Figura N° 324: Código del modelo Requerimiento	300
Figura N° 325: Código del controlador Requerimiento	300
Figura N° 326: Código de la vista Índice Requerimiento	301
Figura N° 327: Código de la vista Crear Requerimiento	302

Figura N° 328: Código de la vista Editar Requerimiento	303
Figura N° 329: Código de la vista Eliminar Requerimiento	304
Figura N° 330: Formulario Index Requerimiento	305
Figura N° 331: Formulario Crear Requerimiento	305
Figura N° 332: Formulario Editar Requerimiento	306
Figura N° 333: Formulario Eliminar Requerimiento	307
Figura N° 335: Procedimiento Almacenado IndiceCoberturaMant	307
Figura N° 337: : Código del modelo Índice de Cobertura	308
Figura N° 338: Código de la Vista Índice de Cobertura	309
Figura N° 339: Indicador de Índice de Cobertura	309
Figura N° 334: Procedimiento Almacenado Reporte General de Activos	310
Figura N° 336: Reporte de activos operativos	310
Figura N° 340: Base de datos completo del proyecto	311
Figura N° 341: Grafica Burn Down del Sprint 3	312

Índice de Tablas

Tabla N° 1: Equipo Scrum	121
Tabla N° 2: Matriz de impacto de prioridades	125
Tabla N° 3: Pila del producto inicial	126
Tabla N° 4: Lista de Sprint	127
Tabla N° 5: Planificación Sprint 1	128
Tabla N° 6: Planificación Sprint 2	129
Tabla N° 7: Planificación Sprint 3	129
Tabla N° 8: TaskBoard Sprint1 Semana 1	131
Tabla N° 9: TaskBoard sprint 1 semana 2	177
Tabla N° 10: TaskBoard Sprint 1 Semana 3	184
Tabla N° 11: Burndown del Sprint 1	185
Tabla N° 12: TaskBoard Sprint 2 Semana 4	188
Tabla N° 13: TaskBoard del Sprint 2 Semana 5	217
Tabla N° 14: TaskBoard del Sprint 2 Semana 6	248
Tabla N° 15: BurnDown del Sprint 2	249
Tabla N° 16: TaskBoard del Sprint 3 Semana 7	252
Tabla N° 17: TaskBoard del Sprint 3 Semana 8	277
Tabla N° 18: TaskBoard del Sprint 3 Semana 9	297
Tabla N° 19: BurnDown del Sprint 3	312

Sistema web para el control de los servicios del área de sistemas en la Compañía Naviera Natalia S.A.C.

Plan de Metodología de Desarrollo

Introducción

En esta documentación se dará a conocer la información correspondiente al desarrollo de un sistema web para el control de los servicios del área de sistemas en la Compañía Naviera Natalia S.A.C. Se utilizó Scrum como metodología de desarrollo y por lo tanto se mostrará los procesos iterativos e incrementales que conlleva el proyecto, así como sus roles y artefactos que lo conforman en cada tarea y avance.

Propósito

Organizar las asignaciones de las tareas y roles para cada uno de los participantes, trabajar de forma colaborativa estableciendo el tiempo de entrega, solucionando cual tipo de percance y entregar la información para llevar el control del desarrollo del proyecto.

Alcance

La Implementación del sistema web para el control de los servicios del área de sistemas, tendrá un plazo máximo de tres meses y contemplará los módulos de Gestión de usuarios, Recursos y Servicios de Tecnología de Información.

Descripción del Proyecto

La Compañía Naviera Natalia SAC. Requiere contar con un sistema que les permita llevar un mejor control de los servicios que brinda el área de sistemas a los equipos informáticos ubicados en cada uno de sus sedes (Lima, Talara y Zorritos), como parte de la mejora y optimización de los servicios brindados por el área de sistemas, requiere tener la información centralizada en un sistema que les permita monitorizar en tiempo real, el control de los recursos de tecnología informática asignados a cada usuario, el estado de los mantenimientos, el inventario de los equipos informáticos, las soluciones aplicadas a los mantenimientos correctivos realizados

en cualquier sede de la compañía, de manera que se pueda disponer de materiales, equipos, repuestos y componentes que se necesiten con anticipación. Es por ello que tiene la necesidad de mejorar el control de los servicios que brindan el área de sistemas de la Compañía Naviera Natalia SAC a los equipos informáticos asignados a los usuarios administrativos de la compañía, a través de la implementación de un sistema web.

El alcance de este proyecto será solo para uso del personal de informática, soporte y sistemas ubicados en las sedes de Lima, Talara y Zorritos, no contemplará la implementación de la infraestructura física, puesto que la compañía ya dispone además de ella de los servicios de publicación web, por lo que solo nos basaremos en el desarrollo del sistema web.

Descripción general de la Metodología

Fundamentación

Se escogió la metodología Scrum porque está más orientado a la acción y permite la entrega al menor costo y tiempo posible de los entregables pues aclara los objetivos, fomenta un espíritu cooperativo de aprendizaje y permite corregir los errores de manera puntual en la revisión de cada entregable iterativo, El Sistema web para el control de los servicios del área de sistemas, permite el desarrollo de manera iterativa de las funcionalidades que se requiera. Por cada módulo terminado se realizarán entregables al cliente para conocer los resultados y mejorar su expectativa, esto se realizará en tiempos cortos durante el desarrollo del sistema web para poder mitigar con anticipación los riesgos asociados al desarrollo del proyecto, asegurándonos de esta el cumplimiento de los requerimientos funcionales solicitados por el cliente y su mejora continua a través de la comunicación efectiva establecida con el cliente y los integrantes del desarrollo del sistema web. Es importante enfatizar que para que la metodología Scrum aplicada a nuestro desarrollo web sea exitosa, cada integrante del equipo tiene que comprender y concientizarse en los principales valores que aporta el Scrum, como el respeto, la delegación de funciones, la autodisciplina y responsabilidad, visibilidad y transparencia y generación de valor para el cliente.

Definición de Roles del proyecto

A continuación, definiremos los roles del proyecto:

Como se puede observar en la tabla 2, **El Product Over** (Dueño del producto) es la persona que representa la voz del cliente y el resto de interesados en el proyecto para listar las funcionalidades del sistema, planificar su inicio, definir los objetivos y garantizar equipo este trabajando de manera adecuado para el cumplimiento de los objetivos asignados, el cual para nuestro proyecto este rol se asignará al Ingeniero Hugo Candela Ríos, persona a cargo de la jefatura corporativa de TI (Tecnología de Información) del MigivaGroup. **El Scrum Master** Es la persona que guiará y ayudará al equipo de trabajo a mantenerse activo y productivo para asegurar el cumplimiento de los objetivos, entre sus funciones está el de realizar seguimiento e informes de los avances del proyecto, validar y supervisar los costos y plazos establecidos para su entrega. **El Team Scrum** Es el equipo asignado para el desarrollo y entrega del producto y de cada Sprint que conforma el producto, este equipo de trabajo tiene una estructura organizacional horizontal, con capacidad de autogestión.

Scrum Team (Equipo Scrum)

Tabla N° 1: Equipo Scrum

Persona	Cargo	Contacto	Rol
Aiquipa Tello Alex	Scrum Master	alex.aiquipa@constructoraroble.com	Scrum Master
Candela Ríos Hugo	Gerente de Informática	Hugo.candela@migivagroup.com	Product Owner
Aiquipa Tello Alex	Investigador	alex.aiquipa@constructoraroble.com	Desarrollador

Fuente: Compañía Naviera Natalia SAC

Historias de Usuario

Historia 1

Figura N° 1: Historia de Usuario 1

Historia de Usuario N° 1	Prioridad 1
Condiciones ✓ El sistema debe contar con una página de inicio de sesión, el cual debe contener al usuario y la contraseña.	T. Estimado 4
Restricciones • Solo podrán acceder los usuarios que administran el sistema.	

© Compañía Naviera Natalia SAC, 2018

Historia 2

Figura N° 2: Historia de Usuario 2

Historia de Usuario N° 2	Prioridad 2
Condiciones ✓ El sistema debe contar con el módulo de mantenimiento para los Tipos de empresa, Empresa, Áreas, Cargos y Usuarios. ✓ El sistema debe permitir la creación de Cuentas de Usuarios y permitir administrar su nivel de acceso al sistema.	T. Estimado 82
Restricciones • Solo podran acceder la los modulos los usuarios que administran el sistema.	

© Compañía Naviera Natalia SAC, 2018

Historia 3

Figura N° 3: Historia de Usuario 3

Historia de Usuario N° 3	Prioridad 3
Condiciones <ul style="list-style-type: none">✓ El sistema debe permitir registrar y dar mantenimiento a los activos de hardware y software.✓ El sistema debe permitir registrar y dar mantenimiento a los insumos y componentes de los equipos informáticos	T. Estimado 44
Restricciones <ul style="list-style-type: none">• Solo podran acceder la los modulos los usuarios que administran el sistema.	

© Compañía Naviera Natalia SAC, 2018

Historia 4

Figura N° 4: Historia de Usuario 4

Historia de Usuario N° 4	Prioridad 4
Condiciones <ul style="list-style-type: none">✓ El sistema debe permitir al Administrador, registrar los componentes que conforman un equipo informático .✓ El sistema debe permitir registrar la asignación de insumos, equipos y licencias de software por usuario.	T. Estimado 56
Restricciones <ul style="list-style-type: none">• Solo podran acceder la los modulos los usuarios que administran el sistema.	

© Compañía Naviera Natalia SAC, 2018

Historia 5

Figura N° 5: Historia de Usuario 5

Historia de Usuario N° 5		Prioridad 5
Condiciones <ul style="list-style-type: none">✓ El sistema deberá permitir al Administrador generar un cronograma de mantenimiento preventivo.✓ El sistema debe permitir registrar los mantenimientos preventivos y las atenciones realizados a los equipos informáticos.		T. Estimado 32
Restricciones <ul style="list-style-type: none">• Solo podran acceder la los modulos los usuarios que administran el sistema.		

© Compañía Naviera Natalia SAC, 2018

Historia 6

Figura N° 6: Historia de Usuario 6

Historia de Usuario N° 6		Prioridad 6
Condiciones <ul style="list-style-type: none">✓ El sistema debe permitir al Administrador registrar las incidencias y problemas de los equipos informáticos.✓ El sistema debe permitir al Administrador registrar los requerimientos de insumos y componentes realizados por el personal de soporte para la reparación de equipos informáticos.		T. Estimado 72
Restricciones <ul style="list-style-type: none">• Solo podran acceder la los modulos los usuarios que administran el sistema.		

© Compañía Naviera Natalia SAC, 2018

Historia 7

Figura N° 7: Historia de Usuario 7

Historia de Usuario N° 7		Prioridad 7
Condiciones ✓ El sistema debe permitir al administrador visualizar el indicador de cumplimiento de las atenciones realizadas. ✓ El sistema debe permitir al administrador visualizar el indicador de índice de cobertura de mantenimiento y grado de cumplimiento.		T. Estimado 32
Restricciones • Solo podran acceder la los modulos los usuarios que administran el sistema.		

© Compañía Naviera Natalia SAC, 2018

Matriz de Impacto

Tabla N° 2: Matriz de impacto de prioridades

Prioridad	
Muy Alta	1
Alta	2
Media	3
Baja	4
Muy Baja	5

Fuente: Compañía Naviera Natalia SAC 2018

Product Backlog

En la siguiente tabla número 3 se muestra todos los requerimientos funcionales, debidamente especificados con su número de historia, tiempo estimado, tiempo realizado y su prioridad, posteriormente en el Product Backlog se detalla la cantidad de Sprints, los requerimientos funcionales de la Pila de Producto y sus respectivos prioridades y tiempos estimados, tal cual se muestra en la tabla número 4.

Pila del Producto (Product Backlog)

Tabla N° 3: Pila del producto inicial

Requerimientos Funcionales	Historias	T. E.	T.R.	P.
RF01: El Sistema debe de tener un formulario que permita autenticar el ingreso a los administradores del Sistema.	H1	4	4	1
RF02: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los tipos de empresas de la Compañía.	H2	8	8	2
RF03: El sistema debe permitir al Administrador, administrar y dar mantenimiento a las empresas colaboradoras de la Compañía	H2	10	10	2
RF04: El sistema debe permitir al Administrador, administrar y dar mantenimiento a las áreas que dispone la Compañía.	H2	8	8	2
RF05: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los cargos de los usuarios que laboraran en la Compañía.	H2	8	8	2
RF06: El sistema debe permitir al Administrador, administrar y dar mantenimiento al registro de usuarios que laboran en la Compañía.	H2	16	16	2
RF07: El sistema debe permitir al Administrador, crear de Cuentas de usuarios del sistema.	H2	16	16	2
RF08: : El sistema debe permitir al Administrador, administrar el nivel de acceso al sistema de cada cuenta de usuario creada.	H2	16	16	2
RF09: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los activos hardware de la Compañía.	H3	12	12	3
RF10: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los activos software de la Compañía.	H3	8	8	3
RF11: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los insumos de los equipos de oficina de la Compañía.	H3	8	8	3
RF12: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los registros de los componentes de los equipos de oficina de la Compañía.	H3	16	16	3
RF13: El sistema debe permitir al Administrador, administrar y registrar los componentes que conforman un equipo.	H4	16	16	4
RF14: El sistema debe permitir al Administrador, registrar la asignación de los insumos a los equipos informáticos de la Compañía.	H4	12	12	4
RF15: El sistema debe permitir al Administrador, registrar los equipos informáticos asignados al personal administrativo de la Compañía.	H4	16	16	4
RF16: El sistema debe permitir registrar los licenciamientos de los recursos asignados al personal administrativo.	H4	12	12	4
RF17: El sistema debe permitir al administrador, administrar y dar mantenimiento al cronograma de mantenimiento preventivo.	H5	8	8	5
RF18: El sistema debe permitir al administrador monitorizar el estado de los mantenimientos realizados.	H5	8	8	5
RF19: El sistema debe permitir, registrar las atenciones solicitadas por los usuarios administrativos de la compañía..	H5	16	16	6
RF20: El sistema debe permitir al administrador, administrar y dar mantenimiento al registro de las incidencias de equipos informáticos reportados.	H6	24	24	6
RF21: El sistema debe permitir al administrador, administrar y dar mantenimiento al registro de los problema reportados.	H6	24	24	6
RF22: El sistema debe permitir al administrador, administrar y dar mantenimiento al registro requerimiento de insumos o componentes solicitados para reparación de los equipos informáticos de la compañía.	H6	24	24	6
RF23: El sistema debe permitir al administrador visualizar el indicador de índice de cobertura de mantenimiento	H7	16	16	7
RF24: El sistema debe permitir al administrador visualizar el indicador de cumplimiento de las atenciones realizadas.	H7	16	16	7

Fuente: Compañía Naviera Natalia SAC, 2018

Entregables por Sprint

A continuación, agruparemos todos los requerimientos funcionales solicitados en 3 sprint, las cuales se muestra en la siguiente tabla.

Tabla N° 4: Lista de Sprint

N° Sprint	Requerimientos Funcionales	Historias	T.E.	T.R.	P.
SPRINT 1 (13 Días)	RF01: El Sistema debe de tener un formulario que permita autenticar el ingreso a los administradores del Sistema.	H1	4	4	1
	RF02: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los tipos de empresas de la Compañía.	H2	8	8	2
	RF03: El sistema debe permitir al Administrador, administrar y dar mantenimiento a las empresas colaboradoras de la Compañía	H2	10	10	2
	RF04: El sistema debe permitir al Administrador, administrar y dar mantenimiento a las áreas que dispone la Compañía.	H2	8	8	2
	RF05: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los cargos de los usuarios que laboraran en la Compañía.	H2	8	8	2
	RF06: El sistema debe permitir al Administrador, administrar y dar mantenimiento al registro de usuarios que laboran en la Compañía.	H2	16	16	2
	RF07: El sistema debe permitir al Administrador, crear de Cuentas de usuarios del sistema	H2	16	16	2
	RF08: : El sistema debe permitir al Administrador, administrar el nivel de acceso al sistema de cada cuenta de usuario creada.	H2	16	16	2
SPRINT 2 (16 Días)	RF09: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los activos hardware de la Compañía.	H3	12	12	3
	RF10: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los activos software de la Compañía.	H3	8	8	3
	RF11: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los insumos de los equipos de oficina de la Compañía.	H3	8	8	3
	RF12: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los registros de los componentes de los equipos de oficina de la Compañía.	H3	16	16	3
	RF13: El sistema debe permitir al Administrador, administrar y registrar los componentes que conforman un equipo.	H4	16	16	4
	RF14: El sistema debe permitir al Administrador, registrar la asignación de los insumos a los equipos informáticos de la Compañía.	H4	12	12	4
	RF15: El sistema debe permitir al Administrador, registrar los equipos informáticos asignados al personal administrativo de la Compañía.	H4	16	16	4
	RF16: El sistema debe permitir registrar los licenciamientos de los recursos asignados al personal administrativo.	H4	12	12	4
SPRINT 3 (17 Días)	RF17: El sistema debe permitir al administrador, administrar y dar mantenimiento al cronograma de mantenimiento preventivo.	H5	8	8	5
	RF18: El sistema debe permitir al administrador monitorizar el estado de los mantenimientos realizados.	H5	8	8	5
	RF19: El sistema debe permitir, registrar las atenciones solicitadas por los usuarios.	H5	16	16	6
	RF20: El sistema debe permitir al administrador, administrar y dar mantenimiento al registro de las incidencias de equipos informáticos reportados.	H6	24	24	6
	RF21: El sistema debe permitir al administrador, administrar y dar mantenimiento al registro de los problema reportados.	H6	24	24	6
	RF22: El sistema debe permitir al administrador, administrar y dar mantenimiento al registro requerimiento de insumos o componentes solicitados para reparación de los equipos informáticos de la compañía.	H6	24	24	6
	RF23: El sistema debe permitir al administrador visualizar el indicador de índice de cobertura de mantenimiento	H7	16	16	7
	RF24: El sistema debe permitir al administrador visualizar el indicador de cumplimiento de las atenciones realizadas.	H7	16	16	7

Fuente: Compañía Naviera Natalia SAC, 2018

Planificación de los Sprints

Una vez concluido la construcción de los Sprints, se procederá a planificar su desarrollo y entrega de cada Sprint, para ello se hará uso de herramientas que nos permitan la visualización del estado de avance de cada Sprints como es el caso del TaskBoard y el BurnDown para mostrar el tiempo con el que se está desarrollando el proyecto, de tal manera que podamos saber que historias o actividades demandan más tiempo para así poder planificar mejor su desarrollo, la entrega y su revisión interactiva de cada Sprints nos permitirá obtener una retrospectiva de las acciones de mejora en los desarrollos posteriores.

A continuación, mostraremos la planificación de los Sprints para este proyecto

Planificación Sprint 1

Tabla N° 5: Planificación Sprint 1

SPRINT 1	
Fecha de inicio:	13/07/2018
Fecha de fin:	31/07/2018
Revisión de los avances:	Las revisiones se realizarán semanalmente. Las fechas de revisión serán las siguientes: 20/07/2018 27/07/2018 31/07/2018
Tareas a desarrollar:	<ul style="list-style-type: none">• Formulario de inicio de sesión para el acceso al Sistema.• CRUD para los tipos de empresas de la Compañía.• CRUD para las empresas colaboradoras de la Compañía• CRUD para las áreas que dispone la Compañía.• CRUD para los cargos de los usuarios que laboraran en la Compañía.• CRUD para el mantenimiento de los usuarios que laboran en la Compañía.• Registrar Cuentas de usuarios del sistema.• Administrar el nivel de acceso de los usuarios al sistema.

Fuente: Elaboración propia

Planificación Sprint 2

Tabla N° 6: Planificación Sprint 2

SPRINT 2	
Fecha de inicio:	01/08/2018
Fecha de fin:	22/08/2018
Revisión de los avances:	Las revisiones se realizarán en las siguientes fechas: 08/08/2018 15/08/2018 22/08/2018
Tareas a desarrollar:	<ul style="list-style-type: none">• CRUD para los activos hardware de la Compañía.• CRUD para los activos software de la Compañía.• CRUD para los insumos de los equipos de impresión.• CRUD para los componentes de las computadoras.• Registrar de los componentes que contiene un equipo de cómputo.• Registrar la asignación de los insumos a los equipos de impresión.• Registrar los equipos informáticos asignados al personal administrativo de la Compañía.• Registrar los licenciamientos de los equipos informáticos.

Fuente: Elaboración propia

Planificación Sprint 3

Tabla N° 7: Planificación Sprint 3

SPRINT 3	
Fecha de inicio:	23/08/2018
Fecha de fin:	14/09/2018
Revisión de los avances:	Las revisiones se realizarán semanalmente. Las fechas de revisión serán las siguientes: 31/08/2018 07/09/2018 14/09/2018
Tareas a desarrollar:	<ul style="list-style-type: none">• Administrar el estado de los mantenimientos realizados.• Registrar las atenciones solicitadas por los usuarios.• Administrar y dar mantenimiento al registro de las incidencias de equipos informáticos reportados.• Administrar y dar mantenimiento al registro de problemas reportados.• Registrar y administrar los requerimientos solicitados para reparación de los equipos informáticos.• Monitorizar el indicador de índice de cobertura de mantenimiento• Monitorizar el indicador de cumplimiento de las atenciones realizadas.

Fuente: Elaboración propia

ACTA DE REUNIÓN N° 1

APERTURA DEL SPRINT 1

DATOS

Empresa / Organización	Compañía Naviera Natalia SAC.
Proyecto	Sistema web para el control de servicios del área de sistemas en la compañía Naviera Natalia SAC

PARTICIPANTES


Rol	Nombre
Dueño del Producto	Candela Ríos Hugo
Equipo	Aiquipa Tello Alex

ACUERDOS

A continuación se listan las funcionalidades que se de desarrollaran e implementaran en el Sprint 1

Historias
El sistema debe de tener un formulario que permita autenticar el ingreso a los usuarios del sistema web.
El sistema debe permitir al usuario autenticado registrar y actualizar los tipos de empresas de la Compañía.
El sistema debe permitir al usuario autenticado registrar y actualizar las empresas colaboradoras de la Compañía.
El sistema debe permitir al usuario autenticado registrar y actualizar las empresas a las áreas que dispone la Compañía.
El sistema debe permitir usuario autenticado registrar y actualizar a los cargos de los usuarios que laboran en la Compañía.
El sistema debe permitir usuario autenticado registrar y actualizar los usuarios que laboran en la Compañía.
El sistema debe permitir al Administrador, crear y modificar cuentas de usuarios del sistema web.
El sistema debe permitir al Administrador clasificar el nivel de acceso de las cuentas de los usuarios del sistema web.

Firmas:


ING. HUGO CANDELA
JEFE DE SISTEMAS CORPORATIVO

DUEÑO DEL PRODUCTO


ALEX ALDO AIQUIPA TELLO
ADMINISTRADOR DE REDES Y COMUNICACIONES

EQUIPO

Sprint N° 1: Story Point 86

TaskBoard Semana 1 del Sprint 1

Tabla N° 8: TaskBoard Sprint1 Semana 1

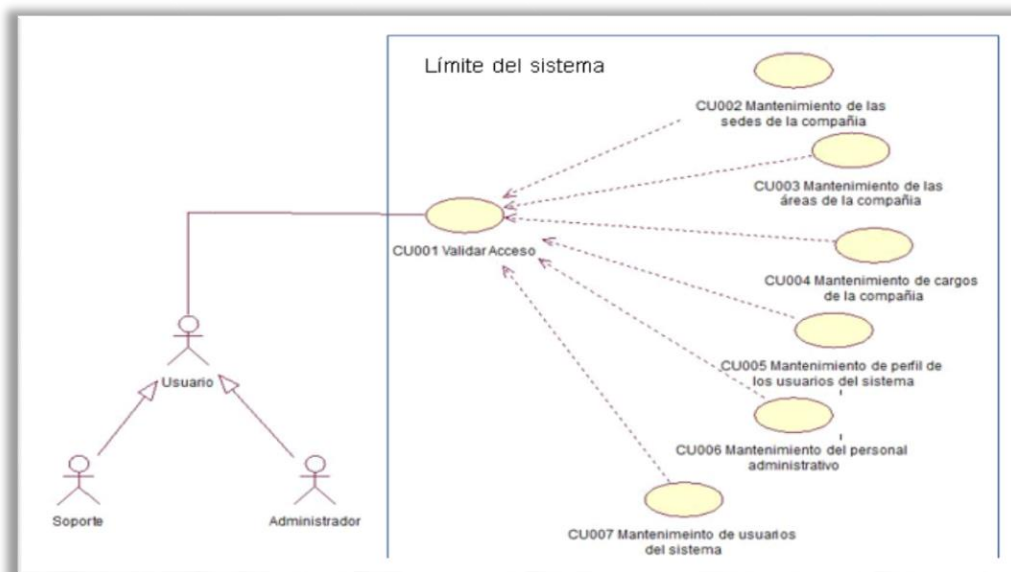
TaskBoard S1				
Sistema web para el control de os servicios de área de sistemas en la compañía Naviera Natalia SAC			Fecha Inicio: 13/07/2018 Fecha final: 19/07/2018	
	Historias por cada Sprint	Pendiente	En curso	Hecho
SPRINT N° 1	Formulario de inicio de sesión para el acceso al Sistema.			X
	CRUD para los tipos de empresas de la Compañía.			X
	CRUD para las empresas colaboradoras de la Compañía			X
	CRUD para las áreas que dispone la Compañía.			X
	CRUD para los cargos de los usuarios que laboraran en la Compañía.		X	
	CRUD para el mantenimiento de los usuarios que laboran en la Compañía.	X		
	Registrar Cuentas de usuarios del sistema.	X		
	Administrar el nivel de acceso de los usuarios al sistema.	X		

Fuente: Elaboración Propia

Análisis Sprint 1

Caso de uso: Diagrama de caso de uso "Sprint 1"

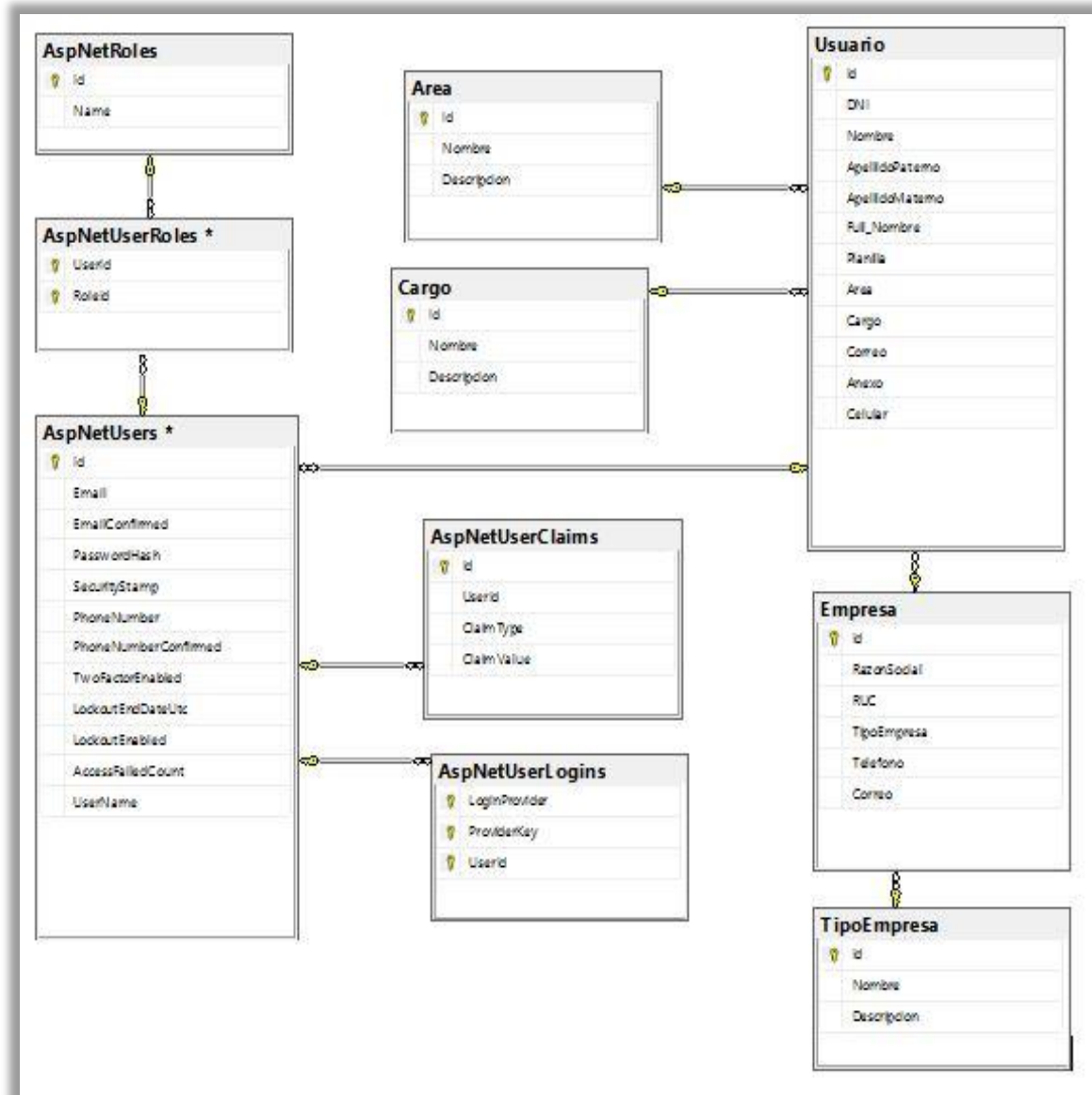
Figura N° 8: Caso de uso Sprint 1



Fuente: Elaboración Propia

Diagrama Físico de la base de datos Sprint 1

Figura N° 9. Diagrama Físico de la base de datos



Fuente: Elaboración Propia

Requerimiento RF1

RF01: El sistema debe de tener un formulario que permita autenticar el ingreso a los usuarios del sistema web.

Diseño

Prototipo 1 RF1.

Figura N° 10: Prototipo de Inicio De Sesión



El prototipo muestra un formulario con el título "Iniciar Sesión". Hay dos campos de entrada: "Nombre usuario" con el tipo de dato "text" y "Contraseña" con caracteres ocultos por asteriscos. Debajo de los campos hay un botón que dice "Iniciar Sesión".

Fuente: Elaboración Propia

Implementación

Codificación del Requerimiento funcional 01

Para la creación de usuarios y roles, ASP.net MVC dispone de una plantilla que permite generar de manera automática la tabla de **AspNetUser**, **AspNetRoles** y **AspNetUserRoles**, para lograr ello, solo basta con cambiar el modo de **Autenticación a Cuentas de Usuarios Individuales** al momento de crear una aplicación web Asp.Net con plantilla MVC, esto habilitara en el menú horizontal 2 link de acción de direccionamiento a la vista parcial de **inicio de sesión** y **registro**, para que el inicio de sesión se realice a través del nombre de usuario y no de la cuenta email que viene predeterminada, solo bastaría con realizar algunas modificaciones en el controlador de **AccountController.cs** en **Post /Account/Login** :

Figura N° 11. Modificación código Post /Account/Login

```
var result = await SignInManager.PasswordSignInAsync(model.UserName,
    model.Password, model.RememberMe, shouldLockout: true);
```

Fuente: Elaboración Propia

en el **Post /Account/Register**:

Figura N° 12. Modificación del código Post /Account/Register

```
var user = new ApplicationUser { UserName = model.UserName, Email = model.Email };
```

Fuente: Elaboración Propia

la vista de **Login** como se muestra a continuación:

Figura N° 13. Modificación del código vista Login

```
@Html.TextBoxFor(m => m.UserName, new { @class = "form-control", placeholder = "username" })
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
```

Fuente: Elaboración Propia

y en la vista de **Register**:

Figura N° 14: Modificación del código vista Register

```
<div class="form-group">
  @Html.LabelFor(m => m.UserName, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    @Html.TextBoxFor(m => m.UserName, new { @class = "form-control" })
  </div>
</div>
```

Fuente: Elaboración Propia

En la siguiente figura se muestra la codificación para el controlador de cuentas:

Figura N° 15. Acción para el botón ingresar del controlador Account

```
public async Task<ActionResult> VerifyCode(VerifyCodeViewModel model)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    var result = await SignInManager.TwoFactorSignInAsync(model.Provider, model.Code, isPersistent: model.RememberMe,
        rememberBrowser: model.RememberBrowser);
    switch (result)
    {
        case SignInStatus.Success:
            return RedirectToLocal(model.ReturnUrl);
        case SignInStatus.LockedOut:
            return View("Lockout");
        case SignInStatus.Failure:
        default:
            ModelState.AddModelError("", "Código no válido.");
            return View(model);
    }
}
```

Fuente: Elaboración Propia

Posteriormente le damos los estilos correspondientes para su personalización:

Figura N° 16: Estilos para la vista Login

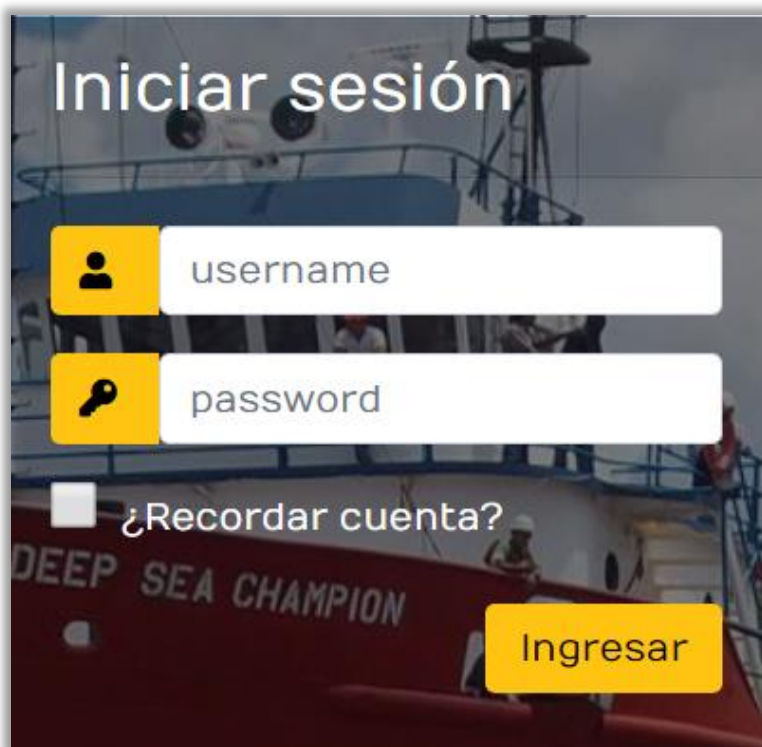
```
<head>
<title>Login Page</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8FJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdKnLPMO" crossorigin="anonymous">
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css"
integrity="sha384-mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU" crossorigin="anonymous">
<link href="~/Content/all.css" rel="stylesheet" />
<link href="~/Content/Styles.css" rel="stylesheet" />
</head>
<body>
@using (Html.BeginForm("Login", "Account", new { returnUrl = ViewBag.ReturnUrl }, FormMethod.Post, new { @class = "form-horizontal", role = "form" }))
{
    @Html.AntiForgeryToken()
    <div class="container">
        <div class="d-flex justify-content-center h-100">
            <div class="card">
                <div class="card-header">
                    <h3>Iniciar sesión</h3>
                </div>
                <div class="card-body">
                    <form>
                        <div class="input-group form-group">
                            <div class="input-group-prepend">
                                <span class="input-group-text"><i class="fas fa-user"></i></span>
                                @Html.TextBoxFor(m => m.UserName, new { @class = "form-control", placeholder = "username" })
                                @Html.ValidationSummary(true, "", new { @class = "text-danger" })
                            </div>
                        </div>
                        <div class="input-group form-group">
                            <div class="input-group-prepend">
                                <span class="input-group-text"><i class="fas fa-key"></i></span>
                                @Html.PasswordFor(m => m.Password, new { @class = "form-control", placeholder = "password" })
                                @Html.ValidationMessageFor(m => m.Password, "", new { @class = "text-danger" })
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>

```

Fuente: Elaboración Propia

GUI

Figura N° 17: Formulario Iniciar Sesión



Fuente: Elaboración Propia

Requerimiento RF2

RF02: El sistema debe permitir al usuario autenticado registrar y actualizar los tipos de empresas de la Compañía Naviera Natalia S.A.C.

Diseño

Prototipo 1 RF02

Figura N° 18: Prototipo 1 Index Tipo De Empresa

Nombre	Descripción	Editar	Detalle	Eliminar
		Editar	Detalle	Eliminar
		Editar	Detalle	Eliminar

Fuente: Elaboración Propia

Prototipo 2 RF02

Figura N° 19: Prototipo 2 Crear Tipo De Empresa

Crear
Tipo de Empresa
Nombre

[Regresar al listado](#)
© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 3 RF02

Figura N° 20: Prototipo 3 Actualizar Tipo De Empresa

Actualizar
Tipo de Empresa
Nombre

[Regresar al listado](#)
© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 4 RF02

Figura N° 21: Prototipo 4 Borrar Tipo De Empresa

El prototipo muestra una interfaz de usuario con el título "Borrar". Debajo del título, se pregunta "¿Estas Seguro que deseas borrarlos?". Hay un campo de entrada etiquetado "Codigo : Label". Debajo del campo, hay un botón "Eliminar" y un enlace "Regresar al listado". En la parte inferior, hay un pie de página que dice "© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas".

Fuente: Elaboración Propia

Implementación

Creación de la tabla TipoEmpresa

El siguiente código crea la tabla TipoEmpresa:

Figura N° 22: Código de creación de la tabla tipo de empresa

```
1 CREATE TABLE [dbo].[TipoEmpresa] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nombre] NVARCHAR (30) NOT NULL,  
4     [Descripcion] NVARCHAR (200) NULL,  
5     PRIMARY KEY CLUSTERED ([Id] ASC)  
6 );  
7
```

Fuente: Elaboración Propia

Codificación del modelo de tipo de empresa:

Figura N° 23: Código del modelo "tipo de empresa"

```
public partial class TipoEmpresa  
{  
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]  
    0 referencias | 0 excepciones  
    public TipoEmpresa()  
    {this.Empresa = new HashSet<Empresa>(); }  
    0 referencias | 0 excepciones  
    public int Id { get; set; }  
    [RegularExpression("[a-zA-Z]+$", ErrorMessage = "Este campo debe contener solo letras")]  
    [Required(ErrorMessage = "Este campo es obligatorio")]  
    0 referencias | 0 excepciones  
    public string Nombre { get; set; }  
    0 referencias | 0 excepciones  
    public string Descripcion { get; set; }  
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]  
    1 referencia | 0 excepciones  
    public virtual ICollection<Empresa> Empresa { get; set; }  
}
```

Fuente: Elaboración Propia

Figura N° 24: Código del controlador Tipo De Empresa

```

public class TipoEmpresasController : Controller
{private WebConSerSisEntities db = new WebConSerSisEntities();
  0 referencias | 3 solicitudes | 0 excepciones
  public ActionResult Index()
  { return View(db.TipoEmpresa.ToList());}
  0 referencias | 0 solicitudes | 0 excepciones
  public ActionResult Details(int? id)
  { if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
    TipoEmpresa tipoEmpresa = db.TipoEmpresa.Find(id);
    if (tipoEmpresa == null)
    { return HttpNotFound();}
    return View(tipoEmpresa);}
  0 referencias | 1 solicitud | 0 excepciones
  public ActionResult Create()
  {return View();}
  [HttpPost]
  [ValidateAntiForgeryToken]
  0 referencias | 1 solicitud | 0 excepciones
  public ActionResult Create([Bind(Include = "Id,Nombre,Descripcion")] TipoEmpresa tipoEmpresa)
  { if (ModelState.IsValid) { db.TipoEmpresa.Add(tipoEmpresa);
    db.SaveChanges();
    return RedirectToAction("Index"); } return View(tipoEmpresa); }
  0 referencias | 2 solicitudes | 0 excepciones
  public ActionResult Edit(int? id)
  { if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
    TipoEmpresa tipoEmpresa = db.TipoEmpresa.Find(id);
    if (tipoEmpresa == null)
    { return HttpNotFound();}
    return View(tipoEmpresa);}
  [HttpPost]
  [ValidateAntiForgeryToken]
  0 referencias | 2 solicitudes | 0 excepciones
  public ActionResult Edit([Bind(Include = "Id,Nombre,Descripcion")] TipoEmpresa tipoEmpresa)
  {if (ModelState.IsValid)
    {db.Entry(tipoEmpresa).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index"); }
    return View(tipoEmpresa);}
  0 referencias | 1 solicitud | 0 excepciones
  public ActionResult Delete(int? id)
  { if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
    TipoEmpresa tipoEmpresa = db.TipoEmpresa.Find(id);
    if (tipoEmpresa == null)
    {return HttpNotFound();}
    return View(tipoEmpresa); }
  [HttpPost, ActionName("Delete")]
  [ValidateAntiForgeryToken]
  0 referencias | 0 solicitudes | 0 excepciones
  public ActionResult DeleteConfirmed(int id)
  {TipoEmpresa tipoEmpresa = db.TipoEmpresa.Find(id);
    db.TipoEmpresa.Remove(tipoEmpresa);
    db.SaveChanges();
    return RedirectToAction("Index");}
}

```

Fuente: Elaboración Propia

Figura N° 25: Código de la vista Index Tipo De Empresa

```
<h2>Tipo de empresa</h2>
<p>
  @Html.ActionLink("Nuevo registro", "Create")
</p>
<table class="table">
  <tr><th>@Html.DisplayNameFor(model => model.Nombre)</th></tr>
  @foreach (var item in Model) {
    <tr><td>@Html.DisplayFor(modelItem => item.Nombre)</td>
      <td>
        @Html.ActionLink("Editar", "Edit", new { id=item.Id }) |
        @Html.ActionLink("Detalles", "Details", new { id=item.Id })
        @Html.ActionLink("Eliminar", "Delete", new { id=item.Id })
      </td>
    </tr>
  }
</table>
```

Fuente: Elaboración Propia

Figura N° 26: Código de la vista Crear Tipo De Empresa

```
<h2>Crear</h2>
@using (Html.BeginForm())
{
  @Html.AntiForgeryToken()
  <div class="form-horizontal">
    <h4>Tipo de Empresa</h4>
    <hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
      @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })
      </div>
    </div>
    <div class="form-group">
      @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })
      </div>
    </div>
    <div class="form-group">
      <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Crear" class="btn btn-default" />
      </div>
    </div>
  </div>
}
```

Fuente: Elaboración Propia

Figura N° 27: Código de la vista Actualizar Tipo De Empresa

```
<h2>Actualizar</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Tipo de Empresa</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)

        <div class="form-group">
            @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Guardar" class="btn btn-default" />
            </div>
        </div>
    </div>
}
```

Fuente: Elaboración Propia

Figura N° 28: Código de la vista Borrar Tipo De Empresa

```
<h2>Delete</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div>
    <h4>Tipo de Empresa</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt> @Html.DisplayNameFor(model => model.Nombre)</dt>
        <dd> @Html.DisplayFor(model => model.Nombre)</dd>
        <dt> @Html.DisplayNameFor(model => model.Descripcion)</dt>
        <dd> @Html.DisplayFor(model => model.Descripcion)</dd>
    </dl>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()
        <div class="form-actions no-color">
            <input type="submit" value="Eliminar" class="btn btn-default" />
            @Html.ActionLink("Regresar al listado", "Index")
        </div>
    }
```

Fuente: Elaboración Propia

GUI

Figura N° 29: Formulario Index Tipo De Empresa

SISCONSER Gestión Usuarios Recursos TI Servicios Reportes

Tipo de empresa

Nuevo registro

Nombre	
Proveedores	Editar Detalles Eliminar
Clientes	Editar Detalles Eliminar
Corporativo	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 30: Formulario Crear Tipo De Empresa

SISCONSER Gestión Usuarios Recursos TI Servicios Reportes

Crear

Tipo de Empresa

Nombre

Descripcion

Fuente: Elaboración Propia

Figura N° 31: Formulario Actualizar Tipo De Empresa

SISCONSER Gestión Usuarios Recursos TI Servicios Reportes

Actualizar

Tipo de Empresa

Nombre

Descripcion

Fuente: Elaboración Propia

Figura N° 32: Formulario Eliminar Tipo De Empresa

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#) [Reportes](#)

Delete

¿Estas seguro que deseas borrarlo?

Tipo de Empresa

Nombre Corporativo
Descripcion Empresas que son parte del grupo Empresarial

| [Regresar al listado](#)

Fuente: Elaboración Propia

Requerimiento RF3

RF03: El sistema debe permitir al usuario autenticado registrar y actualizar las empresas colaboradoras de la Compañía.

Diseño

Prototipo 1 RF03

Figura N° 33: Prototipo 1 Listar Empresa

SisConSer [Gestión de Usuarios](#) [Recursos](#) [Servicios](#) [Reportes](#) [Cerrar Sesión](#)

Tipo de Empresa

Empresa

Área

Cargo

Usuario

Cuentas

Accesos

Empresa

[Crear nuevo](#)

Razón social	Tipo de empresa	RUC	Teléfono	Correo			
					Editar	Detalle	Eliminar
					Editar	Detalle	Eliminar

Fuente: Elaboración Propia

Prototipo 2 RF03

Figura N° 34: Prototipo 2 Crear Empresa

Crear

Empresa

Razón Social

RUC

TipoEmpresa ▼

Teléfono

Correo

[Regresar al listado](#)

Fuente: Elaboración Propia

Prototipo 3 RF03

Figura N° 35: Prototipo 3 Actualizar Empresa

Actualizar
Empresa

Razón Social

RUC

TipoEmpresa ▼

Teléfono

Correo

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 4 RF03

Figura N° 36: Prototipo 4 Eliminar Empresa

Borrar
¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 37: Código para la creación de la tabla Empresa

```
1 CREATE TABLE [dbo].[Empresa] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [RazonSocial] NVARCHAR (50) NOT NULL,  
4     [RUC] NVARCHAR (11) NOT NULL,  
5     [TipoEmpresa] INT NOT NULL,  
6     [Telefono] NVARCHAR (9) NOT NULL,  
7     [Correo] NVARCHAR (100) NOT NULL,  
8     PRIMARY KEY CLUSTERED ([Id] ASC),  
9     CONSTRAINT [FK_Empresa_TipoEmpresa] FOREIGN KEY ([TipoEmpresa]) REFERENCES [dbo].[TipoEmpresa] ([Id])  
10 );  
11
```

Fuente: Elaboración Propia

Figura N° 38: Código del modelo Empresa

```
public partial class Empresa
{[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
  0 referencias | 0 excepciones
  public Empresa() { .. }
  0 referencias | 0 excepciones
  public int Id { get; set; }
  [RegularExpression("^[a-zA-Z]+$", ErrorMessage = "Este campo debe contener solo letras")]
  [Required(ErrorMessage = "Este campo es obligatorio")]
  0 referencias | 0 excepciones
  public string RazonSocial { get; set; }
  [Required(ErrorMessage = "Este campo es obligatorio")]
  [RegularExpression("^[0-9]*$", ErrorMessage = "Este campo debe contener sólo números.")]
  [StringLength(10)]
  0 referencias | 0 excepciones
  public string RUC { get; set; }
  3 referencias | 0 excepciones
  public int TipoEmpresa { get; set; }
  [Required(ErrorMessage = "Este campo es obligatorio")]
  [RegularExpression("^[0-9]*$", ErrorMessage = "Este campo debe contener sólo números.")]
  [StringLength(9)]
  0 referencias | 0 excepciones
  public string Telefono { get; set; }
  [Required(ErrorMessage = "Este campo es obligatorio")]
  [EmailAddress(ErrorMessage = "El formato no es correcto")]
  0 referencias | 0 excepciones
  public string Correo { get; set; }
  1 referencia | 0 excepciones
  public virtual TipoEmpresa TipoEmpresa1 { get; set; }
  [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
  1 referencia | 0 excepciones
  public virtual ICollection<Requerimiento> Requerimiento { get; set; }
  [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
  1 referencia | 0 excepciones
  public virtual ICollection<Usuario> Usuario { get; set; } }
```

Fuente: Elaboración Propia

Figura N° 39: Código del controlador Empresa

```

public class EmpresasController : Controller
{
    private WebConSerSisEntities db = new WebConSerSisEntities();

    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Index()
    {
        var empresa = db.Empresa.Include(e => e.TipoEmpresa);
        return View(empresa.ToList());
    }

    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Details(int? id)
    {
        if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        Empresa empresa = db.Empresa.Find(id);
        if (empresa == null){return HttpNotFound();}
        return View(empresa);
    }

    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Create()
    {
        ViewBag.TipoEmpresa = new SelectList(db.TipoEmpresa, "Id", "Nombre");
        return View();
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Create([Bind(Include = "Id,RazonSocial,RUC,TipoEmpresa,Telefono,Correo")] Empresa empresa)
    {
        if (ModelState.IsValid) {db.Empresa.Add(empresa);
            db.SaveChanges();
            return RedirectToAction("Index");}
        ViewBag.TipoEmpresa = new SelectList(db.TipoEmpresa, "Id", "Nombre", empresa.TipoEmpresa);
        return View(empresa);
    }

    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Edit(int? id)
    {
        if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        Empresa empresa = db.Empresa.Find(id);
        if (empresa == null){return HttpNotFound();}
        ViewBag.TipoEmpresa = new SelectList(db.TipoEmpresa, "Id", "Nombre", empresa.TipoEmpresa);
        return View(empresa);
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Edit([Bind(Include = "Id,RazonSocial,RUC,TipoEmpresa,Telefono,Correo")] Empresa empresa)
    {
        if (ModelState.IsValid){db.Entry(empresa).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");}
        ViewBag.TipoEmpresa = new SelectList(db.TipoEmpresa, "Id", "Nombre", empresa.TipoEmpresa);
        return View(empresa);
    }

    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Delete(int? id)
    {
        if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        Empresa empresa = db.Empresa.Find(id);
        if (empresa == null){return HttpNotFound();}
        return View(empresa);
    }

    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult DeleteConfirmed(int id)
    {
        Empresa empresa = db.Empresa.Find(id);
        db.Empresa.Remove(empresa);
        db.SaveChanges();
        return RedirectToAction("Index");}
    }
}

```

Fuente: Elaboración Propia

Figura N° 40: Código de la vista Index Empresa

```

<h2>Empresa</h2>
<p>@Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
  <tr><th>@Html.DisplayNameFor(model => model.RazonSocial)</th>
    <th>@Html.DisplayNameFor(model => model.RUC)</th>
    <th>@Html.DisplayNameFor(model => model.Telefono)</th>
    <th>@Html.DisplayNameFor(model => model.Correo)</th>
    <th>Tipo de Empresa</th></tr>
  @foreach (var item in Model) {
    <tr><td>@Html.DisplayFor(modelItem => item.RazonSocial)</td>
      <td>@Html.DisplayFor(modelItem => item.RUC)</td>
      <td>@Html.DisplayFor(modelItem => item.Telefono)</td>
      <td>@Html.DisplayFor(modelItem => item.Correo)</td>
      <td>@Html.DisplayFor(modelItem => item.TipoEmpresa.Nombre)</td>
      <td>@Html.ActionLink("Editar", "Edit", new { id=item.Id })
        @Html.ActionLink("Detalles", "Details", new { id=item.Id })
        @Html.ActionLink("Eliminar", "Delete", new { id=item.Id })</td></tr>
  }
</table>

```

Fuente: Elaboración Propia

Figura N° 41: Código de la vista Crear Empresa

```

<h2>Crear</h2>
@using (Html.BeginForm())
{
  @Html.AntiForgeryToken()
  <div class="form-horizontal">
    <h4>Empresa</h4> <hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
      @Html.LabelFor(model => model.RazonSocial, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.RazonSocial, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.RazonSocial, "", new { @class = "text-danger" })
      </div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.RUC, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.RUC, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.RUC, "", new { @class = "text-danger" })
      </div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.TipoEmpresa, "TipoEmpresa", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("TipoEmpresa", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.TipoEmpresa, "", new { @class = "text-danger" })
      </div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Telefono, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Telefono, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Telefono, "", new { @class = "text-danger" })
      </div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Correo, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Correo, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Correo, "", new { @class = "text-danger" })
      </div></div>
    <div class="form-group">
      <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Crear" class="btn btn-default" />
      </div></div></div>
  <div>
    @Html.ActionLink("Regresar al listado", "Index")
  </div>
}

```

Fuente: Elaboración Propia

Figura N° 42: Código de la vista Actualizar Empresa

```

<h2>Actualizar</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Empresa</h4> <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)
        <div class="form-group">
            @Html.LabelFor(model => model.RazonSocial, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.RazonSocial, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.RazonSocial, "", new { @class = "text-danger" })
            </div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.RUC, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.RUC, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.RUC, "", new { @class = "text-danger" })
            </div> </div>
        <div class="form-group">
            @Html.LabelFor(model => model.TipoEmpresa, "TipoEmpresa", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("TipoEmpresa", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.TipoEmpresa, "", new { @class = "text-danger" })
            </div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Telefono, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Telefono, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Telefono, "", new { @class = "text-danger" })
            </div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Correo, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Correo, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Correo, "", new { @class = "text-danger" })
            </div> </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Guardar" class="btn btn-default" />
            </div></div>
    </div>
    @Html.ActionLink("Regresar al listado", "Index")
</div>

```

Fuente: Elaboración Propia

Figura N° 43: Código de la vista Eliminar Empresa

```

<h2>Delete</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div>
    <h4>Empresa</h4><hr />
    <dl class="dl-horizontal">
        <dt>@Html.DisplayNameFor(model => model.RazonSocial)</dt>
        <dd>@Html.DisplayFor(model => model.RazonSocial)</dd>
        <dt>@Html.DisplayNameFor(model => model.RUC)</dt>
        <dd> @Html.DisplayFor(model => model.RUC)</dd>
        <dt>@Html.DisplayNameFor(model => model.Telefono)</dt>
        <dd>@Html.DisplayFor(model => model.Telefono)</dd>
        <dt>@Html.DisplayNameFor(model => model.Correo)</dt>
        <dd>@Html.DisplayFor(model => model.Correo)</dd>
        <dt>@Html.DisplayNameFor(model => model.TipoEmpresa1.Nombre)</dt>
        <dd> @Html.DisplayFor(model => model.TipoEmpresa1.Nombre) </dd>
    </dl>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()
        <div class="form-actions no-color">
            <input type="submit" value="Eliminar" class="btn btn-default" />
            @Html.ActionLink("Regresar al listado", "Index")
        </div>
    }
</div>

```

Fuente: Elaboración Propia

GUI

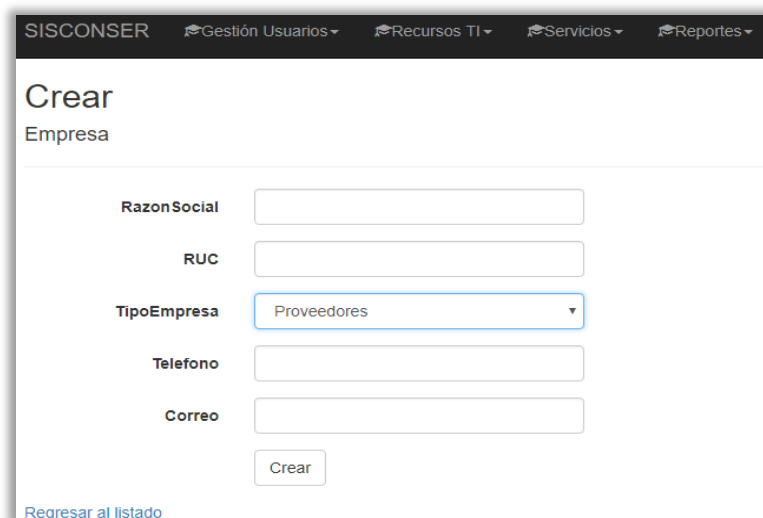
Figura N° 44: Formulario Index Empresa



RazonSocial	Tipo de Empresa	RUC	Telefono	Correo	
Constructora El Roble SAC	Corporativo	20514441163	2015300	facturacion.electronica@construtoraroble.com	Editar Detalles Eliminar
Pesquera Natalia SAC	Corporativo	20513452153	2015300	facturacion.electronica@pesquanatalia.com	Editar Detalles Eliminar
Compañía Naviera Natalia SAC	Corporativo	20537243709	2015300	facturacion.electronica@navieranatalia.com	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 45: Formulario Crear Empresa



Fuente: Elaboración Propia

Figura N° 46: Formulario Actualizar Empresa



Fuente: Elaboración Propia

Figura N° 47: Formulario Eliminar Empresa

SISCONSER Gestión Usuarios Recursos TI Servicios Reportes

Eliminar

¿Estas seguro que deseas borrarlo?

Empresa

RazonSocial	Constructora El Roble SAC
RUC	20514441163
Telefono	2015300
Correo	facturacion.electronica@construtoraroble.com
Nombre	Corporativo

[Regresar al listado](#)

Fuente: Elaboración Propia

Requerimiento RF04

RF04: El sistema debe permitir al usuario autenticado registrar y actualizar las áreas que dispone la Compañía.

Diseño

Prototipo 1 RF04

Figura N° 48: Prototipo 1 Index Áreas

Gestionar Usuarios 88 x 23 SisConSer [Cerrar Sesión](#)

Tipo Empresa
Empresa
Area
Cargo
Usuario
Accesos

Area

[Nuevo Registro](#)

Nombre			
	Editar	Detalle	Eliminar
	Editar	Detalle	Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 2 RF04

Figura N° 49: Prototipo 2 Crear Área

Gestionar Usuarios 88 x 23 SisConSer [Cerrar Sesión](#)

Tipo Empresa
Empresa
Area
Cargo
Usuario
Accesos

Crear

Areas

Nombre

Descripción

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 3 RF04

Figura N° 50: Prototipo 3 Editar Área

Editar
Áreas

Nombre

Descripción

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Prototipo 4 RF04

Figura N° 51: Prototipo 4 Borrar Área

Borrar
¿Estas Seguro que deseas borrarlos ?

Codigo :

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Implementación

Código

Figura N° 52: Código para la creación de la tabla Área

```
1 CREATE TABLE [dbo].[Area] (  
2 [Id] INT IDENTITY (1, 1) NOT NULL,  
3 [Nombre] NVARCHAR (30) NOT NULL,  
4 [Descripcion] NVARCHAR (200) NULL,  
5 PRIMARY KEY CLUSTERED ([Id] ASC)  
6 );
```

Fuente: Elaboración Propia

Figura N° 53: Código del modelo Área

```
public partial class Area
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    0 referencias | 0 excepciones
    public Area()...
    4 referencias | 0 excepciones
    public int Id { get; set; }
    [RegularExpression("^[a-zA-Z]+$", ErrorMessage = "Este campo debe contener solo letras")]
    [Required(ErrorMessage = "Este campo es obligatorio")]
    10 referencias | 0 excepciones
    public string Nombre { get; set; }
    8 referencias | 0 excepciones
    public string Descripcion { get; set; }
}
```

Fuente: Elaboración Propia

Figura N° 54: Código del controlador Área

```
public class AreasController : Controller
{
    private WebConSerSisEntities db = new WebConSerSisEntities();
    0 referencias | 11 solicitudes | 0 excepciones
    public ActionResult Index()
    { return View(db.Area.ToList()); }
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Details(int? id)...
    0 referencias | 2 solicitudes | 0 excepciones
    public ActionResult Create()...
    [HttpPost]
    [ValidateAntiForgeryToken]
    0 referencias | 2 solicitudes | 0 excepciones
    public ActionResult Create([Bind(Include = "Id,Nombre,Descripcion")] Area area)
    { if (ModelState.IsValid)
      { db.Area.Add(area);
        db.SaveChanges();
        return RedirectToAction("Index"); }
      return View(area); }
    0 referencias | 3 solicitudes | 0 excepciones
    public ActionResult Edit(int? id)
    { if (id == null)
      { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }
      Area area = db.Area.Find(id);
      if (area == null)
      { return HttpNotFound(); }
      return View(area); }
    [HttpPost]
    [ValidateAntiForgeryToken]
    0 referencias | 3 solicitudes | 0 excepciones
    public ActionResult Edit([Bind(Include = "Id,Nombre,Descripcion")] Area area)
    { if (ModelState.IsValid)
      { db.Entry(area).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index"); }
      return View(area); }
    0 referencias | 3 solicitudes | 0 excepciones
    public ActionResult Delete(int? id)
    { if (id == null)
      { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }
      Area area = db.Area.Find(id);
      if (area == null)
      { return HttpNotFound(); }
      return View(area); }
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult DeleteConfirmed(int id)
    {
      Area area = db.Area.Find(id);
      db.Area.Remove(area);
      db.SaveChanges();
      return RedirectToAction("Index");
    }
}
```

Fuente: Elaboración Propia

Figura N° 55: Código de la vista Index Área

```
<h2>Area</h2>
<p>@Html.ActionLink("Nuevo registro", "Create")</p>
<table class="table">
  <tr><th> @Html.DisplayNameFor(model => model.Nombre)</th></tr>

  @foreach (var item in Model)
  {
    <tr>
      <td>
        @Html.DisplayFor(modelItem => item.Nombre)
      </td>
      <td>
        @Html.ActionLink("Editar", "Edit", new { id = item.Id }) |
        @Html.ActionLink("Detalles", "Details", new { id = item.Id })
        @Html.ActionLink("Eliminar", "Delete", new { id = item.Id })
      </td>
    </tr>
  }
</table>
```

Fuente: Elaboración Propia

Figura N° 56: Código de la lista Crear Área

```
<h2>Crear</h2>
@using (Html.BeginForm())
{
  @Html.AntiForgeryToken()

  <div class="form-horizontal">
    <h4>Area</h4>
    <hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
      @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })
      </div>
    </div>

    <div class="form-group">
      @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })
      </div>
    </div>

    <div class="form-group">
      <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Crear" class="btn btn-default" />
      </div>
    </div>
  </div>
}
```

Fuente: Elaboración Propia

Figura N° 57: Código de la vista Actualizar Área

```
<h2>Actualizar</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Estado de Atención</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)
        <div class="form-group">
            @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Guardar" class="btn btn-default" />
            </div>
        </div>
    </div>
}
```

Fuente: Elaboración Propia

Figura N° 58: Código de la vista Eliminar Área

```
<h2>Eliminar</h2>
<h3>¿Estas seguro que deseas eliminarlo?</h3>
<div>
    <h4>Area</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>@Html.DisplayNameFor(model => model.Nombre)</dt>
        <dd>@Html.DisplayFor(model => model.Nombre)</dd>
        <dt>@Html.DisplayNameFor(model => model.Descripcion)</dt>
        <dd>@Html.DisplayFor(model => model.Descripcion)</dd>
    </dl>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()
        <div class="form-actions no-color">
            <input type="submit" value="Eliminar" class="btn btn-default" /> |
            @Html.ActionLink("Regresar al listado", "Index")
        </div>}
</div>
```

Fuente: Elaboración Propia

GUI

Figura N° 59: Formulario Index Área

Área	
Nuevo registro	
Nombre	
Contabilidad	Editar Detalles Eliminar
Recepción	Editar Detalles Eliminar
Compras	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 60: Formulario Crear Área

Crear
Area

Nombre

Descripcion

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 61: Formulario Editar Área

Editar
Area

Nombre

Descripcion

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 62: Formulario Eliminar Área

SISCONSER | Gestión Usuarios | Recursos TI | Servicios | Reportes

Eliminar

¿Estas seguro que deseas eliminarlo?

Area

Nombre	Contabilidad
Contabilidad	

Eliminar | [Regresar al listado](#)

Fuente: Elaboración Propia

Requerimiento RF05

RF05: El sistema debe permitir usuario autenticado registrar y actualizar a los cargos de los usuarios que laboran en la Compañía.

Diseño

Prototipo 1 RF05

Figura N° 62: Prototipo 1 Index Cargo

Gestionar Usuarios | 88 x 23 | SisConSer | [Cerrar Sesion](#)

Tipo Empresa
Empresa
Area
Cargo
Usuario
Accesos

Cargo

[Nuevo Registro](#)

Nombre			
	Editar	Detalle	Eliminar
	Editar	Detalle	Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 2 RF05

Figura N° 63: Prototipo 2 Crear Cargo

Crear

Cargo

Nombre:

Descripción:

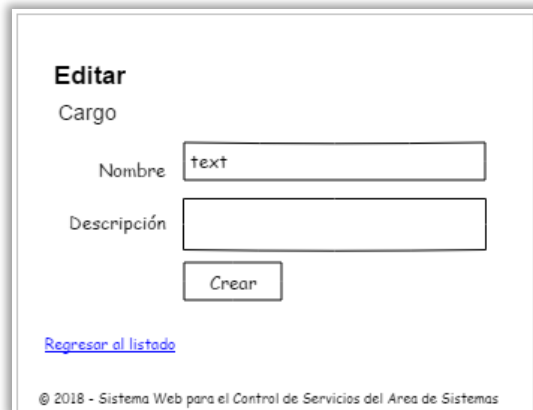
[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 3 RF05

Figura N° 64: Prototipo 3 Editar Cargo



Editar

Cargo

Nombre

Descripción

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 3 RF05

Figura N° 65: Prototipo 3 Eliminar Cargo



Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 66: Código para a creación de a tabla Cargo

```
1 CREATE TABLE [dbo].[Cargo] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nombre] NVARCHAR (30) NOT NULL,  
4     [Descripcion] NVARCHAR (200) NULL,  
5     PRIMARY KEY CLUSTERED ([Id] ASC)  
6 );
```

Fuente: Elaboración Propia

Figura N° 67: Código del controlador Cargo

```

public class CargoesController : Controller
{
    private WebConSerSisEntities db = new WebConSerSisEntities();
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Index()
    {return View(db.Cargo.ToList()); }
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Details(int? id)
    {if (id == null){ return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        Cargo cargo = db.Cargo.Find(id);
        if (cargo == null) { return HttpNotFound();}
        return View(cargo);}
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Create()
    {return View();}
    [HttpPost]
    [ValidateAntiForgeryToken]
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Create([Bind(Include = "Id,Nombre,Descripcion")] Cargo cargo)
    {if (ModelState.IsValid) {db.Cargo.Add(cargo);
        db.SaveChanges();
        return RedirectToAction("Index"); }
        return View(cargo);}
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Createmodal([Bind(Include = "Id,Nombre,Descripcion")] Cargo cargo)
    {if (ModelState.IsValid){db.Cargo.Add(cargo);
        db.SaveChanges();
        return PartialView("View", "Cargoe"); }
        return View(); }
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Edit(int? id)
    { if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        Cargo cargo = db.Cargo.Find(id);
        if (cargo == null){return HttpNotFound();}
        return View(cargo);}
    [HttpPost]
    [ValidateAntiForgeryToken]
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Edit([Bind(Include = "Id,Nombre,Descripcion")] Cargo cargo)
    {if (ModelState.IsValid){db.Entry(cargo).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index"); }
        return View(cargo);}
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Delete(int? id)
    {if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        Cargo cargo = db.Cargo.Find(id);
        if (cargo == null){return HttpNotFound();}
        return View(cargo);}
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult DeleteConfirmed(int id)
    { Cargo cargo = db.Cargo.Find(id);
        db.Cargo.Remove(cargo);
        db.SaveChanges();
        return RedirectToAction("Index"); }
}

```

Fuente: Elaboración Propia

Figura N° 68: Código de la vista Index Cargo

```
<h2>Cargo</h2>
<p>
  @Html.ActionLink("Nuevo Registro", "Create")
</p>
<table class="table">
  <tr><th> @Html.DisplayNameFor(model => model.Nombre)</th>
  @foreach (var item in Model) {
    <tr><td>@Html.DisplayFor(modelItem => item.Nombre)</td>
      <td>
        @Html.ActionLink("Editar", "Edit", new { id=item.Id }) |
        @Html.ActionLink("Detalles", "Details", new { id=item.Id }) |
        @Html.ActionLink("Eliminar", "Delete", new { id=item.Id })
      </td>
    </tr>
  }
</table>
```

Fuente: Elaboración Propia

Figura N° 69: Código de la vista Crear Cargo

```
<h2>Crear</h2>
@using (Html.BeginForm())
{
  @Html.AntiForgeryToken()
  <div class="form-horizontal">
    <h4>Cargo</h4>
    <hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
      @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })
      </div>
    </div>
    <div class="form-group">
      @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })
      </div>
    </div>
    <div class="form-group">
      <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Crear" class="btn btn-default" />
      </div>
    </div>
  </div>
}
```

Fuente: Elaboración Propia

Figura N° 70: Código de la vista Actualizar Cargo

```
<h2>Actualizar</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Cargo</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)
        <div class="form-group">
            @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Guardar" class="btn btn-default" />
            </div>
        </div>
    </div>
}
```

Fuente: Elaboración Propia

Figura N° 71: Código de la vista Eliminar Cargo

```
<h2>Eliminar</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div>
    <h4>Cargo</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>@Html.DisplayNameFor(model => model.Nombre) </dt>
        <dd>@Html.DisplayFor(model => model.Nombre) </dd>
        <dt>@Html.DisplayNameFor(model => model.Descripcion) </dt>
        <dd>@Html.DisplayFor(model => model.Descripcion) </dd>
    </dl>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()
        <div class="form-actions no-color">
            <input type="submit" value="Eliminar" class="btn btn-default" /> |
            @Html.ActionLink("Regresar al listado", "Index")
        </div>
    }
</div>
```

Fuente: Elaboración Propia

GUI

Figura N° 72: Formulario Index Cargo

Cargo	
Nuevo Registro	
Nombre	
Gerente	Editar Detalles Eliminar
Administrador	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 73: Formulario Crear Cargo

SISCONSER Gestión Usuarios Recursos TI Servicios Reportes

Crear Cargo

Nombre

Descripcion

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 74: Formulario Actualizar Cargo

SISCONSER Gestión Usuarios Recursos TI Servicios Reportes

Actualizar Cargo

Nombre

Descripcion

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 75: Formulario Eliminar Cargo

Eliminar Cargo

¿Estas seguro que deseas borrarlo?

Nombre	Descripcion
Gerente	Gerente

Eliminar | Regresar al listado

Fuente: Elaboración Propia

Requerimiento RF06

RF06: El sistema debe permitir usuario autenticado registrar y actualizar los usuarios que laboran en la Compañía.

Diseño

Prototipo 1 RF06

Figura N° 76: Prototipo 1 Index Usuario

Gestionar Usuarios | 88 x 23 | SisConSer | Cerrar Sesion

Usuarios
[Nuevo Registro](#)

Nombre	Apellido	Correo	Anexo	Celular	Area	Cargo	Planilla			
								Editar	Detalle	Eliminar
								Editar	Detalle	Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 2 RF06

Figura N° 77: Prototipo 2 Crear Usuario

Crear Usuario

DNI: text

Nombre: text

Apellido Paterno: text

Apellido Materno: text

Planilla: text goes here ▼

Area: text goes here ▼

Cargo: text goes here ▼

Correo: text

Anexo: text

Celular: text

Crear

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 3 RF06

Figura N° 78: Prototipo 3 Actualizar Usuario

Actualizar

Usuario

DNI	<input type="text" value="text"/>
Nombre	<input type="text" value="text"/>
Apellido Paterno	<input type="text" value="text"/>
Apellido Materno	<input type="text" value="text"/>
Planilla	<input type="text" value="text goes here"/> ▼
Area	<input type="text" value="text goes here"/> ▼
Cargo	<input type="text" value="text goes here"/> ▼
Correo	<input type="text" value="text"/>
Anexo	<input type="text" value="text"/>
Celular	<input type="text" value="text"/>

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 4 RF06

Figura N° 79: Prototipo 4 Eliminar Usuario

Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 80: Código para la creación de la tabla Usuario

```
1 CREATE TABLE [dbo].[Usuario] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [DNI] NVARCHAR (8) NOT NULL,  
4     [Nombre] NVARCHAR (30) NOT NULL,  
5     [ApellidoPaterno] NVARCHAR (30) NOT NULL,  
6     [ApellidoMaterno] NVARCHAR (30) NOT NULL,  
7     [Full_Nombre] AS ((([Nombre]+' ')+[ApellidoPaterno]),  
8     [Planilla] INT NOT NULL,  
9     [Area] INT NOT NULL,  
10    [Cargo] INT NOT NULL,  
11    [Correo] NVARCHAR (100) NOT NULL,  
12    [Anexo] INT NOT NULL,  
13    [Celular] NVARCHAR (9) NULL,  
14    PRIMARY KEY CLUSTERED ([Id] ASC),  
15    CONSTRAINT [FK_Usuario_Planilla] FOREIGN KEY ([Planilla]) REFERENCES [dbo].[Empresa] ([Id]),  
16    CONSTRAINT [FK_Usuario_Area] FOREIGN KEY ([Area]) REFERENCES [dbo].[Area] ([Id]),  
17    CONSTRAINT [FK_Usuario_Cargo] FOREIGN KEY ([Cargo]) REFERENCES [dbo].[Cargo] ([Id])  
18 );
```

Fuente: Elaboración Propia

Figura N° 81: Código del modelo Usuario

```
public partial class Usuario  
{ [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]  
  0 referencias | 0 excepciones  
  public int Id { get; set; }  
  [Required(ErrorMessage = "Este campo es obligatorio")]  
  [RegularExpression(@"^[0-9]*$", ErrorMessage = "Este campo debe contener sólo números.")]  
  [StringLength(8)]  
  0 referencias | 0 excepciones  
  public string DNI { get; set; }  
  [RegularExpression(@"^[a-zA-Z]+$", ErrorMessage = "Este campo debe contener solo letras")]  
  [Required(ErrorMessage = "Este campo es obligatorio")]  
  0 referencias | 0 excepciones  
  public string Nombre { get; set; }  
  [RegularExpression(@"^[a-zA-Z]+$", ErrorMessage = "Este campo debe contener solo letras")]  
  [Required(ErrorMessage = "Este campo es obligatorio")]  
  0 referencias | 0 excepciones  
  public string ApellidoPaterno { get; set; }  
  [RegularExpression(@"^[a-zA-Z]+$", ErrorMessage = "Este campo debe contener solo letras")]  
  [Required(ErrorMessage = "Este campo es obligatorio")]  
  0 referencias | 0 excepciones  
  public string ApellidoMaterno { get; set; }  
  0 referencias | 0 excepciones  
  public string Full_Nombre { get; set; }  
  3 referencias | 0 excepciones  
  public int Planilla { get; set; }  
  3 referencias | 0 excepciones  
  public int Area { get; set; }  
  3 referencias | 0 excepciones  
  public int Cargo { get; set; }  
  [Required(ErrorMessage = "Este campo es obligatorio")]  
  [EmailAddress(ErrorMessage = "El formato no es correcto")]  
  0 referencias | 0 excepciones  
  public string Correo { get; set; }  
  [StringLength(3)]  
  [Required(ErrorMessage = "Este campo es obligatorio")]  
  [RegularExpression(@"^[0-9]*$", ErrorMessage = "Este campo debe contener sólo números.")]  
  0 referencias | 0 excepciones  
  public int Anexo { get; set; }  
  [RegularExpression(@"^[0-9]*$", ErrorMessage = "Este campo debe contener sólo números.")]  
  [StringLength(9)]  
  0 referencias | 0 excepciones  
  public string Celular { get; set; }
```

Fuente: Elaboración Propia

Figura N° 82: Código del controlador Usuario

```

public class UsuariosController : Controller
{
    private WebConSerSisEntities db = new WebConSerSisEntities();
    // Referencias | 7 solicitudes | 0 excepciones
    public ActionResult Index() (var usuario = db.Usuario.Include(u => u.Area1).Include(u => u.Cargo).Include(u => u.Empresa);
        return View(usuario.ToList());
    // Referencias | 0 solicitudes | 0 excepciones
    public ActionResult Details(int? id) (if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        Usuario usuario = db.Usuario.Find(id);
        if (usuario == null){ return HttpNotFound();}
        return View(usuario);
    // Referencias | 5 solicitudes | 0 excepciones
    public ActionResult Create(){ViewBag.Area = new SelectList(db.Area, "Id", "Nombre");
        ViewBag.Cargo = new SelectList(db.Cargo, "Id", "Nombre");
        ViewBag.Planilla = new SelectList(db.Empresa, "Id", "RazonSocial");
        return View();}
    [HttpPost]
    [ValidateAntiForgeryToken]
    // Referencias | 5 solicitudes | 0 excepciones
    public ActionResult Create([Bind(Include = "Id,DNI,Nombre,ApellidoPaterno,ApellidoMaterno,Full_Nombre,Planilla,Area,Cargo,Correo,Anexo,Celular")] Usuario usuario)
    (if (ModelState.IsValid) {db.Usuario.Add(usuario);
        db.SaveChanges();
        return RedirectToAction("Index"); }
        ViewBag.Area = new SelectList(db.Area, "Id", "Nombre", usuario.Area);
        ViewBag.Cargo = new SelectList(db.Cargo, "Id", "Nombre", usuario.Cargo);
        ViewBag.Planilla = new SelectList(db.Empresa, "Id", "RazonSocial", usuario.Planilla);
        return View(usuario);}
    // Referencias | 1 solicitud | 0 excepciones
    public ActionResult Edit(int? id) (if (id == null){ return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }
        Usuario usuario = db.Usuario.Find(id);
        if (usuario == null){ return HttpNotFound();}
        ViewBag.Area = new SelectList(db.Area, "Id", "Nombre", usuario.Area);
        ViewBag.Cargo = new SelectList(db.Cargo, "Id", "Nombre", usuario.Cargo);
        ViewBag.Planilla = new SelectList(db.Empresa, "Id", "RazonSocial", usuario.Planilla);
        return View(usuario); }
    [HttpPost]
    [ValidateAntiForgeryToken]
    // Referencias | 1 solicitud | 0 excepciones
    public ActionResult Edit([Bind(Include = "Id,DNI,Nombre,ApellidoPaterno,ApellidoMaterno,Full_Nombre,Planilla,Area,Cargo,Correo,Anexo,Celular")] Usuario usuario)
    (if (ModelState.IsValid) (db.Entry(usuario).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index"); }
        ViewBag.Area = new SelectList(db.Area, "Id", "Nombre", usuario.Area);
        ViewBag.Cargo = new SelectList(db.Cargo, "Id", "Nombre", usuario.Cargo);
        ViewBag.Planilla = new SelectList(db.Empresa, "Id", "RazonSocial", usuario.Planilla);
        return View(usuario);}
    // Referencias | 2 solicitudes | 0 excepciones
    public ActionResult Delete(int? id)
    (if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        Usuario usuario = db.Usuario.Find(id);
        if (usuario == null) {return HttpNotFound();}
        return View(usuario);}
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    // Referencias | 0 solicitudes | 0 excepciones
    public ActionResult DeleteConfirmed(int id){Usuario usuario = db.Usuario.Find(id);
        db.Usuario.Remove(usuario);
        db.SaveChanges();
        return RedirectToAction("Index"); }
}

```

Fuente: Elaboración Propia

Figura N° 83: Código de la vista Index Usuario

```

<h2>Usuarios</h2>
<p>@Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
    <tr>
        <th>Nombre</th>
        <th>Apellido Paterno</th>
        <th>@Html.DisplayNameFor(model => model.Correo)</th>
        <th>@Html.DisplayNameFor(model => model.Anexo)</th>
        <th>@Html.DisplayNameFor(model => model.Celular)</th>
        <th>Area</th>
        <th>Cargo</th>
        <th>Planilla</th></tr>
    @foreach (var item in Model)
    {<tr><td>@Html.DisplayFor(modelItem => item.Nombre)</td>
        <td>@Html.DisplayFor(modelItem => item.ApellidoPaterno)</td>
        <td>@Html.DisplayFor(modelItem => item.Correo)</td>
        <td>@Html.DisplayFor(modelItem => item.Anexo)</td>
        <td>@Html.DisplayFor(modelItem => item.Celular)</td>
        <td>@Html.DisplayFor(modelItem => item.Area1.Nombre)</td>
        <td>@Html.DisplayFor(modelItem => item.Cargo1.Nombre)</td>
        <td>@Html.DisplayFor(modelItem => item.Empresa.RazonSocial)</td>
        <td>@Html.ActionLink("Editar", "Edit", new { id = item.Id })
            @Html.ActionLink("Detalles", "Details", new { id = item.Id })
            @Html.ActionLink("Eliminar", "Delete", new { id = item.Id })</td></tr>
    }
</table>

```

Fuente: Elaboración Propia

Figura N° 84: Código de la vista Crear Usuario

```

<h2>Crear</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Usuario</h4><h />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.DNI, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.DNI, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.DNI, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.ApellidoPaterno, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.ApellidoPaterno, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.ApellidoPaterno, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.ApellidoMaterno, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.ApellidoMaterno, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.ApellidoMaterno, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Planilla, "Planilla", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Planilla", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Planilla, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Area, "Area", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Area", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Area, "", new { @class = "text-danger" })</div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Cargo, "Cargo", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Cargo", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Cargo, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Correo, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Correo, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Correo, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Anexo, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Anexo, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Anexo, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Celular, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Celular, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Celular, "", new { @class = "text-danger" })</div></div>
            <div class="form-group"><div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Crear" class="btn btn-default" /></div></div>
        </div>
    }
}

```

Fuente: Elaboración Propia

Figura N° 85: Código de la vista Actualizar Usuario

```

<h4>Usuario</h4><hr />
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
@Html.HiddenFor(model => model.Id)
<div class="form-group">
    @Html.LabelFor(model => model.DNI, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.DNI, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.DNI, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.ApellidoPaterno, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.ApellidoPaterno, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.ApellidoPaterno, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.ApellidoMaterno, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.ApellidoMaterno, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.ApellidoMaterno, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Full_Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Full_Nombre, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Full_Nombre, "", new { @class = "text-danger" })</div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.Planilla, "Planilla", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("Planilla", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Planilla, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Area, "Area", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("Area", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Area, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.Cargo, "Cargo", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("Cargo", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Cargo, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Correo, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Correo, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Correo, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Anexo, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Anexo, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Anexo, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Celular, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Celular, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Celular, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10"><input type="submit" value="Guardar" class="btn btn-default" /></div>
</div>

```

Fuente: Elaboración Propia

Figura N° 86: Código la vista Eliminar Usuario

```

<h2>Eliminar</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div>
  <h4>Usuario</h4><hr />
  <dl class="dl-horizontal">
    <dt> @Html.DisplayNameFor(model => model.Full_Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.Full_Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.ApellidoPaterno)</dt>
    <dd>@Html.DisplayFor(model => model.ApellidoPaterno)</dd>
    <dt>@Html.DisplayNameFor(model => model.ApellidoMaterno)</dt>
    <dd>@Html.DisplayFor(model => model.ApellidoMaterno)</dd>
    <dt>@Html.DisplayNameFor(model => model.Full_Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.Full_Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.Correo)</dt>
    <dd>@Html.DisplayFor(model => model.Correo)</dd>
    <dt>@Html.DisplayNameFor(model => model.Anexo)</dt>
    <dd>@Html.DisplayFor(model => model.Anexo)</dd>
    <dt>@Html.DisplayNameFor(model => model.Celular)</dt>
    <dd>@Html.DisplayFor(model => model.Celular) </dd>
    <dt>@Html.DisplayNameFor(model => model.Area1.Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.Area1.Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.Cargo1.Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.Cargo1.Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.Empresa.RazonSocial)</dt>
    <dd>@Html.DisplayFor(model => model.Empresa.RazonSocial)</dd>
  </dl>@using (Html.BeginForm()) { @Html.AntiForgeryToken()
  <div class="form-actions no-color">
    <input type="submit" value="Eliminar" class="btn btn-default" />
    @Html.ActionLink("Regresar al listado", "Index")</div>
</div>

```

Fuente: Elaboración Propia

GUI

Figura N° 87: Formulario Index Usuarios

Nombre	Apellido Paterno	Correo	Anexo	Celular	Area	Cargo	Planilla	
Alex	Aiquipa	alex.aiquipa@constructoraroble.com	191	981043308	Sistemas	Administrador	Constructora El Roble SAC	Editar Detalles Eliminar
Yolanda	Maldonado	yolanda.maldonado@navieranatalia.com	156	981063105	Contabilidad	Auxiliar	Compañía Naviera Natalia SAC	Editar Detalles Eliminar
Raul	Marin	raulroca@navieranatalia.com	196	961112213	Contabilidad	Administrador	Sobitec	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 88: Formulario Crear Usuarios

SISCONSER 🏠 Gestión Usuarios 🏠 Recursos TI 🏠 Servicios

Crear

Usuario

DNI

Nombre

ApellidoPaterno

ApellidoMaterno

Planilla

Area

Cargo

Correo

Anexo

Celular

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 89: Formulario Actualizar Usuarios

SISCONSER Gestión Usuarios Recursos TI Servicios

Actualizar

Usuario

DNI	10627303
Nombre	Alex
ApellidoPaterno	Aiquipa
ApellidoMaterno	Tello
Full_Nombre	Alex Aiquipa
Planilla	Constructora El Roble SAC
Area	Sistemas
Cargo	Administrador
Correo	alex.aiquipa@construtoraroble.com
Anexo	191
Celular	981043308

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 90: Formulario Eliminar Usuario

SISCONSER Gestión Usuarios Recursos TI Servicios

Eliminar

¿Estas seguro que deseas borrarlo?

Usuario

Full_Nombre	Yolanda Maldonado
Nombre	Yolanda
ApellidoPaterno	Maldonado
ApellidoMaterno	Poma
Full_Nombre	Yolanda Maldonado
Correo	yolanda.maldonado@navieranatalia.com
Anexo	156
Celular	981063105
Nombre	Contabilidad
Nombre	Auxiliar
RazonSocial	Compañía Naviera Natalia SAC

| [Regresar al listado](#)

Fuente: Elaboración Propia

Requerimiento RF07

RF007: El sistema debe permitir al Administrador, crear y modificar cuentas de usuarios del sistema web.

Diseño

Prototipo 1 RF007

Figura N° 91: Prototipo 1 Index Cuenta

SisConSer	Gestión de Usuarios ▼	Recursos ▼	Servicios ▼	Reportes ▼	Cerrar Sesión
	Tipo de Empresa	Cuenta			
	Empresa	Crear nuevo			
	Área	Nombre de usuario	Correo		
	Cargo			Eliminar	
	Usuario			Eliminar	
	Cuentas				
	Accesos				

Fuente: Elaboración Propia

Prototipo 2 RF007

Figura N° 92: Prototipo 2 Crear Cuenta

Crear una nueva cuenta

Nombre de usuario

Correo electronico

Contraseña

Confirmar contraseña

Fuente: Elaboración Propia

Prototipo 3 RF007

Figura N° 93: Prototipo 3 Editar Cuenta

Editar cuenta de usuario

Correo

Nombre de usuario

Fuente: Elaboración Propia

Prototipo 4 RF007

Figura N° 94: Prototipo 4 Eliminar Cuenta

Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo :

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 95: Código para la creación de la tabla AspNetUser

```
1 CREATE TABLE [dbo].[AspNetUsers] (  
2     [Id] NVARCHAR (128) NOT NULL,  
3     [Email] NVARCHAR (256) NULL,  
4     [EmailConfirmed] BIT NOT NULL,  
5     [PasswordHash] NVARCHAR (MAX) NULL,  
6     [SecurityStamp] NVARCHAR (MAX) NULL,  
7     [PhoneNumber] NVARCHAR (MAX) NULL,  
8     [PhoneNumberConfirmed] BIT NOT NULL,  
9     [TwoFactorEnabled] BIT NOT NULL,  
10    [LockoutEndDateUtc] DATETIME NULL,  
11    [LockoutEnabled] BIT NOT NULL,  
12    [AccessFailedCount] INT NOT NULL,  
13    [UserName] NVARCHAR (50) NOT NULL,  
14    CONSTRAINT [PK_dbo.AspNetUsers] PRIMARY KEY CLUSTERED ([Id] ASC)  
15 );
```

Fuente: Elaboración Propia

Figura N° 96: Código del controlador Cuenta

```

using System.Data.Entity; using System.Linq; using System.Net; using System.Web.Mvc; using SistemaControlTI.Models;
namespace SistemaControlTI.Controllers
{
    [Authorize(Roles = "Administrador")]
    public class AspNetUsersController : Controller
    {
        private WebConSerSisEntities db = new WebConSerSisEntities();

        // GET: AspNetUsers
        public ActionResult Index()
        {
            return View(db.AspNetUsers.ToList());
        }

        // GET: AspNetUsers/Details/5
        public ActionResult Details(string id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            AspNetUsers aspNetUsers = db.AspNetUsers.Find(id);
            if (aspNetUsers == null)
            {
                return HttpNotFound();
            }
            return View(aspNetUsers);
        }

        // GET: AspNetUsers/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: AspNetUsers/Create
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include = "Id,Email,EmailConfirmed>PasswordHash,SecurityStamp,PhoneNumber,PhoneNumberConfirmed," +
            "TwoFactorEnabled,LockoutEndDateUtc,LockoutEnabled,AccessFailedCount,UserName")] AspNetUsers aspNetUsers)
        {
            if (ModelState.IsValid)
            {
                db.AspNetUsers.Add(aspNetUsers); db.SaveChanges(); return RedirectToAction("Index"); return View(aspNetUsers);
            }
        }

        // GET: AspNetUsers/Edit/5
        public ActionResult Edit(string id)
        {
            if (id == null) return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            AspNetUsers aspNetUsers = db.AspNetUsers.Find(id);
            if (aspNetUsers == null) return HttpNotFound(); return View(aspNetUsers);
        }

        // POST: AspNetUsers/Edit/5
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include = "Id,Email,EmailConfirmed>PasswordHash,SecurityStamp,PhoneNumber,PhoneNumberConfirmed," +
            "TwoFactorEnabled,LockoutEndDateUtc,LockoutEnabled,AccessFailedCount,UserName")] AspNetUsers aspNetUsers)
        {
            if (ModelState.IsValid) { db.Entry(aspNetUsers).State = EntityState.Modified; db.SaveChanges(); return RedirectToAction("Index"); return View(aspNetUsers); }
        }

        // GET: AspNetUsers/Delete/5
        public ActionResult Delete(string id)
        {
            if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }
            AspNetUsers aspNetUsers = db.AspNetUsers.Find(id);
            if (aspNetUsers == null)
            {
                return HttpNotFound();
            }
            return View(aspNetUsers);
        }

        // POST: AspNetUsers/Delete/5
        [HttpPost, ActionName("Delete")]
    }
}

```

Fuente: Elaboración Propia

Figura N° 97: Código de la vista Index Cuenta

```

@model IEnumerable<SistemaControlTI.Models.AspNetUsers>
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Index</h2>
<p> @Html.ActionLink("Create New", "Register", "Account")</p>
<table class="table">
    <tr><th>Nombre de Usuario</th>
    <th>Correo</th></tr>
    @foreach (var item in Model) {
        <tr><td>@Html.DisplayFor(modelItem => item.UserName)</td>
        <td>@Html.DisplayFor(modelItem => item.Email)</td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id = item.Id }) |
            @Html.ActionLink("Details", "Details", new { id = item.Id }) |
            @Html.ActionLink("Delete", "Delete", new { id = item.Id })</td></tr>
    }
</table>

```

Fuente: Elaboración Propia

Figura N° 98: Código de la vista Crear Cuenta

```
@model SistemaControlTI.Models.AspNetUsers
@{ ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml"; }
<h2>Create</h2>
@using (Html.BeginForm())
{ @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>AspNetUsers</h4><hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Id, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Id, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Id, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Email, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Email, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Email, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.EmailConfirmed, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    <div class="checkbox">
                        @Html.EditorFor(model => model.EmailConfirmed)
                        @Html.ValidationMessageFor(model => model.EmailConfirmed, "", new { @class = "text-danger" })</div></div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.PasswordHash, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.PasswordHash, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.PasswordHash, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.SecurityStamp, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.SecurityStamp, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.SecurityStamp, "", new { @class = "text-danger" })</div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.PhoneNumber, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.PhoneNumber, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.PhoneNumber, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.PhoneNumberConfirmed, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    <div class="checkbox">
                        @Html.EditorFor(model => model.PhoneNumberConfirmed)
                        @Html.ValidationMessageFor(model => model.PhoneNumberConfirmed, "", new { @class = "text-danger" })</div> </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.TwoFactorEnabled, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    <div class="checkbox">
                        @Html.EditorFor(model => model.TwoFactorEnabled)
                        @Html.ValidationMessageFor(model => model.TwoFactorEnabled, "", new { @class = "text-danger" })
                    </div>
                </div>
            </div>
        </div>
    }
}
```

Fuente: Elaboración Propia

Figura N° 99: Código de la vista Editar Cuenta

```
@model SistemaControlTI.Models.AspNetUsers
@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Edit</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>AspNetUsers</h4><hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)
        <div class="form-group">
            @Html.LabelFor(model => model.Email, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Email, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Email, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.EmailConfirmed, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                <div class="checkbox">
                    @Html.EditorFor(model => model.EmailConfirmed)
                    @Html.ValidationMessageFor(model => model.EmailConfirmed, "", new { @class = "text-danger" })</div></div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.PasswordHash, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.PasswordHash, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.PasswordHash, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.SecurityStamp, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.SecurityStamp, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.SecurityStamp, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.PhoneNumber, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.PhoneNumber, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.PhoneNumber, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.PhoneNumberConfirmed, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                <div class="checkbox">
                    @Html.EditorFor(model => model.PhoneNumberConfirmed)
                    @Html.ValidationMessageFor(model => model.PhoneNumberConfirmed, "", new { @class = "text-danger" })</div></div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.TwoFactorEnabled, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                <div class="checkbox">
                    @Html.EditorFor(model => model.TwoFactorEnabled)
                    @Html.ValidationMessageFor(model => model.TwoFactorEnabled, "", new { @class = "text-danger" })</div></div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.LockoutEndDateUtc, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.LockoutEndDateUtc, new { htmlAttributes = new { @class = "form-control" } })
            </div>
        </div>
    </div>
}
```

Fuente: Elaboración Propia

Figura N° 100: Código de la vista Eliminar Cuenta

```

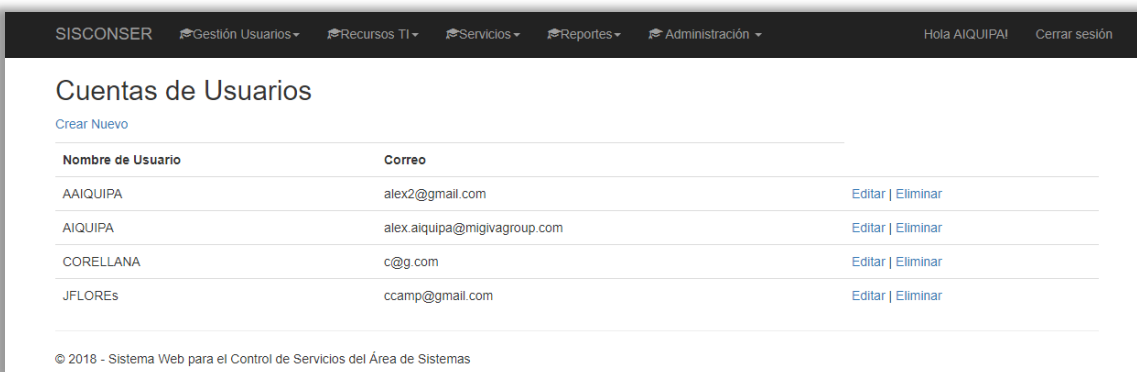
@model SistemaControlTI.Models.AspNetUsers
@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Delete</h2>
<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>AspNetUsers</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt> @Html.DisplayNameFor(model => model.Email)</dt>
        <dd> @Html.DisplayFor(model => model.Email)</dd>
        <dt> @Html.DisplayNameFor(model => model.EmailConfirmed) </dt>
        <dd> @Html.DisplayFor(model => model.EmailConfirmed) </dd>
        <dt> @Html.DisplayNameFor(model => model.PasswordHash)</dt>
        <dd> @Html.DisplayFor(model => model.PasswordHash) </dd>
        <dt> @Html.DisplayNameFor(model => model.SecurityStamp)</dt>
        <dd>@Html.DisplayFor(model => model.SecurityStamp)</dd>
        <dt> @Html.DisplayNameFor(model => model.PhoneNumber)</dt>
        <dd>@Html.DisplayFor(model => model.PhoneNumber)</dd>
        <dt>@Html.DisplayNameFor(model => model.PhoneNumberConfirmed)</dt>
        <dd>@Html.DisplayFor(model => model.PhoneNumberConfirmed)</dd>
        <dt>@Html.DisplayNameFor(model => model.TwoFactorEnabled)</dt>
        <dd>@Html.DisplayFor(model => model.TwoFactorEnabled)</dd>
        <dt>@Html.DisplayNameFor(model => model.LockoutEndDateUtc)</dt>
        <dd>@Html.DisplayFor(model => model.LockoutEndDateUtc)</dd>
        <dt>@Html.DisplayNameFor(model => model.LockoutEnabled)</dt>
        <dd>@Html.DisplayFor(model => model.LockoutEnabled)</dd>
        <dt>@Html.DisplayNameFor(model => model.AccessFailedCount)</dt>
        <dd>@Html.DisplayFor(model => model.AccessFailedCount)</dd>
        <dt>@Html.DisplayNameFor(model => model.UserName)</dt>
        <dd>@Html.DisplayFor(model => model.UserName)</dd>
    </dl>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()
        <div class="form-actions no-color">
            <input type="submit" value="Delete" class="btn btn-default" />
            @Html.ActionLink("Back to List", "Index")
        </div>
    }
</div>

```

Fuente: Elaboración Propia

GUI

Figura N° 101: Código de la vista Eliminar Cuenta



Fuente: Elaboración Propia

Figura N° 102: Código de la vista Registrar Cuenta

Registrarse.
Cree una cuenta nueva.

Nombre Usuario

Correo electrónico

Contraseña

Confirmar contraseña

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Figura N° 103: Código de la vista Editar Cuenta

Editar
Cuenta de Usuario

Email

UserName

[Regresar al Listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Figura N° 104: Código de la vista Eliminar Cuenta

Eliminar
Esta seguro que desea borrarlo?
Cuenta de Usuario

Email alex2@gmail.com
UserName AAIQUIPA

| [Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

TaskBoard Semana 2 del Sprint 1

En la siguiente tabla se muestra los avances realizados durante la segunda semana de iniciado el desarrollo del proyecto.

Tabla N° 9: TaskBoard sprint 1 semana 2

TaskBoard Semana 2				
Sistema web para el control de os servicios de área de sistemas en la compañía Naviera Natalia SAC			Fecha Inicio: 20/07/2018 Fecha final: 26/07/2018	
	Historias por cada Sprint	Pendiente	En curso	Hecho
SPRINT N° 1	Formulario de inicio de sesión para el acceso al Sistema.			X
	CRUD para los tipos de empresas de la Compañía.			X
	CRUD para las empresas colaboradoras de la Compañía			X
	CRUD para las áreas que dispone la Compañía.			X
	CRUD para los cargos de los usuarios que laboraran en la Compañía.			X
	CRUD para el mantenimiento de los usuarios que laboran en la Compañía.			X
	Registrar Cuentas de usuarios del sistema.			X
	Administrar el nivel de acceso de los usuarios al sistema.	X		

Fuente: Elaboración Propia

Requerimiento RF08

RF08: El sistema debe permitir al Administrador, administrar el nivel de acceso al sistema de cada cuenta de usuario creada.

Diseño

Prototipo 1 RF08

Figura N° 105: Código de la vista Index Acceso

The screenshot shows a web application interface for user access management. At the top, there are navigation tabs: 'SisConSer', 'Gestión de Usuarios', 'Recursos', 'Servicios', 'Reportes', and 'Cerrar Sesión'. The 'Gestión de Usuarios' tab is active, showing a dropdown menu with options: 'Tipo de Empresa', 'Empresa', 'Área', 'Cargo', 'Usuario', 'Cuentas', and 'Accesos'. The main content area is titled 'Acceso' and contains a link 'Crear nuevo'. Below this is a table with the following structure:

Roll	Nombre de usuario			
		Editar	Detalle	Eliminar
		Editar	Detalle	Eliminar

At the bottom of the page, there is a copyright notice: '© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas'.

Fuente: Elaboración Propia

Prototipo 2 RF08

Figura N° 106: Código de la vista Agregar Acceso

The screenshot shows a web form titled "Asignar Roles a Usuario". It contains two text input fields: "Nombre de usuario" and "Rol", both with "text" placeholder text. Below the fields is a "Crear" button. At the bottom, there is a copyright notice: "© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas".

Fuente: Elaboración Propia

Prototipo 3 RF08

Figura N° 107: Código de la vista Editar Acceso

The screenshot shows a web form titled "Editar Roles a Usuario". It contains two text input fields: "Nombre de usuario" and "Rol", both with "text" placeholder text. Below the fields is a "Guardar" button. At the bottom, there is a copyright notice: "© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas".

Fuente: Elaboración Propia

Prototipo 4 RF08

Figura N° 108: Código de la vista Eliminar Acceso

The screenshot shows a confirmation dialog titled "Borrar". It asks "¿Estas Seguro que deseas borrarlos?". Below the question, it displays "Codigo : Label". At the bottom, there is an "Eliminar" button and a blue link labeled "Regresar al listado". At the bottom, there is a copyright notice: "© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas".

Fuente: Elaboración Propia

Implementación

Figura N° 109: Código para la creación de la tabla Acceso

```
1 CREATE TABLE [dbo].[AspNetUserRoles] (  
2     [UserId] NVARCHAR (128) NOT NULL,  
3     [RoleId] NVARCHAR (128) NOT NULL,  
4     [Id] INT IDENTITY (1, 1) NOT NULL,  
5     CONSTRAINT [PK_AspNetUserRoles] PRIMARY KEY CLUSTERED ([Id] ASC),  
6     CONSTRAINT [FK_dbo_AspNetUserRoles_dbo_AspNetRoles_RoleId] FOREIGN KEY ([RoleId]) REFERENCES [dbo].[AspNetRoles] ([Id]) ON DELETE CASCADE,  
7     CONSTRAINT [FK_dbo_AspNetUserRoles_dbo_AspNetUsers_UserId] FOREIGN KEY ([UserId]) REFERENCES [dbo].[AspNetUsers] ([Id]) ON DELETE CASCADE  
8 );  
9  
10  
11 GO  
12 CREATE NONCLUSTERED INDEX [IX_AspNetUserRoles]  
13 ON [dbo].[AspNetUserRoles]([UserId] ASC);  
14
```

Fuente: Elaboración Propia

Figura N° 110: Procedimiento almacenado AccesoUsuarios

```
1 CREATE PROCEDURE dbo.AccesosUsuarios  
2     @Username nvarchar(50),  
3     @RoleId int  
4 AS  
5 BEGIN  
6     SET NOCOUNT ON;  
7     INSERT INTO [dbo].[AspNetUserRoles]  
8         ([UserId]  
9         , [RoleId])  
10        VALUES  
11        ((select Id from [dbo].[AspNetUsers] where UserName = @Username), @RoleId);  
12 END
```

Fuente: Elaboración Propia

Figura N° 111: Código del controlador Accesos

```
using System;using System.Collections.Generic;using System.Data;using System.Data.Entity;using System.Linq;
using System.Net;using System.Web;using System.Web.Mvc;using SistemaControlTI.Models;
namespace SistemaControlTI.Controllers
{[Authorize(Roles = "Administrador")]
  0 referencias
  public class AspNetUserRolesController : Controller
  { private WebConSerSisEntities db = new WebConSerSisEntities();
  // GET: AspNetUserRoles
  0 referencias
  public ActionResult Index()
  { var aspNetUserRoles = db.AspNetUserRoles.Include(a => a.AspNetRoles).Include(a => a.AspNetUsers);
    return View(aspNetUserRoles.ToList());}
  // GET: AspNetUserRoles/Create
  0 referencias
  public ActionResult Create()
  {ViewBag.RoleId = new SelectList(db.AspNetRoles, "Id", "Name");
    ViewBag.UserId = new SelectList(db.AspNetUsers, "Id", "UserName"); return View();}
  // POST: AspNetUserRoles/Create
  [HttpPost]
  [ValidateAntiForgeryToken]
  0 referencias
  public ActionResult Create([Bind(Include = "UserId,RoleId,Id")] AspNetUserRoles aspNetUserRoles)
  {if (ModelState.IsValid)
    {db.AspNetUserRoles.Add(aspNetUserRoles); db.SaveChanges();return RedirectToAction("Index");}
    ViewBag.RoleId = new SelectList(db.AspNetRoles, "Id", "Name", aspNetUserRoles.RoleId);
    ViewBag.UserId = new SelectList(db.AspNetUsers, "Id", "UserName", aspNetUserRoles.UserId);
    return View(aspNetUserRoles);}
  // GET: AspNetUserRoles/Edit/5
  0 referencias
  public ActionResult Edit(int? id)
  {if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
    AspNetUserRoles aspNetUserRoles = db.AspNetUserRoles.Find(id);
    if (aspNetUserRoles == null){return HttpNotFound();}
    ViewBag.RoleId = new SelectList(db.AspNetRoles, "Id", "Name", aspNetUserRoles.RoleId);
    ViewBag.UserId = new SelectList(db.AspNetUsers, "Id", "UserName", aspNetUserRoles.UserId);
    return View(aspNetUserRoles);}
  // POST: AspNetUserRoles/Edit/5
  [HttpPost]
  [ValidateAntiForgeryToken]
  0 referencias
  public ActionResult Edit([Bind(Include = "UserId,RoleId,Id")] AspNetUserRoles aspNetUserRoles)
  { if (ModelState.IsValid){db.Entry(aspNetUserRoles).State = EntityState.Modified;
    db.SaveChanges();return RedirectToAction("Index");}
    ViewBag.RoleId = new SelectList(db.AspNetRoles, "Id", "Name", aspNetUserRoles.RoleId);
    ViewBag.UserId = new SelectList(db.AspNetUsers, "Id", "UserName", aspNetUserRoles.UserId);
    return View(aspNetUserRoles); }
  // GET: AspNetUserRoles/Delete/5
  0 referencias
  public ActionResult Delete(int? id)
  {if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
    AspNetUserRoles aspNetUserRoles = db.AspNetUserRoles.Find(id);
    if (aspNetUserRoles == null){return HttpNotFound();}return View(aspNetUserRoles);}
```

Fuente: Elaboración Propia

Figura N° 112: Código de la vista Index Accesos

```

@model IEnumerable<SistemaControlTI.Models.AspNetUserRoles>
@{ ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml"; }
<h2>Registrar Nuevo Usuario del Sistema</h2>
<p>@Html.ActionLink("Create New", "Create")</p>
<table class="table">
    <tr><th>Rol</th>
    <th>Nombre de usuario</th></tr>
    @foreach (var item in Model) {
    <tr>
        <td>@Html.DisplayFor(modelItem => item.AspNetUsers.UserName)</td>
        <td> @Html.DisplayFor(modelItem => item.AspNetRoles.Name)</td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id = item.Id }) |
            @Html.ActionLink("Details", "Details", new { id = item.Id }) |
            @Html.ActionLink("Delete", "Delete", new { id = item.Id })
        </td></tr>
    }</table>

```

Fuente: Elaboración Propia

Figura N° 113: Código de la vista Crear Acceso

```

@model SistemaControlTI.Models.AspNetUserRoles
@{ ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml"; }
<h2>Create</h2>
@using (Html.BeginForm())
{ @Html.AntiForgeryToken()
<div class="form-horizontal">
    <h4>AspNetUserRoles</h4>
    <hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
        @Html.LabelFor(model => model.UserId, "UserId", htmlAttributes: new { @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("UserId", null, htmlAttributes: new { @class = "form-control" })
            @Html.ValidationMessageFor(model => model.UserId, "", new { @class = "text-danger" })
        </div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.RoleId, "RoleId", htmlAttributes: new { @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("RoleId", null, htmlAttributes: new { @class = "form-control" })
            @Html.ValidationMessageFor(model => model.RoleId, "", new { @class = "text-danger" })
        </div> </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Create" class="btn btn-default" />
        </div></div></div>
}
<div>@Html.ActionLink("Back to List", "Index")</div>

```

Fuente: Elaboración Propia

Figura N° 114: Código de la vista Editar Acceso

```

@model SistemaControlTI.Models.AspNetUserRoles
@{ViewBag.Title = "Edit";
Layout = "~/Views/Shared/_Layout.cshtml";}
<h2>Editar</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>AspNetUserRoles</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.UserId, "UserId", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("UserId", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.UserId, "", new { @class = "text-danger" })</div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.RoleId, "RoleId", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("RoleId", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.RoleId, "", new { @class = "text-danger" })</div></div>
            @Html.HiddenFor(model => model.Id)
            <div class="form-group">
                <div class="col-md-offset-2 col-md-10">
                    <input type="submit" value="Save" class="btn btn-default" /></div></div></div>
        </div>
        @Html.ActionLink("Back to List", "Index") </div>
}

```

Fuente: Elaboración Propia

Figura N° 115: Código de la vista Eliminar Acceso

```

@model SistemaControlTI.Models.AspNetUserRoles
@{ViewBag.Title = "Eliminar";
Layout = "~/Views/Shared/_Layout.cshtml";}
<h2>Delete</h2>
<h3>Esta seguro que desea borrarlo?</h3>
<div><h4>AspNetUserRoles</h4>
<hr />
<dl class="dl-horizontal">
    <dt> @Html.DisplayNameFor(model => model.AspNetRoles.Name)</dt>
    <dd> @Html.DisplayFor(model => model.AspNetRoles.Name)</dd>
    <dt> @Html.DisplayNameFor(model => model.AspNetUsers.Email)</dt>
    <dd> @Html.DisplayFor(model => model.AspNetUsers.Email)</dd>
</dl>
@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()
    <div class="form-actions no-color">
        <input type="submit" value="Eliminar" class="btn btn-default" />
        @Html.ActionLink("Regresar al listado", "Index")</div>
}
</div>

```

Fuente: Elaboración Propia

GUI

Figura N° 116: Formulario Index Acceso

Nombre de Usuario	Correo	
AAIQUIPA	alex@gmail.com	Edit Details Delete
AIQUIPA	aiquipa@gmail.com	Edit Details Delete
CORELLANA	c@g.com	Edit Details Delete
JFLORES	ccamp@gmail.com	Edit Details Delete

Fuente: Elaboración Propia

Figura N° 117: Formulario Editar Acceso

Editar
AspNetUserRoles

UserId

RoleId

[Back to List](#)

Fuente: Elaboración Propia

Figura N° 118: Formulario Crear Acceso

Create
AspNetUserRoles

UserId

RoleId

[Back to List](#)

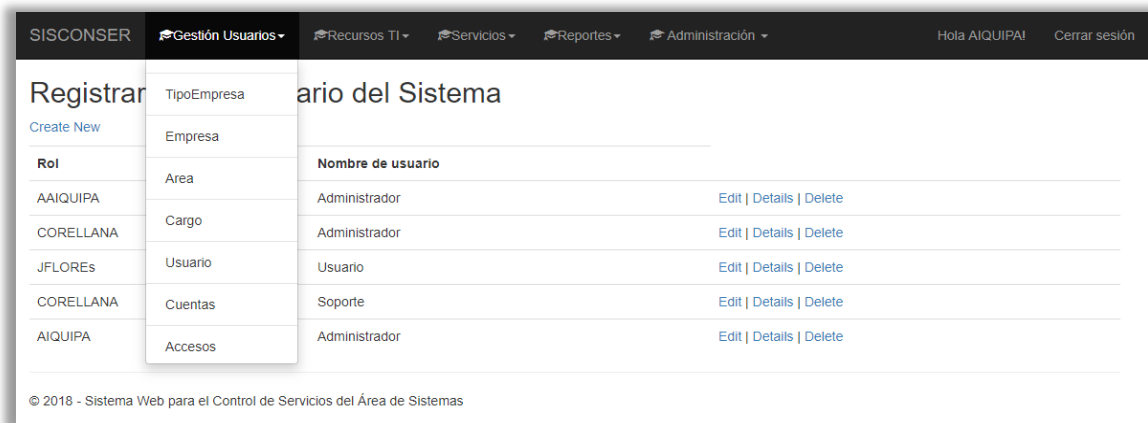
Fuente: Elaboración Propia

Figura N° 119: Formulario Eliminar Acceso



Fuente: Elaboración Propia

Figura N° 120: Formulario Completado del módulo de Gestión de Usuario



Fuente: Elaboración Propia

TaskBoard Semana 3 del Sprint 1

Tabla N° 10: TaskBoard Sprint 1 Semana 3

TaskBoard S3				
Sistema web para el control de servicios del área de sistemas en la compañía Naviera Natalia SAC			Fecha Inicio: 27/07/2018 Fecha final: 31/07/2018	
	Historias por cada Sprint	Pendiente	En curso	Hecho
SPRINT N° 1	Formulario de inicio de sesión para el acceso al Sistema.			X
	CRUD para los tipos de empresas de la Compañía.			X
	CRUD para las empresas colaboradoras de la Compañía			X
	CRUD para las áreas que dispone la Compañía.			X
	CRUD para los cargos de los usuarios que laboraran en la Compañía.			X
	CRUD para el mantenimiento de los usuarios que laboran en la Compañía.			X
	Registrar Cuentas de usuarios del sistema.			X
	Administrar el nivel de acceso de los usuarios al sistema.			X

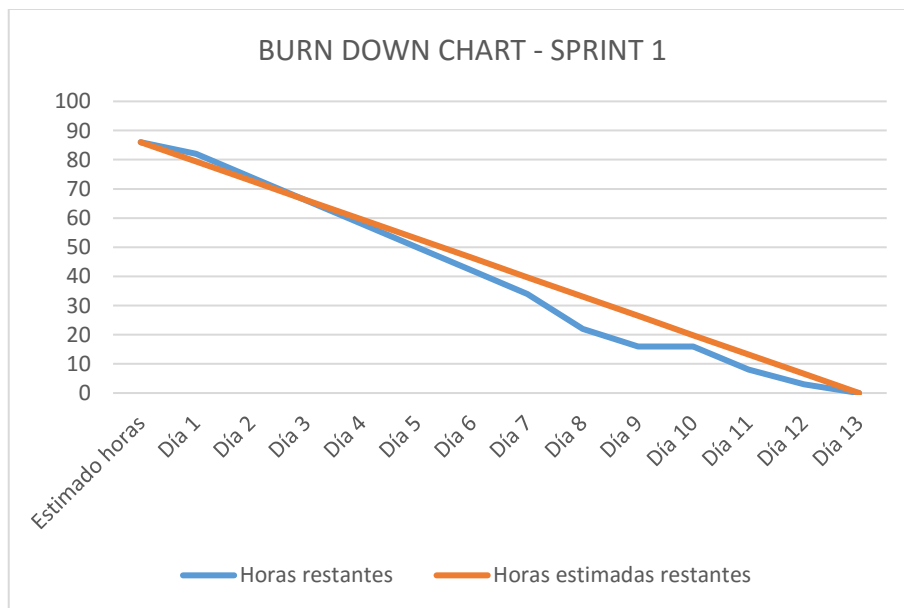
Fuente: Elaboración Propia

Tabla N° 11: Burndown del Sprint 1

BurnDown Sprint 1															
Tareas	Estimado horas	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7	Día 8	Día 9	Día 10	Día 11	Día 12	Día 13	Total horas
		Formulario de inicio de sesión para el acceso al Sistema.	4	4	0	0	0	0	0	0	0	0	0	0	
CRUD para los tipos de empresas de la Compañía.	8	0	6	2	0	0	0	0	0	0	0	0	0	0	8
CRUD para las empresas colaboradoras de la Compañía	10	0	2	0	8	0	0	0	0	0	0	0	0	0	10
CRUD para las áreas que dispone la Compañía.	8	0	0	0	0	8	0	0	0	0	0	0	0	0	8
CRUD para los cargos de los usuarios que laboraran en la Compañía.	8	0	0	6	0	0	2	0	0	0	0	0	0	0	8
CRUD para el mantenimiento de los usuarios que laboran en la Compañía.	16	0	0	0	0	0	0	4	8	4	0	0	0	0	16
Registrar Cuentas de usuarios del sistema.	16	0	0	0	0	0	6	4	4	2	0	0	0	0	16
Administrar el nivel de acceso de los usuarios al sistema.	16	0	0	0	0	0	0	0	0	0	0	8	5	3	16
Horas restantes	86	82	74	66	58	50	42	34	22	16	16	8	3	0	
Horas estimadas restantes	86	79,4	72,8	66,2	59,5	52,9	46,3	39,7	33,1	26,5	19,8	13,2	6,62	0	

Fuente: Elaboración Propia

Figura N° 121: Grafica Burndown del Sprint 1



Fuente: Elaboración Propia

ACTA DE REUNIÓN N° 2

CIERRE DEL SPRINT 1

DATOS

Empresa / Organización	Compañía Naviera Natalia SAC.
Proyecto	Sistema web para el control de servicios del área de sistemas en la compañía Naviera Natalia SAC

PARTICIPANTES

Rol	Nombre
Dueño del Producto	Candela Ríos Hugo
Equipo	Aiquipa Tello Alex

ACUERDOS

Marcar con "X" a razón de cierre de cumplimiento de cada funcionalidad en la apertura del Sprint 1

Historias	Entrega Total
El sistema debe de tener un formulario que permita autenticar el ingreso a los usuarios del sistema web.	X
El sistema debe permitir al usuario autenticado registrar y actualizar los tipos de empresas de la Compañía.	X
El sistema debe permitir al usuario autenticado registrar y actualizar las empresas colaboradoras de la Compañía.	X
El sistema debe permitir al usuario autenticado registrar y actualizar las empresas a las áreas que dispone la Compañía.	X
El sistema debe permitir usuario autenticado registrar y actualizar a los cargos de los usuarios que laboran en la Compañía.	X
El sistema debe permitir usuario autenticado registrar y actualizar los usuarios que laboran en la Compañía.	X
El sistema debe permitir al Administrador, crear y modificar cuentas de usuarios del sistema web.	X
El sistema debe permitir al Administrador clasificar el nivel de acceso de las cuentas de los usuarios del sistema	X

FIRMAS


 ING. HUGO CANDELA
 JEFE DE SISTEMAS CORPORATIVO


 ALEX ALDO AIQUIPA TELLO
 ADMINISTRADOR DE REDES Y COMUNICACIONES

 DUEÑO DEL PRODUCTO

 EQUIPO

ACTA DE REUNIÓN N° 3

APERTURA DEL SPRINT 2

DATOS

Empresa / Organización	Compañía Naviera Natalia SAC.
Proyecto	Sistema web para el control de servicios del área de sistemas en la compañía Naviera Natalia SAC

PARTICIPANTES

Rol	Nombre
Dueño del Producto	Candela Ríos Hugo
Equipo	Aiquipa Tello Alex

ACUERDOS

A continuación se listan las funcionalidades que se de desarrollaran e implementaran en el Sprint 2

Historias
El sistema debe permitir al usuario autenticado, registrar y actualizar los activos de hardware de la Compañía.
El sistema debe permitir al usuario autenticado, registrar y actualizar los activos de software de la Compañía.
El sistema debe permitir al usuario autenticado, registrar y actualizar los insumos de los equipos de impresión de la Compañía.
El sistema debe permitir al usuario autenticado, registrar y actualizar los componentes de los ordenadores.
El sistema debe permitir al usuario autenticado registrar y actualizar los componentes de ensamblaje de computo.
El sistema debe permitir al usuario autenticado registrar y actualizar los cambios de insumos en los equipos de impresión .
El sistema debe permitir al usuario autenticado registrar y actualizar la asignación de los equipos informáticos al personal de la compañía.
El sistema debe permitir al usuario autenticado registrar y actualizar las licencias asignadas a los equipos informáticos .

Firmas:


.....
ING. HUGO CANDELA
JEFE DE SISTEMAS CORPORATIVO

DUÑO DEL PRODUCTO


.....
ALEXALDO AIQUIPA TELLO
ADMINISTRADOR DE REDES Y COMUNICACIONES

EQUIPO

Sprint N° 2: Story point 100

Tabla N° 12: TaskBoard Sprint 2 Semana 4

TaskBoard S4					
Sistema web para el control de os servicios de área de sistemas en la compañía Naviera Natalia SAC			Fecha Inicio: 01/08/2018 Fecha final: 07/08/2018		
Historias por cada Sprint			Pendiente	En curso	Hecho
SPRINT N° 2	CRUD para los activos hardware de la Compañía.				X
	CRUD para los activos software de la Compañía.				X
	CRUD para los insumos de los equipos de impresión.				X
	CRUD para los componentes de las computadoras.			X	
	Registrar de los componentes que contiene un equipo de cómputo.		X		
	Registrar la asignación de los insumos a los equipos de impresión.		X		
	Registrar los equipos informáticos asignados al personal administrativo de la Compañía.		X		
	Registrar los licenciamientos de los equipos informáticos.		X		

Fuente: Elaboración Propia

Análisis Sprint 2

Caso de Uso:

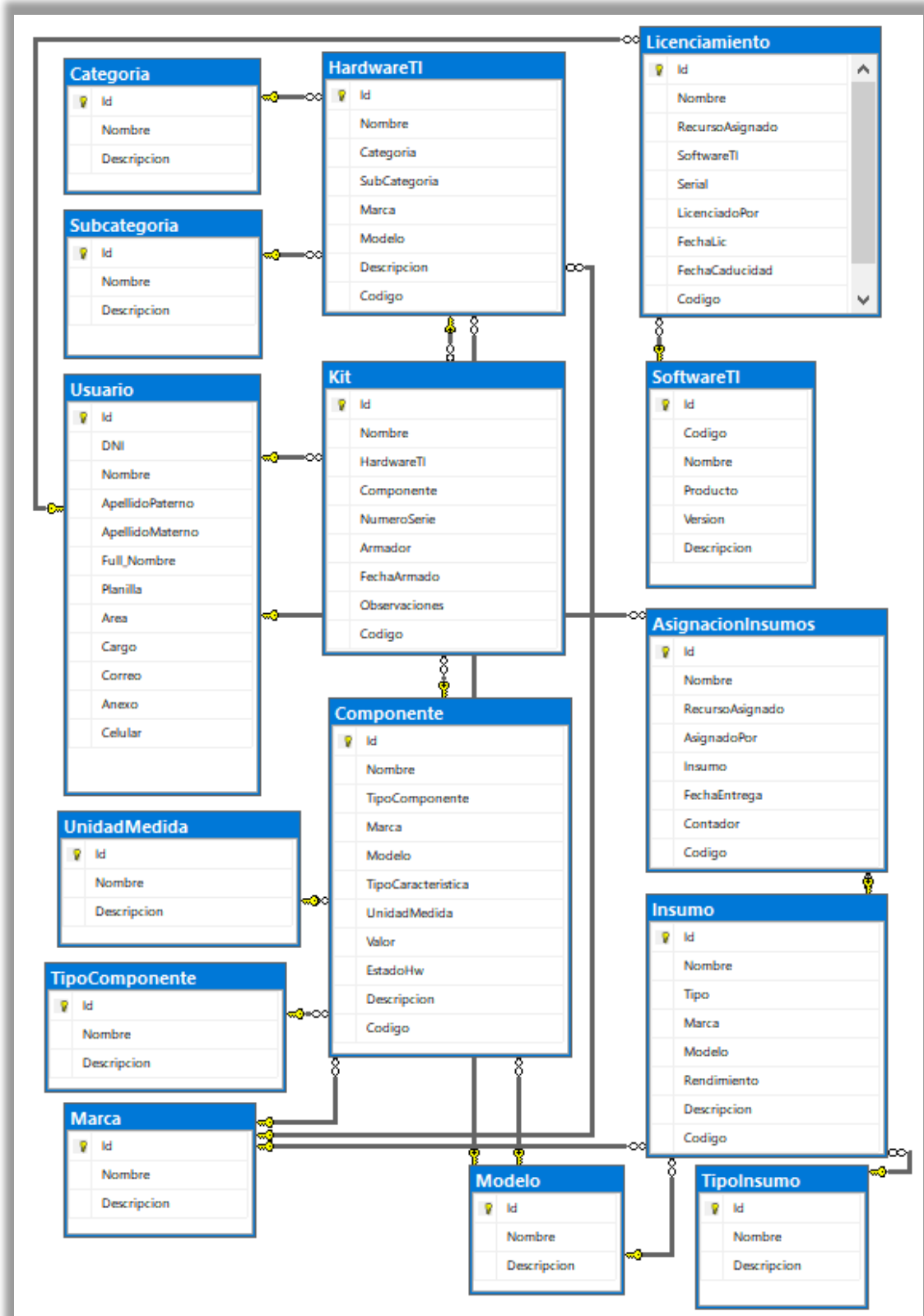
Figura N° 122: Caso de uso del sprint 2



Fuente: Elaboración Propia

Diseño físico de la base de datos del sprint 2

Figura N° 123: Diseño lógico de la base de datos del sprint 2



Fuente: Elaboración Propia

Requerimiento RF9

RF09: El sistema debe permitir al usuario autenticado, registrar y actualizar los activos de hardware de la Compañía.

Prototipo 1 RF9

Figura N° 124: Prototipo 1 Index Hardware

Código	Tipo	Marca	Modelo	Nombre	Editar	Detalle	Eliminar
					Editar	Detalle	Eliminar
					Editar	Detalle	Eliminar

Fuente: Elaboración Propia

Prototipo 2 RF9

Figura N° 125: Prototipo 2 Registrar Hardware

Crear Hardware

Subcategoría: text goes here

Marca: text goes here

Modelo: text goes here

Nombre: text

Descripción:

Crear

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 3 RF9

Figura N° 126: Prototipo 3 Editar Hardware

Editar Hardware

Subcategoria: text goes here ▼

Marca: text goes here ▼

Modelo: text goes here ▼

Nombre: text

Descripción:

Crear

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 4 RF9

Figura N° 127: Prototipo 4 Eliminar Hardware

Borrar

¿Estas Seguro que deseas borrarlos ?

Codigo : Label

Eliminar [Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 128: Código para la creación de la tabla Hardware

```
1 CREATE TABLE [dbo].[HardwareTI] (  
2 [Id] INT IDENTITY (1, 1) NOT NULL,  
3 [Nombre] NVARCHAR (50) NOT NULL,  
4 [Categoria] INT NULL,  
5 [SubCategoria] INT NULL,  
6 [Marca] INT NULL,  
7 [Modelo] INT NULL,  
8 [Descripcion] NVARCHAR (200) NULL,  
9 [Codigo] AS ('H'+right('000000'+CONVERT([varchar],[Id]),(6))),  
10 PRIMARY KEY CLUSTERED ([Id] ASC),  
11 CONSTRAINT [FK_ProductoTI_Categoria] FOREIGN KEY ([Categoria]) REFERENCES [dbo].[Categoria] ([Id]),  
12 CONSTRAINT [FK_ProductoTI_SubCategoria] FOREIGN KEY ([SubCategoria]) REFERENCES [dbo].[Subcategoria] ([Id]),  
13 CONSTRAINT [FK_ProductoTI_Marca] FOREIGN KEY ([Marca]) REFERENCES [dbo].[Marca] ([Id]),  
14 CONSTRAINT [FK_ProductoTI_Modelo] FOREIGN KEY ([Modelo]) REFERENCES [dbo].[Modelo] ([Id])
```

Fuente: Elaboración Propia

Figura N° 129: Código del modelo Hardware

```
public partial class HardwareTI
{[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    0 referencias | 0 excepciones
    public HardwareTI()
    {this.RecursoAsignado = new HashSet<RecursoAsignado>();
      this.Kit = new HashSet<Kit>();
      this.Requerimiento = new HashSet<Requerimiento>();}
    0 referencias | 0 excepciones
    public int Id { get; set; }
    1 referencia | 0 excepciones
    public string Nombre { get; set; }
    3 referencias | 0 excepciones
    public Nullable<int> Categoria { get; set; }
    3 referencias | 0 excepciones
    public Nullable<int> SubCategoria { get; set; }
    3 referencias | 0 excepciones
    public Nullable<int> Marca { get; set; }
    3 referencias | 0 excepciones
    public Nullable<int> Modelo { get; set; }
    1 referencia | 0 excepciones
    public string Descripcion { get; set; }
    1 referencia | 0 excepciones
    public string Codigo { get; set; }
```

Fuente: Elaboración Propia

Figura N° 130: Código del controlador Hardware

```
public class HardwareTIsController : Controller
{private WebConSerSisEntities db = new WebConSerSisEntities();
    0 referencias | 3 solicitudes | 0 excepciones
    public ActionResult Index()
    {var hardwareTI = db.HardwareTI.Include(h => h.Categorial).Include(h => h.Marca1).Include(h => h.Modelo1).Include(h => h.Subcategoria1);
      return View(hardwareTI.ToList());}
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Details(int? id)
    0 referencias | 1 solicitud | 0 excepciones
    public ActionResult Create()
    {ViewBag.Categoria = new SelectList(db.Categoria, "Id", "Nombre");
      ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre");
      ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre");
      ViewBag.SubCategoria = new SelectList(db.Subcategoria, "Id", "Nombre");
      return View();}
    [HttpPost] [ValidateAntiForgeryToken]
    0 referencias | 1 solicitud | 0 excepciones
    public ActionResult Create([Bind(Include = "Id,Nombre,Categoria,SubCategoria,Marca,Modelo,Descripcion,Codigo")] HardwareTI hardwareTI)
    {if (ModelState.IsValid) {db.HardwareTI.Add(hardwareTI);
      db.SaveChanges();
      return RedirectToAction("Index");}
      ViewBag.Categoria = new SelectList(db.Categoria, "Id", "Nombre", hardwareTI.Categoria);
      ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre", hardwareTI.Marca);
      ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre", hardwareTI.Modelo);
      ViewBag.SubCategoria = new SelectList(db.Subcategoria, "Id", "Nombre", hardwareTI.SubCategoria);
      return View(hardwareTI);}
    0 referencias | 1 solicitud | 0 excepciones
    public ActionResult Edit(int? id)
    {if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
      HardwareTI hardwareTI = db.HardwareTI.Find(id);
      if (hardwareTI == null){return HttpNotFound();}
      ViewBag.Categoria = new SelectList(db.Categoria, "Id", "Nombre", hardwareTI.Categoria);
      ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre", hardwareTI.Marca);
      ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre", hardwareTI.Modelo);
      ViewBag.SubCategoria = new SelectList(db.Subcategoria, "Id", "Nombre", hardwareTI.SubCategoria);
      return View(hardwareTI);}
    [HttpPost] [ValidateAntiForgeryToken]
    0 referencias | 1 solicitud | 0 excepciones
    public ActionResult Edit([Bind(Include = "Id,Nombre,Categoria,SubCategoria,Marca,Modelo,Descripcion,Codigo")] HardwareTI hardwareTI)
    {if (ModelState.IsValid){db.Entry(hardwareTI).State = EntityState.Modified;
      db.SaveChanges();
      return RedirectToAction("Index");}
      ViewBag.Categoria = new SelectList(db.Categoria, "Id", "Nombre", hardwareTI.Categoria);
      ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre", hardwareTI.Marca);
      ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre", hardwareTI.Modelo);
      ViewBag.SubCategoria = new SelectList(db.Subcategoria, "Id", "Nombre", hardwareTI.SubCategoria);
      return View(hardwareTI); }
    0 referencias | 2 solicitudes | 0 excepciones
    public ActionResult Delete(int? id)
    {if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
      HardwareTI hardwareTI = db.HardwareTI.Find(id);
      if (hardwareTI == null){ return HttpNotFound(); }
      return View(hardwareTI);}
    [HttpPost, ActionName("Delete")] [ValidateAntiForgeryToken]
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult DeleteConfirmed(int id)
    { HardwareTI hardwareTI = db.HardwareTI.Find(id);
      db.HardwareTI.Remove(hardwareTI);
      db.SaveChanges();
      return RedirectToAction("Index");}
```

Fuente: Elaboración Propia

Figura N° 131: Código de la vista Index Hardware

```

<h2>Hardware</h2>
<p> @Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
  <tr> <th>Codigo</th>
  <th>Tipo</th>
  <th>Marca</th>
  <th>Modelo</th>
  <th>Nombre</th></tr>

  @foreach (var item in Model) {
  <tr><td>@Html.DisplayFor(modelItem => item.Codigo)</td>
  <td>@Html.DisplayFor(modelItem => item.Subcategoria1.Nombre)</td>
  <td>@Html.DisplayFor(modelItem => item.Marca1.Nombre)</td>
  <td>@Html.DisplayFor(modelItem => item.Modelo1.Nombre)</td>
  <td>@Html.DisplayFor(modelItem => item.Nombre)</td>
  <td>@Html.ActionLink("Editar", "Edit", new { id = item.Id })
  @Html.ActionLink("Detalles", "Details", new { id = item.Id })
  @Html.ActionLink("Eliminar", "Delete", new { id = item.Id }) </td></tr>
}
</table>

```

Fuente: Elaboración Propia

Figura N° 132: Código de la vista Crear Hardware

```

<h2>Crear</h2>
@using (Html.BeginForm())
{
  @Html.AntiForgeryToken()
  <div class="form-horizontal">
    <h4>Hardware</h4><hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
      @Html.LabelFor(model => model.SubCategoria, "SubCategoria", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("SubCategoria", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.SubCategoria, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Marca, "Marca", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("Marca", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Marca, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Modelo, "Modelo", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("Modelo", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Modelo, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Categoria, "Categoria", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("Categoria", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Categoria, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Crear" class="btn btn-default" /></div></div>
  </div>
  @Html.ActionLink("Regresar al listado", "Index")</div>
}

```

Fuente: Elaboración Propia

Figura N° 133: Código de la vista Actualizar Hardware

```

<div class="form-horizontal">
  <h4>Hardware</h4><hr />
  @Html.ValidationSummary(true, "", new { @class = "text-danger" })
  @Html.HiddenFor(model => model.Id)
  <div class="form-group">
    @Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.SubCategoria, "SubCategoria", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("SubCategoria", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.SubCategoria, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Marca, "Marca", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("Marca", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Marca, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Modelo, "Modelo", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("Modelo", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Modelo, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Categoria, "Categoria", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("Categoria", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Categoria, "", new { @class = "text-danger" })</div></div>
    <div class="form-group"><div class="col-md-offset-2 col-md-10"><input type="submit" value="Guardar" class="btn btn-default" />
    </div></div></div>
  @Html.ActionLink("Regresar al listado", "Index")</div>

```

Fuente: Elaboración Propia

Figura N° 134: Código de la vista Eliminar Hardware

```

<h2>Eliminar</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div> <h4>Hardware</h4> <hr />
  <dl class="dl-horizontal">
    <dt>Codigo</dt>
    <dd>@Html.DisplayFor(model => model.Codigo)</dd>
    <dt>Tipo</dt>
    <dd>@Html.DisplayFor(model => model.Subcategoria1.Nombre)</dd>
    <dt> Marca</dt>
    <dd> @Html.DisplayFor(model => model.Marca1.Nombre) </dd>
    <dt>Modelo</dt>
    <dd>@Html.DisplayFor(model => model.Modelo1.Nombre)</dd>
    <dt>Nombre</dt>
    <dd>@Html.DisplayFor(model => model.Nombre)</dd>
    <dt>Descripcion</dt>
    <dd>@Html.DisplayFor(model => model.Descripcion)</dd>
    <dt>Categoria</dt>
    <dd>@Html.DisplayFor(model => model.Categoria1.Nombre)</dd>
  </dl>@using (Html.BeginForm())
  {
    @Html.AntiForgeryToken()
    <div class="form-actions no-color">
      <input type="submit" value="Eliminar" class="btn btn-default" /> |
      @Html.ActionLink("Regresar al listado", "Index")
    </div>
  }

```

Fuente: Elaboración Propia

GUI

Figura N° 135: Formulario Index Hardware

Codigo	Tipo	Marca	Modelo	Nombre	
H000001	CPU	HP	IR2530	hp	Editar Detalles Eliminar
H000002	CPU	PcChip	P46G	PcChip i3	Editar Detalles Eliminar
H000003	CPU	Dell	80NR	Dell 8g	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 136: Formulario Crear Hardware

SubCategoria: CPU

Marca: HP

Modelo: IR2530

Nombre:

Descripcion:

Categoria: Hardware

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 137: Formulario Actualizar Hardware

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Actualizar

Hardware

Código

SubCategoría

Marca

Modelo

Nombre

Descripción

Categoría

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 138: Formulario Eliminar Hardware

SISCONSER [Gestión Usuarios](#) [Recursos TI](#)

Eliminar

¿Estas seguro que deseas borrarlo?

Hardware

Código H000001
Tipo CPU
Marca HP
Modelo IR2530
Nombre hp
Descripción
Categoría Hardware

| [Regresar al listado](#)

Fuente: Elaboración Propia

Requerimiento RF10

RF10: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los activos software de la Compañía.

Diseño

Prototipo 1 RF10

Figura N° 139: Prototipo 1 Index Software

SisConSer Gestión de Usuarios Recursos Servicios Reportes Cerrar Sesión

Software
[Crear nuevo](#)

Código	Producto	Versión	Nombre			
				Editar	Detalle	Eliminar
				Editar	Detalle	Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 2 RF10

Figura N° 140: Prototipo 2 Crear Software

Crear Software

Producto

Versión

Nombre

Descripción

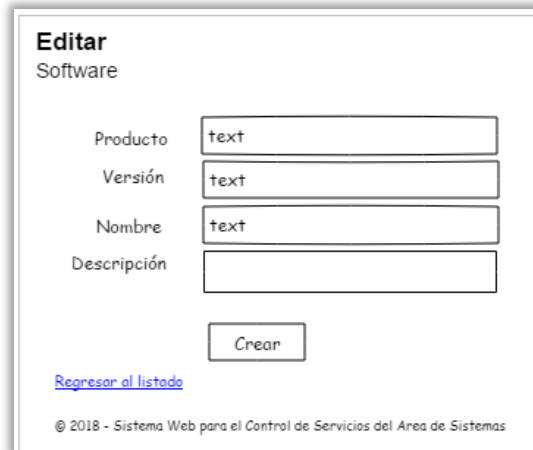
[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 3 RF10

Figura N° 141: Prototipo 3 Actualizar Software



Editar
Software

Producto

Versión

Nombre

Descripción

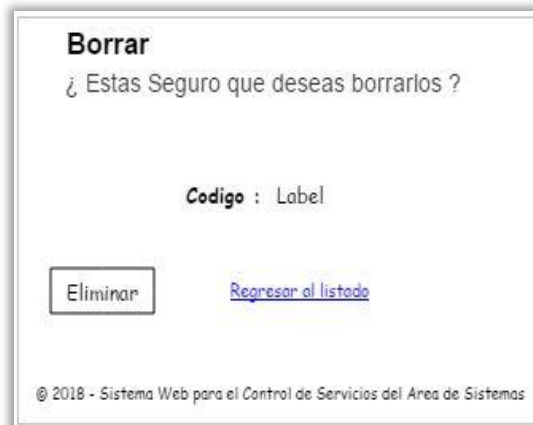
[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Prototipo 4 RF10

Figura N° 142: Prototipo Eliminar Software



Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 143: Código para la creación de la tabla Software

```
1 CREATE TABLE [dbo].[SoftwareTI] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Codigo] AS ('S'+right('000000'+CONVERT([varchar],[Id]),(6))),  
4     [Nombre] NVARCHAR (50) NOT NULL,  
5     [Producto] NVARCHAR (100) NOT NULL,  
6     [Version] NVARCHAR (30) NOT NULL,  
7     [Descripcion] NVARCHAR (200) NULL,  
8     PRIMARY KEY CLUSTERED ([Id] ASC)  
9 );  
10  
11
```

Fuente: Elaboración Propia

Figura N° 144: Código del modelo Software

```
public partial class SoftwareTI
{[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    0 referencias | 0 excepciones
    public SoftwareTI()
    {this.Licenciamiento = new HashSet<Licenciamiento>();}

    0 referencias | 0 excepciones
    public int Id { get; set; }
    2 referencias | 0 excepciones
    public string Codigo { get; set; }
    2 referencias | 0 excepciones
    public string Nombre { get; set; }
    2 referencias | 0 excepciones
    public string Producto { get; set; }
    2 referencias | 0 excepciones
    public string Version { get; set; }
    2 referencias | 0 excepciones
    public string Descripcion { get; set; }
```

Fuente: Elaboración Propia

Figura N° 145: Código del controlador Software

```
public class SoftwareTISController : Controller
{private WebConSerSisEntities db = new WebConSerSisEntities();
    0 referencias | 5 solicitudes | 0 excepciones
    public ActionResult Index()
    {return View(db.SoftwareTI.ToList());}
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult Details(int? id)
    {if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        SoftwareTI softwareTI = db.SoftwareTI.Find(id);
        if (softwareTI == null)
        {return HttpNotFound();}
        return View(softwareTI);}
    0 referencias | 1 solicitud | 0 excepciones
    public ActionResult Create()
    { return View();}
    [HttpPost]
    [ValidateAntiForgeryToken]
    0 referencias | 1 solicitud | 0 excepciones
    public ActionResult Create([Bind(Include = "Id,Codigo,Nombre,Producto,Version,Descripcion")] SoftwareTI softwareTI)
    {if (ModelState.IsValid)
        {db.SoftwareTI.Add(softwareTI);
            db.SaveChanges();
            return RedirectToAction("Index");}
        return View(softwareTI);}
    0 referencias | 1 solicitud | 0 excepciones
    public ActionResult Edit(int? id)
    { if (id == null)
        {return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        SoftwareTI softwareTI = db.SoftwareTI.Find(id);
        if (softwareTI == null) {return HttpNotFound();}
        return View(softwareTI);}
    [HttpPost]
    [ValidateAntiForgeryToken]
    0 referencias | 1 solicitud | 0 excepciones
    public ActionResult Edit([Bind(Include = "Id,Codigo,Nombre,Producto,Version,Descripcion")] SoftwareTI softwareTI)
    {if (ModelState.IsValid)
        {db.Entry(softwareTI).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");}
        return View(softwareTI);}
    0 referencias | 2 solicitudes | 0 excepciones
    public ActionResult Delete(int? id)
    {if (id == null)
        {return new HttpStatusCodeResult(HttpStatusCode.BadRequest);}
        SoftwareTI softwareTI = db.SoftwareTI.Find(id);
        if (softwareTI == null)
        {return HttpNotFound();}
        return View(softwareTI);}
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    0 referencias | 0 solicitudes | 0 excepciones
    public ActionResult DeleteConfirmed(int id)
    {SoftwareTI softwareTI = db.SoftwareTI.Find(id);
        db.SoftwareTI.Remove(softwareTI);
        db.SaveChanges();
        return RedirectToAction("Index");}
```

Fuente: Elaboración Propia

Figura N° 146: Código de la vista Index Software

```
<h2>Software</h2>
<p> @Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
  <tr><th> @Html.DisplayNameFor(model => model.Codigo)</th>
    <th>@Html.DisplayNameFor(model => model.Producto)</th>
    <th>@Html.DisplayNameFor(model => model.Version)</th>
    <th>@Html.DisplayNameFor(model => model.Nombre) </th></tr>
  @foreach (var item in Model) {
    <tr><td> @Html.DisplayFor(modelItem => item.Codigo)</td>
      <td>@Html.DisplayFor(modelItem => item.Producto)</td>
      <td>@Html.DisplayFor(modelItem => item.Version)</td>
      <td>@Html.DisplayFor(modelItem => item.Nombre)</td>
      <td>@Html.ActionLink("Editar", "Edit", new { id = item.Id })
        @Html.ActionLink("Detalles", "Details", new { id = item.Id })
        @Html.ActionLink("Eliminar", "Delete", new { id = item.Id })</td></tr>
  }
```

Fuente: Elaboración Propia

Figura N° 147: Código de la vista Crear Software

```
<h4>SoftwareTI</h4> <hr />
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
<div class="form-group">
  @Html.LabelFor(model => model.Producto, htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.Producto, new { htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.Producto, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
  @Html.LabelFor(model => model.Version, htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.Version, new { htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.Version, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
  @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
  @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
  <div class="col-md-offset-2 col-md-10">
    <input type="submit" value="Crear" class="btn btn-default" /></div></div></div>
<div>@Html.ActionLink("Regresar al listado", "Index")</div>
```

Fuente: Elaboración Propia

Figura N° 148: Código de la vista Editar Software

```
<div class="form-horizontal">
  <h4>SoftwareTI</h4> <hr />
  @Html.ValidationSummary(true, "", new { @class = "text-danger" })
  @Html.HiddenFor(model => model.Id)
  <div class="form-group">
    @Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.Producto, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Producto, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Producto, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.Version, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Version, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Version, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })</div></div>
  <div class="form-group"> <div class="col-md-offset-2 col-md-10">
    <input type="submit" value="Guardar" class="btn btn-default" /></div></div></div>
</div> @Html.ActionLink("Regresar al listado", "Index")</div>
```

Fuente: Elaboración Propia

Figura N° 149: Código de la vista Eliminar Software

```

<h2>Eliminar</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div><h4>SoftwareTI</h4><hr />
  <dl class="dl-horizontal">
    <dt>@Html.DisplayNameFor(model => model.Codigo)</dt>
    <dd>@Html.DisplayFor(model => model.Codigo)</dd>
    <dt>@Html.DisplayNameFor(model => model.Nombre) </dt>
    <dd> @Html.DisplayFor(model => model.Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.Producto)</dt>
    <dd>@Html.DisplayFor(model => model.Producto) </dd>
    <dt>@Html.DisplayNameFor(model => model.Version) </dt>
    <dd>@Html.DisplayFor(model => model.Version)</dd>
    <dt>@Html.DisplayNameFor(model => model.Descripcion) </dt>
    <dd>@Html.DisplayFor(model => model.Descripcion)</dd></dl>
    @using (Html.BeginForm()) {
      @Html.AntiForgeryToken()
      <div class="form-actions no-color">
        <input type="submit" value="Eliminar" class="btn btn-default" />
        @Html.ActionLink("Regresar al listado", "Index") </div></div>

```

Fuente: Elaboración Propia

GUI

Figura N° 150: Formulario Index Software

Codigo	Producto	Version	Nombre	
S000001	SAP ONE	V2,7	SAP	Editar Detalles Eliminar
S000002	SAP ONE	4	STARTSOFT	Editar Detalles Eliminar
S000003	NISIRA ONE	10	NISIRA	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 151: Formulario Crear Software

Crear SoftwareTI

Producto

Version

Nombre

Descripcion

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 152: Formulario Actualizar Software

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Actualizar

SoftwareTI

Codigo	<input type="text" value="S000001"/>
Producto	<input type="text" value="SAP ONE"/>
Version	<input type="text" value="V2,7"/>
Nombre	<input type="text" value="SAP"/>
Descripcion	<input type="text"/>

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 153: Formulario Eliminar Software

SISCONSER [Gestión Usuarios](#) [Recursos TI](#)

Eliminar

¿Estas seguro que deseas borrarlo?
SoftwareTI

Codigo	S000001
Nombre	SAP
Producto	SAP ONE
Version	V2,7
Descripcion	

| [Regresar al listado](#)

Fuente: Elaboración Propia

Requerimiento RF11

RF11: El sistema debe permitir al usuario autenticado, registrar y actualizar los insumos de los equipos de impresión de la Compañía. de la Compañía.

Figura N° 154: Prototipo 1 Index Insumos

Código	Tipo	Marca	Modelo	Nombre	Rendimiento			
						Editar	Detalle	Eliminar
						Editar	Detalle	Eliminar

Fuente: Elaboración Propia

Figura N° 155: Prototipo 2 Crear Insumo

Crear Insumo

Tipo: text goes here

Marca: text goes here

Modelo: text goes here

Nombre: text

Rendimiento: text

Descripción: text

Crear

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 156: Prototipo 3: Actualizar Insumo

Actualizar Insumo

Tipo: text goes here

Marca: text goes here

Modelo: text goes here

Nombre: text

Rendimiento: text

Descripción: text

Crear

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 157: Prototipo 4 Borrar Insumo

Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 158: Código para la creación de la tabla Insumo

```
1 CREATE TABLE [dbo].[Insumo] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nombre] NVARCHAR (50) NOT NULL,  
4     [Tipo] INT NOT NULL,  
5     [Marca] INT NOT NULL,  
6     [Modelo] INT NOT NULL,  
7     [Rendimiento] INT NULL,  
8     [Descripcion] NVARCHAR (100) NULL,  
9     [Codigo] AS ('I'+right('000000'+CONVERT([varchar],[Id]),(6))),  
10    PRIMARY KEY CLUSTERED ([Id] ASC),  
11    UNIQUE NONCLUSTERED ([Nombre] ASC),  
12    CONSTRAINT [FK_Insumos_Marca] FOREIGN KEY ([Marca]) REFERENCES [dbo].[Marca] ([Id]),  
13    CONSTRAINT [FK_Insumos_Modelo] FOREIGN KEY ([Modelo]) REFERENCES [dbo].[Modelo] ([Id]),  
14    CONSTRAINT [FK_Insumos_Tipo] FOREIGN KEY ([Tipo]) REFERENCES [dbo].[TipoInsumo] ([Id])  
15 );
```

Fuente: Elaboración Propia

Figura N° 159: Código del modelo Insumo

```
public Insumo()  
{ this.AsignacionInsumos = new HashSet<AsignacionInsumos>();  
  this.Eventos = new HashSet<Eventos>();  
}  
0 referencias | 0 excepciones  
public int Id { get; set; }  
1 referencia | 0 excepciones  
public string Nombre { get; set; }  
3 referencias | 0 excepciones  
public int Tipo { get; set; }  
3 referencias | 0 excepciones  
public int Marca { get; set; }  
3 referencias | 0 excepciones  
public int Modelo { get; set; }  
1 referencia | 0 excepciones  
public Nullable<int> Rendimiento { get; set; }  
1 referencia | 0 excepciones  
public string Descripcion { get; set; }  
1 referencia | 0 excepciones  
public string Codigo { get; set; }  
  
[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]  
1 referencia | 0 excepciones  
public virtual ICollection<AsignacionInsumos> AsignacionInsumos { get; set; }  
[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]  
1 referencia | 0 excepciones  
public virtual ICollection<Eventos> Eventos { get; set; }  
2 referencias | 0 excepciones  
public virtual Marca Marca1 { get; set; }  
2 referencias | 0 excepciones  
public virtual Modelo Modelo1 { get; set; }  
2 referencias | 0 excepciones  
public virtual TipoInsumo TipoInsumo { get; set; }
```

Fuente: Elaboración Propia

Figura N° 160: Código del controlador Insumo

```

public class InsumoesController : Controller
{
    private WebConSerSisEntities db = new WebConSerSisEntities();

    public ActionResult Index()
    {
        var insumo = db.Insumo.Include(i => i.Marca1).Include(i => i.Modelo1).Include(i => i.TipoInsumo);
        return View(insumo.ToList());
    }

    public ActionResult Details(int? id)
    {
        public ActionResult Create()
        {
            ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre");
            ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre");
            ViewBag.Tipo = new SelectList(db.TipoInsumo, "Id", "Nombre");
            return View();
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include = "Id,Nombre,Tipo,Marca,Modelo,Rendimiento,Descripcion,Codigo")] Insumo insumo)
        {
            if (ModelState.IsValid)
            {
                db.Insumo.Add(insumo);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre", insumo.Marca);
            ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre", insumo.Modelo);
            ViewBag.Tipo = new SelectList(db.TipoInsumo, "Id", "Nombre", insumo.Tipo);
            return View(insumo);
        }

        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Insumo insumo = db.Insumo.Find(id);
            if (insumo == null)
            {
                return HttpNotFound();
            }
            ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre", insumo.Marca);
            ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre", insumo.Modelo);
            ViewBag.Tipo = new SelectList(db.TipoInsumo, "Id", "Nombre", insumo.Tipo);
            return View(insumo);
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include = "Id,Nombre,Tipo,Marca,Modelo,Rendimiento,Descripcion,Codigo")] Insumo insumo)
        {
            if (ModelState.IsValid)
            {
                db.Entry(insumo).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre", insumo.Marca);
            ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre", insumo.Modelo);
            ViewBag.Tipo = new SelectList(db.TipoInsumo, "Id", "Nombre", insumo.Tipo);
            return View(insumo);
        }

        public ActionResult Delete(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Insumo insumo = db.Insumo.Find(id);
            if (insumo == null)
            {
                return HttpNotFound();
            }
            return View(insumo);
        }

        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public ActionResult DeleteConfirmed(int id)
        {
            Insumo insumo = db.Insumo.Find(id);
            db.Insumo.Remove(insumo);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
    }
}

```

Fuente: Elaboración Propia

Figura N° 161: Código de la vista Index Insumo

```

<h2>Insumo</h2>
<p> @Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
    <thead>
        <tr>
            <th>Codigo</th>
            <th>Tipo de Insumo</th>
            <th>Marca</th>
            <th>Modelo</th>
            <th>Nombre</th>
            <th>Rendimiento</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model) {
            <tr>
                <td>@Html.DisplayFor(modelItem => item.Codigo)</td>
                <td>@Html.DisplayFor(modelItem => item.TipoInsumo.Nombre)</td>
                <td>@Html.DisplayFor(modelItem => item.Marca1.Nombre)</td>
                <td>@Html.DisplayFor(modelItem => item.Modelo1.Nombre)</td>
                <td>@Html.DisplayFor(modelItem => item.Nombre)</td>
                <td>@Html.DisplayFor(modelItem => item.Rendimiento)</td>
                <td>@Html.ActionLink("Editar", "Edit", new { id = item.Id })
                    @Html.ActionLink("Detalles", "Details", new { id = item.Id })
                    @Html.ActionLink("Eliminar", "Delete", new { id = item.Id })</td>
            </tr>
        }
    </tbody>
</table>

```

Fuente: Elaboración Propia

Figura N° 162: Código de la vista Crear Insumo

```
<h2>Crear</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Insumo</h4><hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Tipo, "Tipo", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Tipo", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.Tipo, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Marca, "Marca", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Marca", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.Marca, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Modelo, "Modelo", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Modelo", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.Modelo, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Rendimiento, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Rendimiento, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Rendimiento, "", new { @class = "text-danger" })
                Impresiones</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10"><input type="submit" value="Crear" class="btn btn-default" /></div>
        </div>
    </div>
    @Html.ActionLink("Regresar al listado", "Index")</div>
}
```

Fuente: Elaboración Propia

Figura N° 163: Código de la vista Actuaizar Insumo

```
<h2>Actualizar</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Insumo</h4> <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)
        <div class="form-group">
            @Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Tipo, "Tipo", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Tipo", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.Tipo, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Marca, "Marca", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Marca", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.Marca, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Modelo, "Modelo", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Modelo", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.Modelo, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Rendimiento, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Rendimiento, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Rendimiento, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10"><input type="submit" value="Guardar" class="btn btn-default" />
            </div></div>
    </div>
    @Html.ActionLink("Regresar al listado", "Index")</div>
}
```

Fuente: Elaboración Propia

Figura N° 164: Código de la vista Eliminar Insumo

```

<h2>Eliminar</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div><h4>Insumo</h4>
  <hr />
  <dl class="dl-horizontal">
    <dt>Codigo </dt>
    <dd>@Html.DisplayFor(model => model.Codigo)</dd>
    <dt>Tipo de Insumo </dt>
    <dd>@Html.DisplayFor(model => model.TipoInsumo.Nombre)</dd>
    <dt>Marca</dt>
    <dd>@Html.DisplayFor(model => model.Marca1.Nombre)</dd>
    <dt>Modelo</dt>
    <dd>@Html.DisplayFor(model => model.Modelo1.Nombre)</dd>
    <dt> Nombre</dt>
    <dd>@Html.DisplayFor(model => model.Nombre)</dd>
    <dt> Rendimiento</dt>
    <dd>@Html.DisplayFor(model => model.Rendimiento)</dd>
    <dt> Descripcion</dt>
    <dd>@Html.DisplayFor(model => model.Descripcion)</dd></dl>
    @using (Html.BeginForm()) {
      @Html.AntiForgeryToken()
      <div class="form-actions no-color">
        <input type="submit" value="Eliminar" class="btn btn-default" />
        @Html.ActionLink("Regresar al listado", "Index")
      </div>
    }
  </div>

```

Fuente: Elaboración Propia

Figura N° 165: Formulario Index Insumo

Codigo	Tipo de Insumo	Marca	Modelo	Nombre	Rendimiento	
I000001	Toner	Canon	GPR-35	Toner Canon GPR-35	14600	Editar Detalles Eliminar
I000002	Toner	Canon	GPR-39	Toner Canon GPR-39	15000	Editar Detalles Eliminar
I000003	Toner	Canon	GPR-54	Toner Canon GPR-54	17600	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 166: Formulario Crear Insumo

SISCONSER Gestión Usuarios Recursos TI Servicios

Crear

Insumo

Tipo Toner ▼

Marca HP ▼

Modelo IR2530 ▼

Nombre

Rendimiento
Impresiones

Descripción

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 167: Formulario Actualizar Insumo

SISCONSER Gestión Usuarios Recursos TI Servicios

Actualizar

Insumo

Codigo

Tipo Toner ▼

Marca Canon ▼

Modelo GPR-35 ▼

Nombre

Rendimiento

Descripción

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 168: Formulario Eliminar Insumo

SISCONSER | Gestión Usuarios | Recursos TI | Se...

Eliminar

¿Estas seguro que deseas borrarlo?

Insumo

Codigo	I000001
Tipo de Insumo	Toner
Marca	Canon
Modelo	GPR-35
Nombre	Toner Canon GPR-35
Rendimiento	14600
Descripcion	Toner para fotocopiadora Canon IR 2530

| [Regresar al listado](#)

Fuente: Elaboración Propia

Requerimiento RF12

RF12: El sistema debe permitir al Administrador, administrar y dar mantenimiento a los registros de los componentes de los equipos de oficina de la Compañía.

Diseño

Figura N° 169: Prototipo 1 Index Componentes

SisConSer | Gestión de Usuarios | Recursos | Servicios | Reportes | Cerrar Sesión

Componentes

[Crear nuevo](#)

Hardware	Software	Insumo	Componentes	Kits	Asignación de insumos	Asignación de equipos	Licenciamiento
----------	----------	--------	--------------------	------	-----------------------	-----------------------	----------------

Código	Nombre	Tipo	Marca	Modelo	Valor	Unidad medida	Tipo característica	Estado			
									Editar	Detalles	Eliminar
									Editar	Detalles	Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 170: Prototipo 2 Crear Componente

Crear
Componente

Nombre	<input type="text" value="text"/>
Tipo Componente	<input type="text" value="text goes here"/> ▼
Marca	<input type="text" value="text goes here"/> ▼
Modelo	<input type="text" value="text goes here"/> ▼
Tipo característica	<input type="text" value="text goes here"/> ▼
Unidad medida	<input type="text" value="text goes here"/> ▼
Valor	<input type="text" value="text"/>
EstadoHw	<input type="text" value="text goes here"/> ▼
Descripción	<input type="text" value="text"/>

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 171: Prototipo 3 Editar Componente

Editar
Componente

Codigo	<input type="text" value="text"/>
Nombre	<input type="text" value="text"/>
Tipo Componente	<input type="text" value="text goes here"/> ▼
Marca	<input type="text" value="text goes here"/> ▼
Modelo	<input type="text" value="text goes here"/> ▼
Tipo característica	<input type="text" value="text goes here"/> ▼
Unidad medida	<input type="text" value="text goes here"/> ▼
Valor	<input type="text" value="text"/>
EstadoHw	<input type="text" value="text goes here"/> ▼
Descripción	<input type="text" value="text"/>

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 172: Prototipo Borrar Componentes



Fuente: Elaboración Propia

Implementación

Figura N° 173: Código para la creación de la tabla Componente

```
1 CREATE TABLE [dbo].[Componente] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nombre] NVARCHAR (50) NOT NULL,  
4     [TipoComponente] INT NOT NULL,  
5     [Marca] INT NOT NULL,  
6     [Modelo] INT NOT NULL,  
7     [TipoCaracteristica] INT NOT NULL,  
8     [UnidadMedida] INT NOT NULL,  
9     [Valor] DECIMAL (6, 2) NULL,  
10    [EstadoHw] INT NOT NULL,  
11    [Descripcion] NVARCHAR (200) NULL,  
12    [Codigo] AS ('C'+right('00000'+CONVERT([varchar],[Id]),(6))),  
13    PRIMARY KEY CLUSTERED ([Id] ASC),  
14    UNIQUE NONCLUSTERED ([Nombre] ASC),  
15    CONSTRAINT [FK_Componente_TipoComponente] FOREIGN KEY ([TipoComponente]) REFERENCES [dbo].[TipoComponente] ([Id]),  
16    CONSTRAINT [FK_Componente_Marca] FOREIGN KEY ([Marca]) REFERENCES [dbo].[Marca] ([Id]),  
17    CONSTRAINT [FK_Componente_Modelo] FOREIGN KEY ([Modelo]) REFERENCES [dbo].[Modelo] ([Id]),  
18    CONSTRAINT [FK_Componente_TipoCaracteristica] FOREIGN KEY ([TipoCaracteristica]) REFERENCES [dbo].[TipoCaracteristica] ([Id]),  
19    CONSTRAINT [FK_Componente_UnidadMedida] FOREIGN KEY ([UnidadMedida]) REFERENCES [dbo].[UnidadMedida] ([Id]),  
20    CONSTRAINT [FK_Componente_EstadoHw] FOREIGN KEY ([EstadoHw]) REFERENCES [dbo].[EstadoHw] ([Id])  
21 );
```

Fuente: Elaboración Propia

Figura N° 174: Código del modelo Componente

```
public partial class Componente  
{ [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]  
    0 referencias | 0 excepciones  
    public Componente() { }  
    4 referencias | 0 excepciones  
    public int Id { get; set; }  
    5 referencias | 0 excepciones  
    public string Nombre { get; set; }  
    5 referencias | 0 excepciones  
    public int TipoComponente { get; set; }  
    5 referencias | 0 excepciones  
    public int Marca { get; set; }  
    5 referencias | 0 excepciones  
    public int Modelo { get; set; }  
    5 referencias | 0 excepciones  
    public int TipoCaracteristica { get; set; }  
    5 referencias | 0 excepciones  
    public int UnidadMedida { get; set; }  
    5 referencias | 0 excepciones  
    public Nullable<decimal> Valor { get; set; }  
    5 referencias | 0 excepciones  
    public int EstadoHw { get; set; }  
    5 referencias | 0 excepciones  
    public string Descripcion { get; set; }  
    5 referencias | 0 excepciones  
    public string Codigo { get; set; }  
}
```

Fuente: Elaboración Propia

Figura N° 175: Código del controlador Componente

```

namespace WebContServ.Controllers
{
    public class ComponentesController : Controller
    {
        private WebConSerSisEntities db = new WebConSerSisEntities();

        public ActionResult Index()
        {
            var componente = db.Componente.Include(c => c.EstadoHw).Include(c => c.Marca).Include(c => c.Modelo).Include(c => c.TipoCaracteristica).Include(c => c.TipoComponente).Include(c => c.UnidadMedida).ToList();
            return View(componente);
        }

        public ActionResult Details(int? id)
        {
            var componente = db.Componente.Find(id);
            return View(componente);
        }

        public ActionResult Create()
        {
            ViewBag.EstadoHw = new SelectList(db.EstadoHw, "Id", "Nombre");
            ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre");
            ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre");
            ViewBag.TipoCaracteristica = new SelectList(db.TipoCaracteristica, "Id", "Nombre");
            ViewBag.TipoComponente = new SelectList(db.TipoComponente, "Id", "Nombre");
            ViewBag.UnidadMedida = new SelectList(db.UnidadMedida, "Id", "Nombre");
            return View();
        }

        [HttpPost] [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include = "Id,Nombre,TipoComponente,Marca,Modelo,TipoCaracteristica,UnidadMedida,Valor,EstadoHw,Descripcion,Codigo")] Componente componente)
        {
            if (ModelState.IsValid)
            {
                db.Componente.Add(componente);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.EstadoHw = new SelectList(db.EstadoHw, "Id", "Nombre", componente.EstadoHw);
            ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre", componente.Marca);
            ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre", componente.Modelo);
            ViewBag.TipoCaracteristica = new SelectList(db.TipoCaracteristica, "Id", "Nombre", componente.TipoCaracteristica);
            ViewBag.TipoComponente = new SelectList(db.TipoComponente, "Id", "Nombre", componente.TipoComponente);
            ViewBag.UnidadMedida = new SelectList(db.UnidadMedida, "Id", "Nombre", componente.UnidadMedida);
            return View(componente);
        }

        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Componente componente = db.Componente.Find(id);
            if (componente == null)
            {
                return HttpNotFound();
            }
            ViewBag.EstadoHw = new SelectList(db.EstadoHw, "Id", "Nombre", componente.EstadoHw);
            ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre", componente.Marca);
            ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre", componente.Modelo);
            ViewBag.TipoCaracteristica = new SelectList(db.TipoCaracteristica, "Id", "Nombre", componente.TipoCaracteristica);
            ViewBag.TipoComponente = new SelectList(db.TipoComponente, "Id", "Nombre", componente.TipoComponente);
            ViewBag.UnidadMedida = new SelectList(db.UnidadMedida, "Id", "Nombre", componente.UnidadMedida);
            return View(componente);
        }

        [HttpPost] [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include = "Id,Nombre,TipoComponente,Marca,Modelo,TipoCaracteristica,UnidadMedida,Valor,EstadoHw,Descripcion,Codigo")] Componente componente)
        {
            if (ModelState.IsValid)
            {
                db.Entry(componente).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.EstadoHw = new SelectList(db.EstadoHw, "Id", "Nombre", componente.EstadoHw);
            ViewBag.Marca = new SelectList(db.Marca, "Id", "Nombre", componente.Marca);
            ViewBag.Modelo = new SelectList(db.Modelo, "Id", "Nombre", componente.Modelo);
            ViewBag.TipoCaracteristica = new SelectList(db.TipoCaracteristica, "Id", "Nombre", componente.TipoCaracteristica);
            ViewBag.TipoComponente = new SelectList(db.TipoComponente, "Id", "Nombre", componente.TipoComponente);
            ViewBag.UnidadMedida = new SelectList(db.UnidadMedida, "Id", "Nombre", componente.UnidadMedida);
            return View(componente);
        }

        public ActionResult Delete(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Componente componente = db.Componente.Find(id);
            if (componente == null)
            {
                return HttpNotFound();
            }
            db.Componente.Remove(componente);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
    }
}

```

Fuente: Elaboración Propia

Figura N° 176: Código de la vista Index Componente

```

<h2>Index</h2>
<p> @Html.ActionLink("Crear Nuevo", "Create")</p>
<table class="table">
  <tr> <th>Codigo</th>
    <th>Nombre</th>
    <th>Tipo Componente </th>
    <th>Marca</th>
    <th>Modelo</th>
    <th>Valor</th>
    <th>Unidad Medida</th>
    <th>Tipo Caracteristica</th>
    <th>EstadoHw</th></tr>
  @foreach (var item in Model) {
  <tr> <td>@Html.DisplayFor(modelItem => item.Codigo)</td>
    <td>@Html.DisplayFor(modelItem => item.Nombre) </td>
    <td>@Html.DisplayFor(modelItem => item.TipoComponente1.Nombre) </td>
    <td>@Html.DisplayFor(modelItem => item.Marca1.Nombre)</td>
    <td>@Html.DisplayFor(modelItem => item.Modelo1.Nombre)</td>
    <td>@Html.DisplayFor(modelItem => item.Valor)</td>
    <td>@Html.DisplayFor(modelItem => item.UnidadMedida1.Nombre)</td>
    <td>@Html.DisplayFor(modelItem => item.TipoCaracteristica1.Nombre)</td>
    <td>@Html.DisplayFor(modelItem => item.EstadoHw1.Nombre)</td>
    <td>
      @Html.ActionLink("Editar", "Edit", new { id = item.Id })
      @Html.ActionLink("Detalles", "Details", new { id = item.Id })
      @Html.ActionLink("Eliminar", "Delete", new { id = item.Id }) </td></tr>
  }
</table>

```

Fuente: Elaboración Propia

Figura N° 177: Código de la vista Crear Componente

```

<h2>Create</h2>
@using (Html.BeginForm())
@Html.AntiForgeryToken()
<div class="form-horizontal">
  <h4>Componente</h4><br />
  @Html.ValidationSummary(true, "", new { @class = "text-danger" })
  <div class="form-group">
    @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.TipoComponente, "TipoComponente", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("TipoComponente", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.TipoComponente, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.Marca, "Marca", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("Marca", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.Marca, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.Modelo, "Modelo", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("Modelo", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.Modelo, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.TipoCaracteristica, "TipoCaracteristica", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("TipoCaracteristica", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.TipoCaracteristica, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.UnidadMedida, "UnidadMedida", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("UnidadMedida", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.UnidadMedida, "", new { @class = "text-danger" }) </div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.Valor, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Valor, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Valor, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.EstadoHw, "EstadoHw", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("EstadoHw", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.EstadoHw, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    <div class="col-md-offset-2 col-md-10"><input type="submit" value="Create" class="btn btn-default" /> </div></div>
  <div>@Html.ActionLink("Back to List", "Index")</div>

```

Fuente: Elaboración Propia

Figura N° 178: Código de la vista Actualizar Componente

```
<h4>Componente</h4><hr />
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
@Html.HiddenFor(model => model.Id)
<div class="form-group">
  @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
  @Html.LabelFor(model => model.TipoComponente, "TipoComponente", htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.DropDownList("TipoComponente", null, htmlAttributes: new { @class = "form-control" })
    @Html.ValidationMessageFor(model => model.TipoComponente, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
  @Html.LabelFor(model => model.Marca, "Marca", htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.DropDownList("Marca", null, htmlAttributes: new { @class = "form-control" })
    @Html.ValidationMessageFor(model => model.Marca, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
  @Html.LabelFor(model => model.Modelo, "Modelo", htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.DropDownList("Modelo", null, htmlAttributes: new { @class = "form-control" })
    @Html.ValidationMessageFor(model => model.Modelo, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
  @Html.LabelFor(model => model.TipoCaracteristica, "TipoCaracteristica", htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.DropDownList("TipoCaracteristica", null, htmlAttributes: new { @class = "form-control" })
    @Html.ValidationMessageFor(model => model.TipoCaracteristica, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
  @Html.LabelFor(model => model.UnidadMedida, "UnidadMedida", htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.DropDownList("UnidadMedida", null, htmlAttributes: new { @class = "form-control" })
    @Html.ValidationMessageFor(model => model.UnidadMedida, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
  @Html.LabelFor(model => model.Valor, htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.Valor, new { htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.Valor, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
  @Html.LabelFor(model => model.EstadoHw, "EstadoHw", htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.DropDownList("EstadoHw", null, htmlAttributes: new { @class = "form-control" })
    @Html.ValidationMessageFor(model => model.EstadoHw, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
  @Html.LabelFor(model => model.Descripcion, htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.Descripcion, new { htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.Descripcion, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
  @Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
  <div class="col-md-offset-2 col-md-10"><input type="submit" value="Guardar" class="btn btn-default" /></div></div></div>
<div>@Html.ActionLink("Regresar al listado", "Index")</div>
```

Fuente: Elaboración Propia

Figura N° 179: Código de la vista Eliminar Componente

```
<h2>Eliminar</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div><h4>Componente</h4><hr />
<dl class="dl-horizontal">
  <dt> @Html.DisplayNameFor(model => model.Nombre)</dt>
  <dd> @Html.DisplayFor(model => model.Nombre)</dd>
  <dt> @Html.DisplayNameFor(model => model.Valor)</dt>
  <dd> @Html.DisplayFor(model => model.Valor)</dd>
  <dt> @Html.DisplayNameFor(model => model.Descripcion)</dt>
  <dd> @Html.DisplayFor(model => model.Descripcion)</dd>
  <dt> @Html.DisplayNameFor(model => model.Codigo)</dt>
  <dd> @Html.DisplayFor(model => model.Codigo)</dd>
  <dt> @Html.DisplayNameFor(model => model.EstadoHw1.Nombre)</dt>
  <dd> @Html.DisplayFor(model => model.EstadoHw1.Nombre)</dd>
  <dt> @Html.DisplayNameFor(model => model.Marca1.Nombre)</dt>
  <dd> @Html.DisplayFor(model => model.Marca1.Nombre)</dd>
  <dt> @Html.DisplayNameFor(model => model.Modelo1.Nombre) </dt>
  <dd> @Html.DisplayFor(model => model.Modelo1.Nombre)</dd>
  <dt> @Html.DisplayNameFor(model => model.TipoCaracteristica1.Nombre)</dt>
  <dd> @Html.DisplayFor(model => model.TipoCaracteristica1.Nombre)</dd>
  <dt> @Html.DisplayNameFor(model => model.TipoComponente1.Nombre)</dt>
  <dd> @Html.DisplayFor(model => model.TipoComponente1.Nombre)</dd>
  <dt> @Html.DisplayNameFor(model => model.UnidadMedial.Nombre)</dt>
  <dd> @Html.DisplayFor(model => model.UnidadMedial.Nombre) </dd></dl>
  @using (Html.BeginForm()) { @Html.AntiForgeryToken()
  <div class="form-actions no-color"><input type="submit" value="Eliminar" class="btn btn-default" />
  @Html.ActionLink("Regresar al listado", "Index")</div></div>
```

Fuente: Elaboración Propia

Figura N° 180: Formulario Index Componente

Codigo	Nombre	Tipo Componente	Marca	Modelo	Valor	Unidad Medida	Tipo Caracteristica	EstadoHw
C000001	Memoria RAM Kington DDR3 3.2	Memoria RAM	Kington	DDR3	3.20	Gigabyte	Capacidad	Operativo

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 181: Formulario Crear Componente

Nombre

TipoComponente

Marca

Modelo

TipoCaracteristica

UnidadMedida

Valor

EstadoHw

Descripcion

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 182: Formulario Editar Componente

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Editar

Componente

Nombre	<input type="text" value="Memoria RAM Kington DDR3 3.2"/>
TipoComponente	<input type="text" value="Memoria RAM"/>
Marca	<input type="text" value="Kington"/>
Modelo	<input type="text" value="DDR3"/>
TipoCaracteristica	<input type="text" value="Capacidad"/>
UnidadMedida	<input type="text" value="Gigabyte"/>
Valor	<input type="text" value="3,20"/>
EstadoHw	<input type="text" value="Operativo"/>
Descripcion	<input type="text" value="mEMORIA DDR3 PARA ENSAMBLAJE"/>
Codigo	<input type="text" value="C000001"/>

[Regresar al Listado](#)

Fuente: Elaboración Propia

Figura N° 183: Formulario Eliminar Componente

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Eliminar

¿Estas seguro que deseas borrarlo?

Componente

Nombre	Memoria RAM Kington DDR3 3.2
Valor	3,20
Descripcion	mEMORIA DDR3 PARA ENSAMBLAJE
Codigo	C000001
Nombre	Operativo
Nombre	Kington
Nombre	DDR3
Nombre	Capacidad
Nombre	Memoria RAM
Nombre	Gigabyte

[Regresar al listado](#)

Fuente: Elaboración Propia

Tabla N° 13: TaskBoard del Sprint 2 Semana 5

TaskBoard S5				
Sistema web para el control de os servicios de área de sistemas en la compañía Naviera Natalia SAC			Fecha Inicio: 08/08/2018 Fecha final: 14/08/2018	
	Historias por cada Sprint	Pendiente	En curso	Hecho
SPRINT N° 2	CRUD para los activos hardware de la Compañía.			X
	CRUD para los activos software de la Compañía.			X
	CRUD para los insumos de los equipos de impresión.			X
	CRUD para los componentes de las computadoras.			X
	Registrar de los componentes que contiene un equipo de cómputo.			X
	Registrar la asignación de los insumos a los equipos de impresión.			X
	Registrar los equipos informáticos asignados al personal administrativo de la Compañía.		X	
	Registrar los licenciamientos de los equipos informáticos.	X		

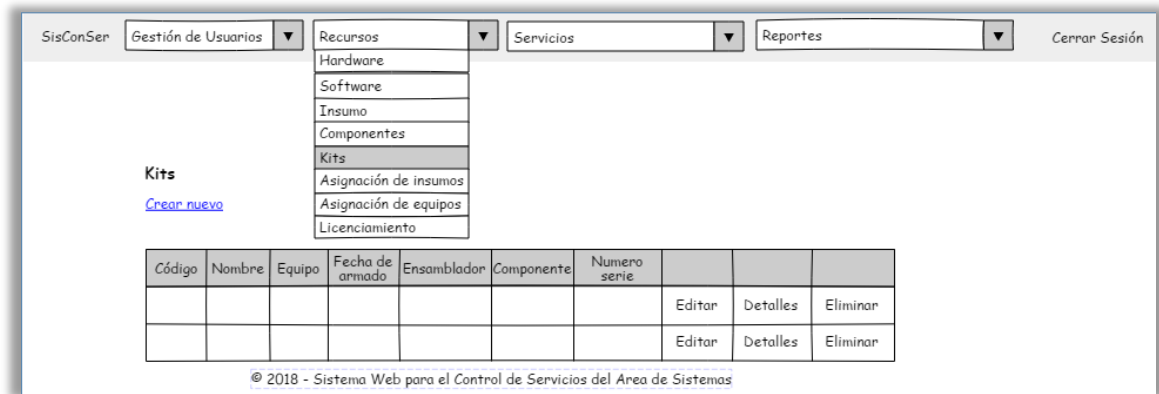
Fuente: Elaboración Propia

Requerimiento RF13

RF13: El sistema debe permitir al usuario autenticado registrar y actualizar los componentes que incluyen en ensamblaje de equipos de cómputos.

Diseño

Figura N° 184: Prototipo 1 Index Kits



Fuente: Elaboración Propia

Figura N° 185: Prototipo 2 Crear Kits

Crear Kits

Nombre	<input type="text"/>
Hardware	<input type="text"/>
FechaDeArmado	<input type="text"/>
Armador	<input type="text"/>
Componente	<input type="text"/>
Numero Serie	<input type="text"/>
Observaciones	<input type="text"/>

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 186: Prototipo 3 Actualizar Kits

Actualizar Kits

Nombre	<input type="text"/>
Hardware	<input type="text"/>
FechaDeArmado	<input type="text"/>
Armador	<input type="text"/>
Componente	<input type="text"/>
Numero Serie	<input type="text"/>
Observaciones	<input type="text"/>

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 187: Prototipo 4 Eliminar Kits

Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 188. Código para la creación de la tabla Kits

```
1 CREATE TABLE [dbo].[Kit] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nombre] NVARCHAR (50) NOT NULL,  
4     [HardwareTI] INT NOT NULL,  
5     [Componente] INT NOT NULL,  
6     [NumeroSerie] NVARCHAR (50) NULL,  
7     [Armador] INT NOT NULL,  
8     [FechaArmado] DATE NULL,  
9     [Observaciones] NVARCHAR (200) NULL,  
10    [Codigo] AS (( 'H'+CONVERT([varchar],[HardwareTI]))+'C'+CONVERT([varchar],[Componente]))),  
11    PRIMARY KEY CLUSTERED ([Id] ASC),  
12    UNIQUE NONCLUSTERED ([Nombre] ASC),  
13    CONSTRAINT [FK_Kit_ProductoTI] FOREIGN KEY ([HardwareTI]) REFERENCES [dbo].[HardwareTI] ([Id]),  
14    CONSTRAINT [FK_Kit_Componente] FOREIGN KEY ([Componente]) REFERENCES [dbo].[Componente] ([Id]),  
15    CONSTRAINT [FK_Kit_Usuario] FOREIGN KEY ([Armador]) REFERENCES [dbo].[Usuario] ([Id])  
16 );
```

Fuente: Elaboración Propia

Figura N° 189: Código del modelo Kits

```
public partial class Kit  
    0 referencias | 0 excepciones  
{ public int Id { get; set; }  
    0 referencias | 0 excepciones  
    public string Nombre { get; set; }  
    3 referencias | 0 excepciones  
    public int HardwareTI { get; set; }  
    3 referencias | 0 excepciones  
    public int Componente { get; set; }  
    0 referencias | 0 excepciones  
    public string NumeroSerie { get; set; }  
    3 referencias | 0 excepciones  
    public int Armador { get; set; }  
    0 referencias | 0 excepciones  
    public Nullable<System.DateTime> FechaArmado { get; set; }  
    0 referencias | 0 excepciones  
    public string Observaciones { get; set; }  
    0 referencias | 0 excepciones  
    public string Codigo { get; set; }
```

Fuente: Elaboración Propia

Figura N° 190: Código del controlador Kits

```

namespace WebContServ.Controllers
{
    [Authorize]
    public class KitsController : Controller
    {
        private WebConSerSisEntities db = new WebConSerSisEntities();

        public ActionResult Index()
        {
            var kit = db.Kit.Include(k => k.Componente1).Include(k => k.HardwareTI1).Include(k => k.Usuario);
            return View(kit.ToList());
        }

        public ActionResult Details(int? id)
        {
            public ActionResult Create()
            {
                ViewBag.Componente = new SelectList(db.Componente, "Id", "Nombre");
                ViewBag.HardwareTI = new SelectList(db.HardwareTI, "Id", "Nombre");
                ViewBag.Armador = new SelectList(db.Usuario, "Id", "Full_Nombre");
                return View();
            }

            [ValidateAntiForgeryToken]
            public ActionResult Create([Bind(Include = "Id,Nombre,HardwareTI,Componente,NumeroSerie,Armador,FechaArmado,Observaciones,Codigo")] Kit kit)
            {
                if (ModelState.IsValid) { db.Kit.Add(kit);
                    db.SaveChanges();
                    return RedirectToAction("Index");
                }
                ViewBag.Componente = new SelectList(db.Componente, "Id", "Nombre", kit.Componente);
                ViewBag.HardwareTI = new SelectList(db.HardwareTI, "Id", "Nombre", kit.HardwareTI);
                ViewBag.Armador = new SelectList(db.Usuario, "Id", "Full_Nombre", kit.Armador);
                return View(kit);
            }

            public ActionResult Edit(int? id)
            {
                if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }
                Kit kit = db.Kit.Find(id);
                if (kit == null) { return HttpNotFound(); }
                ViewBag.Componente = new SelectList(db.Componente, "Id", "Nombre", kit.Componente);
                ViewBag.HardwareTI = new SelectList(db.HardwareTI, "Id", "Nombre", kit.HardwareTI);
                ViewBag.Armador = new SelectList(db.Usuario, "Id", "Full_Nombre", kit.Armador);
                return View(kit);
            }

            [HttpPost]
            [ValidateAntiForgeryToken]
            public ActionResult Edit([Bind(Include = "Id,Nombre,HardwareTI,Componente,NumeroSerie,Armador,FechaArmado,Observaciones,Codigo")] Kit kit)
            {
                if (ModelState.IsValid)
                {
                    db.Entry(kit).State = EntityState.Modified;
                    db.SaveChanges();
                    return RedirectToAction("Index");
                }
                ViewBag.Componente = new SelectList(db.Componente, "Id", "Nombre", kit.Componente);
                ViewBag.HardwareTI = new SelectList(db.HardwareTI, "Id", "Nombre", kit.HardwareTI);
                ViewBag.Armador = new SelectList(db.Usuario, "Id", "Full_Nombre", kit.Armador);
                return View(kit);
            }

            public ActionResult Delete(int? id)
            {
                if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }
                Kit kit = db.Kit.Find(id);
                if (kit == null) { return HttpNotFound(); }
                return View(kit);
            }

            [HttpPost, ActionName("Delete")]
            [ValidateAntiForgeryToken]
            public ActionResult DeleteConfirmed(int id)
            {
                Kit kit = db.Kit.Find(id);
                db.Kit.Remove(kit);
            }
        }
    }
}

```

Fuente: Elaboración Propia

Figura N° 191: Código de la vista Index Kits

```

<h2>Kits</h2>
<p> @Html.ActionLink("Create New", "Create")</p>
<table class="table">
    <tr>
        <th>Codigo</th>
        <th> Nombre</th>
        <th> Equipo</th>
        <th>Fecha Ensamblaje</th>
        <th> Ensamblador </th>
        <th> Componente</th>
        <th>Numero de Serie</th></tr>
    @foreach (var item in Model) {
        <tr><td> @Html.DisplayFor(modelItem => item.Codigo)</td>
        <td>@Html.DisplayFor(modelItem => item.Nombre)</td>
        <td>@Html.DisplayFor(modelItem => item.HardwareTI1.Nombre)</td>
        <td> @Html.DisplayFor(modelItem => item.FechaArmado)</td>
        <td> @Html.DisplayFor(modelItem => item.Usuario.Full_Nombre) </td>
        <td>@Html.DisplayFor(modelItem => item.Componente1.Nombre)</td>
        <td>@Html.DisplayFor(modelItem => item.NumeroSerie)</td>
        <td> @Html.ActionLink("Editar", "Edit", new { id = item.Id })
        @Html.ActionLink("Detalles", "Details", new { id = item.Id })
        @Html.ActionLink("Eliminar", "Delete", new { id = item.Id })</td></tr>
    }
</table>

```

Fuente: Elaboración Propia

Figura N° 192: Código de la vista Crear Kits

```

<h2>Crear</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Kit</h4> <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            Fecha de Ensamblado
            <div class="col-md-10">
                @Html.EditorFor(model => model.FechaArmado, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.FechaArmado, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Armador, "Armador", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Armador", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Armador, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.HardwareTI, "HardwareTI", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("HardwareTI", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.HardwareTI, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Componente, "Componente", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Componente", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Componente, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.NumeroSerie, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.NumeroSerie, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.NumeroSerie, "", new { @class = "text-danger" })
                </div>
            </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Observaciones, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Observaciones, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Observaciones, "", new { @class = "text-danger" })
                </div>
            </div>
            <div class="form-group">
                <div class="col-md-offset-2 col-md-10">
                    <input type="submit" value="Crear" class="btn btn-default" />
                </div>
            </div>
        </div>
    }
}

```

Fuente: Elaboración Propia

Figura N° 193: Código de la vista Editar Kits

```
<h2>Actualizar</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Kit</h4> <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)
        <div class="form-group">
            @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.HardwareTI, "Equipo", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("HardwareTI", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.HardwareTI, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Componente, "Componente", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Componente", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Componente, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.NumeroSerie, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.NumeroSerie, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.NumeroSerie, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.FechaArmado, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.FechaArmado, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.FechaArmado, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Armador, "Armador", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Armador", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Armador, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Observaciones, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Observaciones, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Observaciones, "", new { @class = "text-danger" })
                </div>
            </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" })
                </div>
            </div>
            <div class="form-group">
                <div class="col-md-offset-2 col-md-10">
                    <input type="submit" value="Guardar" class="btn btn-default" />
                </div>
            </div>
        </div>
    </div>
}
```

Fuente: Elaboración Propia

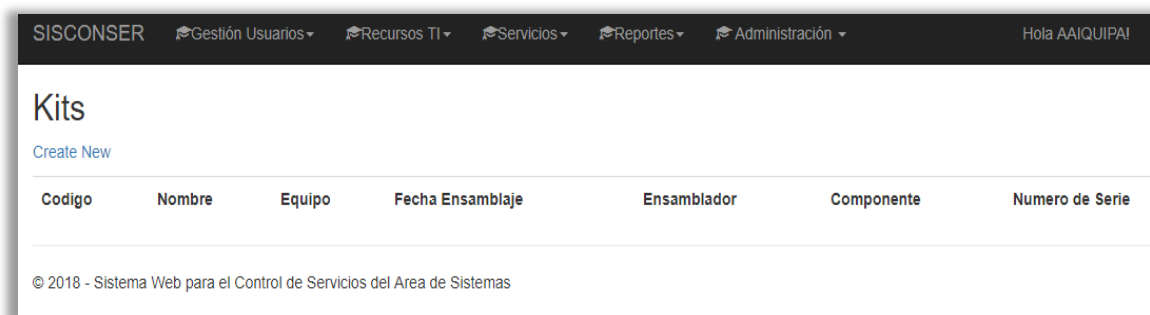
Figura N° 194: Código de la vista Borrar Kits

```
<h2>Delete</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div>
  <h4>Kit</h4><hr />
  <dl class="dl-horizontal">
    <dt>Codigo </dt>
    <dd>@Html.DisplayFor(model => model.Codigo) </dd>
    <dt>Nombre</dt>
    <dd>@Html.DisplayFor(model => model.Nombre)</dd>
    <dt>Fecha de Ensamblado</dt>
    <dd>@Html.DisplayFor(model => model.FechaArmado) </dd>
    <dt>Ensamblador</dt>
    <dd>@Html.DisplayFor(model => model.Usuario.Full_Nombre)</dd>
    <dt>Equipo</dt>
    <dd>@Html.DisplayFor(model => model.HardwareTI1.Nombre)</dd>
    <dt>Componente</dt>
    <dd>@Html.DisplayFor(model => model.Componente1.Nombre)</dd>
    <dt>Numero de Serie</dt>
    <dd>@Html.DisplayFor(model => model.NumeroSerie) </dd>
    <dt>Observaciones</dt>
    <dd>@Html.DisplayFor(model => model.Observaciones) </dd>
  </dl>
  @using (Html.BeginForm()) {
    @Html.AntiForgeryToken()
    <div class="form-actions no-color">
      <input type="submit" value="Eliminar" class="btn btn-default" /> |
      @Html.ActionLink("Regresar al listado", "Index")
    </div>
  }
```

Fuente: Elaboración Propia

GUI

Figura N° 195: Formulario Index Kits



Fuente: Elaboración Propia

Figura N° 196: Formulario Crear Kits

SISCONSER Gestión Usuarios Recursos TI Servicios

Crear Kit

Nombre

HardwareTI

FechaArmado

Armador

Componente

NumeroSerie

Observaciones

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 197: Formulario Editar Kits

SISCONSER Gestión Usuarios Recursos TI Servicios

Editar Kit

Codigo

Nombre

HardwareTI

FechaArmado

Armador

Componente

NumeroSerie

Observaciones

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 198: Formulario Eliminar Kits

Eliminar

¿Estas seguro que deseas borrarlo?
Kit

Nombre PCNAV03
NumeroSerie HUYT65436
FechaArmado 11/09/2018 0:00:00
Observaciones
Codigo H2C1
Nombre Memoria RAM Kington DDR3 3.2
Nombre PcChip i3
DNI 10627303

| [Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Requerimiento RF14

RF14: El sistema debe permitir al usuario autenticado registrar y actualizar los cambios de insumos en los equipos de impresión.

Diseño

Figura N° 199: Prototipo 1 Index Asignar Insumos

SisConSer | Gestión de Usuarios | Recursos | Servicios | Reportes | Cerrar Sesión

Asignación Insumos

[Crear nuevo](#)

Nombre	Fecha de entrega	Contador	Código	Insumo	Equipo	Usuario			
							Editar	Detalles	Eliminar
							Editar	Detalles	Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 200: Prototipo 2 Crear Asignar Insumos

Crear Asignación de Insumos

Nombre	<input type="text"/>
RecursoAsignado	<input type="text"/>
AsignadoPor	<input type="text"/>
Insumo	<input type="text"/>
FechaEntrega	<input type="text"/>
Contador	<input type="text"/>

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 201: Prototipo 3 Asignar Actualizar Insumos

Actualizar Asignación de Insumos

Nombre	<input type="text"/>
RecursoAsignado	<input type="text"/>
AsignadoPor	<input type="text"/>
Insumo	<input type="text"/>
FechaEntrega	<input type="text"/>
Contador	<input type="text"/>

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 202: Prototipo 4 Eliminar Asignar Insumos

Borrar

¿ Estas Seguro que deseas borrarlos ?

Código : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 203: Código para la creación de la tabla Asignar Insumo

```
1 CREATE TABLE [dbo].[AsignacionInsumos] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nombre] NVARCHAR (50) NOT NULL,  
4     [RecursoAsignado] INT NOT NULL,  
5     [AsignadoPor] INT NOT NULL,  
6     [Insumo] INT NOT NULL,  
7     [FechaEntrega] NVARCHAR (50) NOT NULL,  
8     [Contador] INT NOT NULL,  
9     [Codigo] AS ((('I'+CONVERT([varchar],[Insumo]))+('UH'+CONVERT([varchar],[RecursoAsignado])))),  
10    PRIMARY KEY CLUSTERED ([Id] ASC),  
11    CONSTRAINT [FK_AsignacionInsumos_RecursoAsignado] FOREIGN KEY ([RecursoAsignado]) REFERENCES [dbo].[RecursoAsignado] ([Id]),  
12    CONSTRAINT [FK_Asignacion_Insumos_Insumo] FOREIGN KEY ([Insumo]) REFERENCES [dbo].[Insumo] ([Id]),  
13    CONSTRAINT [FK_AsignacionInsumos_Usuario] FOREIGN KEY ([AsignadoPor]) REFERENCES [dbo].[Usuario] ([Id])
```

Controlador Asignar Insumos

Figura N° 204: Código del controlador Asignar Insumo

```
using System.Data.Entity; using System.Linq; using System.Net; using System.Web.Mvc; using SistemaControlTI.Models;  
namespace SistemaControlTI.Controllers  
{  
    [Authorize]  
    public class AsignacionInsumosController : Controller  
    {  
        private WebConSerSisEntities db = new WebConSerSisEntities();  
        // GET: AsignacionInsumos  
        public ActionResult Index()  
        {  
            var asignacionInsumos = db.AsignacionInsumos.Include(a => a.Insumo1).Include(a => a.RecursoAsignado1).Include(a => a.Usuario);  
            return View(asignacionInsumos.ToList());  
        }  
        // GET: AsignacionInsumos/Details/5  
        public ActionResult Details(int? id)  
        {  
            if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }  
            AsignacionInsumos asignacionInsumos = db.AsignacionInsumos.Find(id);  
            if (asignacionInsumos == null) { return HttpNotFound(); }  
            return View(asignacionInsumos);  
        }  
        // GET: AsignacionInsumos/Create  
        public ActionResult Create()  
        {  
            ViewBag.Insumo = new SelectList(db.Insumo, "Id", "Nombre");  
            ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre");  
            ViewBag.AsignadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre");  
            return View();  
        }  
        // POST: AsignacionInsumos/Create  
        [HttpPost][ValidateAntiForgeryToken]  
        public ActionResult Create([Bind(Include = "Id,Nombre,RecursoAsignado,AsignadoPor,Insumo,FechaEntrega,Contador,Codigo")] AsignacionInsumos asignacionInsumos)  
        {  
            if (ModelState.IsValid) { db.AsignacionInsumos.Add(asignacionInsumos); db.SaveChanges(); return RedirectToAction("Index"); }  
            ViewBag.Insumo = new SelectList(db.Insumo, "Id", "Nombre", asignacionInsumos.Insumo);  
            ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", asignacionInsumos.RecursoAsignado);  
            ViewBag.AsignadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", asignacionInsumos.AsignadoPor);  
            return View(asignacionInsumos);  
        }  
        // GET: AsignacionInsumos/Edit/5  
        public ActionResult Edit(int? id)  
        {  
            if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }  
            AsignacionInsumos asignacionInsumos = db.AsignacionInsumos.Find(id);  
            if (asignacionInsumos == null) { return HttpNotFound(); }  
            ViewBag.Insumo = new SelectList(db.Insumo, "Id", "Nombre", asignacionInsumos.Insumo);  
            ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", asignacionInsumos.RecursoAsignado);  
            ViewBag.AsignadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", asignacionInsumos.AsignadoPor);  
            return View(asignacionInsumos);  
        }  
        // POST: AsignacionInsumos/Edit/5  
        [HttpPost][ValidateAntiForgeryToken]  
        public ActionResult Edit([Bind(Include = "Id,Nombre,RecursoAsignado,AsignadoPor,Insumo,FechaEntrega,Contador,Codigo")] AsignacionInsumos asignacionInsumos)  
        {  
            if (ModelState.IsValid) { db.Entry(asignacionInsumos).State = EntityState.Modified; db.SaveChanges(); return RedirectToAction("Index"); }  
            ViewBag.Insumo = new SelectList(db.Insumo, "Id", "Nombre", asignacionInsumos.Insumo);  
            ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", asignacionInsumos.RecursoAsignado);  
            ViewBag.AsignadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", asignacionInsumos.AsignadoPor);  
            return View(asignacionInsumos);  
        }  
        // GET: AsignacionInsumos/Delete/5  
        public ActionResult Delete(int? id) { ... }  
        // POST: AsignacionInsumos/Delete/5  
        [HttpPost, ActionName("Delete")] [ValidateAntiForgeryToken]  
        public ActionResult DeleteConfirmed(int id)  
        {  
            AsignacionInsumos asignacionInsumos = db.AsignacionInsumos.Find(id);  
            db.AsignacionInsumos.Remove(asignacionInsumos); db.SaveChanges(); return RedirectToAction("Index");  
        }  
    }  
}
```

Fuente: Elaboración Propia

Figura N° 205: Código de la vista de Asignación de insumos

```

@model IEnumerable<SistemaControlTI.Models.AsignacionInsumos>
@{ ViewBag.Title = "Index"; Layout = "~/Views/Shared/_Layout.cshtml"; }
<h2>Asignación de Insumos</h2>
<p>@Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
    <tr><th> Nombre</th> <th> Fecha de Entrega </th>
        <th> Contador</th> <th> Codigo</th>
        <th> Insumo</th> <th> Equipo </th>
        <th> Usuario</th><th></th></tr>
    @foreach (var item in Model) {
        <tr> <td> @Html.DisplayFor(modelItem => item.Nombre)</td>
            <td> @Html.DisplayFor(modelItem => item.FechaEntrega)</td>
            <td> @Html.DisplayFor(modelItem => item.Contador)</td>
            <td>@Html.DisplayFor(modelItem => item.Codigo)</td>
            <td>@Html.DisplayFor(modelItem => item.Insumo1.Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.RecursoAsignado1.Nombre) </td>
            <td> @Html.DisplayFor(modelItem => item.Usuario.Full_Nombre)</td>
            <td>@Html.ActionLink("Editar", "Edit", new { id=item.Id }) |
                @Html.ActionLink("Detalles", "Details", new { id=item.Id }) |
                @Html.ActionLink("Eliminar", "Delete", new { id=item.Id })
            </td> </tr>
    }</table>

```

Fuente: Elaboración Propia

Figura N° 206: Código para la vista Crear Asignación de insumos

```

@model SistemaControlTI.Models.AsignacionInsumos
@{ ViewBag.Title = "Create"; }
<h2>Crear</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Asignacion de Insumos</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.RecursoAsignado, "RecursoAsignado", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("RecursoAsignado", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.RecursoAsignado, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.AsignadoPor, "AsignadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("AsignadoPor", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.AsignadoPor, "", new { @class = "text-danger" })</div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Insumo, "Insumo", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Insumo", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Insumo, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.FechaEntrega, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.FechaEntrega, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.FechaEntrega, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Contador, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Contador, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Contador, "", new { @class = "text-danger" }) </div></div>
            <div class="form-group">
                <div class="col-md-offset-2 col-md-10">
                    <input type="submit" value="Crear" class="btn btn-default" /></div> </div></div>
    <div>@Html.ActionLink("Regresar al listado", "Index")</div>
}

```

Fuente: Elaboración Propia

Figura N° 207: Código para la vista Actualizar Asignar Insumos

```

@model SistemaControlTI.Models.AsignacionInsumos
@{ ViewBag.Title = "Edit"; }
<h2>Actualizar</h2>
@using (Html.BeginForm())
{ @Html.AntiForgeryToken()
<div class="form-horizontal">
<h4>Asignacion de Insumos</h4><hr />
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
@Html.HiddenFor(model => model.Id)
<div class="form-group">
@Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.RecursoAsignado, "RecursoAsignado", htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.DropDownList("RecursoAsignado", null, htmlAttributes: new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.RecursoAsignado, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.AsignadoPor, "AsignadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.DropDownList("AsignadoPor", null, htmlAttributes: new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.AsignadoPor, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.Insumo, "Insumo", htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.DropDownList("Insumo", null, htmlAttributes: new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.Insumo, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.FechaEntrega, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.FechaEntrega, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.FechaEntrega, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.Contador, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.Contador, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.Contador, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
<div class="col-md-offset-2 col-md-10">
<input type="submit" value="Guardar" class="btn btn-default" /></div></div>
</div>
@Html.ActionLink("Regresar al listado", "Index")</div>
}

```

Fuente: Elaboración Propia

GUI

Figura N° 208: Formulario Index Asignación de Insumo

SISCONSER
Gestión Usuarios
Recursos TI
Servicios
Reportes
Administración
Hola AIQUIPAI
Cerrar sesión

Asignación de Insumos

[Nuevo Registro](#)

Nombre	Fecha de Entrega	Contador	Codigo	Insumo	Equipo	Usuario	
Instalación tóner canon	15/15/2017	15290	I2UH5	Toner Canon GPR-39	PCNAV55	Alex Aiquipa	Editar Detalles Eliminar

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Figura N° 209: Formulario Crear Asignación de Insumos

The screenshot shows the 'Crear' (Create) form for 'Asignación de Insumos' (Supply Allocation) in the SISCONSER system. The form includes the following fields:

- Nombre:** An empty text input field.
- RecursoAsignado:** A dropdown menu with 'PCNAV11' selected.
- AsignadoPor:** A dropdown menu with 'Alex Aiquipa' selected.
- Insumo:** A dropdown menu with 'Toner Canon GPR-35' selected.
- FechaEntrega:** An empty text input field.
- Contador:** An empty text input field.

At the bottom of the form, there is a 'Crear' button, a link to 'Regresar al listado' (Return to list), and a footer: '© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas'.

Fuente: Elaboración Propia

Figura N° 210: Formulario Actualizar Asignación de Insumo

The screenshot shows the 'Actualizar' (Update) form for 'Asignación de Insumos' (Supply Allocation) in the SISCONSER system. The form includes the following fields:

- Nombre:** A text input field containing 'Instalación tóner canon'.
- RecursoAsignado:** A dropdown menu with 'PCNAV55' selected.
- AsignadoPor:** A dropdown menu with 'Alex Aiquipa' selected.
- Insumo:** A dropdown menu with 'Toner Canon GPR-39' selected.
- FechaEntrega:** A text input field containing '15/15/2017'.
- Contador:** A text input field containing '15290'.
- Codigo:** A text input field containing 'I2UH5'.

At the bottom of the form, there is a 'Guardar' button, a link to 'Regresar al listado' (Return to list), and a footer: '© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas'.

Fuente: Elaboración Propia

Figura N° 211: Formulario Eliminar Asignación de Insumo

SISCONSER 👤Gestión Usuarios ▾ 🛠Recursos TI ▾ 🛠Servicios

Delete

¿Estas seguro que deseas borrarlo?

Asignacion de Insumos

Nombre Instalación tóner canon
FechaEntrega 15/15/2017
Contador 15290
Codigo I2UH5
Nombre Toner Canon GPR-39
Nombre PCNAV55
Full_Nombre Alex Aiquipa

| [Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Requerimiento RF15

RF15: El sistema debe permitir asignar y reasignar recursos T.I. al personal administrativo de la Compañía.

Diseño

Figura N° 212: Prototipo 1 Índice Asignación De Equipos

SisConSer Gestión de Usuarios ▾ Recursos ▾ Servicios ▾ Reportes ▾ Cerrar Sesión

Hardware
Software
Insumo
Componentes
Kits
Asignación de equipos
Asignación de insumos
Asignación de equipos
Licenciamiento

Asignación de equipos
[Crear nuevo](#)

Nombre	Fecha de entrega	Usuario responsable	Entregado por	Técnico	Equipo	Serie	Estado	Fecha caducidad	Nombre				
											Editar	Detalles	Eliminar
											Editar	Detalles	Eliminar

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Figura N° 213: Prototipo 2 Crear Asignación de Recursos

Crear
Asignación de Recursos

Nombre	<input type="text" value="text"/>
Fecha de entrega	<input type="text" value="text"/>
Usuario	<input type="text" value="text goes here"/> ▼
Entregado por	<input type="text" value="text goes here"/> ▼
Facilitador	<input type="text" value="text goes here"/> ▼
Hardware	<input type="text" value="text goes here"/> ▼
Nro Serie	<input type="text" value="text"/>
Estado	<input type="text" value="text goes here"/> ▼
Fecha devolución	<input type="text" value="text"/>
Observaciones	<input type="text" value="text"/>

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 214: Prototipo 3 Editar Asignación de Recursos

Editar
Asignación de Recursos

Nombre	<input type="text" value="text"/>
Fecha de entrega	<input type="text" value="text"/>
Usuario	<input type="text" value="text goes here"/> ▼
Entregado por	<input type="text" value="text goes here"/> ▼
Facilitador	<input type="text" value="text goes here"/> ▼
Hardware	<input type="text" value="text goes here"/> ▼
Nro Serie	<input type="text" value="text"/>
Estado	<input type="text" value="text goes here"/> ▼
Fecha devolución	<input type="text" value="text"/>
Observaciones	<input type="text" value="text"/>

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 215: Prototipo 4 Borrar Asignación de Recursos

Borrar
¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

Eliminar [Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 216: Código para la creación de la tabla Asignación de Recursos

```
1 CREATE TABLE [dbo].[RecursoAsignado] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nombre] NVARCHAR (50) NOT NULL,  
4     [Usuario] INT NOT NULL,  
5     [HardwareTI] INT NOT NULL,  
6     [EstadoHw] INT NOT NULL,  
7     [EntregadoPor] INT NOT NULL,  
8     [Facilitador] INT NOT NULL,  
9     [NumeroSerie] NVARCHAR (30) NULL,  
10    [FechaEntrega] DATE NOT NULL,  
11    [FechaDevolucion] DATE NULL,  
12    [Observaciones] NVARCHAR (200) NULL,  
13    [Codigo] AS ('UH'+right('000000'+CONVERT([varchar],[HardwareTI]),(6))),  
14    PRIMARY KEY CLUSTERED ([Id] ASC),  
15    CONSTRAINT [FK_AsignacionRecursos_Usuario] FOREIGN KEY ([Usuario]) REFERENCES [dbo].[Usuario] ([Id]),  
16    CONSTRAINT [FK_AsignacionRecursos_HardwareTI] FOREIGN KEY ([HardwareTI]) REFERENCES [dbo].[HardwareTI] ([Id]),  
17    CONSTRAINT [FK_AsignacionRecursos_EstadoEC] FOREIGN KEY ([EstadoHw]) REFERENCES [dbo].[EstadoHw] ([Id]),  
18    CONSTRAINT [FK_AsignacionRecursos_UsuarioE] FOREIGN KEY ([EntregadoPor]) REFERENCES [dbo].[Usuario] ([Id]),  
19    CONSTRAINT [FK_AsignacionRecursos_Usuariof] FOREIGN KEY ([Facilitador]) REFERENCES [dbo].[Usuario] ([Id])  
20 );
```

Fuente: Elaboración Propia

Figura N° 217: código del modelo Asignación de Recursos

```
public partial class RecursoAsignado  
{ [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]  
    0 referencias | 0 excepciones  
    public RecursoAsignado() { ... }  
    0 referencias | 0 excepciones  
    public int Id { get; set; }  
    0 referencias | 0 excepciones  
    public string Nombre { get; set; }  
    3 referencias | 0 excepciones  
    public int Usuario { get; set; }  
    3 referencias | 0 excepciones  
    public int HardwareTI { get; set; }  
    3 referencias | 0 excepciones  
    public int EstadoHw { get; set; }  
    3 referencias | 0 excepciones  
    public int EntregadoPor { get; set; }  
    3 referencias | 0 excepciones  
    public int Facilitador { get; set; }  
    0 referencias | 0 excepciones  
    public string NumeroSerie { get; set; }  
    0 referencias | 0 excepciones  
    public System.DateTime FechaEntrega { get; set; }  
    0 referencias | 0 excepciones  
    public Nullable<System.DateTime> FechaDevolucion { get; set; }  
    0 referencias | 0 excepciones  
    public string Observaciones { get; set; }  
    0 referencias | 0 excepciones  
    public string Codigo { get; set; }
```

Fuente: Elaboración Propia

Figura N° 218: Código del controlador Asignación de Recursos

```

public ActionResult Index()
{
    var recursoAsignado = db.RecursoAsignado.Include(r => r.EstadoHw1).Include(r => r.HardwareTI1).Include(r => r.Usuario1).Include(r => r.Usuario2).Include(r => r.Usuario3);
    return View(recursoAsignado.ToList());
}
public ActionResult Details(int? id)
{
    ViewBag.EstadoHw = new SelectList(db.EstadoHw, "Id", "Nombre");
    ViewBag.HardwareTI = new SelectList(db.HardwareTI, "Id", "Nombre");
    ViewBag.Usuario = new SelectList(db.Usuario, "Id", "Full_Nombre");
    ViewBag.EntregadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre");
    ViewBag.Facilitador = new SelectList(db.Usuario, "Id", "Full_Nombre");
    return View();
}
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Id,Nombre,Usuario,HardwareTI,EstadoHw,EntregadoPor,Facilitador,NumeroSerie,FechaEntrega,FechaDevolucion,Observaciones,Codigo")]
{
    if (ModelState.IsValid) { db.RecursoAsignado.Add(recursoAsignado);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.EstadoHw = new SelectList(db.EstadoHw, "Id", "Nombre", recursoAsignado.EstadoHw);
    ViewBag.HardwareTI = new SelectList(db.HardwareTI, "Id", "Nombre", recursoAsignado.HardwareTI);
    ViewBag.Usuario = new SelectList(db.Usuario, "Id", "Full_Nombre", recursoAsignado.Usuario);
    ViewBag.EntregadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", recursoAsignado.EntregadoPor);
    ViewBag.Facilitador = new SelectList(db.Usuario, "Id", "Full_Nombre", recursoAsignado.Facilitador);
    return View(recursoAsignado);
}
public ActionResult Edit(int? id)
{
    if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }
    RecursoAsignado recursoAsignado = db.RecursoAsignado.Find(id);
    if (recursoAsignado == null) { return HttpNotFound(); }
    ViewBag.EstadoHw = new SelectList(db.EstadoHw, "Id", "Nombre", recursoAsignado.EstadoHw);
    ViewBag.HardwareTI = new SelectList(db.HardwareTI, "Id", "Nombre", recursoAsignado.HardwareTI);
    ViewBag.Usuario = new SelectList(db.Usuario, "Id", "Full_Nombre", recursoAsignado.Usuario);
    ViewBag.EntregadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", recursoAsignado.EntregadoPor);
    ViewBag.Facilitador = new SelectList(db.Usuario, "Id", "Full_Nombre", recursoAsignado.Facilitador);
    return View(recursoAsignado);
}
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "Id,Nombre,Usuario,HardwareTI,EstadoHw,EntregadoPor,Facilitador,NumeroSerie,FechaEntrega,FechaDevolucion,Observaciones,Codigo")]
{
    if (ModelState.IsValid) { db.Entry(recursoAsignado).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.EstadoHw = new SelectList(db.EstadoHw, "Id", "Nombre", recursoAsignado.EstadoHw);
    ViewBag.HardwareTI = new SelectList(db.HardwareTI, "Id", "Nombre", recursoAsignado.HardwareTI);
    ViewBag.Usuario = new SelectList(db.Usuario, "Id", "Full_Nombre", recursoAsignado.Usuario);
    ViewBag.EntregadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", recursoAsignado.EntregadoPor);
    ViewBag.Facilitador = new SelectList(db.Usuario, "Id", "Full_Nombre", recursoAsignado.Facilitador);
    return View(recursoAsignado);
}
public ActionResult Delete(int? id)
{
    if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }
    RecursoAsignado recursoAsignado = db.RecursoAsignado.Find(id);
    if (recursoAsignado == null) { return HttpNotFound(); }
    return View(recursoAsignado);
}
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]

```

Fuente: Elaboración Propia

Figura N° 219: Código de la vista Índice Asignación de Recursos

```

<h2>Asignación de Recursos</h2>
<p>@Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
    <tr><th>Nombre </th>
        <th>Fecha de Entrega</th>
        <th>Usuario Responsable</th>
        <th>Entregado por</th>
        <th>Nombre del tecnico que asigna</th>
        <th>Equipo</th>
        <th>Numero de Serie</th>
        <th>Estado</th>
        <th>Fecha de Devolucion</th></tr>
    @foreach (var item in Model) {
        <tr><td>@Html.DisplayFor(modelItem => item.Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.FechaEntrega)</td>
            <td>@Html.DisplayFor(modelItem => item.Usuario1.Full_Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.Usuario2.Full_Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.Usuario3.Full_Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.HardwareTI1.Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.NumeroSerie)</td>
            <td>@Html.DisplayFor(modelItem => item.EstadoHw1.Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.FechaDevolucion)</td>
            <td>@Html.ActionLink("Editar", "Edit", new { id = item.Id })
                @Html.ActionLink("Detalles", "Details", new { id = item.Id })
                @Html.ActionLink("Eliminar", "Delete", new { id = item.Id })</td></tr>
    }
</table>

```

Fuente: Elaboración Propia

Figura N° 220: Código de la vista Crear Asignación de Recursos

```

<h2>Crear</h2>
@using (Html.BeginForm())
{ @Html.AntiForgeryToken()
<div class="form-horizontal">
  <h4>Asignación de Recursos</h4><hr />
  @Html.ValidationSummary(true, "", new { @class = "text-danger" })
  <div class="form-group">
    @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.FechaEntrega, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.FechaEntrega, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.FechaEntrega, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.Usuario, "Usuario", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("Usuario", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.Usuario, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.EntregadoPor, "EntregadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("EntregadoPor", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.EntregadoPor, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.Facilitador, "Facilitador", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("Facilitador", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.Facilitador, "", new { @class = "text-danger" }) </div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.HardwareTI, "HardwareTI", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("HardwareTI", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.HardwareTI, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.NumeroSerie, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.NumeroSerie, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.NumeroSerie, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.EstadoHw, "EstadoHw", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("EstadoHw", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.EstadoHw, "", new { @class = "text-danger" }) </div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.FechaDevolucion, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.FechaDevolucion, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.FechaDevolucion, "", new { @class = "text-danger" })</div></div>
  <div class="form-group">
    @Html.LabelFor(model => model.Observaciones, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Observaciones, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Observaciones, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    <div class="col-md-offset-2 col-md-10"><input type="submit" value="Crear" class="btn btn-default" /></div> </div></div>
  <div> @Html.ActionLink("Regresar al listado", "Index")</div>
}

```

Fuente: Elaboración Propia

Figura N° 221: Código de la vista Editar Asignación de Recursos

```

<h4>Asignación de Recursos</h4><hr />
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
@Html.HiddenFor(model => model.Id)
<div class="form-group">
    @Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.FechaEntrega, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.FechaEntrega, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.FechaEntrega, "", new { @class = "text-danger" })</div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.Usuario, "Usuario", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("Usuario", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Usuario, "", new { @class = "text-danger" })</div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.EntregadoPor, "EntregadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("EntregadoPor", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.EntregadoPor, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Facilitador, "Facilitador", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("Facilitador", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Facilitador, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
    @Html.LabelFor(model => model.HardwareTI, "HardwareTI", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("HardwareTI", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.HardwareTI, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
    @Html.LabelFor(model => model.NumeroSerie, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.NumeroSerie, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.NumeroSerie, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.EstadoHw, "EstadoHw", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("EstadoHw", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.EstadoHw, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.FechaDevolucion, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.FechaDevolucion, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.FechaDevolucion, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Observaciones, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Observaciones, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Observaciones, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group"><div class="col-md-offset-2 col-md-10"><input type="submit" value="Guardar" class="btn btn-default" />

```

Fuente: Elaboración Propia

Figura N° 222: Código de la vista Eliminar Asignación de Recursos

```

<h2>Delete</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div>
  <h4>Asignación de Recursos</h4>
  <hr />
  <dl class="dl-horizontal">
    <dt>Codigo</dt>
    <dd>@Html.DisplayFor(model => model.Codigo)</dd>
    <dt> Nombre</dt>
    <dd>@Html.DisplayFor(model => model.Nombre)</dd>
    <dt> FechaEntrega</dt>
    <dd>@Html.DisplayFor(model => model.FechaEntrega)</dd>
    <dt>Usuario Responsable</dt>
    <dd>@Html.DisplayFor(model => model.Usuario1.Full_Nombre) </dd>
    <dt>Entregado por</dt>
    <dd>@Html.DisplayFor(model => model.Usuario2.Full_Nombre)</dd>
    <dt>Técnico asignador</dt>
    <dd>@Html.DisplayFor(model => model.Usuario3.Full_Nombre)</dd>
    <dt> Equipo</dt>
    <dd>@Html.DisplayFor(model => model.HardwareTI1.Nombre)</dd>
    <dt>Numero de Serie</dt>
    <dd> @Html.DisplayFor(model => model.NumeroSerie)</dd>
    <dt>Estado</dt>
    <dd>@Html.DisplayFor(model => model.EstadoHw1.Nombre)</dd>
    <dt> fecha de Devolucion</dt>
    <dd>@Html.DisplayFor(model => model.FechaDevolucion)</dd>
    <dt>Observaciones</dt>
    <dd> @Html.DisplayFor(model => model.Observaciones)</dd>
  </dl>
  @using (Html.BeginForm()) {
    @Html.AntiForgeryToken()
    <div class="form-actions no-color">
      <input type="submit" value="Eliminar" class="btn btn-default" />
      @Html.ActionLink("Regresar al listado", "Index")
    </div>
  }
</div>

```

Fuente: Elaboración Propia

Diseño

Figura N° 223: Formulario Index Asignación de Recursos

Nombre	Fecha de Entrega	Usuario Responsable	Entregado por	Nombre del tecnico que asigna	Equipo	Numero de Serie	Estado	Fecha de Devolucion	
PCNAV11	12/09/2018 0:00:00	Aide Minaya	Alex Aiquipa	Alex Aiquipa	hp	DFR567457K	Operativo	12/09/2018 0:00:00	Editar Detalles Eliminar
PCNAV55	12/07/2018 0:00:00	Carmen Ortega	Aide Minaya	Alex Aiquipa	hp	asasasa	Operativo	12/08/2018 0:00:00	Editar Detalles Eliminar
PCNAV06	12/07/2018 0:00:00	Yolanda Maldonado	Alex Aiquipa	Alex Aiquipa	PcChip i3	GF45678	Operativo		Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 224: Formulario Crear Asignación de Recursos

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Crear

Asignación de Recursos

Nombre	<input type="text"/>
FechaEntrega	<input type="text"/>
Usuario	<input type="text" value="Alex Aiquipa"/>
EntregadoPor	<input type="text" value="Alex Aiquipa"/>
Facilitador	<input type="text" value="Alex Aiquipa"/>
HardwareTI	<input type="text" value="hp"/>
NumeroSerie	<input type="text"/>
EstadoHw	<input type="text" value="Operativo"/>
FechaDevolucion	<input type="text"/>
Observaciones	<input type="text"/>

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 225: Formulario Actualizar Asignación de Recursos

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Actualizar

Asignación de Recursos

Codigo	<input type="text" value="UH000001"/>
Nombre	<input type="text" value="PCNAV11"/>
FechaEntrega	<input type="text" value="12/09/2018 0:00:00"/>
Usuario	<input type="text" value="Aide Minaya"/>
EntregadoPor	<input type="text" value="Alex Aiquipa"/>
Facilitador	<input type="text" value="Alex Aiquipa"/>
HardwareTI	<input type="text" value="hp"/>
NumeroSerie	<input type="text" value="DFR567457K"/>
EstadoHw	<input type="text" value="Operativo"/>
FechaDevolucion	<input type="text" value="12/09/2018 0:00:00"/>
Observaciones	<input type="text"/>

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 226: Código de la vista Eliminar Asignación de Recursos

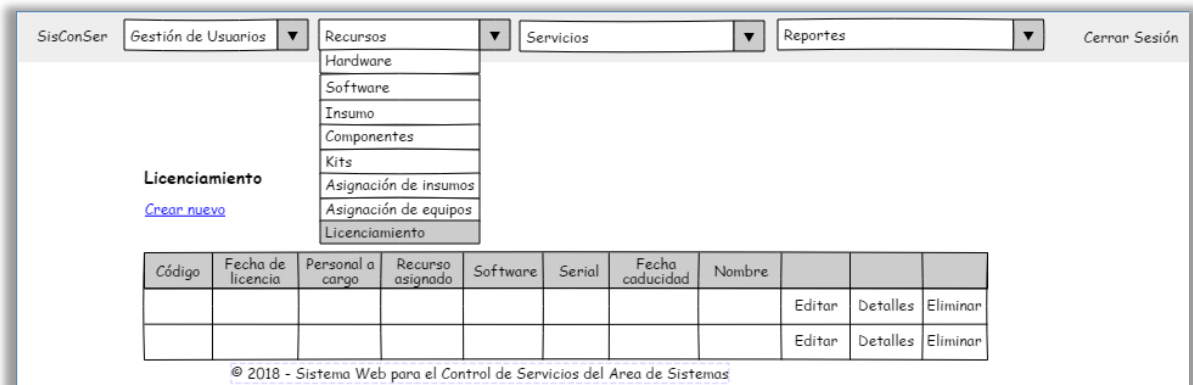


Fuente: Elaboración Propia

Requerimiento RF16

RF16: El sistema debe permitir registrar los licenciamentos de los recursos asignados al personal administrativo.

Figura N° 227: Prototipo 1 Índice Licenciamentos



Fuente: Elaboración Propia

Figura N° 228: Prototipo 2 Crear Licenciamientos

Crear
Licenciamiento

Nombre

FechaLic

Licenciado Por ▼

Recurso Asignado ▼

Software ▼

Serial

Fecha caducidad

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 229: Prototipo 3 Editar Licenciamientos

Editar
Licenciamiento

Nombre

FechaLic

Licenciado Por ▼

Recurso Asignado ▼

Software ▼

Serial

Fecha caducidad

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 230: Prototipo 4 Eliminar Licenciamientos

Borrar
¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 231: Código para la creación de la tabla Licenciamiento

```
1 CREATE TABLE [dbo].[Licenciamiento] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nombre] NVARCHAR (50) NOT NULL,  
4     [RecursoAsignado] INT NOT NULL,  
5     [SoftwareTI] INT NOT NULL,  
6     [Serial] NVARCHAR (50) NULL,  
7     [LicenciadoPor] INT NOT NULL,  
8     [FechaLic] DATE NULL,  
9     [FechaCaducidad] DATE NULL,  
10    [Codigo] AS (('S'+CONVERT([varchar],[SoftwareTI]))+'UH'+CONVERT([varchar],[RecursoAsignado]))),  
11    PRIMARY KEY CLUSTERED ([Id] ASC),  
12    UNIQUE NONCLUSTERED ([Nombre] ASC),  
13    CONSTRAINT [FK_Licenciamiento_RecursoAsignado] FOREIGN KEY ([RecursoAsignado]) REFERENCES [dbo].[RecursoAsignado] ([Id]),  
14    CONSTRAINT [FK_Licenciamiento_SoftwareTI] FOREIGN KEY ([SoftwareTI]) REFERENCES [dbo].[SoftwareTI] ([Id]),  
15    CONSTRAINT [FK_Licenciamiento_Usuario] FOREIGN KEY ([LicenciadoPor]) REFERENCES [dbo].[Usuario] ([Id])  
16 );
```

Fuente: Elaboración Propia

Figura N° 232: Código del modelo Licenciamiento

```
public partial class Licenciamiento  
{ [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]  
    0 referencias | 0 excepciones  
    public Licenciamiento()  
    { this.Eventos = new HashSet<Eventos>(); }  
    0 referencias | 0 excepciones  
    public int Id { get; set; }  
    0 referencias | 0 excepciones  
    public string Nombre { get; set; }  
    3 referencias | 0 excepciones  
    public int RecursoAsignado { get; set; }  
    3 referencias | 0 excepciones  
    public int SoftwareTI { get; set; }  
    0 referencias | 0 excepciones  
    public string Serial { get; set; }  
    3 referencias | 0 excepciones  
    public int LicenciadoPor { get; set; }  
    0 referencias | 0 excepciones  
    public Nullable<System.DateTime> FechaLic { get; set; }  
    0 referencias | 0 excepciones  
    public Nullable<System.DateTime> FechaCaducidad { get; set; }  
    0 referencias | 0 excepciones  
    public string Codigo { get; set; }  
}
```

Fuente: Elaboración Propia

Figura N° 233: Código del controlador Licenciamiento

```

namespace WebContServ.Controllers
{
    [Authorize]
    public class LicenciamientosController : Controller
    {
        private WebContServEntities db = new WebContServEntities();

        public ActionResult Index()
        {
            var licenciamiento = db.Licenciamiento.Include(l => l.RecursoAsignado1).Include(l => l.SoftwareTI1).Include(l => l.Usuario);
            return View(licenciamiento.ToList());
        }

        public ActionResult Details(int? id)
        {
            if (id == null) return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            Licenciamiento licenciamiento = db.Licenciamiento.Find(id);
            if (licenciamiento == null)
            {
                return HttpNotFound();
            }
            return View(licenciamiento);
        }

        public ActionResult Create()
        {
            ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre");
            ViewBag.SoftwareTI = new SelectList(db.SoftwareTI, "Id", "Codigo");
            ViewBag.LicenciadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre");
            return View();
        }

        [HttpPost] [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include = "Id,Nombre,RecursoAsignado,SoftwareTI,Serial,LicenciadoPor,FechaLic,FechaCaducidad,Codigo")] Licenciamiento licenciamiento)
        {
            if (ModelState.IsValid)
            {
                db.Licenciamiento.Add(licenciamiento);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", licenciamiento.RecursoAsignado);
            ViewBag.SoftwareTI = new SelectList(db.SoftwareTI, "Id", "Codigo", licenciamiento.SoftwareTI);
            ViewBag.LicenciadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", licenciamiento.LicenciadoPor);
            return View(licenciamiento);
        }

        public ActionResult Edit(int? id)
        {
            if (id == null) return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            Licenciamiento licenciamiento = db.Licenciamiento.Find(id);
            if (licenciamiento == null) return HttpNotFound();
            ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", licenciamiento.RecursoAsignado);
            ViewBag.SoftwareTI = new SelectList(db.SoftwareTI, "Id", "Codigo", licenciamiento.SoftwareTI);
            ViewBag.LicenciadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", licenciamiento.LicenciadoPor);
            return View(licenciamiento);
        }

        [HttpPost] [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include = "Id,Nombre,RecursoAsignado,SoftwareTI,Serial,LicenciadoPor,FechaLic,FechaCaducidad,Codigo")] Licenciamiento licenciamiento)
        {
            if (ModelState.IsValid)
            {
                db.Entry(licenciamiento).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", licenciamiento.RecursoAsignado);
            ViewBag.SoftwareTI = new SelectList(db.SoftwareTI, "Id", "Codigo", licenciamiento.SoftwareTI);
            ViewBag.LicenciadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", licenciamiento.LicenciadoPor);
            return View(licenciamiento);
        }

        public ActionResult Delete(int? id)
        {
            if (id == null) return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            Licenciamiento licenciamiento = db.Licenciamiento.Find(id);
            if (licenciamiento == null)
            {
                return HttpNotFound();
            }
            return View(licenciamiento);
        }
    }
}

```

Fuente: Elaboración Propia

Figura N° 234: Código de la vista Índice Licenciamiento

```

<h2>Licenciamiento</h2>
<p>@Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
    <tr>
        <th>Codigo </th>
        <th>Fecha de Licenciamiento</th>
        <th>Personal a Cargo</th>
        <th>Recurso Asignado</th>
        <th>Software</th>
        <th>Serial </th>
        <th>Fecha de Caducidad</th>
        <th>Nombre</th></tr>
    @foreach (var item in Model) {
        <tr>
            <td> @Html.DisplayFor(modelItem => item.Codigo)</td>
            <td> @Html.DisplayFor(modelItem => item.FechaLic)</td>
            <td> @Html.DisplayFor(modelItem => item.Usuario.Full_Nombre)</td>
            <td> @Html.DisplayFor(modelItem => item.RecursoAsignado1.Nombre)</td>
            <td> @Html.DisplayFor(modelItem => item.SoftwareTI1.Codigo)</td>
            <td> @Html.DisplayFor(modelItem => item.Serial)</td>
            <td> @Html.DisplayFor(modelItem => item.FechaCaducidad)</td>
            <td> @Html.DisplayFor(modelItem => item.Nombre)</td>
            <td> @Html.ActionLink("Editar", "Edit", new { id = item.Id })
                @Html.ActionLink("Detalles", "Details", new { id = item.Id })
                @Html.ActionLink("Eliminar", "Delete", new { id = item.Id })</td> </tr>
    }
</table>

```

Fuente: Elaboración Propia

Figura N° 235: Código de la vista Crear Licenciamiento

```
<h2>Crear</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Licenciamiento</h4><hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.FechaLic, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.FechaLic, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.FechaLic, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.LicenciadoPor, "LicenciadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("LicenciadoPor", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.LicenciadoPor, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.RecursoAsignado, "RecursoAsignado", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("RecursoAsignado", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.RecursoAsignado, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.SoftwareTI, "SoftwareTI", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("SoftwareTI", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.SoftwareTI, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Serial, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Serial, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Serial, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.FechaCaducidad, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.FechaCaducidad, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.FechaCaducidad, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                <div class="col-md-offset-2 col-md-10"> <input type="submit" value="Crear" class="btn btn-default" /></div></div></div>
    </div>
    @Html.ActionLink("Regresar al listado", "Index")</div>
}
```

Fuente: Elaboración Propia

Figura N° 236: Código de la vista Editar Licenciamiento

```

<h4>Licenciamiento</h4><hr />
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
@Html.HiddenFor(model => model.Id)
<div class="form-group">
    @Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" })</div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.FechaLic, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.FechaLic, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.FechaLic, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.LicenciadoPor, "LicenciadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("LicenciadoPor", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.LicenciadoPor, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.RecursoAsignado, "RecursoAsignado", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("RecursoAsignado", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.RecursoAsignado, "", new { @class = "text-danger" })</div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.SoftwareTI, "SoftwareTI", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("SoftwareTI", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.SoftwareTI, "", new { @class = "text-danger" }) </div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Serial, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Serial, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Serial, "", new { @class = "text-danger" }) </div> </div>
<div class="form-group">
    @Html.LabelFor(model => model.FechaCaducidad, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.FechaCaducidad, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.FechaCaducidad, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10"> <input type="submit" value="Guardar" class="btn btn-default" /> </div> </div></div>
<div>@Html.ActionLink("Regresar al listado", "Index")</div>

```

Fuente: Elaboración Propia

Figura N° 237: Código de la vista Borrar Licenciamiento

```

<h2>Delete</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div>
    <h4>Licenciamiento</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>@Html.DisplayNameFor(model => model.Nombre) </dt>
        <dd> @Html.DisplayFor(model => model.Nombre)</dd>
        <dt> @Html.DisplayNameFor(model => model.Serial) </dt>
        <dd> @Html.DisplayFor(model => model.Serial)</dd>
        <dt> @Html.DisplayNameFor(model => model.FechaLic) </dt>
        <dd> @Html.DisplayFor(model => model.FechaLic) </dd>
        <dt> @Html.DisplayNameFor(model => model.FechaCaducidad)</dt>
        <dd> @Html.DisplayFor(model => model.FechaCaducidad)</dd>
        <dt> @Html.DisplayNameFor(model => model.Codigo)</dt>
        <dd>@Html.DisplayFor(model => model.Codigo) </dd>
        <dt>@Html.DisplayNameFor(model => model.RecursoAsignado1.Nombre) </dt>
        <dd>@Html.DisplayFor(model => model.RecursoAsignado1.Nombre)</dd>
        <dt> @Html.DisplayNameFor(model => model.SoftwareTI1.Codigo)</dt>
        <dd> @Html.DisplayFor(model => model.SoftwareTI1.Codigo)</dd>
        <dt>@Html.DisplayNameFor(model => model.Usuario.Full_Nombre) </dt>
        <dd>@Html.DisplayFor(model => model.Usuario.Full_Nombre)</dd>
    </dl>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()
        <div class="form-actions no-color"> <input type="submit" value="Eliminar" class="btn btn-default" /> |
            @Html.ActionLink("Regresar al listado", "Index")</div>
    }
</div>

```

Fuente: Elaboración Propia

Figura N° 238: Formulario Índice Licenciamiento

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#) [Reportes](#) [Administración](#) Hola AAIQUIPA! [Cerrar sesión](#)

Licenciamiento

[Nuevo Registro](#)

Codigo	Fecha de Licenciamiento	Personal a Cargo	Recurso Asignado	Software	Serial	Fecha de Caducidad	Nombre
S1UH4	06/11/2018 0:00:00	Alex Aiquipa	PCNAV11	S000001	DGTR5	06/11/2019 0:00:00	LicOffice Editar Detalles Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 239: Formulario Crear Licenciamiento

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Crear Licenciamiento

Nombre

FechaLic

LicenciadoPor

RecursoAsignado

SoftwareTI

Serial

FechaCaducidad

Nombre

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 240: Formulario Actualizar Licenciamiento

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicio](#)

Actualizar

Licenciamiento

Codigo	<input type="text" value="S1UH4"/>
FechaLic	<input type="text" value="06/11/2018 0:00:00"/>
LicenciadoPor	<input type="text" value="Alex Aiquipa"/>
RecursoAsignado	<input type="text" value="PCNAV11"/>
SoftwareTI	<input type="text" value="S000001"/>
Serial	<input type="text" value="DGTR5"/>
FechaCaducidad	<input type="text" value="06/11/2019 0:00:00"/>
Nombre	<input type="text" value="LicOffice"/>

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 241: Formulario Borrar Licenciamiento

SISCONSER 👤 Gestión Usuarios 🔧 Recursos TI 🏠 Servicio

Delete

¿Estas seguro que deseas borrarlo?
Licenciamiento

Nombre LicOffice
Serial DGTR5
FechaLic 06/11/2018 0:00:00
FechaCaducidad 06/11/2019 0:00:00
Codigo S1UH4
Nombre PCNAV11
Codigo S000001
Full_Nombre Alex Aiquipa

| [Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Tabla N° 14: TaskBoard del Sprint 2 Semana 6

TaskBoard S6				
Sistema web para el control de os servicios de área de sistemas en la compañía Naviera Natalia SAC		Fecha Inicio: 15/08/2018 Fecha final: 22/08/2018		
	Historias por cada Sprint	Pendiente	En curso	Hecho
SPRINT N° 2	CRUD para los activos hardware de la Compañía.			X
	CRUD para los activos software de la Compañía.			X
	CRUD para los insumos de los equipos de impresión.			X
	CRUD para los componentes de las computadoras.			X
	Registrar de los componentes que contiene un equipo de cómputo.			X
	Registrar la asignación de los insumos a los equipos de impresión.			X
	Registrar los equipos informáticos asignados al personal administrativo de la Compañía.			X
	Registrar los licenciamientos de los equipos informáticos.			X

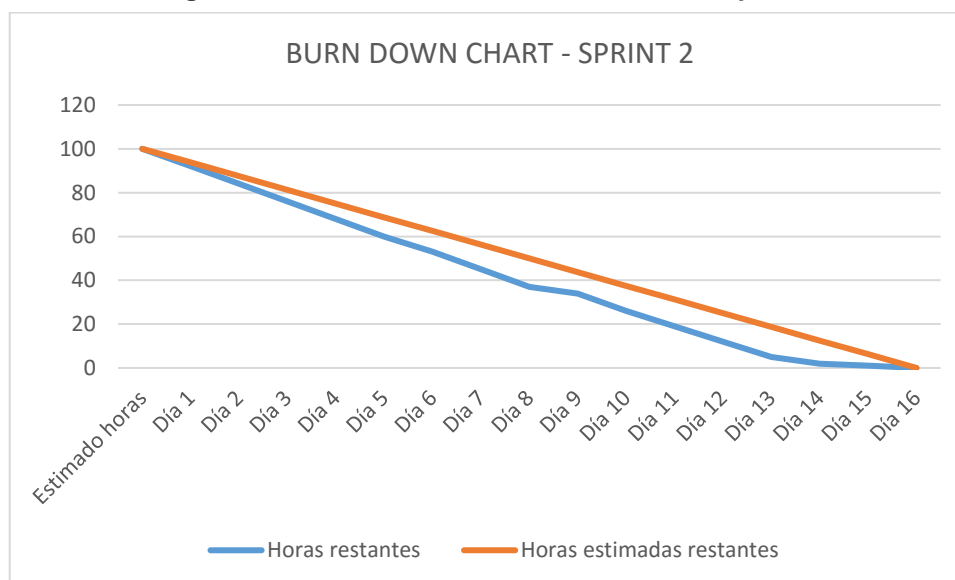
Fuente: Elaboración Propia

Tabla N° 15: BurnDown del Sprint 2

BurnDown Sprint 2																		
Tareas	Estimado horas	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7	Día 8	Día 9	Día 10	Día 11	Día 12	Día 13	Día 14	Día 15	Día 16	Total horas
		CRUD para los activos hardware de la Compañía.	12	8	4	0	0	0	0	0	0	0	0	0	0	0	0	
CRUD para los activos software de la Compañía.	8	0	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	8
CRUD para los insumos de los equipos de impresión.	8	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	8
CRUD para los componentes de las computadoras.	16	0	0	4	0	8	4	0	0	0	0	0	0	0	0	0	0	16
Registrar de los componentes que contiene un equipo de cómputo.	16	0	0	0	0	0	2	8	6	0	0	0	0	0	0	0	0	16
Registrar la asignación de los insumos a los equipos de impresión.	12	0	0	0	0	0	1	0	2	3	6	0	0	0	0	0	0	12
Registrar los equipos informáticos asignados al personal administrativo de la Compañía.	16	0	0	0	0	0	0	0	0	0	2	5	3	4	2	0	0	16
Registrar los licenciamientos de los equipos informáticos.	12	0	0	0	0	0	0	0	0	0	0	2	4	3	1	1	1	12
Horas restantes	100	92	84	76	68	60	53	45	37	34	26	19	12	5	2	1	0	
Horas estimadas restantes	100	93,8	87,5	81,3	75	68,8	62,5	56,3	50	43,8	37,5	31,3	25	18,8	12,5	6,25	0	

Fuente: Elaboración Propia

Figura N° 242: Grafico Burn Down Chart del Sprint 2



Fuente: Elaboración Propia

ACTA DE REUNIÓN N° 4

CIERRE DEL SPRINT 2

DATOS

Empresa / Organización	Compañía Naviera Natalia SAC.
Proyecto	Sistema web para el control de servicios del área de sistemas en la compañía Naviera Natalia SAC

PARTICIPANTES

Rol	Nombre
Dueño del Producto	Candela Ríos Hugo
Equipo	Aiquipa Tello Alex

ACUERDOS

Marcar con "X" a razón de cierre de cumplimiento de cada funcionalidad en la apertura del Sprint 2

Historias	Entrega Total
El sistema debe permitir al usuario autenticado, registrar y actualizar los activos de hardware de la Compañía.	X
El sistema debe permitir al usuario autenticado, registrar y actualizar los activos de software de la Compañía.	X
El sistema debe permitir al usuario autenticado, registrar y actualizar los insumos de los equipos de impresión de la Compañía.	X
El sistema debe permitir al usuario autenticado, registrar y actualizar los componentes de los ordenadores.	X
El sistema debe permitir al al usuario autenticado registrar y actualizar los componentes de ensamblaje de computo.	X
El sistema debe permitir al usuario autenticado registrar y actualizar los cambios de insumos en los equipos de impresión .	X
El sistema debe permitir al usuario autenticado registrar y actualizar la asignación de los equipos informáticos al personal de la compañía.	X
El sistema debe permitir al usuario autenticado registrar y actualizar las licencias asignadas a los equipos informáticos .	X

FIRMAS


 ING. HUGO CANDELA
 JEFE DE SISTEMAS CORPORATIVO

DUEÑO DEL PRODUCTO


 ALEX ALDO AIQUIPA TELLO
 ADMINISTRADOR DE REDES Y COMUNICACIONES

EQUIPO

ACTA DE REUNIÓN N° 5

APERTURA DEL SPRINT 3

DATOS

Empresa / Organización	Compañía Naviera Natalia SAC.
Proyecto	Sistema web para el control de servicios del área de sistemas en la compañía Naviera Natalia SAC

PARTICIPANTES

Rol	Nombre
Dueño del Producto	Candela Ríos Hugo
Equipo	Aiquipa Tello Alex

ACUERDOS

A continuación se listan las funcionalidades que se de desarrollaran e implementaran en el Sprint 3

Historias
El sistema debe permitir al administrador, generar y actualizar el cronograma de mantenimiento preventivo.
El sistema debe permitir al usuario autenticado, registrar y actualizar los servicios de mantenimientos preventivos realizados.
El sistema debe permitir al usuario autenticado, registrar y actualizar las solicitudes de atenciones realizadas por los usuarios.
El sistema debe permitir al usuario autenticado, registrar y actualizar las incidencias de los equipos informáticos.
El sistema debe permitir al usuario autenticado, registrar y actualizar los problemas suscitados.
El sistema debe permitir al usuario autenticado, registrar y actualizar los requerimientos de insumos y/o componentes solicitados para reparación de los equipos informáticos de la compañía.
El sistema debe permitir al administrador monitorizar el indicador de índice de cobertura de mantenimiento
El sistema debe permitir al administrador monitorizar el indicador de cumplimiento de las atenciones realizadas.

Firmas:



 ING. HUGO CANDELA
 JEFE DE SISTEMAS CORPORATIVO

DUEÑO DEL PRODUCTO



 ALEX ALDO AIQUIPA TELLO
 ADMINISTRADOR DE REDES Y COMUNICACIONES

EQUIPO

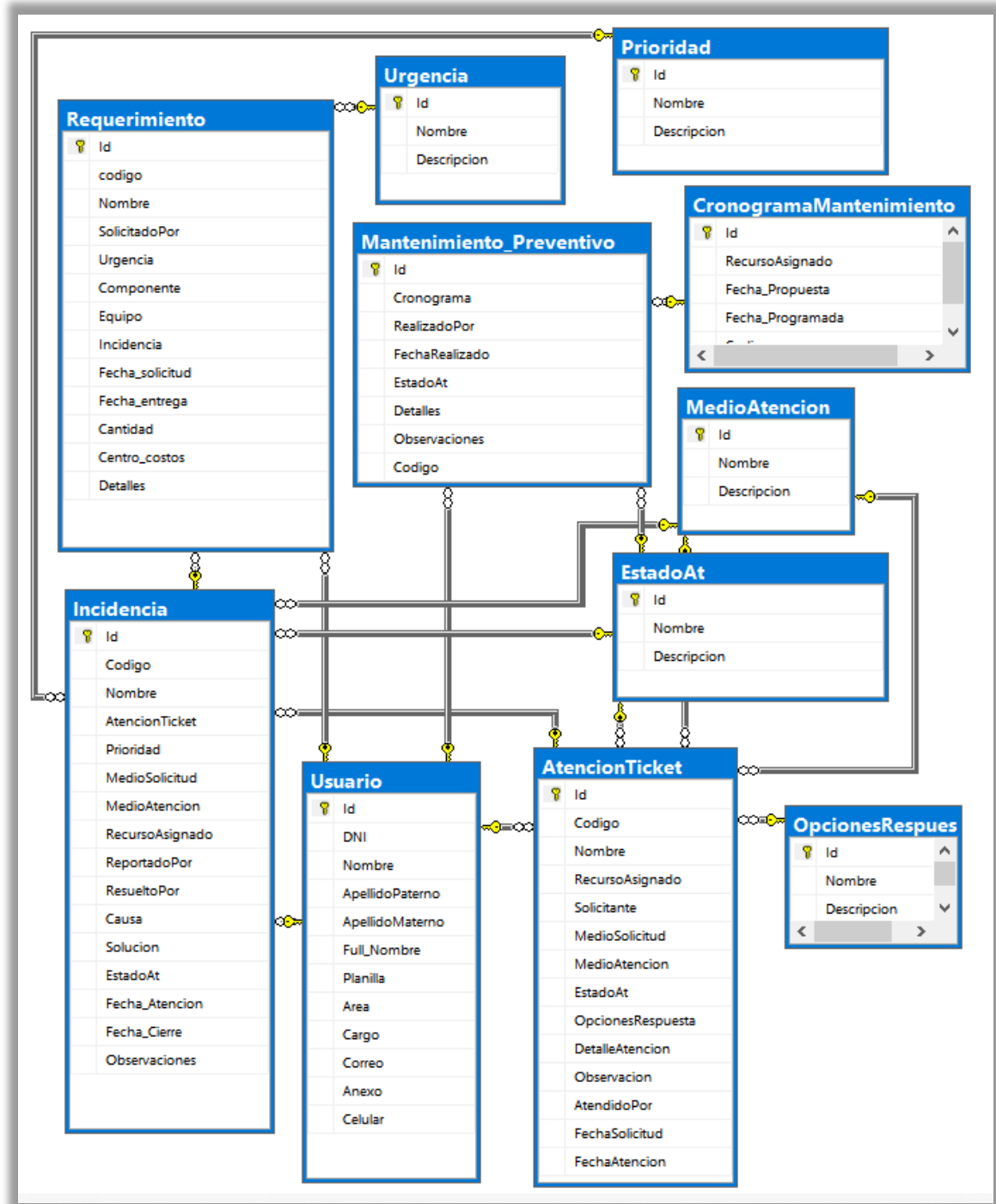
Sprint N° 3: Story Point 136

Tabla N° 16: TaskBoard del Sprint 3 Semana 7

TaskBoard S7					
Sistema web para el control de os servicios de área de sistemas en la compañía Naviera Natalia SAC			Fecha Inicio: 23/08/2018 Fecha final: 29/08/2018		
Historias por cada Sprint			Pendiente	En curso	Hecho
SPRINT N° 3	Registrar e cronograma de mantenimiento				X
	Administrar el estado de los mantenimientos realizados.				X
	Registrar las atenciones solicitadas por los usuarios.				X
	Administrar y dar mantenimiento al registro de las incidencias de equipos informáticos reportados.			X	
	Administrar y dar mantenimiento al registro de los problema reportados.		X		
	Registrar y administrar los requerimientos solicitados para reparación de los equipos informáticos.		X		
	Monitorizar el indicador de índice de cobertura de mantenimiento		X		
	Monitorizar el indicador de cumplimiento de las atenciones realizadas.		X		

Fuente: Elaboración Propia

Figura N° 243: Diagrama físico de a base de datos para e Sprint 3

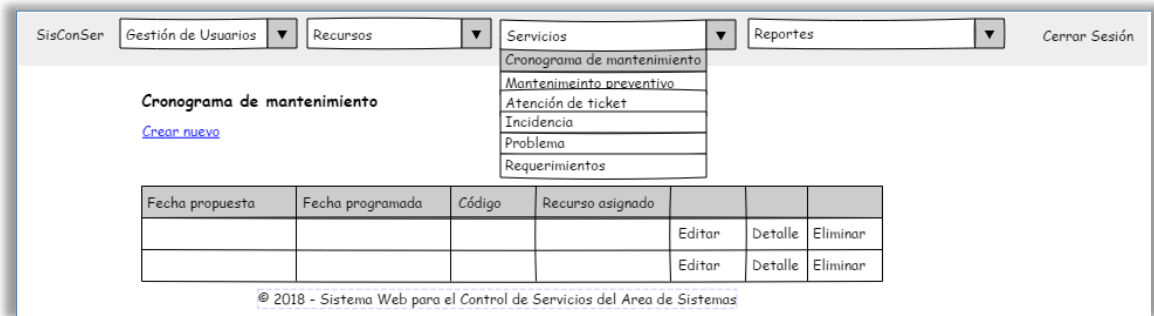


Fuente: Elaboración Propia

Requerimiento RF17

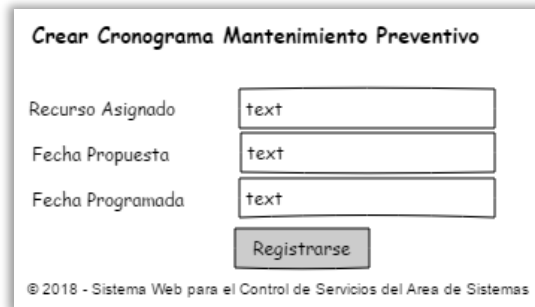
RF17: El sistema debe permitir al administrador, administrar y dar mantenimiento al cronograma de mantenimiento preventivo.

Figura N° 244: Prototipo 1 Index Cronograma de Mantenimiento



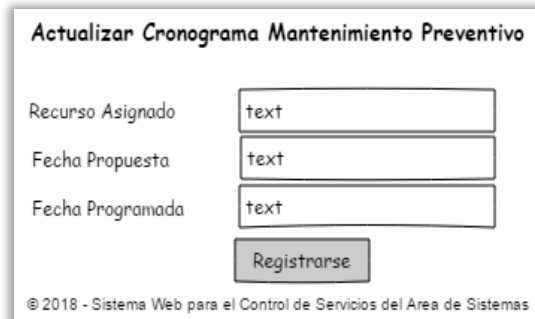
Fuente: Elaboración Propia

Figura N° 245: Prototipo 2 Crear Cronograma de Mantenimiento



Fuente: Elaboración Propia

Figura N° 246: Prototipo 3 Actualizar Cronograma de Mantenimiento



Fuente: Elaboración Propia

Figura N° 247: Prototipo 4 Eliminar Cronograma de Mantenimiento



Fuente: Elaboración Propia

Figura N° 248: Código para la creación de la tabla Cronograma de Mantenimiento

```
1 CREATE TABLE [dbo].[CronogramaMantenimiento] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [RecursoAsignado] INT NOT NULL,  
4     [Fecha_Propuesta] NVARCHAR (10) NOT NULL,  
5     [Fecha_Programada] NVARCHAR (10) NOT NULL,  
6     [Codigo] AS ('CM'+right('000000'+CONVERT([varchar],[Id]),(6))),  
7     PRIMARY KEY CLUSTERED ([Id] ASC),  
8     CONSTRAINT [FK_CronogramaMantenimiento_RecursoAsignado] FOREIGN KEY ([RecursoAsignado]) REFERENCES [dbo].[RecursoAsignado] ([Id])  
9 );  
10
```

Fuente: Elaboración Propia

Figura N° 249: Código del modelo Cronograma de Mantenimiento

```
public partial class CronogramaMantenimiento  
{[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]  
    0 referencias | 0 excepciones  
    public CronogramaMantenimiento()  
    { this.Mantenimiento_Preventivo = new HashSet<Mantenimiento_Preventivo>();  
    0 referencias | 0 excepciones  
    public int Id { get; set; }  
    3 referencias | 0 excepciones  
    public int RecursoAsignado { get; set; }  
    0 referencias | 0 excepciones  
    public string Fecha_Propuesta { get; set; }  
    0 referencias | 0 excepciones  
    public string Fecha_Programada { get; set; }  
    0 referencias | 0 excepciones  
    public string Codigo { get; set; }  
  
    1 referencia | 0 excepciones  
    public virtual RecursoAsignado RecursoAsignado1 { get; set; }  
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]  
    1 referencia | 0 excepciones  
    public virtual ICollection<Mantenimiento_Preventivo> Mantenimiento_Preventivo { get; set; }  
}
```

Fuente: Elaboración Propia

Figura N° 250: Código para el controlador Cronograma de Mantenimiento

```
public class CronogramaMantenimientosController : Controller  
{ private WebConSerSisEntities db = new WebConSerSisEntities();  
    0 referencias | 0 solicitudes | 0 excepciones  
    public ActionResult Index()  
    {var cronogramaMantenimiento = db.CronogramaMantenimiento.Include(c => c.RecursoAsignado1);  
    return View(cronogramaMantenimiento.ToList());  
    0 referencias | 0 solicitudes | 0 excepciones  
    public ActionResult Details(int? id)[...]  
    0 referencias | 0 solicitudes | 0 excepciones  
    public ActionResult Create()  
    {ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre");  
    return View();  
    [HttpPost][ValidateAntiForgeryToken]  
    0 referencias | 0 solicitudes | 0 excepciones  
    public ActionResult Create([Bind(Include = "Id,RecursoAsignado,Fecha_Propuesta,Fecha_Programada,Codigo")] CronogramaMantenimiento cronogramaMantenimiento)  
    {if (ModelState.IsValid){ db.CronogramaMantenimiento.Add(cronogramaMantenimiento);  
    db.SaveChanges();  
    return RedirectToAction("Index");  
    ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", cronogramaMantenimiento.RecursoAsignado);  
    return View(cronogramaMantenimiento);  
    0 referencias | 0 solicitudes | 0 excepciones  
    public ActionResult Edit(int? id)  
    {if (id == null){return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  
    CronogramaMantenimiento cronogramaMantenimiento = db.CronogramaMantenimiento.Find(id);  
    if (cronogramaMantenimiento == null) {return HttpNotFound();}  
    ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", cronogramaMantenimiento.RecursoAsignado);  
    return View(cronogramaMantenimiento);  
    [HttpPost]  
    [ValidateAntiForgeryToken]  
    0 referencias | 0 solicitudes | 0 excepciones  
    public ActionResult Edit([Bind(Include = "Id,RecursoAsignado,Fecha_Propuesta,Fecha_Programada,Codigo")] CronogramaMantenimiento cronogramaMantenimiento)  
    {if (ModelState.IsValid){db.Entry(cronogramaMantenimiento).State = EntityState.Modified;  
    db.SaveChanges();  
    return RedirectToAction("Index");  
    ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", cronogramaMantenimiento.RecursoAsignado);  
    return View(cronogramaMantenimiento);  
    0 referencias | 0 solicitudes | 0 excepciones  
    public ActionResult Delete(int? id)  
    {if (id == null){ return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  
    CronogramaMantenimiento cronogramaMantenimiento = db.CronogramaMantenimiento.Find(id);  
    if (cronogramaMantenimiento == null){return HttpNotFound();}  
    return View(cronogramaMantenimiento);  
    [HttpPost, ActionName("Delete")]  
    [ValidateAntiForgeryToken]  
    0 referencias | 0 solicitudes | 0 excepciones  
    public ActionResult DeleteConfirmed(int id)  
    {CronogramaMantenimiento cronogramaMantenimiento = db.CronogramaMantenimiento.Find(id);  
    db.CronogramaMantenimiento.Remove(cronogramaMantenimiento);  
    db.SaveChanges();  
    return RedirectToAction("Index");}
```

Fuente: Elaboración Propia

Figura N° 251: Código de la vista Índice Cronograma de Mantenimiento

```

<h2>Cronograma de Mantenimiento</h2>
<p> @Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
  <tr> <th> @Html.DisplayNameFor(model => model.Fecha_Propuesta) </th>
    <th> @Html.DisplayNameFor(model => model.Fecha_Programada)</th>
    <th> @Html.DisplayNameFor(model => model.Codigo)</th>
    <th> Recurso Asignado</th></tr>
  @foreach (var item in Model) {
    <tr> <td> @Html.DisplayFor(modelItem => item.Fecha_Propuesta) </td>
      <td> @Html.DisplayFor(modelItem => item.Fecha_Programada) </td>
      <td>@Html.DisplayFor(modelItem => item.Codigo)</td>
      <td> @Html.DisplayFor(modelItem => item.RecursoAsignado1.Nombre)</td>
      <td> @Html.ActionLink("Editar", "Edit", new { id=item.Id })
        @Html.ActionLink("Detalles", "Details", new { id=item.Id })
        @Html.ActionLink("Eliminar", "Delete", new { id=item.Id })</td></tr>
  }
</table>

```

Fuente: Elaboración Propia

Figura N° 252: Código de la vista Crear Cronograma de Mantenimiento

```

<h2>Crear</h2>
@using (Html.BeginForm())
{
  @Html.AntiForgeryToken()
  <div class="form-horizontal">
    <h4>Cronograma de Mantenimiento</h4>
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
      @Html.LabelFor(model => model.RecursoAsignado, "RecursoAsignado", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("RecursoAsignado", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.RecursoAsignado, "", new { @class = "text-danger" }) </div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Fecha_Propuesta, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Fecha_Propuesta, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Fecha_Propuesta, "", new { @class = "text-danger" })</div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.Fecha_Programada, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Fecha_Programada, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Fecha_Programada, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      <div class="col-md-offset-2 col-md-10"><input type="submit" value="Crear" class="btn btn-default" /> </div> </div></div>
    <div>@Html.ActionLink("Regresar al listado", "Index")</div>

```

Fuente: Elaboración Propia

Figura N° 253: Código de la vista Actualizar Cronograma de Mantenimiento

```

<h2>Actualizar</h2>
@using (Html.BeginForm())
{ @Html.AntiForgeryToken()
  <div class="form-horizontal">
    <h4>Cronograma de Mantenimiento</h4><hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    @Html.HiddenFor(model => model.Id)
    <div class="form-group">
      @Html.LabelFor(model => model.RecursoAsignado, "RecursoAsignado", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("RecursoAsignado", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.RecursoAsignado, "", new { @class = "text-danger" })</div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.Fecha_Propuesta, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Fecha_Propuesta, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Fecha_Propuesta, "", new { @class = "text-danger" })</div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.Fecha_Programada, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Fecha_Programada, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Fecha_Programada, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      @Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
      <div class="col-md-offset-2 col-md-10"><input type="submit" value="Guardar" class="btn btn-default" /></div> </div></div>
  </div> @Html.ActionLink("Regresar al listado", "Index")</div>
}

```

Fuente: Elaboración Propia

Figura N° 254: Código de la vista Eliminar Cronograma de Mantenimiento

```

<h2>Eliminar</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div><h4>Cronograma de Mantenimiento</h4><hr />
  <dl class="dl-horizontal">
    <dt>@Html.DisplayNameFor(model => model.Fecha_Propuesta)</dt>
    <dd>@Html.DisplayFor(model => model.Fecha_Propuesta)</dd>
    <dt>@Html.DisplayNameFor(model => model.Fecha_Programada)</dt>
    <dd>@Html.DisplayFor(model => model.Fecha_Programada)</dd>
    <dt>@Html.DisplayNameFor(model => model.Codigo)</dt>
    <dd>@Html.DisplayFor(model => model.Codigo)</dd>
    <dt>@Html.DisplayNameFor(model => model.RecursoAsignado1.Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.RecursoAsignado1.Nombre) </dd></dl>
  @using (Html.BeginForm()) { @Html.AntiForgeryToken()
    <div class="form-actions no-color">
      <input type="submit" value="Eliminar" class="btn btn-default" /> |
      @Html.ActionLink("Regresar al listado", "Index")
    </div>
  }
</div>

```

Fuente: Elaboración Propia

Figura N° 255: Formulario Index Cronograma de Mantenimiento

Fecha_Propuesta	Fecha_Programada	Codigo	Recurso Asignado	
2018-7-11	2018-7-12	CM000003	PCNAV11	Editar Detalles Eliminar
2018-7-11	2018-7-12	CM000004	PCNAV11	Editar Detalles Eliminar
2018-8-11	2018-8-12	CM000005	PCNAV11	Editar Detalles Eliminar
2018-8-11	2018-8-12	CM000006	PCNAV11	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 256: Formulario Crear Cronograma de Mantenimiento

The screenshot shows the 'Crear Cronograma de Mantenimiento' form. At the top, there is a navigation bar with 'SISCONSER' and three menu items: 'Gestión Usuarios', 'Recursos TI', and 'Servicios'. The main heading is 'Crear Cronograma de Mantenimiento'. The form contains three input fields: 'RecursoAsignado' (a dropdown menu with 'PCNAV11' selected), 'Fecha_Propuesta' (an empty text box), and 'Fecha_Programada' (an empty text box). Below these fields is a 'Crear' button. At the bottom left, there is a link 'Regresar al listado'. The footer contains the text '© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas'.

Fuente: Elaboración Propia

Figura N° 257: Formulario Actualizar Cronograma de Mantenimiento

The screenshot shows the 'Actualizar Cronograma de Mantenimiento' form. At the top, there is a navigation bar with 'SISCONSER' and three menu items: 'Gestión Usuarios', 'Recursos TI', and 'Servicios'. The main heading is 'Actualizar Cronograma de Mantenimiento'. The form contains four input fields: 'RecursoAsignado' (a dropdown menu with 'PCNAV11' selected), 'Fecha_Propuesta' (a text box containing '2018-7-11'), 'Fecha_Programada' (a text box containing '2018-7-12'), and 'Codigo' (a text box containing 'CM000003'). Below these fields is a 'Guardar' button. At the bottom left, there is a link 'Regresar al listado'. The footer contains the text '© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas'.

Fuente: Elaboración Propia

Figura N° 258: Formulario Eliminar Cronograma de Mantenimiento

SISCONSER 👤 Gestión Usuarios 📁 Recursos TI 🔒 Ser

Eliminar

¿Estas seguro que deseas borrarlo?
Cronograma de Mantenimiento

Fecha_Propuesta 2018-7-11
Fecha_Programada 2018-7-12
Codigo Html.DisplayFor(model => model.Codigo)
Nombre PCNAV11

| [Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Requerimiento RF18

RF18: El sistema debe permitir al usuario autenticado, registrar y actualizar los servicios de mantenimientos preventivos realizados.

Figura N° 259: Prototipo 1 Index Mantenimiento Preventivo

SisConSer 👤 Gestión de Usuarios 📁 Recursos 📁 Servicios 📁 Reportes 🔒 Cerrar Sesión

Mantenimiento preventivo
[Crear nuevo](#)

- Cronograma de mantenimiento
- Mantenimiento preventivo**
- Atención de ticket
- Incidencia
- Problema
- Requerimientos

Código	Fecha programada	Fecha de realización	Personal a cargo	Estado			
					Editar	Detalle	Eliminar
					Editar	Detalle	Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 260: Prototipo 2 Registrar Mantenimiento Preventivo

Crear Mantenimiento Preventivo

Cronograma	<input type="text"/>
FechaRealizado	<input type="text"/>
RealizadoPor	<input type="text"/>
Detalles	<input type="text"/>
Observaciones	<input type="text"/>
EstadoAt	<input type="text"/>

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 261: Prototipo 3 Actualizar Mantenimiento Preventivo

Actualizar Mantenimiento Preventivo

Cronograma	<input type="text"/>
FechaRealizado	<input type="text"/>
RealizadoPor	<input type="text"/>
Detalles	<input type="text"/>
Observaciones	<input type="text"/>
EstadoAt	<input type="text"/>

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 262: Prototipo 4 Borrar Mantenimiento Preventivo

Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 263: Código para la creación de la tabla Mantenimiento Preventivo

```
1 CREATE TABLE [dbo].[Mantenimiento_Preventivo] (
2     [Id] INT IDENTITY (1, 1) NOT NULL,
3     [Cronograma] INT NOT NULL,
4     [RealizadoPor] INT NOT NULL,
5     [FechaRealizado] DATE NOT NULL,
6     [EstadoAt] INT NOT NULL,
7     [Detalles] NVARCHAR (200) NULL,
8     [Observaciones] NVARCHAR (200) NULL,
9     [Codigo] AS (( 'CM'+CONVERT([varchar],[Cronograma]))+'MP'+CONVERT([varchar],[Id]))),
10    PRIMARY KEY CLUSTERED ([Id] ASC),
11    CONSTRAINT [FK_MantenimientoPreventivo_CronogramaMantenimiento] FOREIGN KEY ([Cronograma]) REFERENCES [dbo].[CronogramaMantenimiento] ([Id]),
12    CONSTRAINT [FK_MantenimientoPreventivo_Usuario] FOREIGN KEY ([RealizadoPor]) REFERENCES [dbo].[Usuario] ([Id]),
13    CONSTRAINT [FK_MantenimientoPreventivo_EstadoAt] FOREIGN KEY ([EstadoAt]) REFERENCES [dbo].[EstadoAt] ([Id])
14 );
```

Fuente: Elaboración Propia

Figura N° 264: código del modelo Mantenimiento Preventivo

```
public partial class Mantenimiento_Preventivo
{
    0 referencias | 0 excepciones
    public int Id { get; set; }
    3 referencias | 0 excepciones
    public int Cronograma { get; set; }
    3 referencias | 0 excepciones
    public int RealizadoPor { get; set; }
    0 referencias | 0 excepciones
    public System.DateTime FechaRealizado { get; set; }
    3 referencias | 0 excepciones
    public int EstadoAt { get; set; }
    0 referencias | 0 excepciones
    public string Detalles { get; set; }
    0 referencias | 0 excepciones
    public string Observaciones { get; set; }
    0 referencias | 0 excepciones
    public string Codigo { get; set; }

    1 referencia | 0 excepciones
    public virtual CronogramaMantenimiento CronogramaMantenimiento { get; set; }
    1 referencia | 0 excepciones
    public virtual EstadoAt EstadoAt1 { get; set; }
    1 referencia | 0 excepciones
    public virtual Usuario Usuario { get; set; }
}
```

Fuente: Elaboración Propia

Figura N° 265: Código del controlador Mantenimiento Preventivo

```

public class Mantenimiento_PreventivoController : Controller
{
    private WebConSerSisEntities db = new WebConSerSisEntities();
    public ActionResult Index()
    {
        var mantenimiento_Preventivo = db.Mantenimiento_Preventivo.Include(m => m.CronogramaMantenimiento).Include(m => m.EstadoAt1).Include(m => m.Usuario);
        return View(mantenimiento_Preventivo.ToList());
    }
    public ActionResult Details(int? id)
    {
        public ActionResult Create()
        {
            ViewBag.Cronograma = new SelectList(db.CronogramaMantenimiento, "Id", "Fecha_Propuesta");
            ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre");
            ViewBag.RealizadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre");
            return View();
        }
        [HttpPost] [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include = "Id,Cronograma,RealizadoPor,FechaRealizado,EstadoAt,Detalles,Observaciones,Codigo")] Mantenimiento_Preventivo mantenimiento_Preventivo)
        {
            if (ModelState.IsValid)
            {
                db.Mantenimiento_Preventivo.Add(mantenimiento_Preventivo);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.Cronograma = new SelectList(db.CronogramaMantenimiento, "Id", "Fecha_Propuesta", mantenimiento_Preventivo.Cronograma);
            ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre", mantenimiento_Preventivo.EstadoAt);
            ViewBag.RealizadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", mantenimiento_Preventivo.RealizadoPor);
            return View(mantenimiento_Preventivo);
        }
        public ActionResult Edit(int? id)
        {
            if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }
            Mantenimiento_Preventivo mantenimiento_Preventivo = db.Mantenimiento_Preventivo.Find(id);
            if (mantenimiento_Preventivo == null)
            {
                return HttpNotFound();
            }
            ViewBag.Cronograma = new SelectList(db.CronogramaMantenimiento, "Id", "Fecha_Propuesta", mantenimiento_Preventivo.Cronograma);
            ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre", mantenimiento_Preventivo.EstadoAt);
            ViewBag.RealizadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", mantenimiento_Preventivo.RealizadoPor);
            return View(mantenimiento_Preventivo);
        }
        [HttpPost] [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include = "Id,Cronograma,RealizadoPor,FechaRealizado,EstadoAt,Detalles,Observaciones,Codigo")] Mantenimiento_Preventivo mantenimiento_Preventivo)
        {
            if (ModelState.IsValid) { db.Entry(mantenimiento_Preventivo).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index"); }
            ViewBag.Cronograma = new SelectList(db.CronogramaMantenimiento, "Id", "Fecha_Propuesta", mantenimiento_Preventivo.Cronograma);
            ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre", mantenimiento_Preventivo.EstadoAt);
            ViewBag.RealizadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", mantenimiento_Preventivo.RealizadoPor);
            return View(mantenimiento_Preventivo);
        }
        public ActionResult Delete(int? id)
        {
            if (id == null) { return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }
            Mantenimiento_Preventivo mantenimiento_Preventivo = db.Mantenimiento_Preventivo.Find(id);
            if (mantenimiento_Preventivo == null) { return HttpNotFound(); }
            return View(mantenimiento_Preventivo);
        }
        [HttpPost, ActionName("Delete")] [ValidateAntiForgeryToken]
        public ActionResult DeleteConfirmed(int id)
        {
            Mantenimiento_Preventivo mantenimiento_Preventivo = db.Mantenimiento_Preventivo.Find(id);
            db.Mantenimiento_Preventivo.Remove(mantenimiento_Preventivo);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
}

```

Fuente: Elaboración Propia

Figura N° 266: Código de la vista Index Mantenimiento Preventivo

```

<h2>Mantenimiento</h2>
<p> @Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
    <tr>
        <th>Codigo</th>
        <th>Fecha Programada</th>
        <th>Fecha de realizacion</th>
        <th>Personal a Cargo</th>
        <th>Estado</th>
    </tr>
    @foreach (var item in Model) {
        <tr>
            <td>@Html.DisplayFor(modelItem => item.Codigo)</td>
            <td>@Html.DisplayFor(modelItem => item.CronogramaMantenimiento.Fecha_Propuesta)</td>
            <td>@Html.DisplayFor(modelItem => item.FechaRealizado)</td>
            <td>@Html.DisplayFor(modelItem => item.Usuario.Full_Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.EstadoAt1.Nombre)</td>
            <td>@Html.ActionLink("Editar", "Edit", new { id = item.Id }) |
                @Html.ActionLink("Detalles", "Details", new { id = item.Id }) |
                @Html.ActionLink("Eliminar", "Delete", new { id = item.Id })</td>
        </tr>
    }
</table>

```

Fuente: Elaboración Propia

Figura N° 267: Código de la vista Crear Mantenimiento Preventivo

```
<h2>Crear</h2>
@using (Html.BeginForm())
{@Html.AntiForgeryToken()
<div class="form-horizontal">
  <h4>Mantenimiento_Preventivo</h4><hr />
  @Html.ValidationSummary(true, "", new { @class = "text-danger" })
  <div class="form-group">
    @Html.LabelFor(model => model.Cronograma, "Cronograma", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("Cronograma", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.Cronograma, "", new { @class = "text-danger" }) </div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.FechaRealizado, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.FechaRealizado, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.FechaRealizado, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.RealizadoPor, "RealizadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("RealizadoPor", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.RealizadoPor, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.Detalles, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Detalles, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Detalles, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.Observaciones, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.Observaciones, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.Observaciones, "", new { @class = "text-danger" })</div> </div>
  <div class="form-group">
    @Html.LabelFor(model => model.EstadoAt, "EstadoAt", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.DropDownList("EstadoAt", null, htmlAttributes: new { @class = "form-control" })
      @Html.ValidationMessageFor(model => model.EstadoAt, "", new { @class = "text-danger" }) </div></div>
  <div class="form-group">
    <div class="col-md-offset-2 col-md-10"><input type="submit" value="Crear" class="btn btn-default" /></div> </div></div>
</div> @Html.ActionLink("Regresar al listado", "Index")</div>
```

Fuente: Elaboración Propia

Figura N° 268: Código de la vista Actualizar Mantenimiento Preventivo

```

<h2>Actualizar</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Mantenimiento_Preventivo</h4><hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)
        <div class="form-group">
            @Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" }) </div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Cronograma, "Cronograma", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Cronograma", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.Cronograma, "", new { @class = "text-danger" }) </div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.FechaRealizado, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.FechaRealizado, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.FechaRealizado, "", new { @class = "text-danger" })</div> </div>
        <div class="form-group">
            @Html.LabelFor(model => model.RealizadoPor, "RealizadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("RealizadoPor", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.RealizadoPor, "", new { @class = "text-danger" })</div> </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Detalles, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Detalles, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Detalles, "", new { @class = "text-danger" })</div> </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Observaciones, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Observaciones, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Observaciones, "", new { @class = "text-danger" })</div> </div>
        <div class="form-group">
            @Html.LabelFor(model => model.EstadoAt, "EstadoAt", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("EstadoAt", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.EstadoAt, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10"> <input type="submit" value="Guardar" class="btn btn-default" /> </div></div></div>
        <div> @Html.ActionLink("Regresar al listado", "Index")</div>
    }
}

```

Fuente: Elaboración Propia

Figura N° 269: Código de la vista Eliminar Mantenimiento Preventivo

```

<h2>Eliminar</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div> <h4>Mantenimiento_Preventivo</h4>
    <hr /><dl class="dl-horizontal">
        <dt>@Html.DisplayNameFor(model => model.FechaRealizado)</dt>
        <dd> @Html.DisplayFor(model => model.FechaRealizado)</dd>
        <dt> @Html.DisplayNameFor(model => model.Detalles)</dt>
        <dd>@Html.DisplayFor(model => model.Detalles) </dd>
        <dt>@Html.DisplayNameFor(model => model.Observaciones)</dt>
        <dd>@Html.DisplayFor(model => model.Observaciones)</dd>
        <dt>@Html.DisplayNameFor(model => model.Codigo)</dt>
        <dd>@Html.DisplayFor(model => model.Codigo) </dd>
        <dt>@Html.DisplayNameFor(model => model.CronogramaMantenimiento.Fecha_Propuesta)</dt>
        <dd>@Html.DisplayFor(model => model.CronogramaMantenimiento.Fecha_Propuesta) </dd>
        <dt>@Html.DisplayNameFor(model => model.EstadoAt1.Nombre)</dt>
        <dd>@Html.DisplayFor(model => model.EstadoAt1.Nombre)</dd>
        <dt>@Html.DisplayNameFor(model => model.Usuario.Full_Nombre) </dt>
        <dd>@Html.DisplayFor(model => model.Usuario.Full_Nombre) </dd></dl>
        @using (Html.BeginForm()) { @Html.AntiForgeryToken()
            <div class="form-actions no-color">
                <input type="submit" value="Eliminar" class="btn btn-default" /> |
                @Html.ActionLink("Regresar al listado", "Index")
            </div>
        }
    </div>
}

```

Fuente: Elaboración Propia

Figura N° 270: Formulario Index Mantenimiento Preventivo

Codigo	Fecha Programada	Fecha de realizacion	Personal a Cargo	Estado	
CM3MP3	2018-7-11	12/07/2018 0:00:00	Alex Aiquipa	Abierto	Editar Detalles Eliminar
CM3MP4	2018-7-11	22/07/2018 0:00:00	Carlos Orellana	Abierto	Editar Detalles Eliminar
CM5MP5	2018-8-11	12/08/2018 0:00:00	Carlos Orellana	Cerrado	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 271: Formulario Crear Mantenimiento Preventivo

Crear
Mantenimiento_Preventivo

Cronograma: 2018-7-11

FechaRealizado:

RealizadoPor: Alex Aiquipa

Detalles:

Observaciones:

EstadoAt: Abierto

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 272: Formulario Actualizar Mantenimiento Preventivo

Actualizar
Mantenimiento_Preventivo

Codigo: CM3MP3

Cronograma: 2018-7-11

FechaRealizado: 12/07/2018 0:00:00

RealizadoPor: Alex Aiquipa

Detalles:

Observaciones:

EstadoAt: Abierto

[Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 273: Formulario Eliminar Mantenimiento Preventivo

SISCONSER Gestión Usuarios Recursos TI

Eliminar

¿Estas seguro que deseas borrarlo?
Mantenimiento_Preventivo

FechaRealizado 12/07/2018 0:00:00

Detalles

Observaciones

Codigo CM3MP3

Fecha_Propuesta 2018-7-11

Nombre Abierto

Full_Nombre Alex AiQUIpa

| [Regresar al listado](#)

Fuente: Elaboración Propia

Requerimiento RF19

RF19: El sistema debe permitir al usuario autenticado, registrar y actualizar las solicitudes de atenciones realizadas por los usuarios.

Figura N° 274: Prototipo 1 Index Atención de Ticket

SisConSer Gestión de Usuarios Recursos Servicios Reportes Cerrar Sesión

Atención de ticket

[Crear nuevo](#)

Código	Asunto	Fecha Solicitud	Solicitante	Medio solicitud	Recurso afectado	Estado	Fecha atención	Medio atención	Opciones de respuesta	Personal a cargo		
											Editar	Eliminar
											Editar	Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 275: Prototipo 2 Agregar Atención de Ticket

Crear Atencion de Solicitudes

Nombre	text
Recurso Asignado	text
Solicitante	text
Medio de Solicitud	text
Medio de Atención	text
EstadoAT	text
Opciones de respuesta	text
Detalle	text
Observación	text
AtendidoPor	text
FechaSolicitud	text
FechaAtención	text

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 276: Prototipo 3 Actualizar Atención de Ticket

Actualizar Atencion de Solicitudes

Nombre	text
Recurso Asignado	text
Solicitante	text
Medio de Solicitud	text
Medio de Atención	text
EstadoAT	text
Opciones de respuesta	text
Detalle	text
Observación	text
AtendidoPor	text
FechaSolicitud	text
FechaAtención	text

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 277: Prototipo 4 Eliminar Atención de Ticket

Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 278: Código para la creación de la tabla Atención de Ticket

```
1 CREATE TABLE [dbo].[AtencionTicket] (
2 [Id] INT IDENTITY (1, 1) NOT NULL,
3 [Codigo] AS ('AT'+right('000000'+CONVERT([varchar],[Id]),(6))),
4 [Nombre] NVARCHAR (50) NOT NULL,
5 [RecursoAsignado] INT NOT NULL,
6 [Solicitante] INT NOT NULL,
7 [MedioSolicitud] INT NOT NULL,
8 [MedioAtencion] INT NOT NULL,
9 [EstadoAt] INT NOT NULL,
10 [OpcionesRespuesta] INT NOT NULL,
11 [DetalleAtencion] NVARCHAR (200) NULL,
12 [Observacion] NVARCHAR (200) NULL,
13 [AtendidoPor] INT NOT NULL,
14 [FechaSolicitud] DATETIME NOT NULL,
15 [FechaAtencion] DATETIME NULL,
16 PRIMARY KEY CLUSTERED ([Id] ASC),
17 CONSTRAINT [FK_Atencion_RecursoAsignado] FOREIGN KEY ([RecursoAsignado]) REFERENCES [dbo].[RecursoAsignado] ([Id]),
18 CONSTRAINT [FK_Atencion_Usuario] FOREIGN KEY ([Solicitante]) REFERENCES [dbo].[Usuario] ([Id]),
19 CONSTRAINT [FK_Atencion_MedioSolicitud] FOREIGN KEY ([MedioSolicitud]) REFERENCES [dbo].[MedioAtencion] ([Id]),
20 CONSTRAINT [FK_Atencion_MedioAtencion] FOREIGN KEY ([MedioAtencion]) REFERENCES [dbo].[MedioAtencion] ([Id]),
21 CONSTRAINT [FK_Atencion_EstadoAt] FOREIGN KEY ([EstadoAt]) REFERENCES [dbo].[EstadoAt] ([Id]),
22 CONSTRAINT [FK_Atencion_OpcionesRespuesta] FOREIGN KEY ([OpcionesRespuesta]) REFERENCES [dbo].[OpcionesRespuesta] ([Id]),
23 CONSTRAINT [FK_Atencion_UsuarioS] FOREIGN KEY ([AtendidoPor]) REFERENCES [dbo].[Usuario] ([Id])
24 );
```

Fuente: Elaboración Propia

Figura N° 279. Código del modelo Atención de Ticket

```
public partial class AtencionTicket
{[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    0 referencias | 0 excepciones
    public AtencionTicket()
    {this.Incidencia = new HashSet<Incidencia>(); }
    0 referencias | 0 excepciones
    public int Id { get; set; }
    0 referencias | 0 excepciones
    public string Codigo { get; set; }
    0 referencias | 0 excepciones
    public string Nombre { get; set; }
    3 referencias | 0 excepciones
    public int RecursoAsignado { get; set; }
    3 referencias | 0 excepciones
    public int Solicitante { get; set; }
    3 referencias | 0 excepciones
    public int MedioSolicitud { get; set; }
    3 referencias | 0 excepciones
    public int MedioAtencion { get; set; }
    3 referencias | 0 excepciones
    public int EstadoAt { get; set; }
    3 referencias | 0 excepciones
    public int OpcionesRespuesta { get; set; }
    0 referencias | 0 excepciones
    public string DetalleAtencion { get; set; }
    0 referencias | 0 excepciones
    public string Observacion { get; set; }
    3 referencias | 0 excepciones
    public int AtendidoPor { get; set; }
    0 referencias | 0 excepciones
    public System.DateTime FechaSolicitud { get; set; }
    0 referencias | 0 excepciones
    public Nullable<System.DateTime> FechaAtencion { get; set; }
}
```

Fuente: Elaboración Propia

Figura N° 280: Código del controlador Atención Ticket

```
public class AtencionTicketsController : Controller
{private WebContext db = new WebContext();
    0 referencias | 0 excepciones
    public ActionResult Index()
    { var atencionTicket = db.AtencionTicket.Include(a => a.EstadoAt).Include(a => a.MedioAtencion).Include(a => a.MedioAtencion2).Include(a => a.OpcionesRespuesta).Include(a => a.RecursoAsignado).Include(a => a.Usuario).Include(a => a.Usuario1);
      return View(atencionTicket.ToList()); }
    0 referencias | 0 excepciones
    public ActionResult Details(int? id)
    { [HttpPost]
      [ValidateAntiForgeryToken]
      public ActionResult Create([Bind(Include = "Id,Codigo,Nombre,RecursoAsignado,Solicitante,MedioSolicitud,MedioAtencion,EstadoAt,OpcionesRespuesta,DetalleAtencion,Observacion,AtendidoPor,FechaSolicitud,FechaAtencion")] AtencionTicket atencionTicket)
      { if (ModelState.IsValid) db.AtencionTicket.Add(atencionTicket);
        db.SaveChanges();
        return RedirectToAction("Index"); }
      ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre", atencionTicket.EstadoAt);
      ViewBag.MedioAtencion = new SelectList(db.MedioAtencion, "Id", "Nombre", atencionTicket.MedioAtencion);
      ViewBag.MedioSolicitud = new SelectList(db.MedioAtencion, "Id", "Nombre", atencionTicket.MedioSolicitud);
      ViewBag.OpcionesRespuesta = new SelectList(db.OpcionesRespuesta, "Id", "Nombre", atencionTicket.OpcionesRespuesta);
      ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", atencionTicket.RecursoAsignado);
      ViewBag.Solicitante = new SelectList(db.Usuario, "Id", "Full_Nombre", atencionTicket.Solicitante);
      ViewBag.AtendidoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", atencionTicket.AtendidoPor);
      return View(atencionTicket); }
    0 referencias | 0 excepciones
    public ActionResult Edit(int? id)
    { [HttpPost]
      [ValidateAntiForgeryToken]
      public ActionResult Edit([Bind(Include = "Id,Codigo,Nombre,RecursoAsignado,Solicitante,MedioSolicitud,MedioAtencion,EstadoAt,OpcionesRespuesta,DetalleAtencion,Observacion,AtendidoPor,FechaSolicitud,FechaAtencion")] AtencionTicket atencionTicket)
      { if (ModelState.IsValid) db.Entry(atencionTicket).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index"); }
      ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre", atencionTicket.EstadoAt);
      ViewBag.MedioAtencion = new SelectList(db.MedioAtencion, "Id", "Nombre", atencionTicket.MedioAtencion);
      ViewBag.MedioSolicitud = new SelectList(db.MedioAtencion, "Id", "Nombre", atencionTicket.MedioSolicitud);
      ViewBag.OpcionesRespuesta = new SelectList(db.OpcionesRespuesta, "Id", "Nombre", atencionTicket.OpcionesRespuesta);
      ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", atencionTicket.RecursoAsignado);
      ViewBag.Solicitante = new SelectList(db.Usuario, "Id", "Full_Nombre", atencionTicket.Solicitante);
      ViewBag.AtendidoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", atencionTicket.AtendidoPor);
      return View(atencionTicket); }
    0 referencias | 0 excepciones
    public ActionResult Delete(int? id)
    { if (id == null) return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
      AtencionTicket atencionTicket = db.AtencionTicket.Find(id);
      if (atencionTicket == null) return HttpNotFound();
      return View(atencionTicket); }
    [HttpPost, ActionName("Delete")] [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    { AtencionTicket atencionTicket = db.AtencionTicket.Find(id);
      db.AtencionTicket.Remove(atencionTicket);
      db.SaveChanges();
      return RedirectToAction("Index"); }
}
```

Activar Windows

Fuente: Elaboración Propia

Figura N° 281: Código de la vista Índice Atención de Ticket

```

<h2>Generación de Tickets</h2>
<p> @Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
  <tr><th>Codigo</th>
    <th>Asunto</th>
    <th>Fecha de Solicitud </th>
    <th>Solicitante</th>
    <th>Medio Solicitud</th>
    <th>Recurso Afectado</th>
    <th>Estado</th>
    <th>Fecha de Atención</th>
    <th>Medio Atención</th>
    <th>Opciones Respuesta </th>
    <th>Personal a cargo</th> </tr>
  @foreach (var item in Model) {
    <tr>
      <td>@Html.DisplayFor(modelItem => item.Codigo) </td>
      <td> @Html.DisplayFor(modelItem => item.Nombre) </td>
      <td>@Html.DisplayFor(modelItem => item.FechaSolicitud) </td>
      <td>@Html.DisplayFor(modelItem => item.Usuario.Full_Nombre) </td>
      <td>@Html.DisplayFor(modelItem => item.MedioAtencion1.Nombre) </td>
      <td> @Html.DisplayFor(modelItem => item.RecursoAsignado1.Nombre) </td>
      <td>@Html.DisplayFor(modelItem => item.EstadoAt1.Nombre)</td>
      <td>@Html.DisplayFor(modelItem => item.FechaAtencion) </td>
      <td>@Html.DisplayFor(modelItem => item.MedioAtencion2.Nombre)</td>
      <td>@Html.DisplayFor(modelItem => item.OpcionesRespuesta1.Nombre) </td>
      <td>@Html.DisplayFor(modelItem => item.Usuario1.Full_Nombre) </td> <td>
        @Html.ActionLink("Editar", "Edit", new { id = item.Id }) |
        @Html.ActionLink("Detalles", "Details", new { id = item.Id }) |
        @Html.ActionLink("Eliminar", "Delete", new { id = item.Id }) </td> </tr>
  }
</table>

```

Fuente: Elaboración Propia

Figura N° 282: Código de la vista Crear Atención de Ticket

```
<h2>Crear</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Atención de Solicitudes</h4><hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.RecursoAsignado, "RecursoAsignado", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("RecursoAsignado", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.RecursoAsignado, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Solicitante, "Solicitante", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Solicitante", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Solicitante, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.MedioSolicitud, "MedioSolicitud", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("MedioSolicitud", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.MedioSolicitud, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.MedioAtencion, "MedioAtencion", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("MedioAtencion", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.MedioAtencion, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.EstadoAt, "EstadoAt", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("EstadoAt", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.EstadoAt, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.OpcionesRespuesta, "OpcionesRespuesta", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("OpcionesRespuesta", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.OpcionesRespuesta, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => modelDetalleAtencion, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => modelDetalleAtencion, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => modelDetalleAtencion, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Observacion, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Observacion, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Observacion, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.AtendidoPor, "AtendidoPor", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("AtendidoPor", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.AtendidoPor, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.FechaSolicitud, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.FechaSolicitud, new { htmlAttributes = new { @class = "form-control" } })
                </div> </div>
        </div>
    }
}
```

Fuente: Elaboración Propia

Figura N° 283: Código de la vista Editar Atención de Ticket

```

<h2>Actualizar</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Atención de Solicitudes</h4> <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)
        <div class="form-group">
            @Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.RecursoAsignado, "RecursoAsignado", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("RecursoAsignado", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.RecursoAsignado, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Solicitante, "Solicitante", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Solicitante", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Solicitante, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.MedioSolicitud, "MedioSolicitud", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("MedioSolicitud", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.MedioSolicitud, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.MedioAtencion, "MedioAtencion", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("MedioAtencion", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.MedioAtencion, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.EstadoAt, "EstadoAt", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("EstadoAt", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.EstadoAt, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.OpcionesRespuesta, "OpcionesRespuesta", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("OpcionesRespuesta", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.OpcionesRespuesta, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => modelDetalleAtencion, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => modelDetalleAtencion, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => modelDetalleAtencion, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.Observacion, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Observacion, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Observacion, "", new { @class = "text-danger" }) </div> </div>
            <div class="form-group">
                @Html.LabelFor(model => model.AtendidoPor, "AtendidoPor", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">

```

Fuente: Elaboración Propia

Figura N° 284: Código de la vista Borrar Atención de Ticket

```

<h2>Delete</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div>
  <h4>Atención de Solicitudes</h4>
  <hr />
  <dl class="dl-horizontal">
    <dt> @Html.DisplayNameFor(model => model.Codigo)</dt>
    <dd> @Html.DisplayFor(model => model.Codigo) </dd>
    <dt> @Html.DisplayNameFor(model => model.Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.Nombre) </dd>
    <dt> @Html.DisplayNameFor(model => model.DetalleAtencion) </dt>
    <dd> @Html.DisplayFor(model => model.DetalleAtencion)</dd>
    <dt> @Html.DisplayNameFor(model => model.Observacion)</dt>
    <dd> @Html.DisplayFor(model => model.Observacion) </dd>
    <dt> @Html.DisplayNameFor(model => model.FechaSolicitud)</dt>
    <dd> @Html.DisplayFor(model => model.FechaSolicitud)</dd>
    <dt> @Html.DisplayNameFor(model => model.FechaAtencion)</dt>
    <dd> @Html.DisplayFor(model => model.FechaAtencion) </dd>
    <dt> @Html.DisplayNameFor(model => model.EstadoAt1.Nombre) </dt>
    <dd> @Html.DisplayFor(model => model.EstadoAt1.Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.MedioAtencion1.Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.MedioAtencion1.Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.MedioAtencion2.Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.MedioAtencion2.Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.OpcionesRespuesta1.Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.OpcionesRespuesta1.Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.RecursoAsignado1.Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.RecursoAsignado1.Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.Usuario.Full_Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.Usuario.Full_Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.Usuario1.Full_Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.Usuario1.Full_Nombre)</dd> </dl>
    @using (Html.BeginForm()) {
      @Html.AntiForgeryToken()
      <div class="form-actions no-color">
        <input type="submit" value="Eliminar" class="btn btn-default" /> |
        @Html.ActionLink("Regresar al listado", "Index")</div>
    }
  </div>

```

Fuente: Elaboración Propia

Figura N° 285: Formulario Índice Generación de Ticket

Codigo	Asunto	Fecha de Solicitud	Solicitante	Medio Solicitud	Recurso Afectado	Estado	Fecha de Atención	Medio Atención	Opciones Respuesta	Personal a cargo	
AT000001	No puede recibir correos	10/11/2018 0:00:00	Aide Minaya	En persona	PCNAV11	Cerrado	10/11/2018 0:00:00	En persona	Ninguna medida	Alex Aiquipa	Editar Detalles Eliminar
AT000002	Lentitud de procesamiento	09/11/2018 0:00:00	Fausto Merino	Teléfono	PCNAV11	Abierto	19/11/2018 0:00:00	Teléfono	Generar registro de incidencia	Alex Aiquipa	Editar Detalles Eliminar
AT000003	No prende CPU	07/11/2018 0:00:00	Maria Maldonado	En persona	PCNAV55	Cerrado	07/11/2018 0:00:00	En persona	Generar registro de incidencia	Raul Marin	Editar Detalles Eliminar
AT000004	Impresiones con rayas negras	10/11/2018 0:00:00	Maria Maldonado	Email	ImpContabilidad	Cerrado	10/11/2018 0:00:00	Email	Respuesta automatica	Alex Aiquipa	Editar Detalles Eliminar

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Figura N° 286: Formulario Crear Atención de Solicitudes

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Crear

Atención de Solicitudes

Nombre	<input type="text"/>
RecursoAsignado	PCNAV11 <input type="button" value="v"/>
Solicitante	Alex Aiquipa <input type="button" value="v"/>
MedioSolicitud	Teléfono <input type="button" value="v"/>
MedioAtencion	Teléfono <input type="button" value="v"/>
EstadoAt	Abierto <input type="button" value="v"/>
OpcionesRespuesta	Ninguna medida <input type="button" value="v"/>
DetalleAtencion	<input type="text"/>
Observacion	<input type="text"/>
AtendidoPor	Alex Aiquipa <input type="button" value="v"/>
FechaSolicitud	<input type="text"/>
FechaAtencion	<input type="text"/>

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Figura N° 287: Formulario Actualizar Atención de Solicitudes

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Actualizar

Atención de Solicitudes

Codigo	<input type="text" value="AT000001"/>
Nombre	<input type="text" value="No puede recibir correos"/>
RecursoAsignado	<input type="text" value="PCNAV11"/>
Solicitante	<input type="text" value="Aide Minaya"/>
MedioSolicitud	<input type="text" value="En persona"/>
MedioAtencion	<input type="text" value="En persona"/>
EstadoAt	<input type="text" value="Cerrado"/>
OpcionesRespuesta	<input type="text" value="Ninguna medida"/>
DetalleAtencion	<input type="text" value="Contraseña boqueado por tiempo expirac"/>
Observacion	<input type="text" value="Se procedio a restabecer e password"/>
AtendidoPor	<input type="text" value="Alex Aiquipa"/>
FechaSolicitud	<input type="text" value="10/11/2018 0:00:00"/>
FechaAtencion	<input type="text" value="10/11/2018 0:00:00"/>

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Figura N° 288: Formulario Eliminar Atención de Solicitudes

SISCONSER | Gestión Usuarios | Recursos TI | Servicios

Delete

¿Estas seguro que deseas borrarlo?
Atención de Solicitudes

Codigo AT000004
Nombre Impresiones con rayas negras
DetalleAtencion Falla causado por cilindro fotoconductor con desgaste
Observacion Se cambio cilindro compatible
FechaSolicitud 10/11/2018 0:00:00
FechaAtencion 10/11/2018 0:00:00
Nombre Cerrado
Nombre En persona
Nombre Email
Nombre Respuesta automatica
Nombre ImpContabilidad
Full_Nombre Maria Maldonado
Full_Nombre Alex Aiquipa

| [Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Tabla N° 17: TaskBoard del Sprint 3 Semana 8

TaskBoard S8				
Sistema web para el control de os servicios de área de sistemas en la compañía Naviera Natalia SAC			Fecha Inicio: 31/08/2018 Fecha final: 06/09/2018	
	Historias por cada Sprint	Pendiente	En curso	Hecho
SPRINT N° 3	Registrar e cronograma de mantenimiento			X
	Administrar el estado de los mantenimientos realizados.			X
	Registrar las atenciones solicitadas por los usuarios.			X
	Administrar y dar mantenimiento al registro de las incidencias de equipos informáticos reportados.			X
	Administrar y dar mantenimiento al registro de los problema reportados.			X
	Registrar y administrar los requerimientos solicitados para reparación de los equipos informáticos.		X	
	Monitorizar el indicador de índice de cobertura de mantenimiento		X	
	Monitorizar el indicador de cumplimiento de las atenciones realizadas.		X	

Fuente: Elaboración Propia

Requerimiento RF20

RF20: El sistema debe permitir al administrador, dar mantenimiento y administrar las incidencias reportadas.

Figura N° 289: Prototipo 1 Index Incidencia

SisConSer Gestión de Usuarios Recursos Servicios Reportes Cerrar Sesión

Incidencias
[Crear nuevo](#)

- Cronograma de mantenimiento
- Mantenimiento preventivo
- Atención de ticket
- Incidencia**
- Problema
- Requerimientos

Código	Nombre	Fecha atención	Ticket	Reportado por	Estado	Medio solicitud	Prioridad	Recurso asignado	Medio atención	Fecha cierre	Resuelto por	
												Editar Detalles Eliminar
												Editar Detalles Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 290: Prototipo 2 Crear Incidencia

Crear Incidencia

Nombre

Fecha_Atención

Número Ticket

Prioridad

Medio de Solicitud

Recurso Implicado

Reportado por

Medio de Atención

Resuelto por

Causa

Solución

Estado de Atención

Fecha Cierre

Observaciones

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 291: Prototipo 3 Actualizar Incidencia

Actualizar Incidencia

Nombre	<input type="text"/>
Fecha_Atención	<input type="text"/>
Número Ticket	<input type="text"/>
Prioridad	<input type="text"/>
Medio de Solicitud	<input type="text"/>
Recurso Implicado	<input type="text"/>
Reportado por	<input type="text"/>
Medio de Atención	<input type="text"/>
Resuelto por	<input type="text"/>
Causa	<input type="text"/>
Solución	<input type="text"/>
Estado de Atención	<input type="text"/>
Fecha Cierre	<input type="text"/>
Observaciones	<input type="text"/>

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 292: Prototipo 4 Borrar Incidencia

Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 293: Código para la creación de la tabla Incidencia

```
1 CREATE TABLE [dbo].[Incidencia] (
2     [Id] INT IDENTITY (1, 1) NOT NULL,
3     [Codigo] AS ('IN'+right('000000'+CONVERT([varchar],[Id]),(6))),
4     [Nombre] NVARCHAR (50) NOT NULL,
5     [AtencionTicket] INT NOT NULL,
6     [Prioridad] INT NOT NULL,
7     [MedioSolicitud] INT NOT NULL,
8     [MedioAtencion] INT NOT NULL,
9     [RecursoAsignado] INT NOT NULL,
10    [ReportadoPor] INT NOT NULL,
11    [ResueltoPor] INT NOT NULL,
12    [Causa] NVARCHAR (200) NOT NULL,
13    [Solucion] NVARCHAR (500) NOT NULL,
14    [EstadoAt] INT NOT NULL,
15    [Fecha_Atencion] DATETIME NOT NULL,
16    [Fecha_Cierre] DATETIME NOT NULL,
17    [Observaciones] NVARCHAR (100) NULL,
18    PRIMARY KEY CLUSTERED ([Id] ASC),
19    CONSTRAINT [FK_Incidencia_AtencionTicket] FOREIGN KEY ([AtencionTicket]) REFERENCES [dbo].[AtencionTicket] ([Id]),
20    CONSTRAINT [FK_Incidencia_Prioridad] FOREIGN KEY ([Prioridad]) REFERENCES [dbo].[Prioridad] ([Id]),
21    CONSTRAINT [FK_Incidencia_RecursoAsignado] FOREIGN KEY ([RecursoAsignado]) REFERENCES [dbo].[RecursoAsignado] ([Id]),
22    CONSTRAINT [FK_Incidencia_MedioSolicitud] FOREIGN KEY ([MedioSolicitud]) REFERENCES [dbo].[MedioAtencion] ([Id]),
23    CONSTRAINT [FK_Incidencia_MedioAtencion] FOREIGN KEY ([MedioAtencion]) REFERENCES [dbo].[MedioAtencion] ([Id]),
24    CONSTRAINT [FK_Incidencia_Usuarios] FOREIGN KEY ([ResueltoPor]) REFERENCES [dbo].[Usuario] ([Id]),
25    CONSTRAINT [FK_Incidencia_Usuario] FOREIGN KEY ([ReportadoPor]) REFERENCES [dbo].[Usuario] ([Id]),
26    CONSTRAINT [FK_Incidencia_EstadoAt] FOREIGN KEY ([EstadoAt]) REFERENCES [dbo].[EstadoAt] ([Id])
)
```

Fuente: Elaboración Propia

Figura N° 294: Código del modelo Incidencia

```
public partial class Incidencia
{
[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    0 referencias | 0 excepciones
    public Incidencia()
    { this.Problema = new HashSet<Problema>(); }
    0 referencias | 0 excepciones
    public int Id { get; set; }
    0 referencias | 0 excepciones
    public string Codigo { get; set; }
    0 referencias | 0 excepciones
    public string Nombre { get; set; }
    3 referencias | 0 excepciones
    public int AtencionTicket { get; set; }
    3 referencias | 0 excepciones
    public int Prioridad { get; set; }
    3 referencias | 0 excepciones
    public int MedioSolicitud { get; set; }
    3 referencias | 0 excepciones
    public int MedioAtencion { get; set; }
    3 referencias | 0 excepciones
    public int RecursoAsignado { get; set; }
    3 referencias | 0 excepciones
    public int ReportadoPor { get; set; }
    3 referencias | 0 excepciones
    public int ResueltoPor { get; set; }
    0 referencias | 0 excepciones
    public string Causa { get; set; }
    0 referencias | 0 excepciones
    public string Solucion { get; set; }
    3 referencias | 0 excepciones
    public int EstadoAt { get; set; }
    0 referencias | 0 excepciones
    public System.DateTime Fecha_Atencion { get; set; }
    0 referencias | 0 excepciones
    public System.DateTime Fecha_Cierre { get; set; }
    0 referencias | 0 excepciones
    public string Observaciones { get; set; }
}
```

Fuente: Elaboración Propia

Figura N° 295: Código del controlador Incidencia

```

public class IncidenciaController : Controller
{
    private ApplicationDbContext db = new ApplicationDbContext();

    public ActionResult Index()
    {
        var incidencia = db.Incidencia.Include(i => i.AtencionTicket).Include(i => i.EstadoAt1).Include(i => i.MedioAtencion).Include(i => i.MedioAtencion2).Include(i => i.Prioridad1).Include(i => i.RecursoAsignado1).Include(i => i.Usuario).Include(i => i.Usuario1);
        return View(incidencia.ToList());
    }

    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return RedirectToAction("Index");
        }
        [ValidateAntiForgeryToken]
        [HttpPost]
        public ActionResult Create([Bind(Include = "Id,Codigo,Nombre,AtencionTicket,Prioridad,MedioSolicitud,MedioAtencion,RecursoAsignado,ReportadoPor,ResueltoPor,Causa,Solucion,EstadoAt,Fecha_Atencion,Fecha_Cierre,Observaciones")] Incidencia incidencia)
        {
            if (ModelState.IsValid)
            {
                db.Incidencia.Add(incidencia);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.AtencionTicket = new SelectList(db.AtencionTicket, "Id", "Codigo", incidencia.AtencionTicket);
            ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre", incidencia.EstadoAt);
            ViewBag.MedioAtencion = new SelectList(db.MedioAtencion, "Id", "Nombre", incidencia.MedioAtencion);
            ViewBag.MedioSolicitud = new SelectList(db.MedioSolicitud, "Id", "Nombre", incidencia.MedioSolicitud);
            ViewBag.Prioridad = new SelectList(db.Prioridad, "Id", "Nombre", incidencia.Prioridad);
            ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", incidencia.RecursoAsignado);
            ViewBag.ReportadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", incidencia.ReportadoPor);
            ViewBag.ResueltoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", incidencia.ResueltoPor);
            return View(incidencia);
        }

        [ValidateAntiForgeryToken]
        [HttpPost]
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return RedirectToAction("Index");
            }
            [ValidateAntiForgeryToken]
            [HttpPost]
            public ActionResult Edit([Bind(Include = "Id,Codigo,Nombre,AtencionTicket,Prioridad,MedioSolicitud,MedioAtencion,RecursoAsignado,ReportadoPor,ResueltoPor,Causa,Solucion,EstadoAt,Fecha_Atencion,Fecha_Cierre,Observaciones")] Incidencia incidencia)
            {
                if (ModelState.IsValid)
                {
                    db.Entry(incidencia).State = EntityState.Modified;
                    db.SaveChanges();
                    return RedirectToAction("Index");
                }
                ViewBag.AtencionTicket = new SelectList(db.AtencionTicket, "Id", "Codigo", incidencia.AtencionTicket);
                ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre", incidencia.EstadoAt);
                ViewBag.MedioAtencion = new SelectList(db.MedioAtencion, "Id", "Nombre", incidencia.MedioAtencion);
                ViewBag.MedioSolicitud = new SelectList(db.MedioSolicitud, "Id", "Nombre", incidencia.MedioSolicitud);
                ViewBag.Prioridad = new SelectList(db.Prioridad, "Id", "Nombre", incidencia.Prioridad);
                ViewBag.RecursoAsignado = new SelectList(db.RecursoAsignado, "Id", "Nombre", incidencia.RecursoAsignado);
                ViewBag.ReportadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", incidencia.ReportadoPor);
                ViewBag.ResueltoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", incidencia.ResueltoPor);
                return View(incidencia);
            }

            public ActionResult Delete(int? id)
            {
                if (id == null)
                {
                    return HttpNotFound();
                }
                Incidencia incidencia = db.Incidencia.Find(id);
                if (incidencia == null)
                {
                    return HttpNotFound();
                }
                db.Incidencia.Remove(incidencia);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            public ActionResult DeleteConfirmed(int id)
            {
                Incidencia incidencia = db.Incidencia.Find(id);
                db.Incidencia.Remove(incidencia);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
        }
    }
}

```

Fuente: Elaboración Propia

Figura N° 296: Código de la vista Index Incidencia

```

<h2>Incidencia</h2>
<p>@Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
    <tr>
        <th>Codigo </th>
        <th> @Html.DisplayNameFor(model => model.Nombre) </th>
        <th>Fecha de Atención </th>
        <th>Fecha de Cierre </th>
        <th>Ticket</th>
        <th> Estado</th>
        <th> Medio de Solicitud </th>
        <th> Medio de Atención</th>
        <th>Prioridad</th>
        <th>Recurso Implicado </th>
        <th> Reportado por </th>
        <th> Resuelto por</th> </tr>
    @foreach (var item in Model) {
        <tr> <td> @Html.DisplayFor(modelItem => item.Codigo) </td>
        <td> @Html.DisplayFor(modelItem => item.Nombre) </td>
        <td>@Html.DisplayFor(modelItem => item.Fecha_Atencion) </td>
        <td> @Html.DisplayFor(modelItem => item.Fecha_Cierre) </td>
        <td> @Html.DisplayFor(modelItem => item.AtencionTicket1.Codigo) </td>
        <td> @Html.DisplayFor(modelItem => item.EstadoAt1.Nombre)</td>
        <td> @Html.DisplayFor(modelItem => item.MedioAtencion1.Nombre) </td>
        <td> @Html.DisplayFor(modelItem => item.MedioAtencion2.Nombre) </td>
        <td> @Html.DisplayFor(modelItem => item.Prioridad1.Nombre) </td>
        <td> @Html.DisplayFor(modelItem => item.RecursoAsignado1.Nombre) </td>
        <td> @Html.DisplayFor(modelItem => item.Usuario.Full_Nombre) </td>
        <td>@Html.DisplayFor(modelItem => item.Usuario1.Full_Nombre) </td>
        <td> @Html.ActionLink("Editar", "Edit", new { id=item.Id }) |
        @Html.ActionLink("Detalles", "Details", new { id=item.Id }) |
        @Html.ActionLink("Eliminar", "Delete", new { id=item.Id }) </td> </tr>
    }
</table>

```

Fuente: Elaboración Propia

Figura N° 297: Código de la vista Crear Incidencia

```

<h2>Crear</h2>
@using (Html.BeginForm())
{ @Html.AntiForgeryToken()
  <div class="form-horizontal">
    <h4>Incidencia</h4><hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
      @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.AtencionTicket, "AtencionTicket", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("AtencionTicket", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.AtencionTicket, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.Prioridad, "Prioridad", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("Prioridad", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Prioridad, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.MedioSolicitud, "MedioSolicitud", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("MedioSolicitud", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.MedioSolicitud, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.MedioAtencion, "MedioAtencion", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("MedioAtencion", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.MedioAtencion, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.RecursoAsignado, "RecursoAsignado", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("RecursoAsignado", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.RecursoAsignado, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.ReportadoPor, "ReportadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("ReportadoPor", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.ReportadoPor, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.ResueltoPor, "ResueltoPor", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("ResueltoPor", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.ResueltoPor, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.Causa, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Causa, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Causa, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.Solucion, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Solucion, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Solucion, "", new { @class = "text-danger" }) </div> </div>
    <div class="form-group">
      @Html.LabelFor(model => model.EstadoAt, "EstadoAt", htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.DropDownList("EstadoAt", null, htmlAttributes: new { @class = "form-control" })
      </div>
    </div>
  </div>
}

```

Fuente: Elaboración Propia

Figura N° 298: Código de la vista Editar Incidencia

```
<h2>Actualizar</h2>
@using (Html.BeginForm())
{@Html.AntiForgeryToken()
<div class="form-horizontal">
<h4>Incidencia</h4>@Html.ValidationSummary(true, "", new { @class = "text-danger" })
@Html.HiddenFor(model => model.Id)
<div class="form-group">
@Html.LabelFor(model => model.Codigo, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.Codigo, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.Codigo, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.AtencionTicket, "AtencionTicket", htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.DropDownList("AtencionTicket", null, htmlAttributes: new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.AtencionTicket, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.Prioridad, "Prioridad", htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.DropDownList("Prioridad", null, htmlAttributes: new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.Prioridad, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.MedioSolicitud, "MedioSolicitud", htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.DropDownList("MedioSolicitud", null, htmlAttributes: new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.MedioSolicitud, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.MedioAtencion, "MedioAtencion", htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.DropDownList("MedioAtencion", null, htmlAttributes: new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.MedioAtencion, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.RecursoAsignado, "RecursoAsignado", htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.DropDownList("RecursoAsignado", null, htmlAttributes: new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.RecursoAsignado, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.ReportadoPor, "ReportadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.DropDownList("ReportadoPor", null, htmlAttributes: new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.ReportadoPor, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.ResueltoPor, "ResueltoPor", htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.DropDownList("ResueltoPor", null, htmlAttributes: new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.ResueltoPor, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.Causa, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.Causa, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.Causa, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
@Html.LabelFor(model => model.Solucion, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.Solucion, new { htmlAttributes = new { @class = "form-control" } })

```

Fuente: Elaboración Propia

Figura N° 299: Código de la vista Eliminar Incidencia

```

<h2>Delete</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div>
  <h4>Incidencia</h4> <hr />
  <dl class="dl-horizontal">
    <dt>@Html.DisplayNameFor(model => model.Codigo) </dt>
    <dd> @Html.DisplayFor(model => model.Codigo)</dd>
    <dt> @Html.DisplayNameFor(model => model.Nombre) </dt>
    <dd>@Html.DisplayFor(model => model.Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.Causa)</dt>
    <dd> @Html.DisplayFor(model => model.Causa)</dd>
    <dt> @Html.DisplayNameFor(model => model.Solucion)</dt>
    <dd>@Html.DisplayFor(model => model.Solucion)</dd>
    <dt>@Html.DisplayNameFor(model => model.Fecha_Atencion)</dt>
    <dd> @Html.DisplayFor(model => model.Fecha_Atencion) </dd>
    <dt> @Html.DisplayNameFor(model => model.Fecha_Cierre)</dt>
    <dd>@Html.DisplayFor(model => model.Fecha_Cierre)</dd>
    <dt>@Html.DisplayNameFor(model => model.Observaciones)</dt>
    <dd> @Html.DisplayFor(model => model.Observaciones) </dd>
    <dt> @Html.DisplayNameFor(model => model.AtencionTicket1.Codigo) </dt>
    <dd> @Html.DisplayFor(model => model.AtencionTicket1.Codigo) </dd>
    <dt>@Html.DisplayNameFor(model => model.EstadoAt1.Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.EstadoAt1.Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.MedioAtencion1.Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.MedioAtencion1.Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.MedioAtencion2.Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.MedioAtencion2.Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.Prioridad1.Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.Prioridad1.Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.RecursoAsignado1.Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.RecursoAsignado1.Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.Usuario.Full_Nombre) </dt>
    <dd>@Html.DisplayFor(model => model.Usuario.Full_Nombre) </dd>
    <dt>@Html.DisplayNameFor(model => model.Usuario1.Full_Nombre) </dt>
    <dd>@Html.DisplayFor(model => model.Usuario1.Full_Nombre)</dd> </dl>
    @using (Html.BeginForm()) {
      @Html.AntiForgeryToken()
      <div class="form-actions no-color">
        <input type="submit" value="Eliminar" class="btn btn-default" /> |
        @Html.ActionLink("Regresar al listado", "Index")
      </div>
    }
  </div>

```

Fuente: Elaboración Propia

Figura N° 300: Formulario Index Incidencia

Codigo	Nombre	Fecha de Atención	Ticket Reportado por	Estado	Medio de Solicitud	Prioridad	Recurso Implicado	Medio de Atención	Fecha de Cierre	Resuelto por
--------	--------	-------------------	----------------------	--------	--------------------	-----------	-------------------	-------------------	-----------------	--------------

Fuente: Elaboración Propia

Figura N° 301: Formulario Registrar Incidencia

Nombre

Fecha_Atencion

Numero de Ticket: AT000001

Prioridad: Baja

Medio de Solicitud: Teléfono

Recurso Implicado: PCNAV11

Reportado Por: Alex Aiquipa

Medio de Atención: Teléfono

Resuelto Por: Alex Aiquipa

Causa

Solucion

EstadoAt: Abierto

Fecha_Cierre

Observaciones

Crear

Fuente: Elaboración Propia

Figura N° 302: Formulario Actualizar Incidencia

SISCONSER 👤 Gestión Usuarios ▾ 👤 Recursos TI ▾ 👤 Servicios ▾

Actualizar

Incidencia

Codigo	<input type="text" value="IN000002"/>
Nombre	<input type="text" value="Error de impresión"/>
AtencionTicket	<input type="text" value="AT000001"/> ▾
Prioridad	<input type="text" value="Baja"/> ▾
MedioSolicitud	<input type="text" value="Teléfono"/> ▾
MedioAtencion	<input type="text" value="En persona"/> ▾
RecursoAsignado	<input type="text" value="PCNAV07"/> ▾
ReportadoPor	<input type="text" value="Maria Maldonado"/> ▾
ResueltoPor	<input type="text" value="Alex Aiquipa"/> ▾
Causa	<input type="text" value="Cambio de dirección ip de impresora"/>
Solucion	<input type="text" value="Se re configuro equipo"/>
EstadoAt	<input type="text" value="Cerrado"/> ▾
Fecha_Atencion	<input type="text" value="11/08/2018 0:00:00"/>
Fecha_Cierre	<input type="text" value="11/08/2018 0:00:00"/>
Observaciones	<input type="text"/>

Fuente: Elaboración Propia

Figura N° 303: Formulario Eliminar Incidencia

Delete

¿Estas seguro que deseas borrarlo?

Incidencia

Codigo IN000001
Nombre Error de impresión
Causa Cambio de dirección ip de impresora
Solucion Se re configuro equipo
Fecha_Atencion 11/08/2018 0:00:00
Fecha_Cierre 11/08/2018 0:00:00

Observaciones

Codigo AT000001
Nombre Cerrado
Nombre En persona
Nombre Teléfono
Nombre Baja
Nombre PCNAV07
Full_Nombre Maria Maldonado
Full_Nombre Alex Aiquipa

| [Regresar al listado](#)

Fuente: Elaboración Propia

Requerimiento RF21

RF21: El sistema debe permitir al usuario autenticado, registrar y actualizar los problemas suscitados.

Figura N° 304: Prototipo 1 Index Problema

SisConSer | Gestión de Usuarios | Recursos | Servicios | Reportes | Cerrar Sesión

Problema

[Crear nuevo](#)

Código	Reportado por	Medio solicitud	Solucionado por	Medio atención	Causa	Prioridad	Fecha cierre	
								Editar Detalles Eliminar
								Editar Detalles Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 305: Prototipo 2 Registrar Problema

Registrar Problema

Incidencia	<input type="text"/>
ReportadoPor	<input type="text"/>
Medio de Solicitud	<input type="text"/>
Medio de Atención	<input type="text"/>
Resuelto por	<input type="text"/>
Causa	<input type="text"/>
Solución temporal	<input type="text"/>
Solución definitiva	<input type="text"/>
Observaciones	<input type="text"/>
EstadoAt	<input type="text"/>
Impacto	<input type="text"/>
Fecha Cierre	<input type="text"/>

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 306: Prototipo 3 Editar Problema

Editar Problema

Incidencia	<input type="text"/>
ReportadoPor	<input type="text"/>
Medio de Solicitud	<input type="text"/>
Medio de Atención	<input type="text"/>
Resuelto por	<input type="text"/>
Causa	<input type="text"/>
Solución temporal	<input type="text"/>
Solución definitiva	<input type="text"/>
Observaciones	<input type="text"/>
EstadoAt	<input type="text"/>
Impacto	<input type="text"/>
Fecha Cierre	<input type="text"/>

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 307: Prototipo 4 Eliminar Problema

Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Implementación

Figura N° 308: Código para la creación de la tabla Problema

```
1 CREATE TABLE [dbo].[Problema] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Incidencia] INT NOT NULL,  
4     [ReportadoPor] INT NOT NULL,  
5     [MedioSolicitud] INT NOT NULL,  
6     [MedioAtencion] INT NOT NULL,  
7     [ResueltoPor] INT NOT NULL,  
8     [Causa] NVARCHAR (200) NOT NULL,  
9     [SolucionTemporal] NVARCHAR (500) NOT NULL,  
10    [SolucionDefinitiva] NVARCHAR (500) NOT NULL,  
11    [Observaciones] NVARCHAR (500) NOT NULL,  
12    [EstadoAt] INT NOT NULL,  
13    [Impacto] INT NOT NULL,  
14    [Fecha_cierre] DATETIME NOT NULL,  
15    PRIMARY KEY CLUSTERED ([Id] ASC),  
16    CONSTRAINT [FK_Problema_Incidencia] FOREIGN KEY ([Incidencia]) REFERENCES [dbo].[Incidencia] ([Id]),  
17    CONSTRAINT [FK_Problema_Usuario] FOREIGN KEY ([ReportadoPor]) REFERENCES [dbo].[Usuario] ([Id]),  
18    CONSTRAINT [FK_Problema_MedioSolicitud] FOREIGN KEY ([MedioSolicitud]) REFERENCES [dbo].[MedioAtencion] ([Id]),  
19    CONSTRAINT [FK_Problema_MedioAtencion] FOREIGN KEY ([MedioAtencion]) REFERENCES [dbo].[MedioAtencion] ([Id]),  
20    CONSTRAINT [FK_Problema_EstadoAt] FOREIGN KEY ([EstadoAt]) REFERENCES [dbo].[EstadoAt] ([Id]),  
21    CONSTRAINT [FK_Problema_UsuarioS] FOREIGN KEY ([ResueltoPor]) REFERENCES [dbo].[Usuario] ([Id]),  
22    CONSTRAINT [FK_Problema_Impacto] FOREIGN KEY ([Impacto]) REFERENCES [dbo].[Impacto] ([Id])  
23 );
```

Fuente: Elaboración Propia

Figura N° 309: Código del modelo Problema

```
public partial class Problema
{
    public int Id { get; set; }

    public int Incidencia { get; set; }
    -referencias
    public int ReportadoPor { get; set; }
    -referencias
    public int MedioSolicitud { get; set; }
    -referencias
    public int MedioAtencion { get; set; }
    3 referencias | 0 excepciones
    public int ResueltoPor { get; set; }
    0 referencias | 0 excepciones
    public string Causa { get; set; }
    0 referencias | 0 excepciones
    public string SolucionTemporal { get; set; }
    0 referencias | 0 excepciones
    public string SolucionDefinitiva { get; set; }
    0 referencias | 0 excepciones
    public string Observaciones { get; set; }
    3 referencias | 0 excepciones
    public int EstadoAt { get; set; }
    3 referencias | 0 excepciones
    public int Impacto { get; set; }
    0 referencias | 0 excepciones
    public System.DateTime Fecha_cierre { get; set; }
    3 referencias | 0 excepciones
    public int Requerimiento { get; set; }
}
```

Fuente: Elaboración Propia

Figura N° 310: Código del controlador Problema

```
namespace WebContServ.Controllers
{
    [Authorize]
    [Route("")]
    public class ProblemasController : Controller
    {
        private WebContServEntities db = new WebContServEntities();

        public ActionResult Index()
        {
            var problema = db.Problema.Include(p => p.EstadoAt).Include(p => p.Impacto).Include(p => p.Incidencia).Include(p => p.MedioAtencion).Include(p => p.MedioSolicitud).Include(p => p.Requerimiento).Include(p => p.ReportadoPor).Include(p => p.ResueltoPor).Include(p => p.Causa).Include(p => p.SolucionTemporal).Include(p => p.SolucionDefinitiva).Include(p => p.Observaciones).Include(p => p.Fecha_cierre).Include(p => p.Requerimiento);
            return View(problema.ToList());
        }

        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return RedirectToAction("Index");
            }
            var problema = db.Problema.Include(p => p.EstadoAt).Include(p => p.Impacto).Include(p => p.Incidencia).Include(p => p.MedioAtencion).Include(p => p.MedioSolicitud).Include(p => p.Requerimiento).Include(p => p.ReportadoPor).Include(p => p.ResueltoPor).Include(p => p.Causa).Include(p => p.SolucionTemporal).Include(p => p.SolucionDefinitiva).Include(p => p.Observaciones).Include(p => p.Fecha_cierre).Include(p => p.Requerimiento);
            return View(problema);
        }

        public ActionResult Create()
        {
            ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre", problema.EstadoAt);
            ViewBag.Impacto = new SelectList(db.Impacto, "Id", "Nombre", problema.Impacto);
            ViewBag.Incidencia = new SelectList(db.Incidencia, "Id", "Codigo", problema.Incidencia);
            ViewBag.MedioAtencion = new SelectList(db.MedioAtencion, "Id", "Nombre", problema.MedioAtencion);
            ViewBag.MedioSolicitud = new SelectList(db.MedioSolicitud, "Id", "Nombre", problema.MedioSolicitud);
            ViewBag.Requerimiento = new SelectList(db.Requerimiento, "Id", "Codigo", problema.Requerimiento);
            ViewBag.ReportadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", problema.ReportadoPor);
            ViewBag.ResueltoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", problema.ResueltoPor);
            return View(problema);
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include = "Id,Incidencia,ReportadoPor,MedioSolicitud,MedioAtencion,ResueltoPor,Causa,SolucionTemporal,SolucionDefinitiva,Observaciones,EstadoAt,Impacto,Fecha_cierre,Requerimiento")] Problema problema)
        {
            if (ModelState.IsValid)
            {
                db.Problema.Add(problema);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre", problema.EstadoAt);
            ViewBag.Impacto = new SelectList(db.Impacto, "Id", "Nombre", problema.Impacto);
            ViewBag.Incidencia = new SelectList(db.Incidencia, "Id", "Codigo", problema.Incidencia);
            ViewBag.MedioAtencion = new SelectList(db.MedioAtencion, "Id", "Nombre", problema.MedioAtencion);
            ViewBag.MedioSolicitud = new SelectList(db.MedioSolicitud, "Id", "Nombre", problema.MedioSolicitud);
            ViewBag.Requerimiento = new SelectList(db.Requerimiento, "Id", "Codigo", problema.Requerimiento);
            ViewBag.ReportadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", problema.ReportadoPor);
            ViewBag.ResueltoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", problema.ResueltoPor);
            return View(problema);
        }

        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return RedirectToAction("Index");
            }
            var problema = db.Problema.Include(p => p.EstadoAt).Include(p => p.Impacto).Include(p => p.Incidencia).Include(p => p.MedioAtencion).Include(p => p.MedioSolicitud).Include(p => p.Requerimiento).Include(p => p.ReportadoPor).Include(p => p.ResueltoPor).Include(p => p.Causa).Include(p => p.SolucionTemporal).Include(p => p.SolucionDefinitiva).Include(p => p.Observaciones).Include(p => p.Fecha_cierre).Include(p => p.Requerimiento);
            return View(problema);
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include = "Id,Incidencia,ReportadoPor,MedioSolicitud,MedioAtencion,ResueltoPor,Causa,SolucionTemporal,SolucionDefinitiva,Observaciones,EstadoAt,Impacto,Fecha_cierre,Requerimiento")] Problema problema)
        {
            if (ModelState.IsValid)
            {
                db.Entry(problema).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.EstadoAt = new SelectList(db.EstadoAt, "Id", "Nombre", problema.EstadoAt);
            ViewBag.Impacto = new SelectList(db.Impacto, "Id", "Nombre", problema.Impacto);
            ViewBag.Incidencia = new SelectList(db.Incidencia, "Id", "Codigo", problema.Incidencia);
            ViewBag.MedioAtencion = new SelectList(db.MedioAtencion, "Id", "Nombre", problema.MedioAtencion);
            ViewBag.MedioSolicitud = new SelectList(db.MedioSolicitud, "Id", "Nombre", problema.MedioSolicitud);
            ViewBag.Requerimiento = new SelectList(db.Requerimiento, "Id", "Codigo", problema.Requerimiento);
            ViewBag.ReportadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", problema.ReportadoPor);
            ViewBag.ResueltoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", problema.ResueltoPor);
            return View(problema);
        }

        public ActionResult Delete(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Problema problema = db.Problema.Find(id);
            if (problema == null)
            {
                return HttpNotFound();
            }
            return View(problema);
        }

        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public ActionResult DeleteConfirmed(int id)
        {
            Problema problema = db.Problema.Find(id);
            db.Problema.Remove(problema);
            db.SaveChanges();
        }
    }
}
```

Fuente: Elaboración Propia

Figura N° 311: Código de la vista Index Problema

```
<h2>Problemas</h2>
<p>@Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
  <tr>
    <th> Fecha de cierre</th>
    <th>Codigo de Incidencia</th>
    <th> Estado</th>
    <th> Impacto</th>
    <th> Reportado por</th>
    <th> Medio de Solicitud</th>
    <th> Medio de Atencion</th>
    <th> Requerimiento Asociado</th>
    <th>Solucionado por</th>
  </tr>
  @foreach (var item in Model) {
    <tr>
      <td>@Html.DisplayFor(modelItem => item.Fecha_cierre)</td>
      <td> @Html.DisplayFor(modelItem => item.Incidencia1.Codigo)</td>
      <td> @Html.DisplayFor(modelItem => item.EstadoAt1.Nombre) </td>
      <td> @Html.DisplayFor(modelItem => item.Impacto1.Nombre)</td>
      <td> @Html.DisplayFor(modelItem => item.Usuario.Full_Nombre)</td>
      <td> @Html.DisplayFor(modelItem => item.MedioAtencion1.Nombre) </td>
      <td> @Html.DisplayFor(modelItem => item.MedioAtencion2.Nombre)</td>
      <td>@Html.DisplayFor(modelItem => item.Requerimiento1.codigo) </td>
      <td> @Html.DisplayFor(modelItem => item.Usuario1.Full_Nombre) </td>
      <td>
        @Html.ActionLink("Editar", "Edit", new { id = item.Id })
        @Html.ActionLink("Detalles", "Details", new { id = item.Id })
        @Html.ActionLink("Eliminar", "Delete", new { id = item.Id })
      </td>
    </tr>
  }
</table>
```

Fuente: Elaboración Propia

Figura N° 312: Código de la vista Crear Problema

```

<h2>Crear</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Problema</h4><hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Incidencia, "Incidencia", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Incidencia", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.Incidencia, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.ReportadoPor, "ReportadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("ReportadoPor", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.ReportadoPor, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.MedioSolicitud, "MedioSolicitud", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("MedioSolicitud", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.MedioSolicitud, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.MedioAtencion, "MedioAtencion", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("MedioAtencion", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.MedioAtencion, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.ResueltoPor, "ResueltoPor", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("ResueltoPor", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.ResueltoPor, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Causa, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Causa, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Causa, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.SolucionTemporal, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.SolucionTemporal, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.SolucionTemporal, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.SolucionDefinitiva, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.SolucionDefinitiva, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.SolucionDefinitiva, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Observaciones, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Observaciones, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Observaciones, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.EstadoAt, "EstadoAt", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("EstadoAt", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.EstadoAt, "", new { @class = "text-danger" })</div></div>
        <div class="form-group">
            @Html.LabelFor(model => model.Impacto, "Impacto", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Impacto", null, htmlAttributes: new { @class = "form-control" })
            </div>
        </div>
    </div>
}

```

Fuente: Elaboración Propia

Figura N° 313: Código de la vista Actualizar Problema

```

<h4>Problema</h4><hr />
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
@Html.HiddenFor(model => model.Id)
<div class="form-group">
    @Html.LabelFor(model => model.Incidencia, "Incidencia", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("Incidencia", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Incidencia, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.ReportadoPor, "ReportadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("ReportadoPor", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.ReportadoPor, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.MedioSolicitud, "MedioSolicitud", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("MedioSolicitud", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.MedioSolicitud, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.MedioAtencion, "MedioAtencion", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("MedioAtencion", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.MedioAtencion, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.ResueltoPor, "ResueltoPor", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("ResueltoPor", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.ResueltoPor, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Causa, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Causa, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Causa, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.SolucionTemporal, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.SolucionTemporal, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.SolucionTemporal, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.SolucionDefinitiva, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.SolucionDefinitiva, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.SolucionDefinitiva, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Observaciones, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Observaciones, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Observaciones, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.EstadoAt, "EstadoAt", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("EstadoAt", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.EstadoAt, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Impacto, "Impacto", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("Impacto", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.Impacto, "", new { @class = "text-danger" })</div></div>
<div class="form-group">
    @Html.LabelFor(model => model.Fecha cierre, htmlAttributes: new { @class = "control-label col-md-2" })

```

Fuente: Elaboración Propia

Figura N° 314: Código de la vista Eliminar Problema

```

<h2>Eliminar</h2>
<h3>¿Estas seguro que deseas borrarlo?</h3>
<div><h4>Problema</h4><hr />
  <dl class="dl-horizontal">
    <dt> @Html.DisplayNameFor(model => model.Causa)</dt>
    <dd>@Html.DisplayFor(model => model.Causa)</dd>
    <dt>@Html.DisplayNameFor(model => model.SolucionTemporal)</dt>
    <dd>@Html.DisplayFor(model => model.SolucionTemporal)</dd>
    <dt>@Html.DisplayNameFor(model => model.SolucionDefinitiva) </dt>
    <dd>@Html.DisplayFor(model => model.SolucionDefinitiva)</dd>
    <dt> @Html.DisplayNameFor(model => model.Observaciones) </dt>
    <dd>@Html.DisplayFor(model => model.Observaciones)</dd>
    <dt>@Html.DisplayNameFor(model => model.Fecha_cierre)</dt>
    <dd>@Html.DisplayFor(model => model.Fecha_cierre)</dd>
    <dt>@Html.DisplayNameFor(model => model.EstadoAt1.Nombre) </dt>
    <dd>@Html.DisplayFor(model => model.EstadoAt1.Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.Impacto1.Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.Impacto1.Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.Incidencia1.Codigo)</dt>
    <dd> @Html.DisplayFor(model => model.Incidencia1.Codigo) </dd>
    <dt>@Html.DisplayNameFor(model => model.MedioAtencion1.Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.MedioAtencion1.Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.MedioAtencion2.Nombre) </dt>
    <dd> @Html.DisplayFor(model => model.MedioAtencion2.Nombre)</dd>
    <dt>@Html.DisplayNameFor(model => model.Requerimiento1.codigo)</dt>
    <dd>@Html.DisplayFor(model => model.Requerimiento1.codigo)</dd>
    <dt>@Html.DisplayNameFor(model => model.Usuario.Full_Nombre)</dt>
    <dd>@Html.DisplayFor(model => model.Usuario.Full_Nombre)</dd>
    <dt> @Html.DisplayNameFor(model => model.Usuario1.Full_Nombre)</dt>
    <dd> @Html.DisplayFor(model => model.Usuario1.Full_Nombre)</dd></dl>
    @using (Html.BeginForm()) { @Html.AntiForgeryToken()
    <div class="form-actions no-color">
      <input type="submit" value="Eliminar" class="btn btn-default" /> |
      @Html.ActionLink("Regresar al listado", "Index")</div> }
  </div>

```

Fuente: Elaboración Propia

Figura N° 315: Formulario Index Problema

Causa	SolucionTemporal	SolucionDefinitiva	Observaciones	Fecha_cierre	Nombre	Nombre	Codigo	Nombre	Nombre	Reportado por	Solucionado por	
No prende la computadora	Cambio de fuente	instalación de upc	Equipo en buen estado	12/08/2018 0:00:00	Cerrado	Media	IN000001	En persona	Teléfono	Alex Aiquipa	Alex Aiquipa	Editar Detalles Eliminar

Fuente: Elaboración Propia

Figura N° 316: Formulario Crear Problema

Crear Problema

Incidencia

ReportadoPor

MedioSolicitud

MedioAtencion

ResueltoPor

Causa

SolucionTemporal

SolucionDefinitiva

Observaciones

EstadoAt

Impacto

Fecha_cierre

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 317: Formulario Editar Problema

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Editar Problema

Incidencia	<input type="text" value="IN000001"/>
ReportadoPor	<input type="text" value="Alex Aiquipa"/>
MedioSolicitud	<input type="text" value="Teléfono"/>
MedioAtencion	<input type="text" value="En persona"/>
ResueltoPor	<input type="text" value="Alex Aiquipa"/>
Causa	<input type="text" value="No prende la computadora"/>
SolucionTemporal	<input type="text" value="Cambio de fuente"/>
SolucionDefinitiva	<input type="text" value="instalación de upc"/>
Observaciones	<input type="text" value="Equipo en buen estado"/>
EstadoAt	<input type="text" value="Cerrado"/>
Impacto	<input type="text" value="Media"/>
Fecha_cierre	<input type="text" value="12/08/2018 0:00:00"/>

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 318: Formulario Eliminar Problema

SISCONSER Gestión Usuarios Recursos TI

Eliminar

¿Estas seguro que deseas borrarlo?
Problema

Causa No prende la computadora
SolucionTemporal Cambio de fuente
SolucionDefinitiva instalación de upc
Observaciones Equipo en buen estado
Fecha_cierre 12/08/2018 0:00:00
Nombre Cerrado
Nombre Media
Codigo IN000001
Nombre En persona
Nombre Teléfono
Reportado por Alex Aiquipa
Solucionado por Alex Aiquipa

| [Regresar al listado](#)

Fuente: Elaboración Propia

Tabla N° 18: TaskBoard del Sprint 3 Semana 9

TaskBoard S9				
Sistema web para el control de os servicios de área de sistemas en la compañía Naviera Natalia SAC			Fecha Inicio: 07/09/2018 Fecha final: 14/09/2018	
	Historias por cada Sprint	Pendiente	En curso	Hecho
SPRINT N° 3	Registrar e cronograma de mantenimiento			X
	Administrar el estado de los mantenimientos realizados.			X
	Registrar las atenciones solicitadas por los usuarios.			X
	Administrar y dar mantenimiento al registro de las incidencias de equipos informáticos reportados.			X
	Administrar y dar mantenimiento al registro de los problema reportados.			X
	Registrar y administrar los requerimientos solicitados para reparación de los equipos informáticos.			X
	Monitorizar el indicador de índice de cobertura de mantenimiento			X
	Monitorizar el indicador de cumplimiento de las atenciones realizadas.			X

Fuente: Elaboración Propia

Requerimiento RF22

RF22: El sistema debe permitir al usuario autenticado, registrar y actualizar los requerimientos de insumos y/o componentes solicitados para reparación de los equipos informáticos de la compañía.

Diseño

Figura N° 319: Prototipo 1 Índice Requerimientos

SisConSer | Gestión de Usuarios | Recursos | Servicios | Reportes | Cerrar Sesión

Servicios desplegado:

- Cronograma de mantenimiento
- Mantenimeinto preventivo
- Atención de ticket
- Incidencia
- Problema
- Requerimientos

Requerimientos

[Crear nuevo](#)

Código	Descripción	Fecha Solicitud	Fecha Entrega	Cantidad	Detalles	Razon Social	Equipo	Incidencia	Prioridad				
											Editar	Detalles	Eliminar
											Editar	Detalles	Eliminar

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 320: Prototipo 2 Crear Requerimientos

Crear Requerimiento

Nombre:

Solicitado por:

Urgencia:

Componente:

Equipo:

Incidencia:

Fecha Solicitud:

Fecha Entrega:

Cantidad:

Centro costos:

Detalles:

Estado de Atención:

Fecha Cierre:

Observaciones:

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 321: Prototipo 3 Editar Requerimientos

Editar Requerimiento

Nombre	<input type="text" value="text"/>
Solicitado por	<input type="text" value="text"/>
Urgencia	<input type="text" value="text"/>
Componente	<input type="text" value="text"/>
Equipo	<input type="text" value="text"/>
Incidencia	<input type="text" value="text"/>
Fecha Solicitud	<input type="text" value="text"/>
Fecha Entrega	<input type="text" value="text"/>
Cantidad	<input type="text" value="text"/>
Centro costos	<input type="text" value="text"/>
Detalles	<input type="text" value="text"/>
Estado de Atención	<input type="text" value="text"/>
Fecha Cierre	<input type="text" value="text"/>
Observaciones	<input type="text" value="text"/>

© 2018 - Sistema Web para el Control de Servicios del Area de Sistemas

Fuente: Elaboración Propia

Figura N° 322: Prototipo 4 Eliminar Requerimientos

Borrar

¿ Estas Seguro que deseas borrarlos ?

Codigo : Label

 [Regresar al listado](#)

Fuente: Elaboración Propia

Figura N° 323: Código para la creación de la tabla Requerimiento

```

1 CREATE TABLE [dbo].[Requerimiento] (
2     [Id] INT IDENTITY (1, 1) NOT NULL,
3     [codigo] AS ('RE'+right('00000'+CONVERT([varchar],[Id]),(6))),
4     [Nombre] NVARCHAR (50) NOT NULL,
5     [SolicitadoPor] INT NOT NULL,
6     [Urgencia] INT NOT NULL,
7     [Componente] INT NOT NULL,
8     [Equipo] INT NULL,
9     [Incidencia] INT NULL,
10    [Fecha_solicitud] DATETIME NOT NULL,
11    [Fecha_entrega] DATETIME NOT NULL,
12    [Cantidad] INT NOT NULL,
13    [Centro_costos] INT NOT NULL,
14    [Detalles] NVARCHAR (100) NOT NULL,
15    PRIMARY KEY CLUSTERED ([Id] ASC),
16    CONSTRAINT [FK_Requerimiento_Componente] FOREIGN KEY ([Componente]) REFERENCES [dbo].[Componente] ([Id]),
17    CONSTRAINT [FK_Requerimiento_Usuario] FOREIGN KEY ([SolicitadoPor]) REFERENCES [dbo].[Usuario] ([Id]),
18    CONSTRAINT [FK_Requerimiento_Urgencia] FOREIGN KEY ([Urgencia]) REFERENCES [dbo].[Urgencia] ([Id]),
19    CONSTRAINT [FK_Requerimiento_HardwareTI] FOREIGN KEY ([Equipo]) REFERENCES [dbo].[HardwareTI] ([Id]),
20    CONSTRAINT [FK_Requerimiento_Centro_costos] FOREIGN KEY ([Centro_costos]) REFERENCES [dbo].[Empresa] ([Id]),
21    CONSTRAINT [FK_Requerimiento_Incidencia] FOREIGN KEY ([Incidencia]) REFERENCES [dbo].[Incidencia] ([Id])
22 );

```

Fuente: Elaboración Propia

Figura N° 324: Código del modelo Requerimiento

```

namespace SistemaControlTI.Models
{
    using System;
    using System.Collections.Generic;
    19 referencias
    public partial class Requerimiento
    {
        0 referencias
        public int Id { get; set; }
        0 referencias
        public string codigo { get; set; }
        0 referencias
        public string Nombre { get; set; }
        3 referencias
        public int SolicitadoPor { get; set; }
        3 referencias
        public int Urgencia { get; set; }
        3 referencias
        public int Componente { get; set; }
        3 referencias
        public Nullable<int> Equipo { get; set; }
        3 referencias
        public Nullable<int> Incidencia { get; set; }
        0 referencias
        public System.DateTime Fecha_solicitud { get; set; }
        0 referencias
        public System.DateTime Fecha_entrega { get; set; }
        0 referencias
        public int Cantidad { get; set; }
        3 referencias
        public int Centro_costos { get; set; }
        0 referencias
        public string Detalles { get; set; }
        1 referencia
        public virtual Componente Componente1 { get; set; }
        1 referencia
        public virtual Empresa Empresa { get; set; }
        1 referencia
        public virtual HardwareTI HardwareTI { get; set; }
        1 referencia
        public virtual Incidencia Incidencia1 { get; set; }
        1 referencia
        public virtual Urgencia Urgencia1 { get; set; }
    }
}

```

Fuente: Elaboración Propia

Figura N° 325: Código del controlador Requerimiento

```

using System.Net;
using System.Web.Mvc;
using SistemaControlTI.Models;
namespace SistemaControlTI.Controllers
{
    0 referencias
    public class RequerimientosController : Controller
    {
        [private WebConSerSisEntities db = new WebConSerSisEntities();]
        // GET: Requerimientos
        public ActionResult Index()
        {
            0 referencias
            (var requerimiento = db.Requerimiento.Include(r => r.Componente1).Include(r => r.Empresa).Include(r => r.HardwareTI).Include(r => r.Incidencia1).Include(r => r.Urgencia1).Include(r => r.Usuario);
            return View(requerimiento.ToList());
        }
        // GET: Requerimientos/Details/5
        public ActionResult Details(int? id)
        {
            0 referencias
            // GET: Requerimientos/Create
            public ActionResult Create()
            {
                0 referencias
                (ViewBag.Componente = new SelectList(db.Componente, "Id", "Nombre");
                ViewBag.Centro_costos = new SelectList(db.Empresa, "Id", "RazonSocial");
                ViewBag.Equipo = new SelectList(db.HardwareTI, "Id", "Nombre");
                ViewBag.Incidencia = new SelectList(db.Incidencia, "Id", "Codigo");
                ViewBag.Urgencia = new SelectList(db.Urgencia, "Id", "Nombre");
                ViewBag.SolicitadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre");
                return View();
            }
            // POST: Requerimientos/Create
            [HttpPost]
            [ValidateAntiForgeryToken]
            public ActionResult Create([Bind(Include = "Id,codigo,Nombre,SolicitadoPor,Urgencia,Componente,Equipo,Incidencia,Fecha_solicitud,Fecha_entrega,Cantidad,Centro_costos,Detalles")] Requerimiento requerimiento)
            {
                0 referencias
                (db.Requerimiento.Add(requerimiento);db.SaveChanges();return RedirectToAction("Index"));
                ViewBag.Componente = new SelectList(db.Componente, "Id", "Nombre", requerimiento.Componente);
                ViewBag.Centro_costos = new SelectList(db.Empresa, "Id", "RazonSocial", requerimiento.Centro_costos);
                ViewBag.Equipo = new SelectList(db.HardwareTI, "Id", "Nombre", requerimiento.Equipo);
                ViewBag.Incidencia = new SelectList(db.Incidencia, "Id", "Codigo", requerimiento.Incidencia);
                ViewBag.Urgencia = new SelectList(db.Urgencia, "Id", "Nombre", requerimiento.Urgencia);
                ViewBag.SolicitadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", requerimiento.SolicitadoPor);
                return View(requerimiento);
            }
            // GET: Requerimientos/Edit/5
            public ActionResult Edit(int? id)
            {
                0 referencias
                (if (id == null) (return new HttpStatusCodeResult(HttpStatusCode.BadRequest));
                Requerimiento requerimiento = db.Requerimiento.Find(id);
                if (requerimiento == null){return HttpNotFound();}
                ViewBag.Componente = new SelectList(db.Componente, "Id", "Nombre", requerimiento.Componente);
                ViewBag.Centro_costos = new SelectList(db.Empresa, "Id", "RazonSocial", requerimiento.Centro_costos);
                ViewBag.Equipo = new SelectList(db.HardwareTI, "Id", "Nombre", requerimiento.Equipo);
                ViewBag.Incidencia = new SelectList(db.Incidencia, "Id", "Codigo", requerimiento.Incidencia);
                ViewBag.Urgencia = new SelectList(db.Urgencia, "Id", "Nombre", requerimiento.Urgencia);
                ViewBag.SolicitadoPor = new SelectList(db.Usuario, "Id", "Full_Nombre", requerimiento.SolicitadoPor);
                return View(requerimiento);
            }
            // POST: Requerimientos/Edit/5
        }
    }
}

```

Fuente: Elaboración Propia

Figura N° 326: Código de la vista Índice Requerimiento

```

@model IEnumerable<SistemaControlTI.Models.Requerimiento>
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Requerimientos</h2>
<p>@Html.ActionLink("Nuevo Registro", "Create")</p>
<table class="table">
    <tr>
        <th>@Html.DisplayNameFor(model => model.codigo)</th>
        <th>Descripción </th>
        <th>@Html.DisplayNameFor(model => model.Fecha_solicitud)</th>
        <th>@Html.DisplayNameFor(model => model.Fecha_entrega)</th>
        <th>@Html.DisplayNameFor(model => model.Cantidad)</th>
        <th>@Html.DisplayNameFor(model => model.Detalles)</th>
        <th>Componente</th>
        <th>@Html.DisplayNameFor(model => model.Empresa.RazonSocial)
        </th><th>Equipo</th>
        <th>Incidencia</th>
        <th>Prioridad</th>
        <th>Solicitante </th><th></th>
    </tr>
    @foreach (var item in Model) {
        <tr>
            <td>@Html.DisplayFor(modelItem => item.codigo)</td>
            <td>@Html.DisplayFor(modelItem => item.Nombre) </td>
            <td>@Html.DisplayFor(modelItem => item.Fecha_solicitud)</td>
            <td>@Html.DisplayFor(modelItem => item.Fecha_entrega)</td>
            <td>@Html.DisplayFor(modelItem => item.Cantidad)</td>
            <td>@Html.DisplayFor(modelItem => item.Detalles)</td>
            <td>@Html.DisplayFor(modelItem => item.Componente1.Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.Empresa.RazonSocial) </td>
            <td>@Html.DisplayFor(modelItem => item.HardwareTI.Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.Incidencia1.Codigo)</td>
            <td>@Html.DisplayFor(modelItem => item.Urgencia1.Nombre)</td>
            <td>@Html.DisplayFor(modelItem => item.Usuario.Full_Nombre) </td>
            <td>
                @Html.ActionLink("Editar", "Edit", new { id=item.Id }) |
                @Html.ActionLink("Detalles", "Details", new { id=item.Id }) |
                @Html.ActionLink("Eliminar", "Delete", new { id=item.Id })
            </td>
        </tr>
    }
}

```

Fuente: Elaboración Propia

Figura N° 327: Código de la vista Crear Requerimiento

```

@model SistemaControlTI.Models.Requerimiento
@{ViewBag.Title = "Create"; Layout = "~/Views/Shared/_Layout.cshtml";}
<h2>Crear</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal"> <h4>Requerimiento</h4><hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
        @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
            @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.SolicitadoPor, "SolicitadoPor", htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
            @Html.DropDownList("SolicitadoPor", null, htmlAttributes: new { @class = "form-control" })
            @Html.ValidationMessageFor(model => model.SolicitadoPor, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.Urgencia, "Urgencia", htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
            @Html.DropDownList("Urgencia", null, htmlAttributes: new { @class = "form-control" })
            @Html.ValidationMessageFor(model => model.Urgencia, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.Componente, "Componente", htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
            @Html.DropDownList("Componente", null, htmlAttributes: new { @class = "form-control" })
            @Html.ValidationMessageFor(model => model.Componente, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.Equipo, "Equipo", htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
            @Html.DropDownList("Equipo", null, htmlAttributes: new { @class = "form-control" })
            @Html.ValidationMessageFor(model => model.Equipo, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.Incidencia, "Incidencia", htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
            @Html.DropDownList("Incidencia", null, htmlAttributes: new { @class = "form-control" })
            @Html.ValidationMessageFor(model => model.Incidencia, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.Fecha_solicitud, htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
            @Html.EditorFor(model => model.Fecha_solicitud, new { htmlAttributes = new { @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Fecha_solicitud, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.Fecha_entrega, htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
            @Html.EditorFor(model => model.Fecha_entrega, new { htmlAttributes = new { @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Fecha_entrega, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.Cantidad, htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
            @Html.EditorFor(model => model.Cantidad, new { htmlAttributes = new { @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Cantidad, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.Centro_costos, "Centro_costos", htmlAttributes: new { @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("Centro_costos", null, htmlAttributes: new { @class = "form-control" })
            @Html.ValidationMessageFor(model => model.Centro_costos, "", new { @class = "text-danger" })</div></div>
    <div class="form-group">
        @Html.LabelFor(model => model.Detalles, htmlAttributes: new { @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Detalles, new { htmlAttributes = new { @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Detalles, "", new { @class = "text-danger" })</div></div>
    }
}

```

Fuente: Elaboración Propia

Figura N° 328: Código de la vista Editar Requerimiento

```
@model SistemaControlTII.Models.Requerimiento
@{ViewBag.Title = "Edit";Layout = "~/Views/Shared/_Layout.cshtml";}
<h2>Editar</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Requerimiento</h4><hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)
        <div class="form-group">
            @Html.LabelFor(model => model.codigo, htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
                @Html.EditorFor(model => model.codigo, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.codigo, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })<div class="col-md-10">
                    @Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.SolicitadoPor, "SolicitadoPor", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("SolicitadoPor", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.SolicitadoPor, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Urgencia, "Urgencia", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Urgencia", null, htmlAttributes: new { @class = "form-control" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Componente, "Componente", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Componente", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Componente, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Equipo, "Equipo", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Equipo", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Equipo, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Incidencia, "Incidencia", htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.DropDownList("Incidencia", null, htmlAttributes: new { @class = "form-control" })
                    @Html.ValidationMessageFor(model => model.Incidencia, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Fecha_solicitud, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Fecha_solicitud, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Fecha_solicitud, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Fecha_entrega, htmlAttributes: new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.Fecha_entrega, new { htmlAttributes = new { @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.Fecha_entrega, "", new { @class = "text-danger" })</div></div>
            <div class="form-group">
                @Html.LabelFor(model => model.Cantidad, htmlAttributes: new { @class = "control-label col-md-2" })
```

Fuente: Elaboración Propia

Figura N° 329: Código de la vista Eliminar Requerimiento

```

@model SistemaControlTI.Models.Requerimiento
@{ ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml"; }
<h2>Eliminar</h2>
<h3>Esta seguro que desea borrarlo?</h3>
<div>
    <h4>Requerimiento</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>@Html.DisplayNameFor(model => model.codigo)</dt>
        <dd>@Html.DisplayFor(model => model.codigo)</dd>
        <dt>@Html.DisplayNameFor(model => model.Nombre)</dt>
        <dd>@Html.DisplayFor(model => model.Nombre)</dd>
        <dt>@Html.DisplayNameFor(model => model.Fecha_solicitud)</dt>
        <dd>@Html.DisplayFor(model => model.Fecha_solicitud)</dd>
        <dt>@Html.DisplayNameFor(model => model.Fecha_entrega)</dt>
        <dd>@Html.DisplayFor(model => model.Fecha_entrega)</dd>
        <dt>@Html.DisplayNameFor(model => model.Cantidad)</dt>
        <dd>@Html.DisplayFor(model => model.Cantidad)</dd>
        <dt>@Html.DisplayNameFor(model => model.Detalles)</dt>
        <dd>@Html.DisplayFor(model => model.Detalles)</dd>
        <dt>@Html.DisplayNameFor(model => model.Componente1.Nombre) </dt>
        <dd>@Html.DisplayFor(model => model.Componente1.Nombre)</dd>
        <dt>@Html.DisplayNameFor(model => model.Empresa.RazonSocial)</dt>
        <dd>@Html.DisplayFor(model => model.Empresa.RazonSocial)</dd>
        <dt>@Html.DisplayNameFor(model => model.HardwareTI.Nombre)</dt>
        <dd>@Html.DisplayFor(model => model.HardwareTI.Nombre)</dd>
        <dt>@Html.DisplayNameFor(model => model.Incidencia1.Codigo)</dt>
        <dd>@Html.DisplayFor(model => model.Incidencia1.Codigo)</dd>
        <dt>@Html.DisplayNameFor(model => model.Urgencia1.Nombre)</dt>
        <dd>@Html.DisplayFor(model => model.Urgencia1.Nombre)</dd>
        <dt>@Html.DisplayNameFor(model => model.Usuario.DNI)</dt>
        <dd>@Html.DisplayFor(model => model.Usuario.DNI)</dd>
    </dl>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()
        <div class="form-actions no-color">
            <input type="submit" value="Eliminar" class="btn btn-default" />
            @Html.ActionLink("Regresar al listado", "Index")
        </div>
    }
</div>

```

Fuente: Elaboración Propia

Figura N° 330: Formulario Index Requerimiento

codigo	Descripción	Fecha_solicitud	Fecha_entrega	Cantidad	Detalles	Componente	RazonSocial	Equipo	Incidencia	Prioridad	Solicitante	
RE000002	Reparacion	12/05/2018 0:00:00	12/05/2018 0:00:00	2	adada	Memoria RAM Kington DDR3 4GB	Constructora El Roble SAC	hp	IN000001	Media	Alex Aiquipa	Editar Detalles Eliminar

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Figura N° 331: Formulario Crear Requerimiento

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Crear

Requerimiento

Nombre

SolicitadoPor

Urgencia

Componente

Equipo

Incidencia

Fecha_solicitud

Fecha_entrega

Cantidad

Centro_costos

Detalles

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Figura N° 332: Formulario Editar Requerimiento

SISCONSER [Gestión Usuarios](#) [Recursos TI](#) [Servicios](#)

Editar

Requerimiento

codigo	<input type="text" value="RE000002"/>
Nombre	<input type="text" value="Reparacion"/>
SolicitadoPor	<input type="text" value="10627303"/>
Urgencia	<input type="text" value="Media"/>
Componente	<input type="text" value="Memoria RAM Kington DDR3 4GB"/>
Equipo	<input type="text" value="hp"/>
Incidencia	<input type="text" value="IN000001"/>
Fecha_solicitud	<input type="text" value="12/05/2018 0:00:00"/>
Fecha_entrega	<input type="text" value="12/05/2018 0:00:00"/>
Cantidad	<input type="text" value="2"/>
Centro_costos	<input type="text" value="Constructora El Roble SAC"/>
Detalles	<input type="text" value="adada"/>

[Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Figura N° 333: Formulario Eliminar Requerimiento

SISCONSER | Gestión Usuarios | Recursos TI | Servicios

Eliminar

Esta seguro que desea borrarlo?
Requerimiento

codigo	RE000002
Nombre	Reparacion
Fecha_solicitud	12/05/2018 0:00:00
Fecha_entrega	12/05/2018 0:00:00
Cantidad	2
Detalles	adada
Nombre	Memoria RAM Kington DDR3 4GB
RazonSocial	Constructora El Roble SAC
Nombre	hp
Codigo	IN000001
Nombre	Media
DNI	10627303

| [Regresar al listado](#)

© 2018 - Sistema Web para el Control de Servicios del Área de Sistemas

Fuente: Elaboración Propia

Requerimiento RF23

RF23: El sistema debe permitir al administrador monitorizar el indicador de índice de cobertura de mantenimiento

Implementación

Figura N° 334: Procedimiento Almacenado IndiceCoberturaMant

```
1 CREATE procedure IndiceCoberturaMant
2 AS
3 SELECT
4 year(Fecha_Programada) as Anio,
5 month(Fecha_Programada) as Mes,
6 Count(*) as TotalPlanificado,
7 SUM(iif(Cronograma is null,0,1)) as TotalRealizado,
8 Round((SUM(iif(Cronograma is null,0,1))*100.00/Count(*)),2) as Ratio
9 from
10 (
11 select
12 cp.*,
13 mp.Cronograma
14 from
15 CronogramaMantenimiento cp left join (select Cronograma from Mantenimiento_Preventivo where EstadoAt='4') mp
16 on cp.id = mp.Cronograma
17 ) as c
18 where
19 Fecha_Programada is not null
20 group by
21 year(Fecha_Programada),
22 month(Fecha_Programada)
```

Fuente: Elaboración Propia

Figura N° 335: : Código del modelo Índice de Cobertura

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6 using SistemaControlTI.Models;
7 namespace SistemaControlTI.Controllers
8 {
9     [Authorize]
10    public class Indicador_Grado_CumplimientoController : Controller
11    {
12        private WebConSerSisEntities db = new WebConSerSisEntities();
13        // GET: ICM
14        public ActionResult Index()
15        {
16            return View(db.Indicador_incidencia_mensual(DateTime.Now.AddMonths(-1)).ToList());
17        }
18        // GET: ICM/Details/5
19        public ActionResult Details(int id)
20        {
21            return View();
22        }
23        // GET: ICM/Create
24        public ActionResult Create()
25        {
26            return View();
27        }
28        // POST: ICM/Create
29        [HttpPost]
30        public ActionResult Create(FormCollection collection)
31        {
32            try
33            {
34                // TODO: Add insert logic here
35                return RedirectToAction("Index");
36            }
37            catch
38            {
39                return View();
40            }
41        }
42        // GET: ICM/Edit/5
43        public ActionResult Edit(int id)
44        {
45            return View();
46        }
47        // POST: ICM/Edit/5
48        [HttpPost]
49        public ActionResult Edit(int id, FormCollection collection)
50        {
51            try { return RedirectToAction("Index");
52            } catch { return View(); }
53        }
54        // GET: ICM/Delete/5
55        public ActionResult Delete(int id)
56        {
57            return View();
58        }
59        // POST: ICM/Delete/5
60        [HttpPost]
61        public ActionResult Delete(int id, FormCollection collection)
62        {
63            try { return RedirectToAction("Index");
64            } catch { return View(); }
65        }
66    }
67 }
```

Fuente: Elaboración Propia

Figura N° 336: Código de la Vista Índice de Cobertura

```

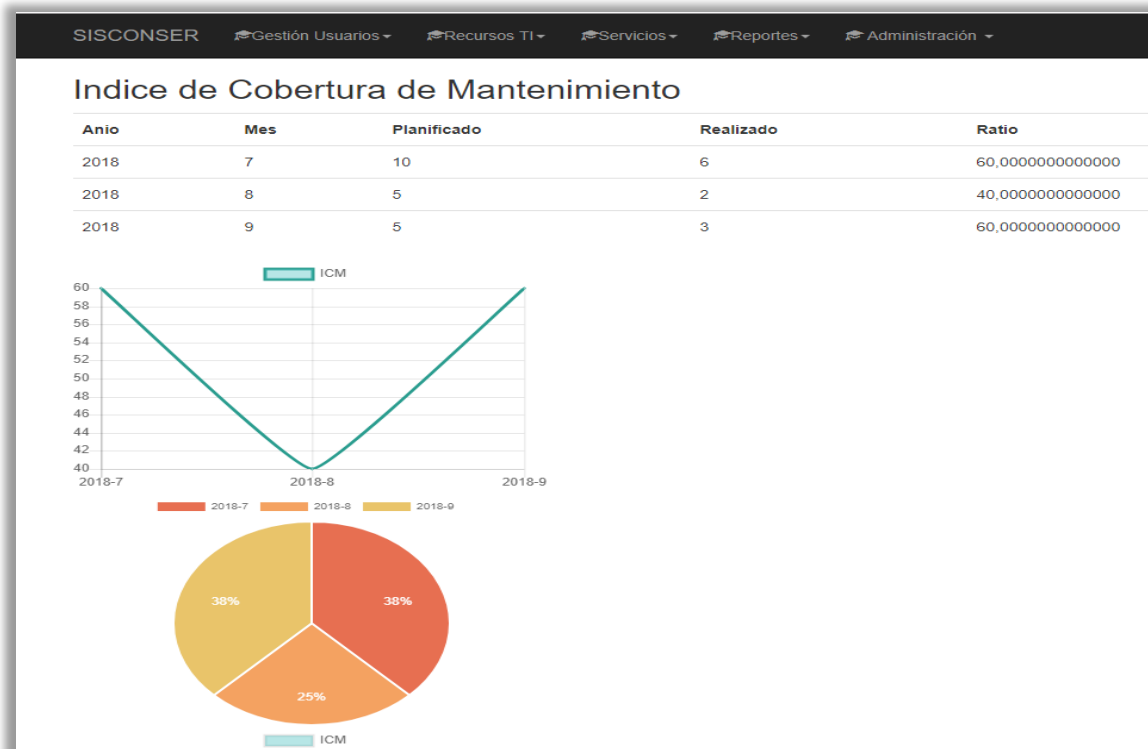
@model IEnumerable<SistemaControlTI.Models.Indicador_incidencia_mensual_Result>
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width" />
<title>Index</title>
</head>
<body>
<h2>Estado de atenciones</h2>
@using (Html.BeginForm("Index", "Home", FormMethod.Get))
{
    <table cellpadding="0" cellspacing="0" class="table">
    <tr><th>Estado</th><th>Total</th></tr>
    @foreach (SistemaControlTI.Models.Indicador_incidencia_mensual_Result item in Model)
    {
        <tr><td>@item.EstadoAt </td><td>@item.Total </td></tr>
    }
    </table>
    <div class="informesmantenimientos" style="width:400px !important">
    <canvas id="informe1" height="200"></canvas>
    </div>
    <div class="informesmantenimientos" style="width:400px !important">
    <canvas id="informe2" height="200"></canvas>
    </div>
    <div class="informesmantenimientos" style="width:400px !important">
    <canvas id="informe3" height="200"></canvas>
    </div>
}
<script src="~/Scripts/jquery-3.3.1.min.js"></script>
<script src="~/Scripts/Chart.bundle.min.js"></script>
<script src="~/Scripts/cercleCharts.js"></script>
<script type="text/javascript">
$(document).ready(function () {
    var x = '@string.Join(",", Model.OrderBy(y=>y.EstadoAt).Select(x => x.EstadoAt.ToString() + '-' + x.EstadoAt.ToString()).ToArray())';
    var y = '@string.Join(",", Model.OrderBy(y => y.EstadoAt).Select(x =>x.Total.Value).ToArray())';
    linesSimple("informe1", x, y, "Estado de atenciones");
    pie("informe2", x, y, "Estado de atenciones");
    bar("informe3", x, y, "Estado de atenciones");});
}

```

Fuente: Elaboración Propia

GUI

Figura N° 337: Indicador de Índice de Cobertura



Fuente: Elaboración Propia

Figura N° 338: Procedimiento Almacenado Reporte General de Activos

```
1 CREATE Proc [dbo].[Reporte_General_Activos]
2 AS
3 SELECT RazonSocial, su.Nombre as Activo, es.Nombre as Estado, Sum(1) as Total from
4 RecursoAsignado re inner join Usuario us on re.Usuario = us.Id inner join Empresa em on us.Planilla=em.Id inner join Estadohw es on re.Estadohw = es.Id
5 inner join HardwareTI hw on re.HardwareTI=hw.Id inner join Subcategoria su on hw.SubCategoria = su.id
6 Group by RazonSocial,su.Nombre, es.Nombre
```

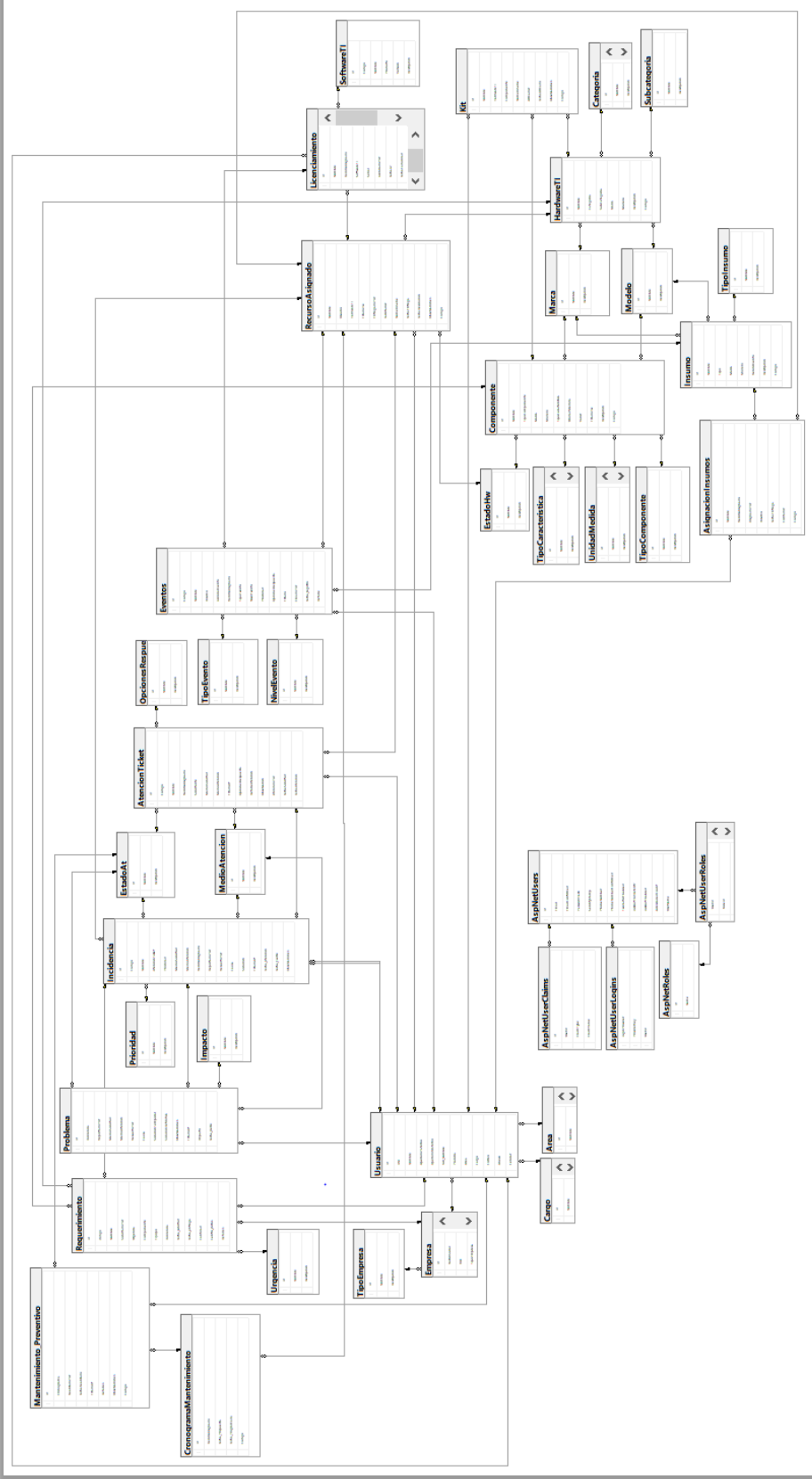
Fuente: Elaboración Propia

Figura N° 339: Reporte de activos operativos

RazonSocial	Activo	Estado	Total
Compañía Naviera Natalia SAC	CPU	Operativo	3
Compañía Naviera Natalia SAC	Impresora	Operativo	1
Constructora El Roble SAC	CPU	Operativo	1
Constructora El Roble SAC	Laptop	Operativo	1
Sobitec	CPU	Operativo	3

Fuente: Elaboración Propia

Figura N° 340: Base de datos completo del proyecto



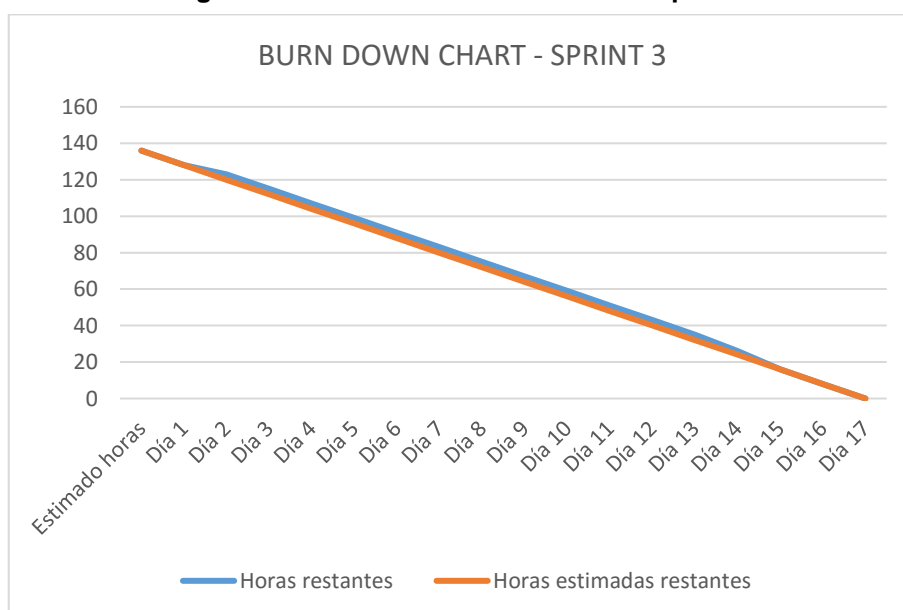
Fuente: Elaboración Propia

Tabla N° 19: BurnDown del Sprint 3

BurnDown Sprint 3																			
Tareas	Estimado horas	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7	Día 8	Día 9	Día 10	Día 11	Día 12	Día 13	Día 14	Día 15	Día 16	Día 17	Total horas
	Administrar y dar mantenimiento al cronograma de mantenimiento preventivo.	8	7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Administrar el estado de los mantenimientos realizados.	8	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
Registrar las atenciones solicitadas por los usuarios.	16	0	4	0	4	8	0	0	0	0	0	0	0	0	0	0	0	0	16
Administrar y dar mantenimiento al registro de las incidencias de equipos informáticos reportados.	24	0	0	0	0	0	8	8	4	4	0	0	0	0	0	0	0	0	24
Administrar y dar mantenimiento al registro de los problema reportados.	24	0	0	0	0	0	0	0	0	0	8	8	7	1	0	0	0	0	24
Registrar y administrar los requerimientos solicitados para reparación de los equipos informáticos.	24	0	0	0	0	0	0	0	0	0	0	0	0	7	8	8	1	0	24
Monitorizar el indicador de índice de cobertura de mantenimiento	16	1	0	0	4	0	0	0	0	3	0	0	1	0	0	0	7	0	16
Monitorizar el indicador de cumplimiento de las atenciones realizadas.	16	0	0	0	0	0	0	0	4	1	0	0	0	0	1	2	0	8	16
Horas restantes	136	128	123	115	107	99	91	83	75	67	59	51	43	35	26	16	8	0	
Horas estimadas restantes	136	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0	

Fuente: Elaboración Propia

Figura N° 341: Grafica Burn Down del Sprint 3



Fuente: Elaboración Propia

ACTA DE REUNIÓN N° 6

CIERRE DEL SPRINT 3

DATOS

Empresa / Organización	Compañía Naviera Natalia SAC.
Proyecto	Sistema web para el control de servicios del área de sistemas en la compañía Naviera Natalia SAC

PARTICIPANTES

Rol	Nombre
Dueño del Producto	Candela Ríos Hugo
Equipo	Aiquipa Tello Alex

ACUERDOS

Marcar con "X" a razón de cierre de cumplimiento de cada funcionalidad en la apertura del Sprint 3

Historias	Entrega Total
El sistema debe permitir al administrador, generar y actualizar el cronograma de mantenimiento preventivo.	X
El sistema debe permitir al usuario autenticado, registrar y actualizar los servicios de mantenimientos preventivos realizados.	X
El sistema debe permitir al usuario autenticado, registrar y actualizar las solicitudes de atenciones realizadas por los usuarios.	X
El sistema debe permitir al usuario autenticado, registrar y actualizar las incidencias de los equipos informáticos.	X
El sistema debe permitir al usuario autenticado, registrar y actualizar los problemas suscitados.	X
El sistema debe permitir al usuario autenticado, registrar y actualizar los requerimientos de insumos y/o componentes solicitados para reparación de los equipos informáticos de la compañía.	X
El sistema debe permitir al administrador monitorizar el indicador de índice de cobertura de mantenimiento	X
El sistema debe permitir al administrador monitorizar el indicador de cumplimiento de las atenciones realizadas.	X

FIRMAS



 ING. HUGO CANDELA
 JEFE DE SISTEMAS CORPORATIVO



 ALEX ALDO AIQUIPA TELLO
 ADMINISTRADOR DE REDES Y COMUNICACIONES

 DUEÑO DEL PRODUCTO

 EQUIPO

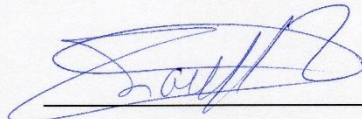
Acta de Aprobación de Originalidad de Tesis

Yo, **Mgtr. HUAROTE ZEGARRA RAUL EDUARDO**, asesor del curso de Desarrollo de Proyecto de Investigación, revisor de la tesis del estudiante **AIQUIPA TELLO ALEX ALDO**, titulada: **SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL AREA DE SISTEMAS EN LA COMPAÑÍA NAVIERA NATALIA S.A.C.**; constato que la misma tiene un índice de similitud del 25% verificable en el reporte de originalidad del programa Turnitin.

El suscrito analizó dicho reporte y concluyó que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la tesis cumple con todas las normas para el uso de citas y referencias establecida por la Universidad César Vallejo.

Lima, jueves 10 de julio del 2019

Atentamente,



Mgtr. HUAROTE ZEGARRA RAUL EDUARDO

DOCENTE ASESOR DE TESIS

DNI:37985830.....



FACULTAD DE INGENIERIA

ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL
AREA DE SISTEMAS EN LA COMPAÑIA NAVIERA NATALIA
S.A.C.

TESIS PARA OBTENER EL TITULO PROFESIONAL DE
INGENIERIA DE SISTEMAS

AUTOR:

AQUIPA TELLO ALEX ALDO

ASESOR:

MG. HUAROTE ZEGARRA RAUL EDUARDO

LINEA DE INVESTIGACION/
SISTEMAS DE INFORMACION Y COMUNICACIONES

LIMA - PERU

Resumen de coincidencias

25 %

Se están viendo fuentes estándar

Ver fuentes en inglés (Beta)

Coincidencias

1	Entregado a Universid...	10 %	>
2	repositorio ucv.edu.pe	6 %	>
3	Entregado a CONACYT	1 %	>
4	www.proactivanet.com	1 %	>
5	Entregado a ESIC Busin...	<1 %	>
6	Entregado a Universid...	<1 %	>
7	mateo.pbworks.com	<1 %	>
8	Entregado a UNAPEC	<1 %	>



Centro de Recursos para el Aprendizaje y la Investigación (CRAI)
"César Acuña Peralta"

FORMULARIO DE AUTORIZACIÓN PARA LA PUBLICACIÓN ELECTRÓNICA DEL TRABAJO DE INVESTIGACIÓN O LA TESIS

1. DATOS PERSONALES

Apellidos y Nombres: (solo los datos del que autoriza)

Aiquipa Tello Alex Aldo
D.N.I. : 1.06.27308
Domicilio : 174° 30Te 35° urb. Los Libertadores S.M.P.
Teléfono : Fijo : Móvil 981043308
E-mail : alex773at@gmail.com

2. IDENTIFICACIÓN DE LA TESIS

Modalidad:

- Trabajo de Investigación de Pregrado
 Tesis de Pregrado

Facultad : Ingeniería
Escuela : Ingeniería de Sistemas
Carrera : Sistemas
 Grado Título

Tesis de Post Grado

- Maestría Doctorado
Grado :
Mención :

3. DATOS DE LA TESIS

Autor (es) Apellidos y Nombres:

Aiquipa Tello Alex Aldo

Título del trabajo de investigación o de la tesis:

Sistema Web para el control de los Servicios del
Área de Sistemas en la Compañía Naviera Natalia SAC.

Año de publicación : 2019

4. AUTORIZACIÓN DE PUBLICACIÓN DE LA TESIS EN VERSIÓN ELECTRÓNICA:

A través del presente documento,

- Si autorizo a publicar en texto completo mi trabajo de investigación o tesis.
 No autorizo a publicar en texto completo mi trabajo de investigación o tesis.

Firma : 

Fecha : 08-06-2019



UNIVERSIDAD CÉSAR VALLEJO

AUTORIZACIÓN DE LA VERSIÓN FINAL DEL TRABAJO DE INVESTIGACIÓN

CONSTE POR EL PRESENTE EL VISTO BUENO QUE OTORGA EL ENCARGADO DE INVESTIGACIÓN DE

La Escuela de Ingeniería de Sistemas

A LA VERSIÓN FINAL DEL TRABAJO DE INVESTIGACIÓN QUE PRESENTA:

Aiquipa Tello, Alex Aldo

INFORME TÍTULADO:

SISTEMA WEB PARA EL CONTROL DE LOS SERVICIOS DEL ÁREA DE SISTEMAS EN LA COMPAÑÍA NAVIERA NATALIA S.A.C

PARA OBTENER EL TÍTULO O GRADO DE:

Ingeniero de Sistemas

SUSTENTADO EN FECHA: 20/12/2018

NOTA O MENCIÓN: 13



FIRMA DEL ENCARGADO DE INVESTIGACIÓN