



UNIVERSIDAD CÉSAR VALLEJO

**FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**Sistema web open source vue Js para el proceso de pruebas
de calidad de software en la empresa NextPerience**

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:
Ingeniero de Sistemas**

AUTOR

Joseph Olegario Flores Castillejo (ORCID: 0000-0001-8211-2409)

ASESOR

Magister Orleans Moisés Gálvez Tapia (ORCID: 0000-0002-0006-0973)

LÍNEA DE INVESTIGACIÓN

Sistema De Información Y Comunicación

LIMA – PERÚ

2020

Dedicatoria

Esta investigación se la dedico a mis estimados padres Doris y Cesar, debido a que su esfuerzo pude lograr esta meta.

Agradecimiento

Doy gracias a mi familia por el sustento incondicional y el aliento que nos brindan siempre para lograr nuestra meta. Al Mg. Orleans Gálvez Tapia, por el apoyo como asesor y conocimientos brindados en el desarrollo de la tesis. También agradecer al Doc. Adilio Christian Ordoñez Pérez, por el apoyo como asesor en la asignatura Proyecto de Investigación.

ÍNDICE DE CONTENIDOS

Contenido	Página
ÍNDICE DE CONTENIDOS.....	iv
ÍNDICE DE TABLAS.....	v
ÍNDICE DE FIGURAS.....	vi
I. INTRODUCCIÓN.....	ix
II. MARCO TEÓRICO.....	18
III. METODOLOGÍA.....	38
3.1. Tipo y diseño de investigación.....	39
3.2. Variables y Operacionalización.....	41
3.3. Población (criterios de selección), muestra y muestro, unidad de análisis.	45
3.4. Técnicas e instrumentos de recolección de datos.....	46
3.5. Procedimientos.....	48
3.6. Métodos de análisis de datos.....	51
3.7. Aspectos Éticos.....	55
IV. RESULTADOS.....	56
V. DISCUSIÓN.....	68
ANEXOS.....	78

ÍNDICE DE TABLAS

Tabla 1 Criterios de evaluación	30
Tabla 2 Cuadro comparativo de Metodologías bajo juicio de expertos.	31
Tabla 3 Lifecycle Hooks	34
Tabla 4: Operacionalización de variable	43
Tabla 5: Operacionalización de indicadores	44
Tabla 6: Recolección de datos	48
Tabla 7: Validación de la prueba de expertos	48
Tabla 8 Interpretación del coeficiente de confiabilidad	49
Tabla 9 Resultado de confiabilidad para el nivel de cobertura de las pruebas	50
Tabla 10 Resultado de confiabilidad para el nivel de madurez de las pruebas	50

ÍNDICE DE FIGURAS

Figura 1: Valor de la cobertura de las pruebas	13
Figura 2: Valor de la madurez de las pruebas	13
Figura 3 Patrón de diseño MVVM	29
Figura 4 Fases de RUP	32
Figura 5: Diseño del estudio	40
Figura 6: Estadística de Prueba	53
Figura 7: Gráfica del T- Student	54

RESUMEN

La presente investigación apunta el avance y ejecución de un sistema web open source Vue js para el proceso de pruebas de calidad de software en la empresa NextPerience. El ámbito previo contaba un inconveniente, el cual viene generado a raíz de la carencia de control y el rastreo en las pruebas de calidad que se hacen en la empresa NextPerience, así como la carencia de una metodología que logre validar la calidad de los productos manejados y desarrollados por el negocio.

El propósito de la investigación, es determinar la influencia de un sistema web open source Vue js para el proceso de pruebas de calidad de software en la empresa NextPerience, teniendo en cuenta los objetivos específicos para la realización de dicho sistema. El Sistema Web hará uso de la metodología RUP (Rational Unified Process) y creada a travez de la tecnología JavaScript por medio del framework Vue.js, así como la construcción de servicios api rest desarrollada en Node.js usando la librería Express.js y con uso del motor de base de datos MySQL, facilitará hacer de manera automática los casos de pruebas, desde el registro de estos, hasta la validez de la calidad de los productos y/o proyectos.

Se aplicará una investigación aplicada, experimental y como diseño de investigación, se hará uso del pre-experimental. Donde se tomó como indicador el nivel de cobertura de las pruebas y el nivel de madurez de las pruebas. Para las cuales se utilización 20 fichas de registro. Posteriormente se determinó que el Sistema Web Open Source Vue mejoro el nivel de cobertura de las pruebas, así como el nivel de madurez de las pruebas en el proceso de pruebas de calidad en la empresa NextPerience, debido a que se logró un aumento de 0.142 (14.2%) para el nivel de cobertura de las pruebas y un 0.218 (21.8%) para el nivel de madurez del proceso de pruebas de calidad.

Palabras claves: sistema web open source Vue, proceso de pruebas de calidad, nivel de cobertura, nivel de madurez

ABSTRACT

This research points to the advancement and execution of an open source Vue js web system for the software quality testing process in NextPerience. The previous scope had an inconvenience, which is generated by the lack of control and tracking in the quality tests that are done in the company NextPerience, as well as the lack of a methodology that manages to validate the quality of the products managed and developed by the business.

The purpose of the research is to determine the open source Vue js web system for the software quality testing process in NextPerience, taking into account the specific objectives for the realization of such a system. The Web System will make use of the RUP (Rational Unified Process) methodology and created through JavaScript technology through the Vue.js framework, as well as the construction of api rest services developed in Node.js using the Express.js library and with the use of the MySQL database engine, will make it easier to automatically make test cases, from registering them, to the validity of the quality of products and/or projects.

An investigation will be applied, applied, experimental and as a research design, will be used by the pre-experimental. Where the level of coverage of the tests and the maturity level of the tests were taken as an indicator. For which 20 registration cards are used. It was subsequently determined that the WebOpen Source Vue system improved the level of test coverage, as well as the maturity level of the tests in the NextPerience company's quality testing process, due to an increase of 0.142 (14.2%) for the level of test coverage and 0.218 (21.8%) for the maturity level of the quality testing process.

Words Keyers: Open source Vue system, quality testing process, coverage level, maturity level

I. INTRODUCCIÓN

Realidad Problemática

Durante el transcurso de la historia, la ingeniería del software ha empleado pruebas en los productos, convirtiéndolos así en una importante herramienta para asegurar la calidad del producto final, debido a que permite estas permiten validar los requerimientos iniciales del cliente. En una investigación realizado por la Standish Group, demuestra que el 29% de los proyectos realizados entre grandes y medianas empresas que desarrollan software, resultan exitosos. Indicando que solo ciertos proyectos que son entregados a tiempo, no superan el presupuesto y de las que cumplen con los requerimientos implantados por el cliente cumplen con este porcentaje. Los estudios realizados en España sobre la madurez y las buenas prácticas de los procesos de pruebas (cerca del 50% de las respuestas), revelan que solo el 65,69% resultan ser validaciones de las expectativas del cliente, 49,02% especificaciones documentadas y el 49.02 son informes de resultados de las pruebas al grupo desarrollador. En estudios previos, se logran identificar un porcentaje de 64% de organizaciones encuestadas, logran llevar procedimientos documenta para pruebas, el 25% cuenta con un equipo encargado de las pruebas, donde se concluye que el mismo personal que se encarga del desarrollo son los mismos en realizar las pruebas, y que un 67% confirman certificar el nivel de calidad de los productos (Blanquicett, Bonfante y Acosta-Solano, 2018, p. 2).

En el Perú existe un problema durante el proceso del desarrollo de un software, que es el aseguramiento de esta, realizar el cumplimiento de los requisitos en la mayoría de las cuestiones no es suficiente, es por ello que es necesario definir un proceso de pruebas serio, equitativo para cada proyecto, que incluya en ella las mejores prácticas de pruebas, siempre bajo un modelo de mejora continua. Lo cual es necesario desarrollar y aplicar un modelo basado en metodologías para el planeamiento, la especificación y la ejecución de las pruebas de software, donde permita llegar a criterios de aceptación para las empresas desarrolladoras de software. Generalmente, en este ámbito, las actividades relacionadas a las pruebas no son aplicadas, debido a que nos y crean caso de pruebas, donde permitan asegurar la calidad del software. Lo que se necesita es obter por una metodología, donde defina actividades y responsabilidad para el equipo, lo que es claro que esta tarea no se debe realizar de manera improvisada, sino que es un proceso gradual (Valdivia, 2017, p.2).

Esta investigación se realiza en la empresa NextPerience Perú S.A.C el cual se encuentra ubicada en Av. Javier Prado Oeste Nro. 757 Int. 1806, la cual se encuentra en el rubro del desarrollo de software a medida. Esta organización sustenta su funcionamiento en base a sus clientes y al servicio de calidad que ofrece; y maneja un área de desarrollo de software, el cual cuenta con funciones como: desarrollo de software, control de pruebas de calidad, etc.

Según la entrevista que se realizó al ingeniero Renzo Castillo Ibarra, jefe de proyectos (Ver anexo 08), nos indica que el proceso de pruebas de calidad se viene realizando hace aproximadamente dos años y medio, y que en el transcurso de seis meses se presentaron problemas durante la realización de las pruebas de calidad ya que no existía un control sistematizado de muestras que se encargara de registrar adecuadamente el número de pruebas realizadas y pruebas exitosas que se ejecutaban al día.

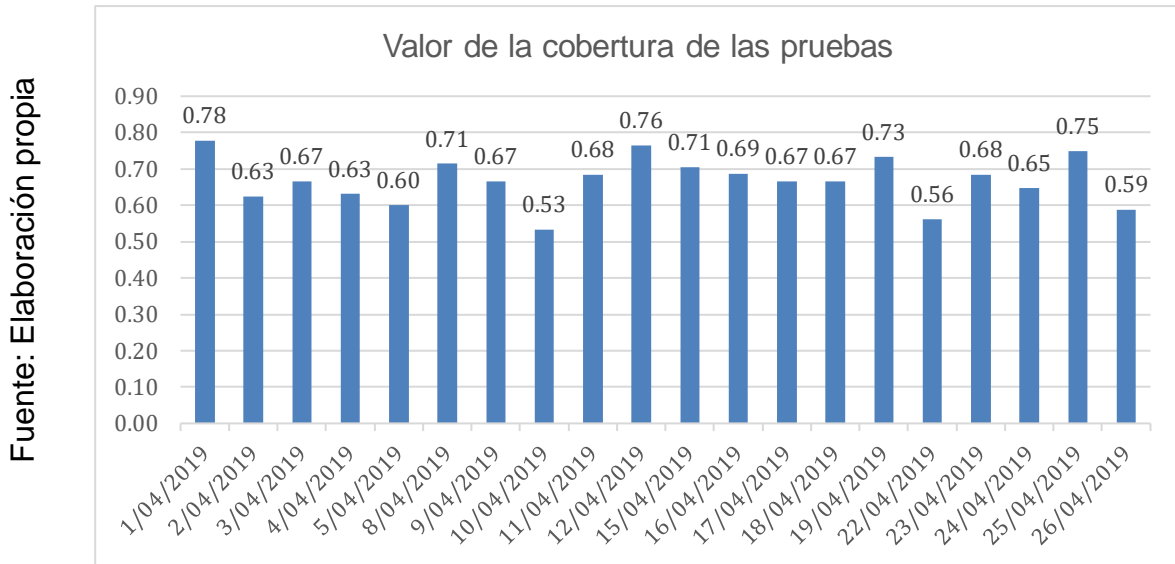
El proceso de las pruebas de calidad se inicia con la toma de los requerimientos funcionales recogidos durante la entrevista al cliente en la primera fase del ciclo de vida de un proyecto. Aquí se determina el nivel de prueba al que va a pertenecer dicho requerimiento, el cual ayudará a priorizar que prueba debe ser más importante que otra. Las pruebas que pasarán a manos de los desarrolladores, los cuales se encargarán de ejecutar y verificar el funcionamiento de la mismas, se consideran pruebas iniciales, ya que, durante los entregables al cliente, este puede encontrar nuevos problemas o errores en el funcionamiento, fallas de tipeo o validación de campos de textos.

Las pruebas iniciales son enviadas a los programadores, los cuales son los mismos que estuvieron encargados en el desarrollo del software a evaluar, ellos realizan las pruebas, determinando así la validez y funcionalidad de cada una de ellas. Al día se entregan de 10 a 30 pruebas, las cuales no son abarcadas al 100%, generando así que estas pruebas sean llenadas en un documento Excel para ser enviados al jefe de proyectos y poder reprogramarlas en siguientes días, lo cual da como resultado una acumulación progresiva de pruebas para días posteriores, así como el retraso a la hora de entrega del producto al cliente. Un factor que influye a este problema es el método de validación de pruebas que se utiliza, el cual puede durar entre 30 a 45 minutos generado por el sistema rudimentario del uso de aplicaciones que no son

destinadas para esta actividad, como lo es un documento de Excel o Word. Las pruebas realizadas al día también son registradas en otro documento Excel, en los cuales se seleccionan las pruebas satisfactorias y se colocan en otro documento, para luego ser enviadas al cliente con el fin de que sean verificadas en totalidad. El cliente se encarga de validar cada prueba realizada, en base al uso iterativo del sistema entregado junto con ellas. En ciertas ocasiones, el cliente puede tener nuevos requerimientos, que en algún caso se encolan o se priorizan al conjunto de pruebas del día, el cual será determinado por el cliente, quien puede definir cuáles son más relevantes que otras. Entre los nuevos requerimientos se pueden encontrar: errores de tipeo, nuevas validaciones de campos de textos, nueva distribución de elementos ya colocados y la creación de nuevos módulos o apartados que puedan generar conflictos o errores con alguna funcionalidad ya realizada para el sistema. Los nuevos requerimientos pasan al área de desarrollo, logrando así que el programador detenga la realización de las pruebas, para poder hacer los cambios solicitados por el cliente. Este último escenario se replica entre 2 a 4 veces por semana.

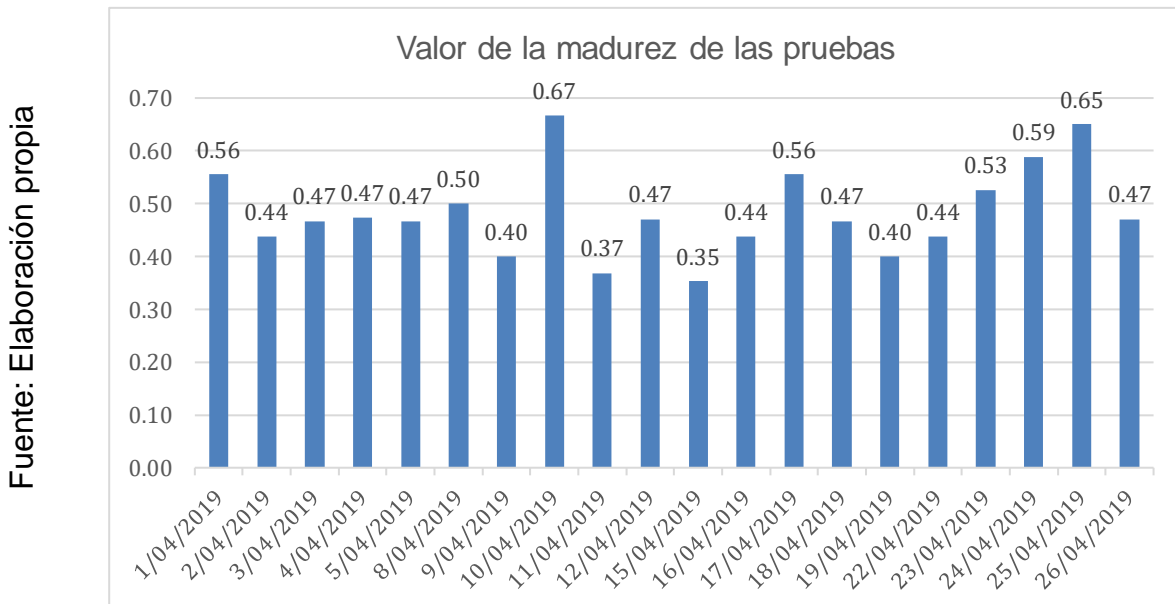
De igual modo, las evaluaciones aplicadas a unas 333 pruebas que determina el nivel de cobertura (pruebas totales que se realizan al día) como en el pre-test de la presente investigación, refleja niveles no esperados, ya que en promedio no abarca el 100% de las pruebas, tanto por día como en su totalidad. El promedio total es de 67%, esto es debido a que regularmente cada prueba dura entre 30 a 45 minutos, lo cual no cubre por completo la jornada laboral, también porque a veces la cantidad de pruebas puede incrementar a causa de las nuevas observaciones del cliente, esto se logra visualizar en la Figura 1.

Figura 1: Valor de la cobertura de las pruebas



Asimismo, el nivel de madurez de prueba hallado en el pre-test refleja que el total de pruebas realizadas satisfactorias tampoco cubre el 100%, sino que el resultado obtenido cubre tan solo un 48%, puesto que no se logra evaluar por completo las pruebas con soluciones satisfactorias debido a que no todos los resultados han sido determinados por completo. Estos resultados se pueden ver en la figura 2.

Figura 2: Valor de la madurez de las pruebas



Por lo tanto, si se sigue manteniendo esta situación donde los niveles de cobertura y madurez de las pruebas, se mantengan en niveles por debajo de lo esperado, se formularía la siguiente pregunta: ¿Qué sucederá si se siguen teniendo los mismos problemas? La respuesta a dicha pregunta resultaría en una pérdida de tiempo valioso al realizar cada prueba, el retraso de tiempo de entrega del producto al cliente, la baja mejoría en los niveles de calidad y lo más perjudicial, que los clientes de la compañía queden insatisfechos.

Formulación del problema

Problema Principal

¿De qué manera influye un sistema web en el proceso de pruebas de calidad de software en la empresa NextPerience?

Problemas Secundarios

¿De qué manera influye un sistema web en el nivel de cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience?

¿De qué manera influye un sistema web en el nivel de madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience?

Objetivos

Objetivo General

Determinar de qué manera influye un sistema web en el proceso de pruebas de calidad de software en la empresa NextPerience.

Objetivos Específicos

Determinar la influencia de un sistema web en el nivel de cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

Determinar la influencia de un sistema web en el nivel de madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

Hipótesis

Hipótesis General

El sistema web mejora el proceso de pruebas de calidad de software en la empresa NextPerience

Hipótesis Específicas

El sistema web aumenta el nivel de cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

El sistema web aumenta el nivel de madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

Justificación de la investigación

Justificación Institucional

Los objetivos y los fines estratégicos de una entidad originan y proporciona el convenio, pertenencia y eliminación de políticas internas, en los trabajadores, reforzando la ilustración institucional a clientes y proveedores (Fuentes, 2014, p.56).

Es así como la presente investigación aportará en gran medida al cumplimiento de metas y objetivos de la empresa NextPerience, por medio de herramientas que logran impulsar el negocio al mercado. Así como seguir cumpliendo el compromiso

que existe con los clientes, brindando un servicio de calidad e involucrándose completamente en su satisfacción.”

Justificación Tecnológica

La mayoría de las organizaciones deben contar con tecnología de última generación, para que a través de ella se pueda obtener data práctica, efectiva y concisa. En los sistemas informáticos, se ponen en práctica los procedimientos y políticas orientadas a asegurar la ejecución constante de los sistemas y al aseguramiento de su uso, donde esta constituye una herramienta de apoyo y logra agilizar y optimizar todos los procesos (Aguilar, 2016, p. 87).

Es por eso que esta investigación contribuirá con la mejora de calidad de softwares, asegurando la funcionalidad de los requerimientos del cliente, así como consolidar la integridad, seguridad y la disponibilidad de los datos, tanto para el usuario mismo como para todas las áreas que se encargarán del manejo del sistema web, haciendo uso de nuevas tecnologías en el proceso de pruebas de calidad.

Justificación Operativa

Según Salas Rueda (2016), define que: “El aspecto de la accesibilidad debe ser considerado durante la planeación y organización de los sistemas web con la finalidad de mejorar el canal de comunicación de información y el usuario final” (p.57).

La presente investigación hará uso de un sistema web que estará disponible las 24 horas, el cual tendrá una interfaz acorde a las últimas tendencias web, llegando a ser un sitio amigable y dinámico, el cual será de fácil interacción generando una rápida comprensión para cada actor del negocio. De esta manera, la operatividad del sistema web para el proceso de pruebas de calidad estará garantizada.

Justificación Económica

“Todo proyecto implica una inversión para la compañía y toda inversión debe tener un entorno o beneficios que justifique el proyecto”. (Brojt, 2015, p. 69).

Actualmente en la empresa, el proceso de pruebas de calidad requiere de 2 programadores, los cuales tienen un sueldo aproximado de S/. 1,200.00 cada uno, generando un pago mensual de S/. 2,400.00, lo que dejaría un gasto anual cercano a S/. 28,800.00. Se debe tener en cuenta que con la implementación de este sistema web ya no sería necesario contratar a dos programadores, sino sólo a uno que fácilmente podría hacerse cargo del proceso; de esta manera, el gasto por pago al personal sería de S/. 14,000.00 anualmente. Por consiguiente, con la implementación de este sistema web, se generaría un ahorro de S/. 14,800.00 por año. De esta manera, el negocio se beneficiaría a nivel económico y mejora de servicio.

II. MARCO TEÓRICO

Trabajos previos

A nivel nacional

Fernández Avalos (2018) en trabajo previo que lleva como título Automatización de Procesos para mejorar las Pruebas de Software en el área de calidad del Banco de Crédito para lograr obtener el grado académico de maestro en Ingeniero de Sistemas, trabajada en la Universidad Cesar Vallejo con sede Lima en Perú. La exploración tiene como propósito saber en qué medida va a mejorar la Automatización de Procesos desde adentro de las Pruebas de Programa en el sector de calidad del Banco de Crédito, ya que el aplicativo SAP que se usa, crea en el proceso de pruebas de programa unas estimaciones por horas, minutos y segundos los cuales desarrollan costos superiores en concordancia al personal solicitado. De igual modo, el inconveniente se lleva a cabo cerca del conjunto de Calidad de Programa, en donde se impulsa un tiempo definido y comienza la organización de la data, dejando claro el tiempo de ejecución del aplicativo a certificar. La metodología empleada es de es cuantitativa, debido a la recolección de datos para evidenciar las hipótesis. La técnica empleada para la recolección de datos fue de fichas de observación. El cual fue aplicado directamente a los trabajadores del área de calidad de software del Banco de Crédito. El estudio fue aplicado con diseño pre experimental. La población fue de un total de 32 participantes y aplicaron una muestra de 30 participantes como mínimo. Los resultados obtenidos fueron que la cantidad de datos formada a partir de las pruebas de software, en el pre test, se obtuvo un total de 217.50 segundos. Sin embargo, con la automatización, el post test logro alcanzar un total de 94 segundos, señalando una disminución de 123.5 segundos, un total de 56.78% de mejora. Esta investigación se concluye que al sistematizar los procesos disminuyen el tiempo e incrementa la productividad, aprobando así que la automatización de procesos mejora las pruebas de software en el área de calidad del Banco de Crédito, reduciendo el tiempo de la generación de data de prueba en un 56.78%, incrementando la cantidad de data creada en un 7.71%, consecuentemente, se puede finiquitar que la automatización de procesos permitió la mejora de carácter significativa las pruebas de software.

El mencionado antecedente pudo aportar un entendimiento mejor para el proceso de pruebas de software, teniendo como base un banco, que es un lugar en donde las pruebas deben ser más intensas.

Baldeón (2015) en su trabajo previo, que lleva como título Método para la evaluación de calidad de software basado en ISO/IEC 25000, para lograr obtener el grado académico de maestro en computación y sistemas, el cual contaba como objetivo el de diseñar un método que esté basado en el estándar ISO/IEC 25000-2005. Se utilizó una metodología en base al enfoque cuantitativo, debido a la recopilación de datos y se empleó un diseño de tipo. La población total fue de 28 proyectos financieros, con una muestra también de 28 proyectos financieros para poder aplicar un análisis cuantitativo, y haciendo uso de un instrumento de reporte de errores y bitácora de pases de producción para la obtención de la información. Los resultados finales demuestran que, a la mejora en la calidad del software, se encontró un 95% de confianza en que el método para evaluación de calidad basado en ISO/IEC 25000 puede mejorar la calidad del software, esto se vio demostrado en una menor proporción de retrabajo para que el usuario de la conformidad del software y éste presente una menor proporción de fallos después del pase a producción del software.

El mencionado antecedente aporta con el análisis de un método que permita definir la evaluación de la calidad de software, basada en la ISO 25000, ayudando a mejorar comprensión y aseguramiento del cumplimiento de los requisitos del usuario.

Lazo (2017) en su trabajo previo, que lleva como título Aplicación informática para mejorar las pruebas de software en un banco privado, Lima 2017, para lograr obtener el título profesional de Ingeniero de sistemas e informática, realizado en la Universidad Nibert Wiener – Perú, el cual tuvo como objetivo el de planear un sistema de información para la realización de mejora para las pruebas de software del banco. Ya que el banco se requiere realizar una cantidad considerable de pruebas de sistema, al uso de un aplicativo que pueda gestionar los requerimientos. Lo cual el inconveniente se lleva cerca de los mientras de equipo de certificación de software que conforman el equipo de Certificación de Software. La metodología empleada es de carácter proyectiva, ya que se empleará una iniciativa con misión a ser aplicada más adelante, de esta forma como la utilización de los estándares de la metodología

RUP. Se empleó una técnica de encuesta, a través de una entrevista. El diseño utilizado es experimental, ya que no se manejará ni se somete a cambios a las variables del estudio. La población total fue de 180 miembros del equipo de Certificación de Software y se empleó una muestra de 20 miembros. Se pudo tener resultados lo cual indican que las pruebas de software tuvieron un 30% de nivel eficiente en el desarrollo del software. Al final se puede concluir, de que los bancos privados, los procesos de las pruebas de software tienen un nivel regular de desempeño, en la cual se consiguió una falta de comunicación entre áreas, poca participación del área de desarrollo de aplicaciones, los cuellos de botella generados por la lentitud de respuesta, el poco reabastecimiento de las ocupaciones entre los integrantes del conjunto y falta de estandarización de procesos, tuvieron problemas sobre el cliente

El mencionado antecedente aporta la importancia de una implementación del sistema de información frente a actividades manuales y repetitivas, que restringe la evolución del organismo. También se extrajo algunas definiciones del marco teórico.

Patiño & Suarez (2014) en su trabajo previo, que lleva como título Optimización del Proceso de Pruebas de Software, ejercido en la Universidad Peruana de Ciencias Aplicadas, el cual tuvo como objetivo principal el de poder mejorar el proceso de pruebas de software. La investigación cuenta con un diseño pre experimental, el cual permitió realizar el análisis con respecto al desarrollo de software especialmente en la calidad, en el que se demostró el grado en los artefactos que estas puedan conformar un producto de software. Se logró implementar un proceso eficientemente sobre las pruebas de softwares, organizado a través de una política de calidad establecida para el desarrollo de software basada en normas internacionales ISO y en los estándares de IEEE, se empleó una dimensión productividad, en el cual, se logró obtener como resultado un total de 29% de incremento en cantidad de data obtenida. Teniendo información para los trabajos previos con respecto a la variable dependiente de Pruebas de Software, verificaciones en los productos entregados al usuario con el fin de brindarle una mejor satisfacción y poder medir la productividad de data generada. La investigación concluye, que, a pesar de la implementación de la propuesta de mejora, esta pueda incidir en un aumento de costo total, mientras que esta es bien aprovechada esta puede crear un

enorme ahorro a la organización tanto en el mismo desarrollo como a los otros procesos del negocio que son soportados por el mismo.

El citado antecedente dar para conseguir información en relación a la administración de calidad del programa, el cómo y en qué circunstancia aplicarlo.

A nivel internacional

Espinosa (2016) en su trabajo previo, que lleva como título Desarrollo De Un Modelo De Pruebas Y Calidad De Software Para La Empresa Seguros Atlas S.A, para lograr obtener el título de Licenciado en Ciencias de la Información, del cual se tiene como objetivo, el de emplear un modelo de pruebas de calidad con la intención de poder mejorar de una manera eficaz los softwares en cuestión, dando como resultado, una notable mejora para el nivel de calidad del producto final, así como la optimización de los tiempos de entrega. Dado a que los proyecto desarrollados solo la cuarta parte de estas, llegan a ser terminados a tiempo y que cumplen con el presupuesto. La metodología aplicada para la este antecedente es SCRUM. Se tiene un diseño en donde se puede presenciar, el uso de un cronograma de actividades en formato Gantt. La población al cual va destinada para este proyecto, son los desarrolladores del área de sistemas de la organización. Los resultados pudieron lograr demostrar que es altamente posible establecer el modelo de pruebas de calidad de software dentro de un área de información. Se concluye que la investigación logro mejorar el proceso de pruebas de software a través de un modelo que terminó mayormente servible y fue aplicada a algunos proyectos y los resultados lanzaron resultados satisfactorios, por lo cual concluyo que sí se cumple la conjetura citada.

El citado antecedente contribuye para gestionar la creación del programa, modelos de creación de programa (cascada, espiral, incremental y evolutivo), el cual se considerará, adjuntado con la metodología elegida, en el momento de crear el programa que va a proporcionar satisfacción a la circunstancia de esta exploración.

Martínez & Morales (2015) en su trabajo previo, que lleva como título Diseño e implementación de mejoras a la plataforma para la gestión de pruebas de proyectos de software MANTEST, para lograr obtener el título de profesional de Ingeniero de

sistemas e informática, trabajada en la Universidad Tecnológica de Pereira en Colombia, donde se tuvo como objetivo, el de poder diseñar y a la vez implementar posibles mejoras a la plataforma que gestiona las pruebas del proyecto de software – MANTEST. Este antecedente empleó la metodología MANTEST, en donde se realizó una gestión para la documentación generada por el proceso de pruebas dentro del desarrollo de software en un proyecto, de tal modo que se logra visualizar el empleo de la metodología Scrum, el cual puede agilizar la construcción de software, y poder enfocar objetivos establecidos. Teniendo como resultados a lo largo de la secuencia de pruebas un porcentaje de éxito del 85%, dejando un total de 5 incidencias lo semejante en 15% del total de los casos ejecutados, en relación a la productividad se incrementó un 75% a un 80% que equivale un aumento del 5%.

El citado antecedente dar como referencia los resultados que se consiguieron de la utilización de mejoras para el capítulo de discusión.

Stephan (2009), en su trabajo previo, que lleva como título Modelos de prueba y criterios de madurez para la generación automática de pruebas basadas en modelos con máquinas de estado UML, para lograr obtener el grado académico doctor. En el campo de la informática titulada desarrollada en la Universidad Humboldt de Berlín. Tuvo como objetivo combinación de diferentes modelos de prueba, las relaciones de los modelos de prueba y los criterios de cobertura, y la combinación de criterios de cobertura. El tipo de investigación es cualitativa. La obtención de la población de esta investigación es producto de la realización del testeo y de los resultados obtenidos por el modelo de pruebas aplicados, el cual determina el valor de la población. Teniendo como resultado durante la implementación del modelo, que el nivel de madurez obtenido de los resultados lanza un promedio de 0.530 para las pruebas ejecutadas satisfactoriamente.

El citado antecedente dar como referencia el indicador nivel de madurez en los casos de prueba para el capítulo de discusión.

Vaca (2017) en su trabajo previo, que lleva como título Modelo De Calidad De Software Aplicado Al Módulo De Talento Humano Del Sistema Informático Integrado universitario UTN, para lograr obtener el título de Magister en Ingeniería del Software,

el cual tuvo como objetivo, el de poder aplicar y determinar un modelo de calidad de software para el módulo de talento humano, correspondiente al sistema informático integrado universitario perteneciente a la Universidad Técnica del Norte, dado que la problemática parte de la inconsistencia de información, la validación y estandarización para los inputs de ingresos, las cuales deben asegurar información de calidad. En este antecedente se expresa de modo descriptiva y documental, apoyada en métodos, técnicas e instrumentos que permitan el seleccionado y diseño según los objetivos presentados, apoyándose del de la metodología RUP. Se empleo una técnica de encuesta según los miembros de la comisión de evaluación interna, los funcionarios de talento humano y director del departamento de desarrollo tecnológico. El antecedente opta por un diseño documental descriptiva. La población a valorar es de 40 miembros. Con una muestra de un total de 40. Los resultados finales presentan la detección de fortalezas y debilidades del software, generando así mejoras, en donde se presenta un incremento en la calidad del software y consiguientemente de los datos. Se concluye mencionando que la implementación del criterio de calidad del software por medio de modelos o estándares, necesita tiempo y concienciación de los comprometidos, a la vez que constituye la base para aumentar la calidad del SIIU y consecuentemente de sus datos.

Este antecedente contribuye a alimentar la investigación con información metodológica, elementos que constituyen un Sistema de Información y el proceso de evaluación de calidad de producto programa.

Teorías relacionadas al tema

Variable Dependiente: Proceso de pruebas de calidad de software

El proceso de pruebas de calidad es la manera de valorar la calidad de un software y evitar el riesgo de encontrar errores en el software durante el funcionamiento, en donde las pruebas se encargan de validar si el software cumple o no con los requerimientos (ISTQB, 2018, p. 19).

Realizar el aseguramiento de la calidad de un sistema, da como resultado una tarea compleja. Las pruebas radican en validar si el software o parte de ella, cumple con

las condiciones determinadas, para probar si los entregables son las adecuadas. Dichas pruebas tienen como objetivo de descubrir errores que se puedan identificar durante el ensayo (Gómez y Moraleda, 2014, p.20)

Actividades del proceso de Pruebas de Calidad de Software

ISTQB (2018, p.17) indica que no hay un único método de proceso de prueba para un software en el mundo, no obstante, existen conjunto de tareas o actividades, que sin ellas las pruebas serían menos expuestas a lograr los objetivos establecido. Esta aglomeración de actividades, se pueden discutir durante la estrategia de la prueba en una empresa.

Planificación

ISTQB (2018, p.18) indica que planificación de las pruebas, involucra actividades donde estas delimitan los objetivos de las mismas. En esta actividad se definen previos objetivos de las pruebas, y el rumbo para poder lograr concluir los objetivos de la prueba.

Monitoreo y control.

La actividad de monitoreo y control, consiste en comparar constantemente el proceso real con lo planeado, usando las métricas de monitoreo definido en el plan de prueba (ISTQB, 2018, p. 19).

Análisis

Esta tarea consiste en analizar los resultados de las pruebas para capturar características probadas y poder concluir las condiciones de las pruebas. Determina el que probar (ISTQB, 2018, p. 19).

Diseño

Es aquí donde se responde la pregunta de ¿Cómo se va evaluar? (ISTQB, 2018, p. 20).

Implementación

En esta actividad se realiza la creación y/o se completa el sistema de prueba que sera necesario durante la ejecución. Esta incluye una se cuenta de procedimientos para la prueba. Aquí se responde a la pregunta ¿Tenemos todo listo para ejecutar las pruebas? (ISTQB, 2018, p. 21).

Ejecución

En esta actividad se realiza la ejecución de la prueba, según el cronograma establecido durante las anteriores tareas (ISTQB, 2018, p. 21).

Terminación

Durante esta actividad se recopila la información de las pruebas completas, para poder reforzar la experiencia, el sistema de prueba o cualquier información que sera relevante para la empresa (ISTQB, 2018, p. 22).

Dimensiones e indicadores del proceso de pruebas de calidad

Dimensión: Ejecución

Indicador: Nivel de cobertura de las pruebas

La cobertura de las pruebas indica en cómo se están cumpliendo los casos de prueba registrados, dando como entendimiento, que una mayor cobertura indica un buen desarrollo de las pruebas (Freitas Vieira, 2015, p.33) .

Fórmula

$$Cp = CPE / CPR$$

Dónde:

CP: cobertura de las pruebas

CPE: casos de prueba que han sido ejecutados

CPR: casos de prueba a ejecutar requeridos para cubrir todos los requerimientos.

Dimensión: Ejecución

Indicador: Nivel de madurez de las pruebas

Para el indicador nivel de madurez de las pruebas, se establece que este indicador está en la dimensión de ejecución de Sistema, debido a que la medición de la misma se basará en los resultados de las pruebas que se realizarán.

La madurez de las pruebas comprende los casos de pruebas que han dado como resultados un retorno satisfactorio (Freitas Vieira, 2015, p.12) .

Fórmula:

$$MP = CPS / CPR$$

Dónde:

MP: madurez de las pruebas

CPS: casos de prueba que han dado resultados satisfactorios

CPR: casos de prueba diseñados para cubrir todos los requerimientos

Variable independiente: Sistema Web

Dueñas Noguera (2014, p. 114) sostiene que los sistemas web son uno de las nuevas maneras para el usuario de obtener información y que está compuesta por tecnologías que están en constante evolución.

El sistema web es un aglomerado de información como instrumento para el manejo y generación de información, todo esto para el manejo de decisiones de una empresa (Según Forexeco, 2016, p. 46).

Un sistema web es aquello que es instaurado en una plataforma o sistema operativo y se puede utilizar a través de un navegador web. No es necesario instalarse en cada computadora, debido a que el usuario se conecta al servidor alojado del sistema (Báez, 2014, p.43)

Arquitectura MODELO–VISTA–MODELO DE VISTA (MVVM)

Según Peppers (2015) nos especifica que el patrón de diseño MVVM sirve “para separar la interfaz de usuario de la lógica empresarial y aprovechar al máximo el enlace de datos. Las aplicaciones diseñadas de esta manera tienen una capa ViewModel que no depende de sus interfaces de usuario.[...]”

El patrón de diseño MVVM también es muy similar al patrón de diseño MVC discutido en los capítulos anteriores.” (p. 47 - 48).

- **Modelo (Model)**

“Es la lógica de negocio del backend que impulsa la aplicación y cualquier objeto de la empresa que la acompañe. Esto puede ser cualquier cosa, desde hacer solicitudes web a un servidor hasta usar una base de datos de fondo.” (Peppers 2015, p. 48).

- **Vista (View)**

“Esta capa es la interfaz de usuario real que se ve en la pantalla. En el caso del desarrollo multiplataforma, incluye cualquier código específico de plataforma para conducir la interfaz de usuario de la aplicación” (Peppers 2015, p. 48).

- **VistaModelo (ViewModel)**

“Esta capa actúa como pegamento en aplicaciones MVVM. Las capas de ViewModel coordinan operaciones entre las capas de Vista y Modelo. Una capa ViewModel contendrá propiedades que la Vista obtendrá o establecerá, y funciones para cada operación que el usuario pueda realizar en cada Vista.” (Peppers 2015, p. 48).

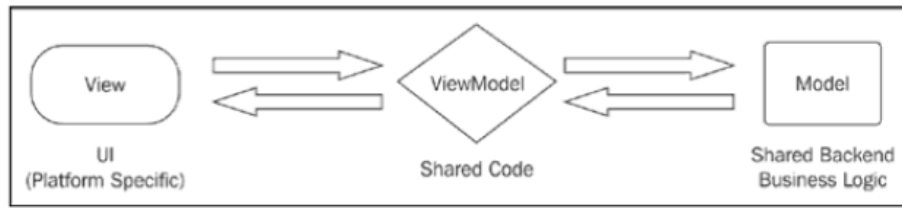


Figura 3 Patrón de diseño MVVM

Patrón de diseño

Según Vora (2017) define que los patrones de diseño “se enfocan explícitamente en el problema dentro del contexto de uso y guían a los diseñadores sobre cuándo, cómo y por qué se puede aplicar la solución, los patrones son prácticos y describen instancias de "buen" diseño al tiempo que incorporan principios y estrategias de alto nivel”.

Así mismo según Vora, se puede interpretar que los patrones de diseño son una forma de reutilización para resolver problemas comunes.

Metodologías para el desarrollo de Sistemas Web

OOHDM (Object Oriented Hypermedia Design Methodology)

El modelo OOHDM es una metodología diseñada para la creación de sistemas multimedia, de las cuales tienen un objetivo de simplificar y al mismo tiempo de poder hacer más eficaz la aplicación. Se definen en estos pasos, donde esta centralmente el diseño (Ricardo y Frank, 2014, p.2).

UWE (UML-based Web Engineering)

Según Maximillians (2016) define que es “un enfoque basado en estándares que no se limita al uso de UML "lingua franca" sino que también utiliza XMI como formato de intercambio de modelo, MOF para metamodelo, los principios guiados por modelo de

El enfoque MDA, el lenguaje de transformación del modelo QVT y XML.” (p. 3).

Rational Unified Process (RUP)

Según Zykov, Sergey (2018) nos define que RUP es “una metodología de proceso iterativa. Es compatible con la elaboración secuencial del producto bajo modelos de desarrollo de software como espiral o incremental. La evaluación de riesgos es un componente importante de la ruptura. Esto es válido para todas las metodologías, especialmente las ágiles” (p. 38).

ELECCIÓN DE LA METODOLOGÍA PARA EL SISTEMA WEB

Para poder elegir de entre las 3 metodologías son la adecuada para el despliegue del sistema web, se optó por los siguientes criterios a evaluar:

Tabla 1 Criterios de evaluación

ÍTEM	CRITERIO	DESCRIPCIÓN
1	Brinda una adaptación a los objetivos del negocio	La metodología tiene la capacidad de adaptarse a los objetivos del negocio.
2	Cubre por completo el ciclo de vida el software	La metodología tiende a imitar el ciclo de vida del software
3	Reduce el costo de los cambios en cualquier etapa del ciclo de vida	La metodología ayuda a reducir costos de implementación en cualquier ciclo de vida del proyecto.
4	Utiliza las mejores prácticas para sistemas webs	La metodología usa en si misma las buenas prácticas de desarrollo para sitios webs
5	Adaptabilidad	La metodología se adapta a cualquier entorno web.
6	Los requisitos pueden	La metodología ayuda a poder cambiar

	cambiar en cualquier etapa del ciclo de vida	los requisitos en cualquier momento sin alterar el ciclo de vida del proyecto.
--	--	--

Fuente: Elaboración propia

EVALUACIÓN DE LA METODOLOGÍA

Es una metodología es una guía de trabajo, destinada a la programación, es decir, lo importante es realizar el producto en un corto tiempo, mediante herramientas y modelos que ayuden a documentar lo avanzado.

Tabla 2 Cuadro comparativo de Metodologías

Experto	METOLOGÍAS		
	OOHDM	UWE	RUP
Dra. Yesenia Del Rosario Vásquez Valencia	12	14	18
Mgtr. Perez Rojas Even Deyser	16	16	117
Mgtr. Ruben Alexander More Valencia	17	17	18
Total	45	47	54

Fuente: Elaboración propia

La tabla anterior (Tabla N.º 2) se apreciar cómo cada evaluador da como resultado adecuado la metodología **RATIONAL UNIFIED PROCESS (RUP)** como la necesaria para poder cubrir los aspectos evaluados y requeridos para este proyecto.

RATIONAL UNIFIED PROCESS (RUP)

Para Hashmi y Choudhury (2020) la metodología RUP es “una metodología de proceso iterativa para el desarrollo basada en arquitectura y casos de uso. La metodología RUP se basa en el lenguaje de modelado unificado (UML). UML es usado para especificación, visualización, construcción y documentación del desarrollo de software” (p. 499). Asimismo, Hashmi y Choudhury (2020) nos menciona que “la

metodología RUP tiene control, puntos clave o críticos a través del desarrollo. En otras palabras, cada fase [...] debe terminar con algún control, puntos clave o críticos donde se sumen los resultados obtenidos y se planifiquen las futuras direcciones en función de los resultados. La metodología RUP tiene artefactos (documentos), modelos y elementos del modelo.” (p. 499).

RUP cuenta con 4 fases:

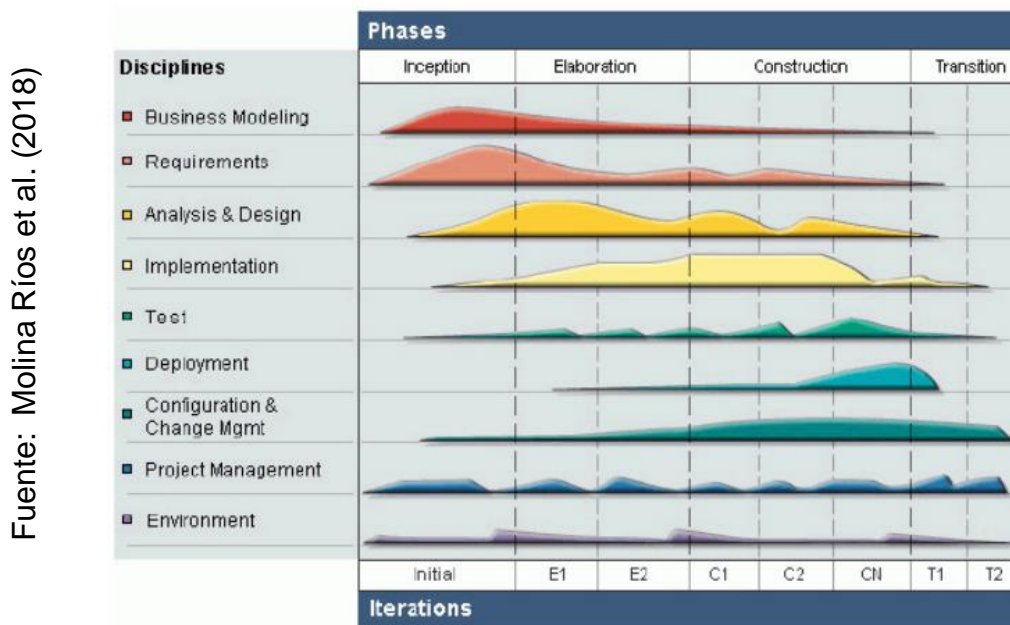


Figura 4 Fases de RUP

Fase de iniciación

Según Ashfa (2019) que nos comenta que la esta fase “es donde se define el alcance del proyecto en el contexto del presupuesto y el esquema del proyecto. Se establece una carta del proyecto que consta de factores de éxito, caso de negocios y previsión financiera.” (p. 375).

Fase de elaboración

Según Ashfa (2019) “en la fase de elaboración, el diseño y la arquitectura del producto toman forma. El hito después de la fase de elaboración es conocido como la piedra angular de la arquitectura del ciclo de vida.” (p. 375).

Fase de construcción

Según Ashfa (2019) “La fase de construcción es donde el diseño del software se convierte en código de software. Es la fase más laboriosa en todo el proyecto de software. Para encargarse del gran requisito de creación de código, se puede utilizar ingeniería concurrente y otras técnicas para el desarrollo rápido de aplicaciones.” (p. 375).

Fase de transición

Ashfa (2019) nos comenta que “una vez que el producto está hecho, debe estar disponible para su implementación para que los usuarios finales puedan usarlo. Sin embargo, antes de eso, la aplicación debe ser probada en beta para garantizar que no haya errores críticos en la aplicación. La aplicación se evalúa para validar si el nivel de calidad cumple con el acuerdo realizado durante la fase inicial. La capacitación del usuario también debe organizarse.” (p. 375).

VUE.JS

Vue es un marco progresivo para la construcción de interfaces de usuario, diseñado desde cero para ser adaptable, así mismo es capaz de impulsar aplicaciones sofisticados de una sola página, combinando consigo mismo herramientas modernas y bibliotecas de soporte (Dayan, 2020).

Componentes de VUE.JS

Según Maniatis, Kyriakidis y Styde (2017) nos define los componentes como “una de las características más poderosas de Vue.js. Ellos te ayudan a extender elementos básicos de HTML para encapsular código reusable. En un nivel alto, los componentes son elementos personalizados a los que el compilador de Vue.js asignará un comportamiento específico” (p. 92).

Lifecycle Hooks (Ciclo de vida de VUE.JS)

Para Maniatis, Kyriakidis y Styde (2017) los lifecycle hooks son funciones que se ejecutan cuando suceden eventos relacionados a Vue.

Tabla 3 Lifecycle Hooks

Hook	Se ejecuta
beforeCreate	Después de la inicialización de la instancia y antes de la observación de datos y la configuración de evento/observador.
created	Después de que la instancia es creada
beforeMount	Justo antes de comenzar el montaje
mounted	Después de que la instancia ha sido montada al DOM.
beforeUpdate	Cuando los datos cambian, antes de que el DOM virtual se vuelva a renderizar y parchear
updated	Después de que un cambio en los datos cause que el DOM virtual se vuelva a renderizar y parchear
activated	Cuando un componente kept-alive es activado
deactivated	Cuando un componente kept-alive es desactivado
beforeDestroy	Justo antes de que una instancia de Vue sea destruida
destroyed	Después de que una instancia de Vue ha sido destruida.

Fuente: Maniatis, Kyriakidis y Styde (2017)

NUXT.JS

Según Bush. et al. (2019) nos define que “Nuxt.js es un marco que lo ayuda a construir aplicaciones Vue.js procesadas por el servidor fácilmente. Resume la mayor parte de la configuración compleja involucrada en la gestión de cosas como datos asíncronos, middleware y enrutamiento. Es similar a Angular Universal para Angular y Next.js para React.” (p. 362)

Single File Components (SFC)

Los componentes de archivo único son archivos con una extensión .Vue que contienen la definición completa de un solo componente y se pueden importar a su aplicación Vue js. Los SFC's simplifican la creación y el uso de componentes, y algunos con una variedad de otros beneficios que pronto exploraremos

SINGLE PAGE APPLICATION (SPA)

Fink y Flatow (2014) nos define que una single page application “es una aplicación web que utiliza solo una web HTML como Shell para todas las páginas web de la aplicación y cuyas interacciones con el usuario final se implementan mediante JavaScript, HTML y CSS.” (p. 3)

API REST

Según Gilchrist (2015) nos define que “una API es el acrónimo de Application Programming Interface (Interfaz de programación de aplicaciones) y, como tal, se compone de una interfaz programable para una aplicación: permite que una aplicación cliente externa se comuniquen e integre con el recurso de servicio” (p.32).

NODE JS

Azaustre (2016) define que “Node es un entorno de ejecución para JavaScript en servidor a partir del motor V8 de renderizado de JavaScript que utiliza el

navegador Chrome de Google. Node nos facilita la creación de aplicaciones de servidor altamente escalables.” (p. 10)

EXPRESS JS

Según Azat (2014), nos explica que “express.js es un marco web que se basa en el núcleo del módulo **http** de Node.js y conecta componentes. Esos componentes se denominan middlewares. En otras palabras, los desarrolladores son libres de elegir las bibliotecas que necesiten para un proyecto en particular que les brinde flexibilidad y alta personalización” (p. 2).

MYSQL

Según Combaudon (2017) nos menciona que “MYSQL es el sistema de gestión de base de datos Open Source más popular del mundo y es conocido por su rendimiento y fiabilidad. [...] Así, hoy en día, la inmensa mayoría de sitios de gran densidad de tráfico, como las redes sociales [...], ha alcanzado su nivel de rendimiento mediante el uso intensivo de MySQL”

SEQUELIZE JS

Según Bojinov, Herron and Resende (2018) “el módulo de Sequelize está basado en Promise, ofrece características ORM sólidas y bien desarrolladas, y puede conectarse con SQLite 3, MySQL, PostgresSql, MariaDB y MSSQL. Debido a que Sequelize se basa en Promise, se ajustará naturalmente con el código de aplicación basado en Promise que estamos escribiendo.” (p. 242)

ORM

Según Mittal (2017) “ORM o Mapeadores de Objetos Relacionales es un proceso de mapeo entre objetos y sistemas de bases de datos de relaciones. Un ORM actúa como una interfaz entre dos sistemas. ORM ofrece ventajas para los desarrolladores, [...] ahorro de tiempo y esfuerzo [...]. El código es robusto en lugar de redundante. ORM ayuda a gestionar consultas para múltiples tablas de manera efectiva.

ECMASCRIPT

Prusty (2015) nos menciona que ECMAScript “es un lenguaje de secuencias de comandos estandarizado por Ecma International en la especificación Ecma-262. Los lenguajes de secuencias de comandos como JavaScript, JScript y ActionScript son superconjuntos de ECMAScript, al definir más objetos y métodos, las características principales de estos idiomas son las mismas como ECMAScript.” (p. 11)

ZEIT NOW

Para Arora, Hennessy and Noring (2018) Zeit - NOW “es un servicio multi-nube que permite la implementación de aplicaciones global en tiempo real directamente desde la CLI. NOW trabaja con aplicaciones que implementan correctamente package.json o un Dockerfile.” (p. 428)

III. METODOLOGÍA

3.1. Tipo y diseño de investigación

Método de investigación: Hipotético Deductivo

Moisevich y Fedorovich (2018, p. 290) indica que este método de investigación consta en realizar un planteamiento de verdades en base a la hipótesis, y poder comprobarlas en base a la deducción, en principio con las conclusiones obtenidas y el enfrentamiento de los hechos últimos.

Para la actual investigación se determinó en usar el método hipotético deductivo, debido a que las evaluaciones y el análisis se podrá confirmar las hipótesis determinadas.

Tipo de estudio

Explicativa

Según Díaz y Narvárez (2016, p. 118) nos indica que la indicación explicativa, es aquella que tiene como fin, el hallazgo de leyes fundamentales que tienen la posibilidad de hablar del porque hay cuales y tales características, y del porqué, son consideradas características que tienen la posibilidad de asociarse entre si

Experimental

Para Baena (2014, p. 14) se representa por medio del manejo de una variable en base de prueba no comprobada, en escenarios controladas, con el objetivo de detallar el modo o por qué se genera un escenario o un suceso muy particular.

Aplicada

Según Lozada (2014), la investigación aplicada “tiene como objeto el estudio de un problema destinado a la acción. La investigación aplicada puede aportar hechos nuevos... sí proyectamos suficientemente bien nuestra investigación aplicada, de modo que podamos confiar en los hechos puestos al descubierto, la nueva información puede ser útil y estimable para la teoría.

Según Lozada (2014, p. 28), la exploración aplicada, cuanta como fin, un análisis de un inconveniente designado a la acción. En esta se puede dar escenarios nuevos. Si durante la investigación aplicada , se logra proyectar confiar en los escenarios, la información resultante, puede llegar a ser útil y amigable para la teoría.

Para la de hoy, esta investigación tiene como tipo de estudio la investigación explicativo experimenta y aplicada, dado que se construirá un sistema web para el proceso de control de las pruebas, el cual va poder corregir la eventualidad que hay en el organismo NextPerience.

Diseño de estudio: Pre experimental

El diseño preexperimental es el estudio de un caso de una sola ejecución en la que los datos están sometidos a una condición y se mide posteriormente si existió algún cambio (Kaufman, 2019, p. 1).

El diseño de estudio es un diseño pre experimental ya que se prueba realizar un sistema web de control de pruebas empleando el modo pretest y post-test.

Fuente: Sreejesh,
Mohapatra y
Anusree 2016



Figura 5: Diseño del estudio

Donde:

EG: Grupo experimental: casos de pruebas

O1: Resultados de las pruebas previas: Pre-test

X: Variable independiente: Sistema Web

O2: Resultados de las pruebas posteriores: Post-test.

En la Figura 4, se logra apreciar el proceso que el diseño pre experimental compone. EG es compuesta por el grupo de estudios, el cual está compuesta por los casos de

pruebas para esta investigación, dicho grupo de estudio pasa por una prueba previa, el cual está representada por O_1 , la cual son los resultados obtenidos por el cálculo de los indicadores de cobertura y madurez de las pruebas. Posteriormente el grupo de estudio se le administraran un tratamiento, el cual le es representado por X , dicho tratamiento es la variable independiente de la investigación, que en su caso es el sistema web a desarrollar. Finalmente, el grupo de estudio después de haber pasado por el tratamiento se vuelve a realizar el análisis por los indicadores de cobertura y madurez de las pruebas, logrando así un nuevo resultado O_2 , los cuales son sometidos a una comparación con O_1 .

3.2. Variables y Operacionalización

Definición Conceptual

Variable Independiente (VI): Sistema Web

Un sistema web es aquello que es desarrollado en una interfaz o S.O. y se puede usar a través de un navegador web. No es requisito instalarse en cada PC, ya que el usuario se conecta al servidor alojado del sistema (Báez ,2014, p.43)

Variable Dependiente (VD.): Proceso de pruebas de calidad de software

Según Granados La Paz (2014, p.104) nos define que son un carácter, estos caracteres son medibles, según es dependiente el programa y el sistema donde se ejecuta. No es semejante, la calidad esperada de un sistema ocasional, construida para una exclusiva funcionalidad, que la de un sistema que debe trabajar sin algún error, a lo largo de bastante más de 5 años.

Definición Operacional

Variable Independiente (VI): Sistema Web

Es una utilidad que facilita el registro de las pruebas de calidad llevadas a cabo, de esta forma como las pruebas ejecutadas con resultados satisfactorios para la compañía NextPerience, la cual en esta exploración se busca hacer mejor el

desarrollo de pruebas de calidad, que en la actualidad se hace de manera rudimentaria por medio de sistemas no premeditados al desarrollo.

Variable Dependiente (VD.): Proceso de pruebas de calidad de software

Son todos aquellos procesos o actividades cuya ejecución nos dé como resultado, conocer la calidad del software, así como reconocer los posibles fallos que puedan existir en un corto, medio o largo plazo después de la entrega del software.

Tabla 4: Operacionalización de variable

Variable	Descripción Conceptual	Definición operacional	Dimensión	Indicador	Escala de medición
Variable independiente (VI): Sistema Web	Un sistema web es aquello que es desarrollado en una interfaz o S.O. y se puede usar a través de un navegador web. No es requisito instalarse en cada PC, ya que el usuario se conecta al servidor alojado del sistema (Báez ,2014, p.43)	Es una utilidad que facilita el registro de las pruebas de calidad llevadas a cabo, de esta forma como las pruebas ejecutadas con resultados satisfactorios para la compañía NextPerience, la cual en esta exploración se busca hacer mejor el desarrollo de pruebas de calidad, que en la actualidad se hace de manera rudimentaria por medio de sistemas no premeditados al desarrollo			
Variable dependiente (VD.): Proceso de pruebas de calidad de software	Según Granados La Paz (2014, p.104) nos define que son un carácter, estos caracteres son medibles, según es dependiente el programa y el sistema donde se ejecuta. No es semejante, la calidad esperada de un sistema ocasional, construida para una exclusiva funcionalidad, que la de un sistema que debe trabajar sin algún error, a lo largo de bastante más de 5 años.	Son todos aquellos procesos o actividades cuya ejecución nos dé como resultado, conocer la calidad del software, así como reconocer los posibles fallos que puedan existir en un corto, medio o largo plazo después de la entrega del software.	Fase de Ejecución	Nivel cobertura de las pruebas	Razón
				Nivel madurez de las pruebas	Razón

Fuente: Elaboración Propia

Tabla 5: Operacionalización de indicadores

Dimensión	Indicador	Descripción	Instrumento	Unidad De Medida	Fórmula
Pruebas de Sistema	Nivel de cobertura de las pruebas	La cobertura de las pruebas indica en cómo se están cumpliendo los casos de prueba registrados, dando como entendimiento, que una mayor cobertura indica un buen desarrollo de las pruebas (Freitas Vieira, 2015, p.33)	Ficha	Unidad	$Cp = CPE / CPR$ <p>Dónde:</p> <p>CP: cobertura de las pruebas</p> <p>CPE: casos de prueba que han sido ejecutados</p> <p>CPR: casos de prueba a ejecutar requeridos para cubrir todos los requerimientos.</p>
Pruebas de Sistema	Nivel de madurez de las pruebas	La madurez de las pruebas comprende los casos de pruebas que han dado como resultados un retorno satisfactorio (Freitas Vieira, 2015, p.12)	Ficha	Unidad	$MP = CPS / CPR$ <p>MP: madurez de las pruebas</p> <p>CPS: casos de prueba que han dado resultados satisfactorios</p> <p>CPR: casos de prueba diseñados para cubrir todos los requerimientos</p>

Fuente: Elaboración propia

3.3. Población (criterios de selección), muestra y muestro, unidad de análisis.

Población

Arias, Villasis y Miranda (2016, p. 202) la población en estudio es un aglomerado de casos, definido, con limite y alcanzable, generara el lider para la selección de la muestra, donde se cumpla la serie de criterios preestablecidos

En esta presente investigación se hará uso de una población total de 2500 casos de pruebas agrupados en 20 reportes de pruebas, distribuidas en el rango de 1 mes.

Muestra

Arias, Villasis y Miranda (2016, p. 206) Es el numero especifico de comprometidos, donde estas conformaran lo primordial a fin de poder lograr los objetivos desde el inicio. Esta estimación se calcula por medio de las fórmulas matemáticas o estadísticos. Este resultado, es completamente distinto para cada investigacion, y va depender de su diseño conjetura, grupos a estudiar y la escala de medición de cambiantes.

Operación del tamaño de la muestra.

$$n = \frac{Z^2 N}{Z^2 + 4(N - 1)(EE^2)}$$

Donde :

N = Tamaño de muestra

Z = Nivel de confianza al 95% (1.96) elegido para esta investigación

N = Población total del estudio

EE=Error estimado (al 5%)

Empleando la fórmula:

$$n = \frac{(1.96)^2 * 2500}{(1.96)^2 + 4 * (2500 - 1) * (0.05^2)}$$

$$n = \frac{3.8416 * 2500}{3.8416 * 9996(0.0025)}$$

$$n = 333.4 \cong 333$$

El tamaño de la muestra estuvo conformado por 333 casos de prueba, agrupados en 20 reportes de pruebas durante el periodo de un mes.

Muestreo: Aleatorio - simple

El tipo de muestreo a usar, es el muestreo no probabilista, puesto que la gente a investigar es de carácter finito y de que todas ellas tienen la misma posibilidad de ser elegida.

Muestreo probabilístico

Diaz, Ojeda y Valderrábanos (2016) nos “define que este tipo de muestreo los datos no se seleccionan de manera aleatorio, sino que son seleccionadas de manera que el investigador sea representativo.” (p. 20).

3.4. Técnicas e instrumentos de recolección de datos.

Hernández, Fernández, Baptista (2016, p. 466) nos indica que son elementos que utilizan en una investigación para juntar información y/o datos relacionados con el tema a estudiar. Por estos medios, el investigador obtiene información que después va a poder utilizar y mostrar en su marco teórico

Técnicas

Cuauro (2014, p. 40) nos indica que es una aglomeración de entendimientos o instrumentos para la obtención de resultados deseados. Esta puede ser provechada en algún tema del a ciencia.

La técnica nos ayudara a decidir qué instrumento nos es más útil y fácil para la investigación, el cual se puede aplicar en cualquier ámbito.

Fichaje

Hortencia (2017, p. 17) nos indica que constituye una técnica que facilita amontonar datos, agarrar ideas y organizarlo todo en un fichero.

La técnica logrará acumular datos para los indicadores nivel de cobertura de las pruebas y madurez de las pruebas

Instrumentos

Cuauro (2014, p. 40) es el intermediario en donde se ocupará de registrar la información obtenida a lo largo de la investigación. Es una aglomeración de efectos tangibles, donde facilita el registro, la conservación y el plasmado de todo lo investigado, por medio de las técnicas usadas que lograron la recolección de información.

El instrumento sera fase inicial de la investigación, el cual guardara los datos almacenados del grupo de estudio observado.

Ficha de registro

En esta investigación se fabricó una ficha de registro para el indicador Nivel de Cobertura de las Pruebas, donde se registró los resultados obtenidos los casos de pruebas requeridos (Anexo 2) y se contrastó con los resultados esperados de los

casos de pruebas ejecutadas, y se matizó la cobertura de las pruebas de los casos de pruebas durante 20 días.

Se desarrollo una ficha de registro para el indicador de Nivel de Madurez de las pruebas, en donde se registró los casos de pruebas requeridos (anexo 3) y se contrastó con los resultados as pruebas satisfactorias, y se matizó la madurez de las pruebas en los casos de pruebas durante 20 días.

Tabla 6: Recolección de datos

Dimensión	Indicador	Descripción	Técnica	Instrumento	Formula
Ejecución	Nivel cobertura de las pruebas	Se evaluará la cobertura de los casos de pruebas	Fichaje	Ficha de registro	$CP = CPE / CPR$
Ejecución	Nivel madurez de las pruebas	Se evaluará la madurez de los casos de pruebas	Fichaje	Ficha de registro	$MP = CPS / CPR$

Fuente: Elaboración propia

3.5. Procedimientos

Validez

Gavagnin (2015, p. 54) nos indica que la validez, es el nivel donde un instrumento mide la variable que se quiere medir.

Tabla 7: Validación de la prueba de expertos

Experto	Nivel cobertura de las pruebas	Nivel madurez de las pruebas
Mg. Ormeño Rojas, Robert	85%	85%

Mg. Galvez Tapia Orleans Moisés	80%	80%
Mg. Huarote Zegarra Raúl	90.5%	89.4%
TOTAL	88.5% (Aceptable)	84.8% (Aceptable)

Elaboración Propia

En la Tabla N.º 9 se logra apreciar el total realizado a los indicadores: Nivel de Cobertura de las pruebas, así como el Nivel de madurez de las pruebas, dan como resultado un porcentaje aceptable, el cual, en términos simples, validan el instrumento para que pueda ser aplicado.

Confiabilidad

Zamora (2015, p.156) Es la seguridad de las evaluaciones resultadas por los individuos en diferentes escenarios o con diferentes conjuntos de equivalentes.

La confiabilidad se representa a través del coeficiente de correlación, donde sus valores oscilan entre cero (0) y uno (1.00) (Ruiz, 2015, p. 11).

Tabla 8 Interpretación del coeficiente de confiabilidad

Rangos	Magnitud
0,81 a 1,00	Muy Alta
0,61 a 0,80	Alta
0,41 a 0,60	Moderada
0,21 a 0,40	Baja
0,01 a 0,20	Muy baja

Fuente: Ruiz (2015)

Según Ruiz (2015, p. 11) un valor de coeficiente, se considera aceptado siempre y cuando este por menos del límite superior a 0,80 de la categoría Alta. Sin embargo,

la regla se exhuma dependiendo del propósito, tipo de instrumento y del tipo de confiabilidad en la que se trabaje.

Tabla 9 Resultado de confiabilidad para el nivel de cobertura de las pruebas

Correlaciones			
		Test_CP	Retest__CP
Test_CP	Correlación de Pearson	1	,621**
	Sig.(bilateral)		,003
	N	20	20
Retest__CP	Correlación de Pearson	,621**	1
	Sig. (bilateral)	,003	
	N	20	20

Fuente: Elaboración propia

El indicador de nivel de cobertura de las pruebas, dio como consecuencia un total de 0,689 indicando un valor alto aceptable. Por lo tanto, el instrumento para el indicador de nivel cobertura de las pruebas es confiable. Donde se visualiza en la tabla 9.

Tabla 10 Resultado de confiabilidad para el nivel de madurez de las pruebas

Correlaciones			
		Test_MP	Retest_MP
Test_MP	Correlación de Pearson	1	,669**
	Sig.(bilateral)		,001
	N	20	20
Retest_MP	Correlación de Pearson	,669**	1
	Sig. (bilateral)	,001	
	N	20	20

Fuente: Elaboración propia

El indicador de nivel de madurez de las pruebas, dio como resultado un total de 0,614 indicando un valor alto aceptable. Por lo tanto, el instrumento para el indicador de nivel madurez de las pruebas es confiable. Como se aprecia en la tabla 10.

3.6. Métodos de análisis de datos

El método de análisis usada para la investigación es de modo cuantitativo, pues según Venemedia (2019, p. 1) es aquel que se apoya de los numero para investigar y corroborar datos e información, intentando especificar y delimitar la correlación o asociación.

Hipótesis de la investigación 1

a. Hipotesis Especifica (HE1)

El sistema web aumenta la cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience.

b. Indicador 1: Nivel de cobertura

INCa: Nivel de cobertura antes de utilizar el sistema web

INCd: Nivel de cobertura antes de utilizar el sistema web

c. Hipotesis Estadístico 1:

Hipótesis Nula H_{10} : El sistema web no aumenta la cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

$$H_{10}: INCa \geq INCd$$

La hipótesis nula (H_{10}) se hará efecto si los resultados obtenidos del Indicador de Nivel de Cobertura de las pruebas antes del utilizar el sistema web (INCa) son mayores a los datos del indicador de Nivel de Cobertura de las pruebas después del sistema web (INCd).

Hipótesis Alternativa H1a: El sistema web aumenta la cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

$$H1_a: INCa < INCd$$

La hipótesis alternativa ($H1_a$) se hará efecto si los resultados obtenidos del Indicador de Nivel de Cobertura de pruebas antes del utilizar el sistema web ($INCa$) son menores a los datos del indicador de Nivel de Cobertura de las pruebas después del sistema web ($INCd$).

Hipotesis de la investigación 2

a. Hipotesis Especifica (HE2)

El sistema web aumenta la madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

b. Indicador 1: Nivel de cobertura

INMa: Nivel de madurez de las pruebas antes de utilizar el sistema web

INMd: Nivel de madurez de las pruebas antes de utilizar el sistema web

c. Hipotesis Estadístico 2:

Hipótesis Nula $H2_0$: El sistema web no aumenta la madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

$$H2_0: INMa \geq INMd$$

La hipótesis nula ($H2_0$) se hará efecto si los resultados obtenidos del Indicador de Nivel de Madurez de las pruebas antes del utilizar el sistema web ($INMa$) son mayores a los datos del indicador de Nivel de Madurez de las pruebas después del sistema web ($INMd$).

Hipótesis Alternativa H2a: El sistema web aumenta la madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

H2a: INMa < INMd

La hipótesis alternativa (H2a) se hará efecto si los resultados obtenidos del Indicador de Nivel de Madurez de las pruebas antes del utilizar el sistema web (INMa) son menores a los datos del indicador de Nivel de Madurez de las pruebas después del sistema web (INMa).

Nivel de Significancia

Margen de error: $X = 0.05 = 5\%$ (error)

Nivel de confiabilidad: $1 - X = 0.95 = 95\%$

Figura 6: Estadística de Prueba

$$T = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

T- Student

Donde:

n1 = Tamaño de la muestra pre test

n2 = Tamaño de la muestra post Test

S1 = Varianza-pre test

S2 = Varianza-post test

X = Media-pre test

Y = Media-post test

Región de Rechazo

Región Rechazo es $t = t_x$

Donde t_x es tal que:

$$P [T > t_x] = 0.05$$

Donde t_x = Valor Tabular

Luego RR: $t > t_x$

Varianza

$$S^2 = \frac{\sum(\bar{x} - \bar{x})^2}{K - 1}$$

Dónde:

x = Numero de la muestra

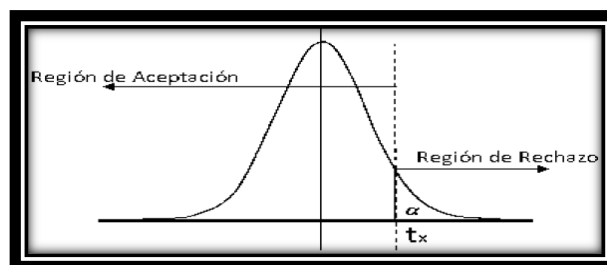
k = Muestra

Análisis de resultado:

Los resultados que se consiguieron van a ser comparados por medio de la prueba de T, donde se verificará la formulación de la conjetura, dejando claro el rechazo o aceptación de la conjetura nula. (Valenzuela, 2011, p. 21)

Figura 7: T- Student

Guisande (2006)



Grafica del t- Student

La tabulación, examen y la interpretación de los datos seleccionados se han realizado por medio de utilidades SPSS, Con el acompañamiento y la asesoría del profesional en el sector de estadística.

3.7. Aspectos éticos

Los datos representados en esta investigación fueron recopilados del grupo de control y en fase de prueba de la investigación y se procesaron de forma correcta sin ninguna adulteración, ya que estos datos están basados en el instrumento aplicado a los grupos de estudio.

Se siguió una investigación de acuerdo a los lineamientos y estatutos de la universidad Cesar Vallejo.

El trabajo que se realizó es original y no existe otro semejante en la institución de estudio de exploración.

Para deducir, los resultados de la exploración se afirman que no son adulteradas o plagiadas de otras indagaciones.

IV. RESULTADOS

Análisis descriptivo

En la investigación se aplicó un Sistema Web para considerar el Nivel de cobertura de pruebas y Nivel de madurez de pruebas en el desarrollo de pruebas de calidad, para eso se empleó un Pre-Test que permitió comprender los resultados iniciales del indicador; consecutivamente se implementó el Sistema Web y de nuevo se registró el nivel de cobertura de pruebas y el nivel de madurez de pruebas en el desarrollo de pruebas de calidad. Los resultados de estas medidas se observan en las tablas 11 y 12 .

- **INDICADOR: Nivel de Cobertura de las pruebas**

Los resultados descriptivos del nivel de cobertura de las pruebas de estas medidas se observan en la Tabla 11 .

Tabla 11 Medidas descriptivas del Nivel de cobertura de las pruebas antes y después de implementar el Sistema Web Open Source Vue

Estadísticos descriptivos					
	N	Mínimo	Máximo	Media	Desv. Desviación
PreTest_CP	20	.56	.78	.6750	.05643
PostTest_CP	20	.72	.94	.8170	.05676
N válido (por lista)	20				

Fuente: Elaboración propia

Para el nivel de cobertura de las pruebas en el proceso de pruebas de calidad en el pre-test se consiguió un valor de 0.6750, en oposición al post-test, que fue de 0.8170 de la misma forma que se visualiza en la figura 8; esto exhibe una distingue antes y luego de la utilización del Sistema Web; de esta forma mismo; el nivel de cobertura de las pruebas mínima es de 0.56 antes y 0.72 posterior a la utilización del Sistema Web (Tabla 11).

A la dispersión del nivel de cobertura, el pre-test se pudo una variabilidad de 0.5643; concurrentemente, en el post-test se tuvo un valor de 0.5676.

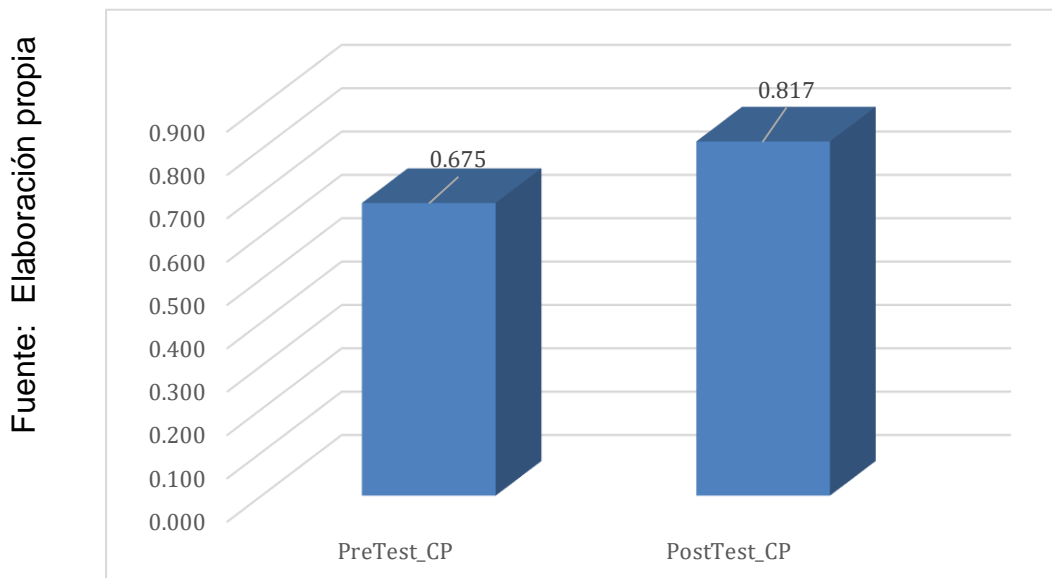


Figura 8 Nivel de madurez de las pruebas antes y después de implementar el Sistema Web Open Source Vue

- **INDICADOR : Nivel de madurez de las pruebas**

Los resultados descriptivos del nivel de madurez de las pruebas de estas medidas se observan en la Tabla 12 .

Tabla 12 Medidas descriptivas del Nivel de madurez de las pruebas antes y después de implementar el Sistema Web Open Source Vue

Estadísticos descriptivos

	N	Mínimo	Máximo	Media	Desv. Desviación
PreTest_MP	20	.35	.65	.4790	.07483
PostTest_MP	20	.59	.82	.7000	.06113
N válido (por lista)	20				

Fuente: Elaboración propia

En la situación del nivel de madurez de las pruebas en el proceso de pruebas de calidad en el pre-test se consiguió un valor de 0.4790, en oposición al post-test, que fue de 0.7000 de la misma forma que se visualiza en la figura 9; esto revela una distingue antes y luego de la utilización del Sistema Web; de esta forma mismo; el nivel de cobertura de las pruebas mínima fue de 0.35 antes y 0.59 posterior a la utilización del Sistema Web (Tabla 12).

A la dispersión del nivel de cobertura, en el pre-test se pudo una variabilidad de 0.7483; no obstante, en el post-test se tuvo un valor de 0.6113.

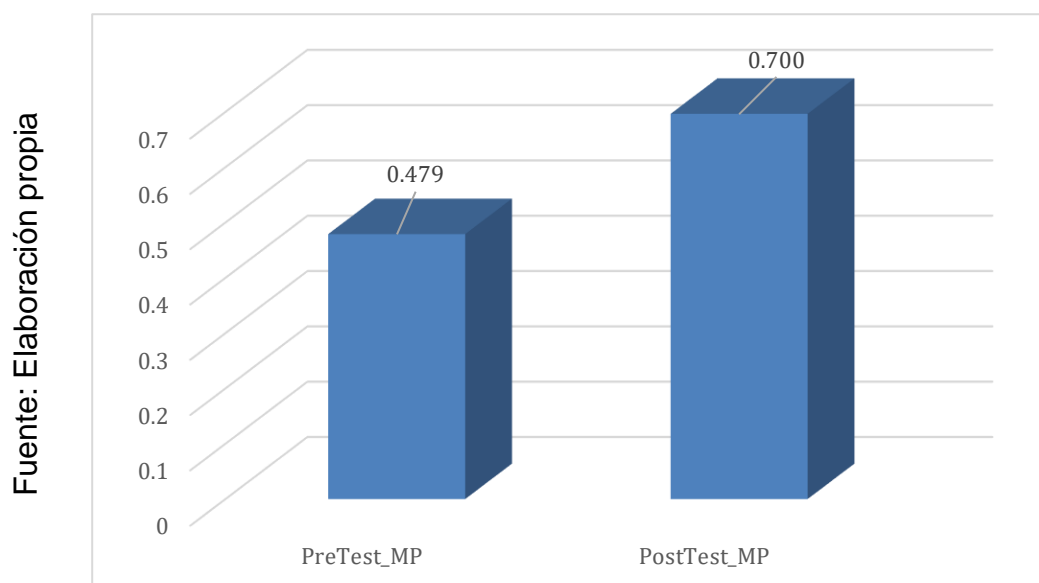


Figura 9 Nivel de madurez de las pruebas antes y después de implementar el Sistema Web Open Source Vue

1.1. Análisis Inferencial

Prueba de Normalidad

A continuación, se realizó la prueba de normalidad para el indicador de nivel de cobertura de las pruebas y nivel de madurez de las pruebas usando el método Shapiro-Wilk, consecuentemente a que el volumen de la muestra está conformado por 20 fichas y es menor a 50, como lo indica Hernández, Fernández y Baptista (2014, p. 376). La prueba se realizó llenando los datos de cada indicador en el software SPSS versión 25.0 para un nivel de confianza del 95%, frente las siguientes situaciones:

Si :

Sig. < 0.05 adopta una distribución no normal.

Sig. \geq 0.05 adopta una distribución normal.

Dónde:

Sig. : p-valor o nivel crítico del contraste.

Los resultados fueron los siguientes

Indicador : Nivel de Cobertura de las pruebas

Con el propósito de elegir la prueba de hipótesis; los datos fueron sujetos a la validación de la organización, exactamente si los datos del nivel de cobertura de las pruebas describían con organización habitual.

Tabla 13 Prueba de Normalidad del Nivel de cobertura de las pruebas antes y después de implementar el Sistema Web Open Source Vue

	Pruebas de normalidad		
	Shapiro-Wilk		
	Estadístico	gl	Sig.
PreTest_CP	,974	20	,827
PostTest_CP	,974	20	,841

Fuente: Elaboración propia

Como se puede ver en la tabla 13 los resultados del estudio revelan que el sig. del nivel de cobertura fue de 0.827 en el Pre-test, donde el valor es más grande que 0.05. Por lo cual, el nivel de cobertura distribuye normalmente. Los resultados de las pruebas del Post-Test detallan que el Sig. del nivel de cobertura de las pruebas es de 0.841, cuyo valor es más grande que 0.05, por lo cual exhibe que el nivel de coberturas de las pruebas se distribuye normalmente. Lo que, valida la distribución normal de los datos de la exhibe. Se consigue ver en las figuras 10 y 11 .

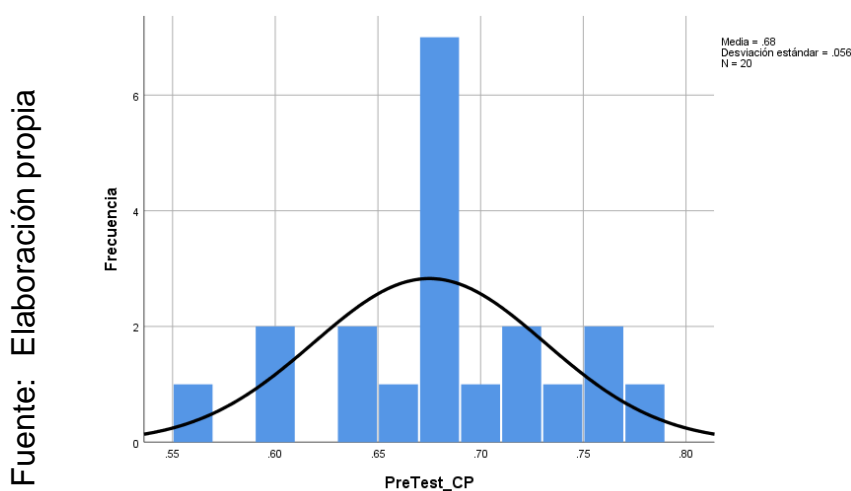


Figura 10 Prueba de Normalidad del Nivel de cobertura de las pruebas antes de implementar el Sistema Web Open Source Vue

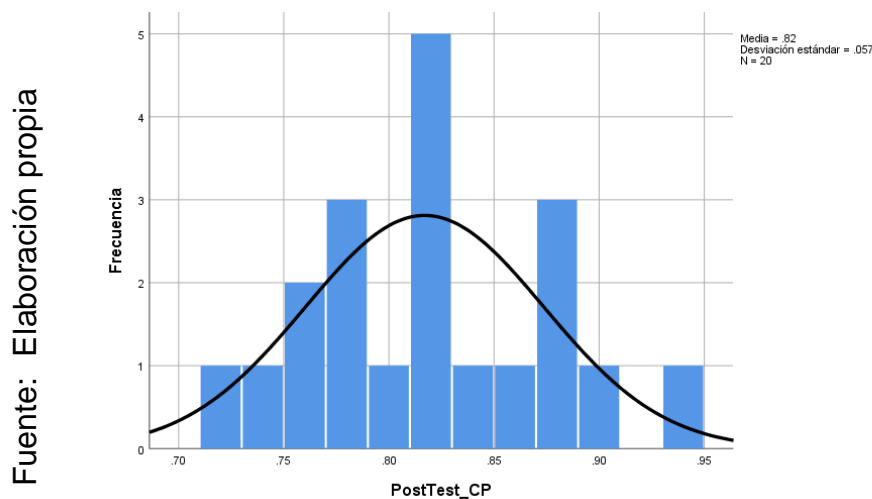


Figura 11 Prueba de Normalidad del Nivel de cobertura de las pruebas después de implementar el Sistema Web Open Source Vue

- **INDICADOR : Nivel de madurez de pruebas**

Con el propósito de elegir la prueba de hipótesis; los datos fueron sujetos a la validación de su organización, exactamente si los datos del nivel de madurez de las pruebas describían con organización habitual.

Tabla 14 Prueba de Normalidad del Nivel de madurez de las pruebas antes y después de implementar el Sistema Web Open Source Vue

Pruebas de normalidad			
	Shapiro-Wilk		
	Estadístico	gl	Sig.
PreTest_MP	.964	20	.637
PostTest_MP	.974	20	.841

Fuente: Elaboración propia

Como se puede ver en la tabla 14 los resultados de la prueba revelan que el Sig. del nivel de madurez fue de 0.637 en el Pre-test, donde valor es más grande que 0.05. Por lo cual, el nivel de cobertura se distribuye normalmente. Los resultados de las pruebas del Post-Test detallan que el Sig. del nivel de madurez de las pruebas fue 0.841, cuyo valor es más grande que 0.05, por lo cual exhibe que el nivel de

madurez de las pruebas se distribuye normalmente. Lo que, valida la organización habitual de los datos de la exhibe. Se consigue ver en las figuras 12 y 13 .

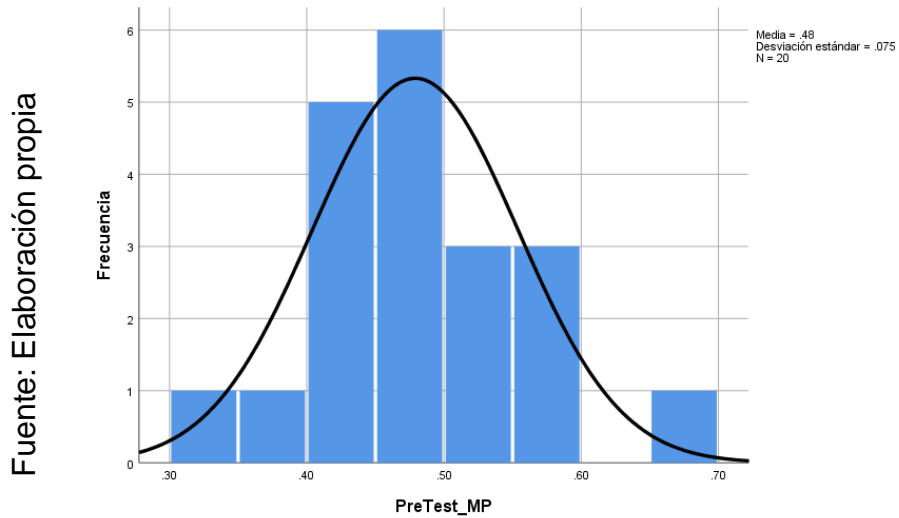


Figura 12 Prueba de Normalidad del Nivel de madurez de las pruebas antes de implementar el Sistema Web Open Source Vue

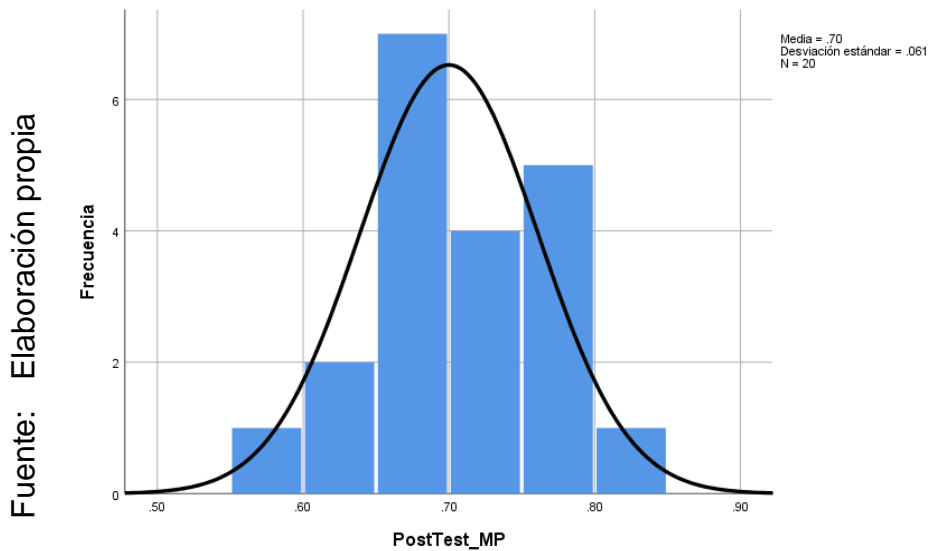


Figura 13 Prueba de Normalidad del Nivel de madurez de las pruebas después de implementar el Sistema Web Open Source Vue

1.2. Prueba de Hipótesis

Hipótesis de la investigación 1

- **H1:** El sistema web aumenta la cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience
- **Indicador:** Nivel de Cobertura de las pruebas

Hipótesis Estadísticas

INCa: Nivel de cobertura antes de utilizar el sistema web

INCd: Nivel de cobertura después de utilizar el sistema web.

- **H1₀:** El sistema web no aumenta la cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

H1₀: INCa \geq INCd

- **H1_a:** El sistema web aumenta la cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

H1_a: INCa $<$ INCd

La Figura 14, el nivel de cobertura de las pruebas del Pre-Test es de 67.50% y el Post-Test es de 81.95%.

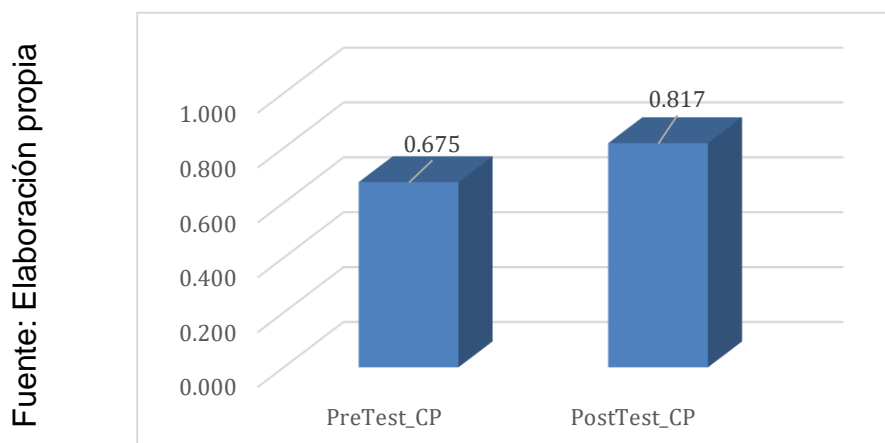


Figura 14 Nivel de cobertura de las pruebas - Comparativa General

En la figura 14 se puede concluir el aumento del nivel de la cobertura de las pruebas, en donde se puede corroborar comparando las medias, que se desprende un aumento de 0.675 al valor de 0.817.

Además, se empleó la Prueba de T-Student al resultado de la hipótesis, debido a que los resultados del Pre-Test y Post-Test se distribuyen normalmente. El valor resultante del T es igual a -9.091, dando como finiquitado que es menor que -1.7291 (Tabla 15).

Tabla 15 Prueba de T-Student para el Nivel de cobertura de las pruebas antes y después de implementar el Sistema Web Open Source Vue

		Media	t	gl	Sig. (bilateral)
Par 1	PreTest_CP - PostTest_CP	-0.1420000	-9,091	19	,000

Fuente: Elaboración propia

Posteriormente, se refuta la hipótesis nula, y dando como resultado aceptar la hipótesis alterna. Con un T logrado, en donde se puede visualizar en la figura 15, se puede localizar en la zona de rechazo. Por lo cual, el sistema web incrementa el nivel de cobertura de las pruebas en el proceso de pruebas de calidad en la empresa NextPerience.

Resolviendo la formula

$$TC = \frac{x - u}{S/\sqrt{n}}$$

$$TC = \frac{-0.1420000}{0.069986/\sqrt{20}}$$

$$TC = \frac{-0.142000}{0.01562117}$$

$$TC = -9.091$$

Fuente: Elaboración propia



Figura 15 T-Student - Nivel de cobertura de las pruebas

Hipotesis de la investigación 2

- **H2:** El sistema web aumenta la madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience
- **Indicador:** Nivel de Madurez de las pruebas

Hipotesis Estadístico

INMa: Nivel de madurez de las pruebas antes de utilizar el sistema web

INMd: Nivel de madurez de las pruebas después de utilizar el sistema web

- **H2₀:** El sistema web no aumenta la madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

$$\mathbf{H2_0: INMa \geq INMd}$$

- **H2a:** El sistema web aumenta la madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience

$$\mathbf{H2a: INMa < INMd}$$

En la Figura 16, el nivel de madurez de las pruebas del Pre-Test es de 0.479 y el Post-Test es de 0.700.

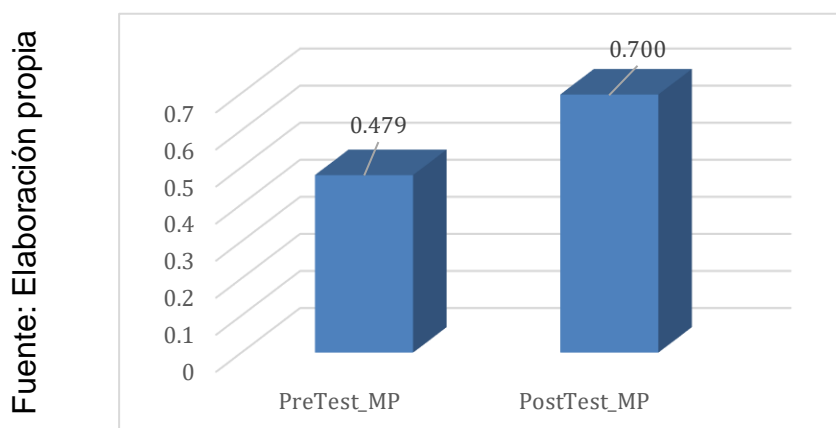


Figura 16 Nivel de madurez de las pruebas - Comparativa General

En la figura 14 se puede concluir el aumento del nivel de la madurez de las pruebas, en donde se puede corroborar comparando las medias , que se desprende un aumento de 0.479 al valor de 0.7000.

Además, se empleó la Prueba de T-Student al resultado de la hipótesis, debido a que los resultados del Pre-Test y Post-Test se distribuyen normalmente. El valor resultante del T es igual a -12.335, dando como finiquitado que es menor que -1.7291(Tabla 16).

Tabla 16 Prueba de T-Student para el Nivel de madurez de las pruebas antes y después de implementar el Sistema Web Open Source Vue

		Media	t	gl	Sig. (bilateral)
Par 1	PreTest_MP - PostTest_MP	-0.221000	-12.335	19	,000

Fuente: Elaboración propia

Posteriormente, se refuta la hipótesis nula, y dando como resultado aceptar la hipótesis alterna. Con un T logrado, en donde se puede visualizar en la figura 16, se

puede localizar en la zona de rechazo. Por lo cual, el sistema web incrementa el nivel de madurez de las pruebas en el proceso de pruebas de calidad en la empresa NextPerience.

Resolviendo la formula

$$TC = \frac{x - u}{S/\sqrt{n}}$$
$$TC = \frac{-0.221000}{0.08012/\sqrt{20}}$$
$$TC = \frac{-0.221000}{0.01791538}$$
$$TC = -12.3357719$$

Fuente: Elaboración propia



Figura 17 T-Student - Nivel de madurez de las pruebas

V. DISCUSIÓN

Basado en los resultados de esta exploración se examina una comparativa sobre el nivel de cobertura de las pruebas y el nivel de madurez de las pruebas

Hipotesis específica 1:

La utilización de un sistema web incrementa el nivel de cobertura de las pruebas en el desarrollo de pruebas de calidad en la empresa NextPerience.

El nivel de cobertura de las pruebas antes de utilizar el sistema era de 0.68, y después de su incorporación se creció un 0.82 en promedio, lo que otorga como resultado un aumento de 14.2%. De hecho, se puede deducir que el incremento del nivel de cobertura de las pruebas es de 14.2%, y según este resultado se concluye que se creció el nivel de cobertura de las pruebas de manera correcta en un 14.2%. Y esto a consecuencia de la utilización de un sistema web para el proceso de pruebas de calidad.

En el intervalo de nuestra exploración no se lograron encontrar estudios completados en el rubro del nivel de cobertura de pruebas de una compañía, no obstante, los resultados se tienen que confrontar con los resultados que se consiguieron en la misma extensión de ejecución, indicado por Martínez & Morales (2015) en la proposición titulada: Diseño e implementación de mejoras a la plataforma para la gestión de pruebas de proyectos de programa – MANTEST, quien consiguió un aumento de productividad en un 5%; y en nuestra investigación creció el nivel de cobertura de pruebas 14.2% .

Hipotesis específica 2:

La utilización de un sistema web incrementa el nivel de madurez de las pruebas en el desarrollo de pruebas de calidad en la empresa NextPerience.

El nivel de cobertura de las madureces antes de utilizar el sistema era de 0.479, y después de su incorporación se creció un 0.700, lo que proporciona como resultado

un aumento de 22.1 %. De hecho, se puede deducir que el incremento del nivel de madurez de las pruebas es de 22.1%, y según este resultado se concluye que se creció el nivel de madurez de las pruebas de manera correcta en un 22.1%. Y esto a consecuencia de la utilización de un sistema web para el proceso de pruebas de calidad.

En el intervalo de nuestra implementación se consigue adquirir los resultados que se consiguieron por Stephan (2009), en su investigación llamada “Modelos de prueba y criterios de madurez para la generación automática de pruebas fundamentadas en modelos con máquinas de estado UML”, quien logro un nivel de madurez de 0.530 en oposición al nivel de madurez de 0.700 obtenido en esta exploración, logrando un aumento de 0.700.

VI. CONCLUSIONES

- Primero: se deduce que el nivel de cobertura de las pruebas en el proceso de pruebas de calidad en la empresa NextPerience, se aumentó con la utilización de un sistema web open source Vue, dado que el nivel de cobertura antes de la utilización era de 0.675 y el nivel de cobertura de las pruebas luego de la utilización fue de 0.81, lo cual se traduce en un aumento de 0.142 (14.2%) en el nivel de cobertura de las pruebas.
- Segundo: se deduce que el nivel de madurez de las pruebas en el proceso de pruebas de calidad en la empresa NextPerience, se aumentó con la utilización de un sistema web open source Vue, dado que el nivel de madurez antes de la utilización era de 0.479 y el nivel de cobertura de las pruebas luego de la utilización fue de 0.7000, lo cual significa un aumento de 0.221 (22.1%) en el nivel de madurez de las pruebas.
- Tercera: En conclusión, luego de haber logrado resultados satisfactorios para los indicadores presentados de esta tesis, se concluye que un sistema web open source mejora el desarrollo de pruebas de calidad en la empresa NextPerience

VII. Recomendaciones

- Primera: El sistema implementado usa una tecnología nueva e innovado: Vue, que tiene como base el lenguaje de programación JavaScript, el cual tiene la ventaja de ser multiplataforma y escalable, mejoras y/o transición a otros dispositivos para poder incrementar los módulos y actividades del sistema pueda tener.
- Segunda: El Perú hay pocos o ningún sistema web orientados al desarrollo de pruebas de calidad, por lo cual se sugiere realizar y utilizar una optimización para su venta.
- Tercera: La realización de una tesis de este tipo implica una exhaustiva investigación con respecto a indicadores para el análisis, por lo que se recomienda aplicar los indicadores aplicados en esta tesis con investigación anticipada.

REFERENCIAS

- AHMED, A. 2019. SOFTWARE PROJECT MANAGEMENT. [S.I.]: CRC PRESS.

- ANWAR, A. 2020. A Review of RUP (Rational Unified Process) [en línea]. Atlanta: Department of Computer Science & MIS. [Consulta: 17 Abril 2020]. Disponible en: https://www.academia.edu/11981553/A_Review_of_RUP_Rational_Unified_Process_-_2014.

- ARORA, C., HENNESSY, K. and NORING, C. 2018. Building Large-Scale Web Applications with Angular. S.I.: Packt Publishing.

- ARVIZU IBARRA, C. 2019. TÉCNICA DE INVESTIGACIÓN DOCUMENTAL. [en línea]. [Consulta: 11 junio 2019]. Disponible en: http://www.juridicaformativa.uson.mx/materialdidactico/Taller_de_Instrumentacion_Juridica/5_Unidad_3/1_Documental/1_Investigacion_documental.pdf.

- AZAUSTRE, C. 2016. Aprendiendo JavaScript: Desde cero hasta ECMAScript 6. S.I.: carlosazaustre.es.

- BALDEÓN VILLANES, E. 2015. Método Para La Evaluación De Calidad De Software Basado En ISO/IEC 25000. Maestro. S.I.: Universidad San Martin de Porres.

- BLANQUICETT, Luis A.; BONFANTE, María C. y ACOSTA-SOLANO, Jairo. Prácticas de Pruebas desde la Industria de Software. La Plataforma ASISTO como Caso de Estudio. Inf. tecnol. [online]. 2018, vol.29, n.1, pp.11-18. ISSN 0718-0764. <http://dx.doi.org/10.4067/S0718-07642018000100011>.

- BOJINOV, V., HERRON, D. and RESENDE, D. 2018. Node.js complete reference guide. S.I.: s.n.

- BROJT, D. [sin fecha]. *Project management*. S.I.: Ediciones Granica S.A.

- BUSH., D., PERNA., M., GERCHEV., I., ATOYEBI., D., VUNDI., C., ELUWANDE., Y., MULDER., M., JACQUES., N., OMOLE., O., CAMPBELL., C., SILAS., K., BOUCHEFRA., A., WANYOIKE., M., FRANKLIN., J. and HIBBARD., J. 2019. Learn Vue.js: The Collection. S.I.: SitePoint.

- CARDADOR CABELLO, A. 2015. *Implantación de aplicaciones web en entornos Internet, intranet y extranet*. Antequera, Málaga: IC Editorial.

- CARRASCO CASTILLO, G. 2019. Muestreo de poblaciones: Tamaño de la muestra. *Recursostic.educacion.es* [en línea]. [Consulta: 11 junio 2019]. Disponible en:
http://recursostic.educacion.es/descartes/web/materiales_didacticos/muestreo_poblaciones_ccg/tamano_muestra.htm.

- CHAVEZ, L. 2017. Aplicación informática para mejorar las pruebas de software en un banco privado. Ingeniero. S.I.: Universidad Nortbert Wiener.

- COMBAUDON, S. 2017. MySQL 5.7. Cornellà de Llobregat (Espagne): Ediciones ENI.

- Dayan, S., 2020. Vue.Js. [online] Vuejs.org. Available at: <<https://vuejs.org>> [Accessed 4 January 2020].

- DUEÑAS NOGUERAS, J. 2014. *Sistemas de información y bases de datos en consumo*. Antequera, Málaga: IC Editorial.

- El Ciclo de Vida de Pruebas de TMap® Next en Microsoft Test Manager y Visual Studio. Tu centro de expertise en España sobre Digital Assurance y Testing [en línea] 2014.

- ESPINOSA GARCÍA, G. 2016. Desarrollo De Un Modelo De Pruebas Y Calidad De Software Para La Empresa Seguros Atlas S.A. Licenciado. S.I.: Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas.

- FERNANDEZ AVALOS, J. 2018. Automatización de Procesos para mejorar las Pruebas de Software en el área de calidad del Banco de Crédito. Maestría en ingeniería de sistemas con mención en tecnologías de la información. S.I.: Universidad César Vallejo.

- FINK, G. and FLATOW, I. 2014. Pro single page application development. 1. S.I.: Apress.

- GILCHRIST, A. 2015. *REST API Design Control and Management*. S.I.: s.n.

- GISBERT VERCHER, B. 2015. *UF1272 - Administración y auditoría de los servicios web*. S.I.: Editorial Elearning, S.L.

- GÓMEZ PALOMO, S. and MORALEDA GIL, E. 2014. *Aproximación a la ingeniería del software*. Madrid: Editorial Universitaria Ramón Areces.

- GRANADOS LA PAZ, R. 2015. *Desarrollo de aplicaciones web en el entorno servidor*. Málaga: IC Editorial.

- HERNÁNDEZ SAMPIERI, R., FERNÁNDEZ COLLADO, C. and PILAR BAPTISTA LUCIO, M. 2014. *Metodología de la investigación*. México: McGraw-Hill.

- ISTQB Glossary. *Glossary.istqb.org* [en línea] 2019.

- JAIME GUZMÁN, S. 2018. Automatización de Procesos para mejorar las Pruebas de Software en el área de calidad del Banco de Crédito. Bachiller. S.I.: Universidad Cesar Vallejo.

- KOOMEN, T., AALST, L. and BROEKMAN, B. 2007. TMap® Next for result-driven testing. 's-Hertogenbosch: Tutein Nolthenius.

- KOPAL, N. 2020. Rational Unified Process [en línea]. Kassel: Universitat Kassel. [Consulta: 18 abril 2020]. Disponible en: <https://arxiv.org/pdf/1609.07350.pdf>.

- MARDANOV, A. 2014. Express.js guide. [Vancouver, B.C.]: Lean Publishing.

- MARTÍNEZ GUTIÉRREZ, D. and MORALES QUINTERO, B. 2015. Diseño e implementación de mejoras a la plataforma para la gestión de pruebas de proyectos de software - MANTEST. S.I.: Universidad Tecnológica de Pereira.

- MITTAL, A. 2017. Getting Started with Sequelize for Nodejs Applications | Hacker Noon. Hackernoon.com [en línea]. [Consulta: 18 abril 2020]. Disponible en: <https://hackernoon.com/getting-started-with-sequelize-for-nodejs-applications-2854c58ffb8c>.

- MOLINA RÍOS, J., ZEA ORDÓÑEZ, M., REDROVÁN CASTILLO, F., LOJA MORA, N., VALAREZO PARDO, M. and HONORES TAPIA, J. 2018. *SNAIL, Una metodología híbrida para el desarrollo de aplicaciones web*. S.I.: 3Ciencias.

- Método hipotético-deductivo en el Diccionario soviético de filosofía. *Filosofía.org* [en línea] 2019.

- PANTALEO, G. y RINAUDO, L. 2015. *Ingeniería de Software*. S.I.: Alfaomega Grupo Editor.

- PATIÑO CAMARGO, P. and SUÁREZ VILLEGAS, R. 2014. Optimización del proceso de pruebas de software. S.I.: Universidad Peruana de Ciencias Aplicadas.

- PEPPERS, J. 2015. *Xamarin Cross-platform Application Development - Second Edition*. S.I.: Packt Publishing.

- PHAKITI, A. 2014. *Experimental Research Methods in Language Learning*. S.I.: Bloomsbury Publishing.

- Pre-Experimental Design definition | Psychology Glossary | alleydog.com. *Alleydog.com*[en línea] 2019.

- PRESSMAN, R. 2015. *Software engineering*. Boston: McGraw-Hill Education.

- PRUSTY, N. 2015. *Learning ECMAScript 6*. Birmingham, UK: Packt Publishing.

- RODRÍGUEZ, P. 2019. Niveles de Pruebas: Calidad en el ciclo de vida de una aplicación. *QA:news* [en línea]. [Consulta: 11 junio 2019]. Disponible en: <https://qanewsblog.com/2016/02/09/niveles-de-pruebas-calidad-en-el-ciclo-de-vida-de-una-aplicacion/>.

- RUIZ BOLÍVAR, C. 2015. Confiabilidad [en línea]. S.l.: s.n. [Consulta: 5 mayo 2019]. Disponible en: <http://200.11.208.195/blogRedDocente/alexisduran/wp-content/uploads/2015/11/CONFIABILIDAD.pdf>.

- SREEJESH, S., MOHAPATRA, S. and ANUSREE, M. 2013. *Business research methods*. S.l.: Springer Science & Business Media.

- The Vue Instance — Vue.js. *Vuejs.org* [en línea] 2019.

- VACA SIERRA, T. 2019. Modelo De Calidad De Software; aplicado Al Módulo De Talento Humano Del Sistema Informático Integrado Universitario. S.l.: Universidad Técnica del Norte.

- VALDIVIA ESPINOZA, D. 2017. Estándares de calidad para pruebas de software. Ingeniero de Sistemas. S.l.: Universidad Nacional Mayor de San Marcos.

- VORA, P. 2009. *Web application design patterns*. Amsterdam: Elsevier.

- ZAMBRANO LOOR, J. and ECHEVERRÍA HIDROVO, J. 2019. Aplicación web para la administración de los materiales almacenados en las bodegas de la empresa constructora Coinfra S.A. S.l.: Escuela Superior Politécnica Agropecuaria de Manabí.

- ZYKOV, S., GROMOFF, A. and KAZANTSEV, N. 2018. Software engineering for enterprise system agility. S.l.: s.n.

- WEIßLEDER, S. 2009. Modelos de prueba y criterios de madurez para la generación automática de pruebas basadas en modelos con máquinas de estado UML. Grado académico doctor. S.l.: Universidad Humboldt de Berlín.

ANEXOS

Anexo 1: Matriz de Consistencia

Problemas	Objetivos	Hipótesis	Variables	Variable Dependiente				Método
Principal	General	General	Independiente					Tipo de investigación Explicativa - Experimental - Aplicativa Diseño de Investigación Pre-experimental Método de investigación Hipotética-deductiva Población 2500 casos de prueba Muestra 20 fichas de recolección de datos
¿De qué manera influye un sistema web en el proceso de pruebas de calidad de software en la empresa NextPerience?	Determinar de qué manera influye un sistema web en el proceso de pruebas de calidad de software en la empresa NextPerience	El sistema web mejora el proceso de pruebas de calidad de software en la empresa NextPerience	Sistema web					
Secundario	Específicos	Específicos	Dependiente					
¿De qué manera influye un sistema web en el nivel de cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience?	Determinar la influencia de un sistema web en el nivel de cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience	El sistema web aumenta el nivel de cobertura de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience	Proceso de pruebas de calidad de software	Dimensión	Indicador	Técnica	Instrumento	
¿De qué manera influye un sistema web en el nivel de madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience?	Determinar a influencia de un sistema web en el nivel de madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience	El sistema web aumenta el nivel de madurez de pruebas en el proceso de pruebas de calidad de software en la empresa NextPerience		Pruebas de Sistema	Nivel de cobertura de las pruebas	Fichaje	Ficha	
				Pruebas de Sistema	Nivel de madurez de las pruebas	Fichaje	Ficha	


Anexo 2 Ficha técnica. Instrumento de recolección de datos

Autor	Flores Castillejo Joseph Olegario	
Nombre del instrumento	Ficha de registro	
Lugar	NextPerience	
Fecha de Aplicación	Pretest	Postest
	01 de abril del 2019	
Objetivo	Determinar de qué manera influye un sistema web en el proceso de pruebas de calidad de software en la empresa NextPerience.	
Tiempo de duración	20 días (lunes a viernes)	
Elección de técnica de instrumento		
Variable	Técnica	Instrumento
Variable dependiente Proceso de pruebas de calidad	Fichaje	Ficha de registro
Variable independiente Sistema Web	--	--

Anexo 3: Ficha de registro: Nivel de cobertura de las pruebas (Pre-Test)

Anexo 3: Ficha de registro: Nivel de cobertura de las pruebas (Pre-Test)

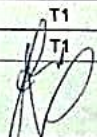
Investigador	Joseph Flores Castillejo		Tipo de Prueba	Pre test	
Empresa investigada	NextPerience				
Motivo de Investigación	Nivel de cobertura de las pruebas				
Fecha Inicio	1/04/2019		Fecha fin	31/04/2019	
Variable	Indicador		Medida	Fórmula	
	Nivel de cobertura de las pruebas		Porcentaje	CP=CPE/CPR	
Ítem	Fecha	COD. Trabajador	Número de casos de prueba que han sido ejecutados	Número de casos de prueba a ejecutar requeridos para cubrir todos los requerimientos.	Valor de la cobertura de las pruebas
1	1/04/2019	T1	14	18	0.78
2	2/04/2019	T1	10	16	0.63
3	3/04/2019	T1	10	15	0.67
4	4/04/2019	T1	12	19	0.63
5	5/04/2019	T1	9	15	0.60
6	8/04/2019	T1	10	14	0.71
7	9/04/2019	T1	10	15	0.67
8	10/04/2019	T1	8	15	0.53
9	11/04/2019	T1	13	19	0.68
10	12/04/2019	T1	13	17	0.76
11	15/04/2019	T1	12	17	0.71
12	16/04/2019	T1	11	16	0.69
13	17/04/2019	T1	12	18	0.67
14	18/04/2019	T1	10	15	0.67
15	19/04/2019	T1	11	15	0.73
16	22/04/2019	T1	9	16	0.56
17	23/04/2019	T1	13	19	0.68
18	24/04/2019	T1	11	17	0.65
19	25/04/2019	T1	15	20	0.75
20	26/04/2019	T1	10	17	0.59


Renzo Castillo Ibarra
 Jefe de proyectos
 NextPerience

Anexo 4 Ficha de registro: Nivel de madurez de las pruebas (Pre-Test)

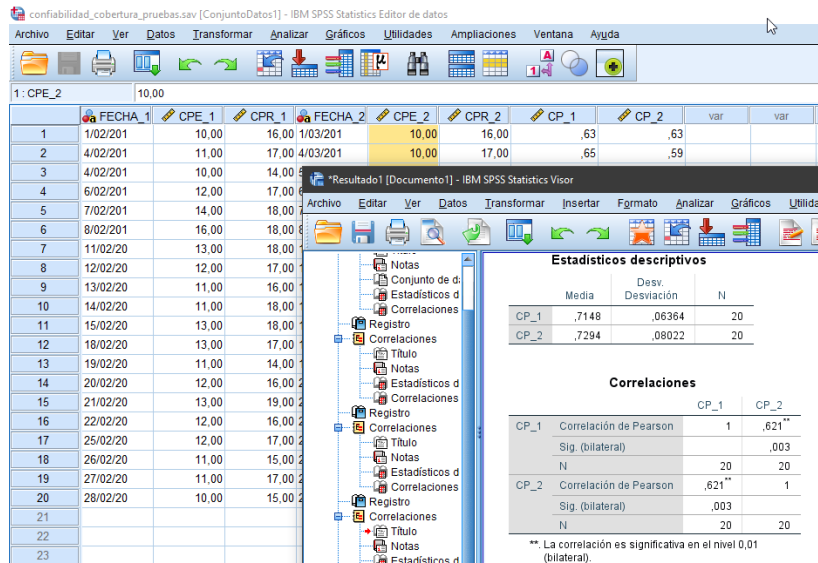
Anexo 4 Ficha de registro: Nivel de madurez de las pruebas (Pre-Test)

Investigador	Joseph Flores Castillejo		Tipo de Prueba	Pre test	
Empresa investigada	NextPerience				
Motivo de Investigación	Nivel de madurez de las pruebas				
Fecha Inicio	1/04/2019		Fecha fin	31/04/2019	
Variable	Indicador		Medida	Fórmula	
	Nivel de madurez de las pruebas		Porcentaje	MP=CPSCPR	
Ítem	Fecha	COD. Trabajador	Número de casos de prueba que han dado resultados satisfactorios	Número de casos de prueba diseñados para cubrir todos los requerimientos	Valor de la madurez de las pruebas
1	1/04/2019	T1	10	18	0.56
2	2/04/2019	T1	7	16	0.44
3	3/04/2019	T1	7	15	0.47
4	4/04/2019	T1	9	19	0.47
5	5/04/2019	T1	7	15	0.47
6	8/04/2019	T1	7	14	0.50
7	9/04/2019	T1	6	15	0.40
8	10/04/2019	T1	10	15	0.67
9	11/04/2019	T1	7	19	0.37
10	12/04/2019	T1	8	17	0.47
11	15/04/2019	T1	6	17	0.35
12	16/04/2019	T1	7	16	0.44
13	17/04/2019	T1	10	18	0.56
14	18/04/2019	T1	7	15	0.47
15	19/04/2019	T1	6	15	0.40
16	22/04/2019	T1	7	16	0.44
17	23/04/2019	T1	10	19	0.53
18	24/04/2019	T1	10	17	0.59
19	25/04/2019	T1	13	20	0.65
20	26/04/2019	T1	8	17	0.47


 Renzo Castillo Ibarra
 Jefe de proyectos
 NextPerience

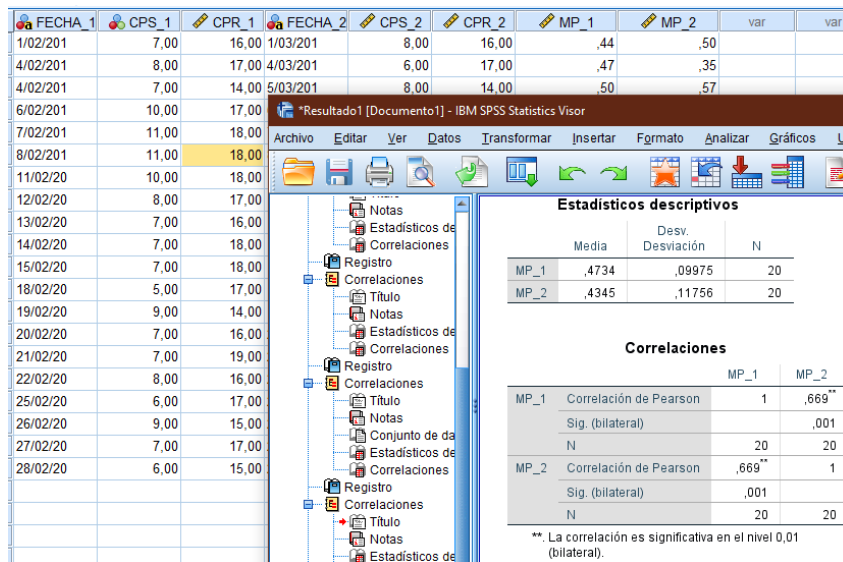
Anexo 5 Confiabilidad del instrumento de investigación

Para el indicador de nivel de cobertura de las pruebas



Como es resultado 0,621 y es mayor a 0,6. Por lo tanto, el instrumento es confiable para medir el nivel de cobertura de las pruebas en la investigación

Para el indicador de nivel de madurez de las pruebas



Como es resultado 0,614 y es mayor a 0,6. Por lo tanto, el instrumento es confiable para medir el nivel de madurez de las pruebas en la investigación

Anexo 6 Evaluación de las metodologías para el desarrollo



TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombres del Experto: Yesenia Del Rosario Vásquez Valencia

Título y/o Grado:

Ph . D ()	Doctor (X)	Magister ()	Licenciado ()	Otros () Especifique: _____
------------	--------------	--------------	----------------	------------------------------

Universidad que labora : Universidad Cesar Vallejo

Fecha : 16/06/2020

SISTEMA WEB CON OPEN SOURCE VUE JS PARA EL PROCESO DE PRUEBAS DE CALIDAD DE SOFTWARE EN LA EMPRESA NEXTPERIENCE

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar las metodologías involucradas, mediante una serie de criterios con puntuaciones especificadas al final de la tabla. Asimismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias, con la finalidad de mejorar la coherencia de los criterios.

ITEMS	CRITERIOS	METODOLOGÍA			
		OOHDM	UWE	RUP	OBSERVACIONES
1	Permite manejar la gestión de prioridades	2	2	3	
2	Permite gestionar el trabajo en grupos	2	3	3	
3	Permite manejar el enfoque a usuarios	2	3	3	
4	Permite manejar la documentación formal	2	2	3	
5	Permite manejar la orientación a la calidad	2	2	3	
6	Utiliza UML	2	2	3	
	TOTAL	12	14	18	

Evaluar con la siguiente puntuación: 1: Malo - 2: Regular - 3: Bueno

SUGERENCIAS:

Los Olivos, junio del 2020

Firma del experto



TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombres del Experto: Mg. Perez Rojas Even Deysier

Título y/o Grado: Magister en Gestión de Tecnologías de Información

Ph . D ()	Doctor ()	Magister (X)	Licenciado ()	Otros () Especifique: _____
------------	------------	--------------	----------------	------------------------------

Universidad que labora : Universidad Cesar Vallejo – “Lima Este”

Fecha : 16/06/2020

SISTEMA WEB CON OPEN SOURCE VUE JS PARA EL PROCESO DE PRUEBAS DE CALIDAD DE SOFTWARE EN LA EMPRESA NEXTPERIENCE

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar las metodologías involucradas, mediante una serie de criterios con puntuaciones especificadas al final de la tabla. Asimismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias, con la finalidad de mejorar la coherencia de los criterios.

ITEMS	CRITERIOS	METODOLOGÍA			
		OOHDM	UWE	RUP	OBSERVACIONES
1	Permite manejar la gestión de prioridades	3	3	2	
2	Permite gestionar el trabajo en grupos	2	2	3	
3	Permite manejar el enfoque a usuarios	3	3	3	
4	Permite manejar la documentación formal	2	2	3	
5	Permite manejar la orientación a la calidad	3	3	3	
6	Utiliza UML	3	3	3	
	TOTAL	16	16	17	

Evaluar con la siguiente puntuación: 1: Malo - 2: Regular - 3: Bueno

SUGERENCIAS:

Los Olivos, junio del 2020


Firma del experto



TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombres del Experto: Ruben Alexander More Valencia

Título y/o Grado: Ingeniero Informático Magíster Administración Educación

Ph . D ()	Doctor ()	Magister (X)	Licenciado ()	Otros () Especifique: _____
------------	------------	----------------	----------------	------------------------------

Universidad que labora : Universidad Cesar Vallejo

Fecha : 06/07/2020

SISTEMA WEB CON OPEN SOURCE VUE JS PARA EL PROCESO DE PRUEBAS DE CALIDAD DE SOFTWARE EN LA EMPRESA NEXTPERIENCE

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar las metodologías involucradas, mediante una serie de criterios con puntuaciones especificadas al final de la tabla. Asimismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias, con la finalidad de mejorar la coherencia de los criterios.

ITEMS	CRITERIOS	METODOLOGÍA			
		OOHDM	UWE	RUP	OBSERVACIONES
1	Permite manejar la gestión de prioridades	3	3	3	
2	Permite gestionar el trabajo en grupos	3	3	3	
3	Permite manejar el enfoque a usuarios	3	3	3	
4	Permite manejar la documentación formal	3	3	3	
5	Permite manejar la orientación a la calidad	3	3	3	
6	Utiliza UML	2	2	3	
	TOTAL	17	17	18	RUP

Evaluar con la siguiente puntuación: 1: Malo - 2: Regular - 3: Bueno

SUGERENCIAS:

_____ MD RUP _____

Los Olivos, 06-07 - 2020

Firma del experto



Validación del Instrumento

Título de la tesis: Sistema web con open source VUE js para el proceso de pruebas de calidad de software en la empresa NextPerience.

Autor: Joseph Olegario Flores Castillejo

Nombre del instrumento de Evaluación: Ficha de registro

Indicador: Nivel de cobertura de las pruebas

Datos del Experto:

1. Apellido y nombre: Ormeño Rojas, Robert
2. Cargo: DTA - UCV
3. Título y/o grado: Magister
4. Fecha: 11/6/2019

Indicador	Criterios	Deficiente 0%-19%	Regular 20%-39%	Bueno 40%-60%	Muy Bueno 61%-80%	Excelente 81%-100%
Claridad	Esta formulad con lenguaje apropiado					85%
Objetividad	Esta expresado en conducta observable					85%
Organización	Existe una organización lógica					85%
Suficiencia	Comprende los aspectos de calidad					85%
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico.					85%
Consistencia	Está basado en aspectos teóricos y científicos					85%
Coherencia	Existe relación de los contenidos con los indicadores de la variable					85%
Metodología	Responde al propósito del trabajo con los objetivos a lograr					85%
Pertinencia	El instrumento es adecuado al tipo de investigación					85%
TOTAL						85%

Aplicabilidad: El instrumento puede ser aplicado
El instrumento debe ser mejorado

Observaciones:


Firma del experto



Validación del Instrumento

Título de la tesis: Sistema web con open source VUE js para el proceso de pruebas de calidad de software en la empresa NextPerience.

Autor: Joseph Olegario Flores Castillejo

Nombre del Instrumento de Evaluación: Ficha de registro

Indicador: Nivel de madurez de las pruebas

Datos del Experto:

1. Apellido y nombre: Ormeño Rojas, Robert
2. Cargo: DTP-UCV
3. Título y/o grado: Magister
4. Fecha: 11/04/2019

Indicador	Criterios	Deficiente 0%-19%	Regular 20%-39%	Bueno 40%-60%	Muy Bueno 61%-80%	Excelente 81%-100%
Claridad	Esta formulado con lenguaje apropiado					85%
Objetividad	Esta expresado en conducta observable					85%
Organización	Existe una organización lógica					85%
Suficiencia	Comprende los aspectos de calidad					85%
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico.					85%
Consistencia	Está basado en aspectos teóricos y científicos					85%
Coherencia	Existe relación de los contenidos con los indicadores de la variable					85%
Metodología	Responde al propósito del trabajo con los objetivos a lograr					85%
Pertinencia	El instrumento es adecuado al tipo de investigación					85%
TOTAL						85%

Aplicabilidad: El instrumento puede ser aplicado []
El instrumento debe ser mejorado []

Observaciones:


Firma del experto



Validación del Instrumento

Título de la tesis: Sistema web para el proceso de pruebas de calidad de software en la empresa NextPerience.

Autor: Joseph Olegario Flores Castillejo

Nombre del instrumento de Evaluación: Ficha de registro

Indicador: Nivel de cobertura de las pruebas

Formula:

$$CP = CPE / CPR$$


Datos del Experto:

1. Apellido y nombre: Galvez Tapia Orleans Moisés
2. Cargo: Docente
3. Título y/o grado: Magister en Ingeniería de Sistemas
4. Fecha: 11/09 2019

Indicador	Criterios	Deficiente 0%-19%	Regular 20%-39%	Bueno 40%-60%	Muy Bueno 61%-80%	Excelente 81%-100%
Claridad	Esta formulad con lenguaje apropiado				80%	
Objetividad	Esta expresado en conducta observable				80%	
Organización	Existe una organización lógica				80%	
Suficiencia	Comprende los aspectos de calidad				80%	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico.				80%	
Consistencia	Está basado en aspectos teóricos y científicos				80%	
Coherencia	Existe relación de los contenidos con los indicadores de la variable				80%	
Metodología	Responde al propósito del trabajo con los objetivos a lograr				80%	
Pertinencia	El instrumento es adecuado al tipo de investigación				80%	
TOTAL						

Aplicabilidad: El instrumento puede ser aplicado
 El instrumento debe ser mejorado

Observaciones:


 Firma del experto



Validación del Instrumento

Título de la tesis: Sistema web para el proceso de pruebas de calidad de software en la empresa NextPerience.

Autor: Joseph Olegario Flores Castillejo

Nombre del Instrumento de Evaluación: Ficha de registro

Indicador: Nivel de cobertura de las pruebas

Formula:

$$CP = \frac{CPE}{CPR}$$

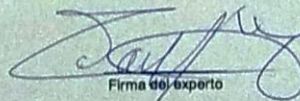
Datos del Experto:

1. Apellido y nombre: Huante Fegarra Raul
2. Cargo: Docente
3. Título y/o grado: Magister
4. Fecha: 4/24/19

Indicador	Criterios	Deficiente 0%-19%	Regular 20%-39%	Bueno 40%-60%	Muy Bueno 61%-80%	Excelente 81%-100%
Claridad	Esta formulad con lenguaje apropiado					85
Objetividad	Esta expresado en conducta observable					90
Organización	Existe una organización lógica					95
Suficiencia	Comprende los aspectos de calidad					90
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico.					95
Consistencia	Está basado en aspectos teóricos y científicos					95
Coherencia	Existe relación de los contenidos con los indicadores de la variable					90
Metodología	Responde al propósito del trabajo con los objetivos a lograr					85
Pertinencia	El instrumento es adecuado al tipo de investigación					90.
TOTAL						

Aplicabilidad: El instrumento puede ser aplicado
 El instrumento debe ser mejorado

Observaciones:


Firma del experto

Validación del Instrumento

Título de la tesis: Sistema web para el proceso de pruebas de calidad de software en la empresa NextPerience.

Autor: Joseph Olegario Flores Castillejo

Nombre del Instrumento de Evaluación: Ficha de registro

Indicador: Nivel de madurez de las pruebas

Formula:

$$MP = CPS / CPR$$

Datos del Experto:

1. Apellido y nombre: Huarote Zegava Raúl
2. Cargo: Docente
3. Título y/o grado: Magister
4. Fecha: 14/06/19

Indicador	Criterios	Deficiente 0%-19%	Regular 20%-39%	Bueno 40%-60%	Muy Bueno 61%-80%	Excelente 81%-100%
Claridad	Esta formulado con lenguaje apropiado					90
Objetividad	Esta expresado en conducta observable					95
Organización	Existe una organización lógica					95
Suficiencia	Comprende los aspectos de calidad					90
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico.					85
Consistencia	Está basado en aspectos teóricos y científicos					90
Coherencia	Existe relación de los contenidos con los indicadores de la variable					90
Metodología	Responde al propósito del trabajo con los objetivos a lograr					85
Pertinencia	El instrumento es adecuado al tipo de investigación					85
TOTAL						

 Aplicabilidad: El instrumento puede ser aplicado
 El instrumento debe ser mejorado

Observaciones:



Firma del experto

Anexo 7 Entrevista

Anexo 7: Entrevista al cliente

Investigador : Joseph Flores Castillejo
Entrevistado : Ing. Renzo Castillo Ibarra
Cargo del Entrevistado : Jefe de proyectos
Fecha de Entrevista : 08 de abril de 2019

Preguntas:

1. ¿Cuál es la razón social y ubicación actual de la empresa?

NextPerience es una empresa del rubro de tecnologías de la información, ubicada en Magdalena

2. ¿Cuál es el sector empresarial a la que pertenece?

El sector es informático y tecnologías de la información

3. ¿A qué se dedica la empresa y cuáles son las funciones y actividades de la empresa?

La empresa se dedica principalmente al desarrollo de software a medida tales como: páginas web, aplicaciones web, aplicaciones de escritorio, aplicaciones móviles, etc. E implementación de tecnologías de la información.

4. Podría explicar brevemente como realiza su proceso principal

El proceso principal que es de desarrollo de software a medida, inicia con el análisis de requerimientos, seguida por la creación de los mockups, base de datos y la codificación del software, luego o al mismo tiempo realizar las pruebas y finalmente la implementación y puesta en producción.

5. En base a lo mencionado anteriormente, ¿Cuáles son los problemas más resaltantes que ha identificado?

proceso principal el mayor problema recae en que siempre se encuentran, errores, validaciones o no se llega (a veces) a satisfacer las necesidades del cliente, o incluso el cliente realiza muchos cambios.

Al día se registran las pruebas por realizar, pero no se logra ejecutar todas las pruebas, lo cual causa que las pruebas restantes, sean trasladadas al siguiente día, prolongándose el tiempo de entrega.

Actualmente las observaciones se describen en un documento Word o Excel y luego se resuelven, el problema es que por no existir un correcto seguimiento de las pruebas, genera pérdida de tiempo al momento de resolver nuevos errores o validaciones que no se contemplaron en un inicio, muchas veces hasta este tiempo puede durar semanas, ya que el trabajo se acumula, y no se logra terminar por completo el proyecto, y esto genera incomodidad en el cliente, demora en el pago del entregable, pérdida de dinero por la pérdida de tiempo, por las horas adicionales que se emplean.

También no se pueden validar correctamente las pruebas realizadas satisfactorias, ya que al no ejecutar las pruebas correspondientes al día, no tenemos un número exacto de pruebas satisfactorias.

6. ¿Desde hace cuánto tiempo se presentan estos problemas?

Hace aproximadamente 2 años

7. ¿Tiene un estimado en dinero, de cuanto se pierde a diario por estos problemas?

En general un promedio de 250 soles diarios, que son un estimado de 5000 soles mensuales

8. ¿Qué medidas utiliza para evitar o solucionar estos problemas?

Este problema se ha reducido, pero aún se mantiene en el tiempo, tratamos de organizar mejor el proyecto y hacer pruebas más seguidas, pero son pruebas generales.

9. ¿Autoriza las visitas o entrevistas con las demás personas que intervienen en el proceso de producción en la empresa?

Si


Renzo Castillo Ibarra
Jefe de proyectos
Hex Perlen

Anexo 8. Análisis, diseño y desarrollo del sistema

ÍNDICE DE TABLAS

Tabla 1 Requerimientos Funcionales del Sistema	99
Tabla 2 Requerimientos no Funcionales del Sistema	100
Tabla 3 Relación entre los requerimientos funcionales y los casos de uso	102
Tabla 4 Especificaciones del caso de uso "Autenticar Usuario"	107
Tabla 5 Especificaciones del caso de uso "Registrar Casos de Pruebas"	110
Tabla 6 Especificaciones del caso de uso "Registrar Resultados"	112
Tabla 7 Especificaciones del caso de uso " Cerrar Proyecto"	114
Tabla 8 Especificaciones del caso de uso "Registrar Errores"	115
Tabla 9 Especificaciones del caso de uso "Registrar Versiones"	117
Tabla 10 Detalle tabla Case	172
Tabla 11 Detalle tabla CaseHistory	173
Tabla 12 Detalle tabla CaseHistoryMessage	173
Tabla 13 Detalle tabla CaseHistoryStatus	174
Tabla 14 Detalle tabla CaseHistoryStep	174
Tabla 15 Detalle tabla CaseHistoryStepStatus	175
Tabla 16 Detalle tabla CasePostCondition	175
Tabla 17 Detalle tabla CasePreCondition	176
Tabla 18 Detalle tabla CaseSpecialCondition	176
Tabla 19 Detalle tabla CaseStatus	177
Tabla 20 Detalle tabla CaseStep	177
Tabla 21 Detalle tabla Complexity	178
Tabla 22 Detalle tabla Error	178
Tabla 23 Detalle tabla ErrorMessage	179
Tabla 24 Detalle tabla ErrorStatus	179
Tabla 25 Detalle tabla File	180
Tabla 26 Detalle tabla Project	180
Tabla 27 Detalle tabla ProjectFile	181
Tabla 28 Detalle tabla ProjectStatus	181
Tabla 29 Detalle tabla Report	182

Tabla 30 Detalle tabla ReportFiles	182
Tabla 31 Detalle tabla User.....	183
Tabla 32 Detalle tabla UserRole	183
Tabla 33 Detalle tabla UserStatus	184
Tabla 34 Detalle tabla Version	184

ÍNDICE DE FIGURAS

Figura 1 Actores del sistema.....	101
Figura 2 Vista de Casos de Uso del Sistema.....	104
Figura 3 Diagrama de Casos de Uso de Sistema del Rol Administrador	105
Figura 4 Diagrama de Casos de Uso de Sistema del Jefe de Proyecto	106
Figura 5 Diagrama de Casos de Uso de Sistema del Tester	106
Figura 6 Diagrama de Casos de Uso de Sistema del Desarrollador	107
Figura 7 Realización de Caso de Uso del sistema: Autenticar Usuario.....	118
Figura 8 Realización de Caso de Uso del sistema: Recuperar Contraseña	119
Figura 9 Realización de Caso de Uso del sistema: Consultar Usuarios.....	119
Figura 10 Realización de Caso de Uso del sistema: Registrar Usuarios	119
Figura 11 Realización de Caso de Uso del sistema: Consultar Roles	120
Figura 12 Realización de Caso de Uso del sistema: Consultar Proyectos	120
Figura 13 Realización de Caso de Uso del sistema: Registrar Proyectos.....	120
Figura 14 Realización de Caso de Uso del sistema: Consultar Casos de Pruebas	121
Figura 15 Realización de Caso de Uso del sistema: Registrar Casos de Pruebas	121
Figura 16 Realización de Caso de Uso del sistema: Consultar Reportes	121
Figura 17 Realización de Caso de Uso del sistema: Generar Reportes	122
Figura 18 Realización de Caso de Uso del sistema: Resolver Casos de Pruebas	122
Figura 19 Realización de Caso de Uso del sistema: Cerrar Proyecto.....	122
Figura 20 Realización de Caso de Uso del sistema: Consultar Errores	123
Figura 21 Realización de Caso de Uso del sistema: Registrar Errores.....	123
Figura 22 Realización de Caso de Uso del sistema: Consultar Versiones	123
Figura 23 Realización de Caso de Uso del sistema: Registrar Versiones.....	124
Figura 24 Realización de Caso de Uso del sistema: Asignar Desarrollador a proyecto.....	124
Figura 25 Realización de Caso de Uso del sistema: Asignar Tester a Casos de Pruebas .	124
Figura 26 Realización de Caso de Uso del sistema: Notificar Correo electrónico	125
Figura 27 Diagrama de Clases de análisis "Autenticar Usuario".....	126
Figura 28 Diagrama de Clases de análisis "Registro Proyecto".....	127
Figura 29 Diagrama de Clases de análisis "Registrar Casos de pruebas"	128
Figura 30 Diagrama de Clases de análisis "Resolver Casos de prueba"	129
Figura 31 Diagrama de Clases de análisis "Cerrar Proyecto".....	130
Figura 32 Diagrama de Clases de análisis "Registrar Errores".....	131
Figura 33 Diagrama de Clases de análisis "Registrar Versiones".....	132
Figura 34 Lista de interfaces.....	133
Figura 35 Interfaz: Ingreso.....	134

Figura 36 Interfaz Dashboard	134
Figura 37 Interfaz Caso de prueba	135
Figura 38 Interfaz Registrar Caso de prueba	135
Figura 39 Interfaz Proyectos	136
Figura 40 Interfaz Registrar Proyecto	136
Figura 41 Interfaz Reportes	137
Figura 42 Interfaz Generar reportes	137
Figura 43 Interfaz Versiones	138
Figura 44 Interfaz Registrar Versiones	138
Figura 45 Interfaz Errores	139
Figura 46 Registrar Errores	139
Figura 47 Interfaz Casos de prueba asignadas	140
Figura 48 Interfaz Resolver Casos de pruebas	140
Figura 49 Interfaz Indicadores	141
Figura 50 Interfaz Usuarios	141
Figura 51 Interfaz Agregar Usuario	142
Figura 52 Interfaz Perfil	142
Figura 53 Lista de Modelos	143
Figura 54 Lista de Vista-Modelo	144
Figura 55 Lista de Routes	145
Figura 56 Lista de Controladores	146
Figura 57 Lista de Entidades	147
Figura 58 Diagrama de Secuencia del caso de uso del Sistema "Autenticar Usuario"	148
Figura 59 Diagrama de Secuencia del caso de uso del Sistema "Registrar proyecto"	149
Figura 60 Diagrama de Secuencia del caso de uso del Sistema "Registrar Casos de pruebas"	150
Figura 61 Diagrama de Secuencia del caso de uso del Sistema "Resolver casos de prueba"	151
Figura 62 Diagrama de Secuencia del caso de uso del Sistema "Cerrar proyecto"	152
Figura 63 Diagrama de Secuencia del caso de uso del Sistema "Registrar errores"	153
Figura 64 Diagrama de Secuencia del caso de uso del Sistema "Registrar versión"	154
Figura 65 Diagrama de Colaboración del caso de uso de sistema "Autenticar usuario"	155
Figura 66 Diagrama de Colaboración del caso de uso de sistema "Registrar proyecto"	156
Figura 67 Diagrama de Colaboración del caso de uso de sistema "Registrar casos de prueba"	157

Figura 68 Diagrama de Colaboración del caso de uso de sistema "Resolver caso de prueba"	158
Figura 69 Diagrama de Colaboración del caso de uso de sistema "Cerrar proyecto"	159
Figura 70 Diagrama de Colaboración del caso de uso de sistema "Registrar error"	160
Figura 71 Diagrama de Colaboración del caso de uso de sistema "Registrar versiones"	161
Figura 72 Diagrama de Actividades del caso de uso de sistema "Autenticar Usuario"	162
Figura 73 Diagrama de Actividades del caso de uso de sistema "Registrar proyecto"	163
Figura 74 Diagrama de Actividades del caso de uso de sistema "Registrar caso de prueba"	164
Figura 75 Diagrama de Actividades del caso de uso de sistema "Resolver caso de prueba"	165
Figura 76 Diagrama de Actividades del caso de uso de sistema "Cerrar proyecto"	166
Figura 77 Diagrama de Actividades del caso de uso de sistema "Registrar error"	167
Figura 78 Diagrama de Actividades del caso de uso de sistema "Registrar Versión"	168
Figura 79 Modelo Conceptual	169
Figura 80 Modelo lógico de base de datos	170
Figura 81 Modelo físico de base de datos	171
Figura 82 Diagrama de Componentes	185
Figura 83 Diagrama de despliegue	185

Modelado del Sistema

A continuación, se describió, el desarrollo de la metodología RUP empleada para la construcción del sistema web open source Vue js para el proceso de pruebas de calidad de software en la empresa NextPerience.

A. Cuadro de requerimientos funcionales

Tabla 17 Requerimientos Funcionales del Sistema

Código	Requerimiento funcional	Prioridad
RF01	El sistema debe mostrar una pantalla de inicio de sesión para que valide a los usuarios mediante una contraseña	ALTA
RF02	El sistema debe permitir la recuperación de contraseña a los usuarios	MEDIO
RF03	El sistema debe permitir la consulta y el registro de datos de los usuarios	ALTA
RF04	El sistema debe permitir la consulta y el registro de datos de los roles de los usuarios	ALTA
RF05	El sistema debe permitir la consulta, el registro de proyectos	ALTA
RF06	El sistema debe permitir la consulta, el registro de casos de prueba	ALTA
RF07	El sistema debe permitir generar y consultar reportes	MEDIO
RF08	El sistema debe permitir resolver los casos de prueba	ALTA
RF09	El sistema debe permitir el cierre de un proyecto	ALTA
RF10	El sistema debe permitir consultar y registrar errores.	ALTA
RF11	El sistema debe permitir la consulta y registro de versiones por proyecto	ALTO
RF12	El sistema debe permitir asignar un desarrollador al proyecto	MEDIO
RF13	El sistema debe permitir asignar un tester al caso de prueba	MEDIO

RF17	El sistema debe permitir enviar correos automáticamente de acuerdo al estado de los casos de prueba	MEDIO
RF18	El sistema debe permitir enviar correos después de realizar una auditoría por proyecto	MEDIO

Fuente: Elaboración propia

Explicación: Se muestra el listado de requerimientos funcionales con los que debe contar el sistema web.

B. Cuadro de requerimientos no funcionales

Tabla 18 Requerimientos no Funcionales del Sistema

Código	Requerimiento	Prioridad
RNF01	El sistema debe ser confiable, restringiendo el acceso a usuarios no autorizados.	ALTA
RNF02	El sistema debe estar disponible las 24 horas del día y los 365 días del año.	ALTA
RNF03	El sistema debe contar con interfaces fáciles de comprender	MEDIA
RNF04	El sistema debe funcionar en un entorno web.	ALTA
RNF05	Se debe capacitar a los usuarios durante la primera semana después de su implementación en el proceso.	MEDIA

Fuente: Elaboración propia

Explicación: Se muestra el listado de requerimientos no funcionales con los que debe contar el sistema web.

C. Actores del negocio

Los actores del sistema se identifican a partir de los actores y trabajadores del negocio. Por ello, los actores del sistema, son el administrador del sistema, el director, el solicitante, y mesa de partes.

Fuente: Elaboración propia

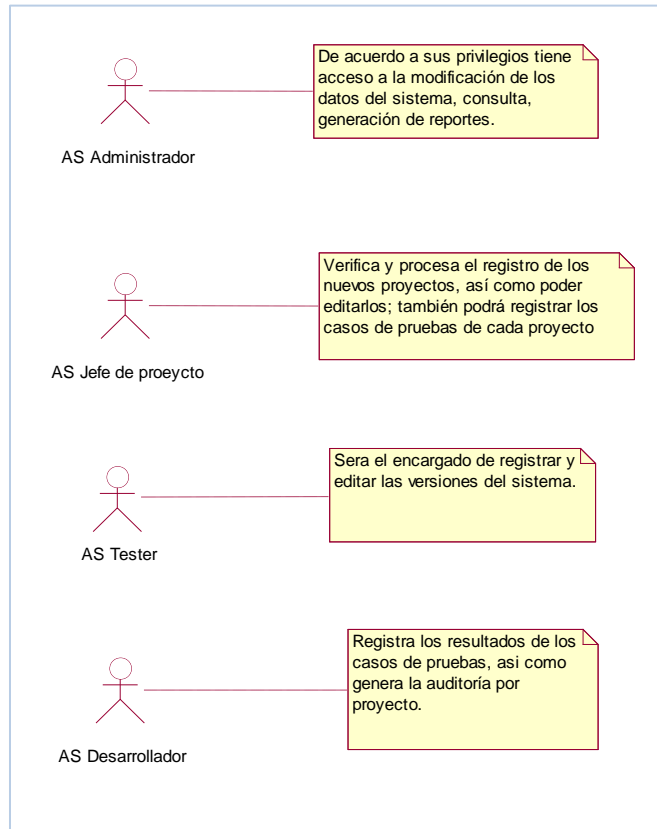





















Figura 18 Actores del sistema


Explicación: Se muestran los actores del sistema web con su respectiva descripción y funciones a realizar.

D. Casos de Uso del Sistema

Tabla 19 Relación entre los requerimientos funcionales y los casos de uso

Código	Caso de uso	Requerimiento Funcional	Representación
CUS01	Autenticar Usuario	RF01	 Autenticar Usuario
CUS02	Recuperar contraseña	RF02	 Recuperar contraseña
CUS03	Consultar Usuarios	RF03	 Consultar Usuarios
CUS04	Registrar Usuarios		 Registrar Usuarios
CUS05	Consultar Roles	RF04	 Consultar Roles
CUS06	Consultar Proyectos	RF05	 Consultar Proyectos
CUS07	Registro Proyecto		 Registrar Proyectos
CUS08	Consultar Casos de Pruebas	RF06	 Consultar Casos de pruebas
CUS09	Registrar Casos de Pruebas		 Registrar Casos de pruebas

CUS10	Consultar Reportes	RF07	 Consultar Reportes
CUS11	Generar Reportes		 Generar Reportes
CUS12	Resolver Casos de prueba	RF08	 Resolver Casos de Pruebas
CUS13	Cerrar Proyecto	RF09	 Cerrar Proyecto
CUS14	Consultar Errores	RF10	 Consultar Errores
CUS15	Registrar Errores		 Registrar Errores
CUS16	Consultar Versiones	RF11	 Consulta Versiones
CUS17	Registrar Versiones		 Registrar Versiones
CUS18	Asignar Desarrollador a Proyecto	RF12	 Asignar Desarrollador a proyecto
CUS19	Asignar Tester a Casos de Pruebas	RF13	 Asignar Tester a Casos de Pruebas

CUS20	Notificar Correo Electrónico	RF14	 Notificar Correo electrónico
		RF15	

Fuente: Elaboración propia

Explicación: Se muestra el listado casos de uso del sistema y los requerimientos funcionales que cumplen.

E. Casos de Uso del Sistema

El diagrama de casos de uso del sistema muestra las relaciones y los casos de uso de los requerimientos identificados.

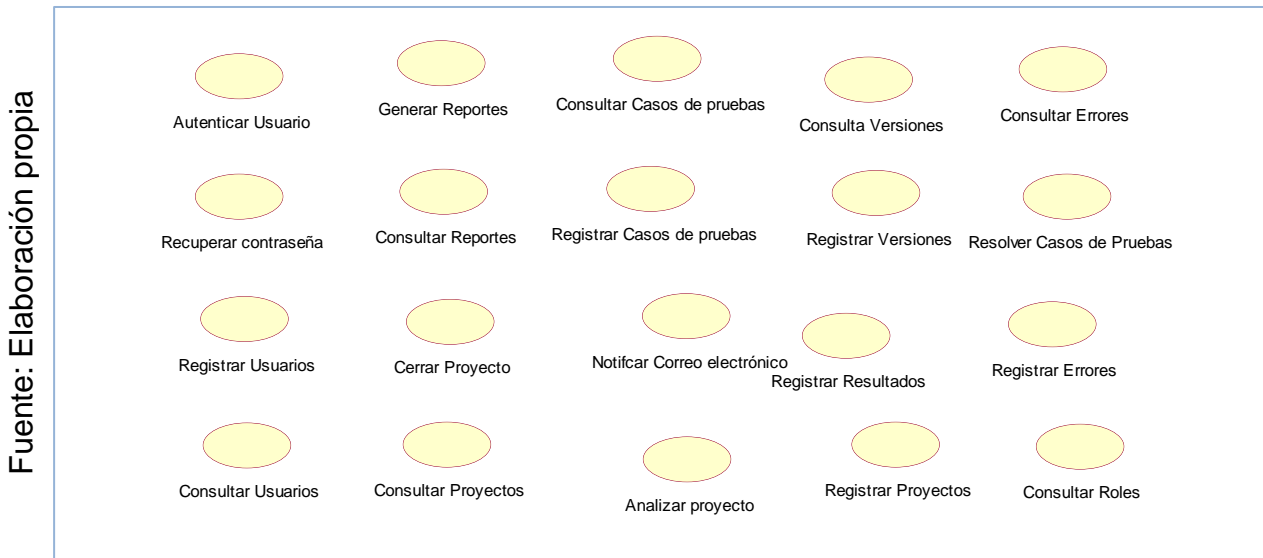


Figura 19 Vista de Casos de Uso del Sistema

Explicación: Se muestra el listado de casos de uso del sistema que se analizaron para el desarrollo del sistema web.

F. Diagrama de Casos de Uso del Sistema

Los diagramas de caso de uso de sistema pueden ser usados para describir la funcionalidad de un sistema en base a funciones que realiza cada usuario del sistema.

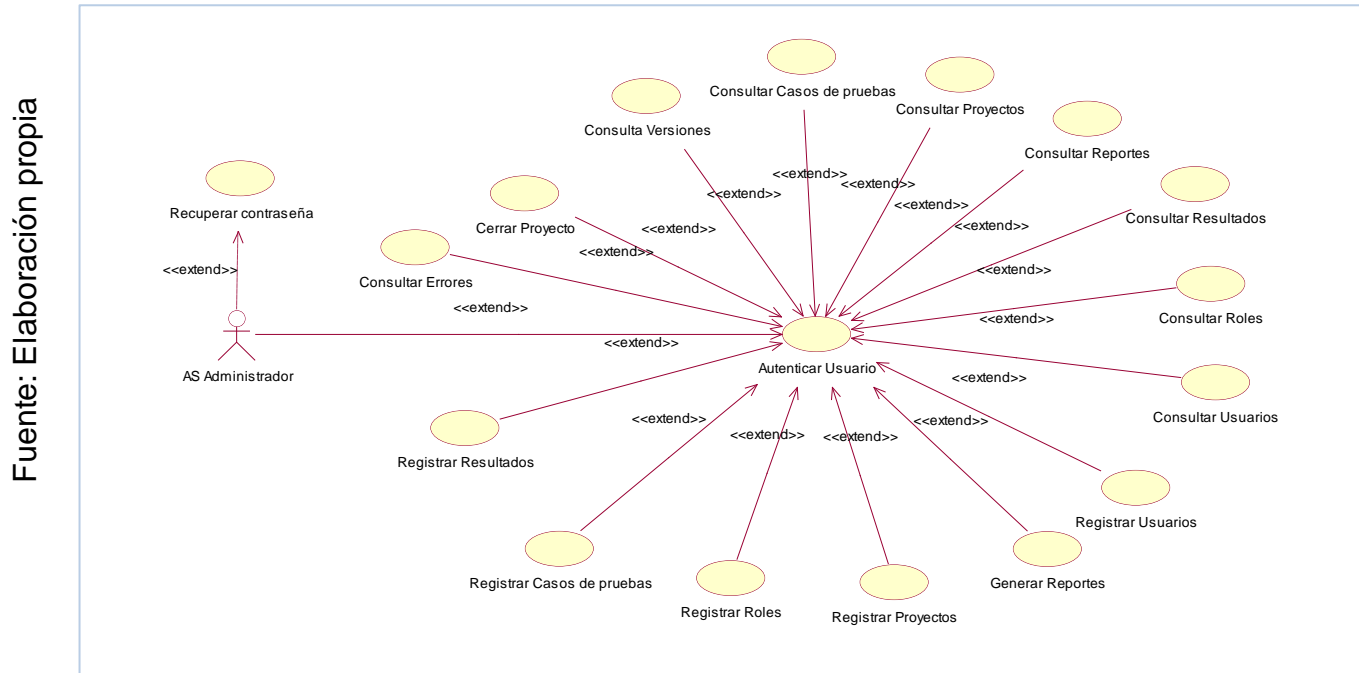


Figura 20 Diagrama de Casos de Uso de Sistema del Rol Administrador

Explicación: En la Figura N° 3, se puede observar la relación entre el actor Administrador y los casos de uso del sistema.

Fuente: Elaboración propia

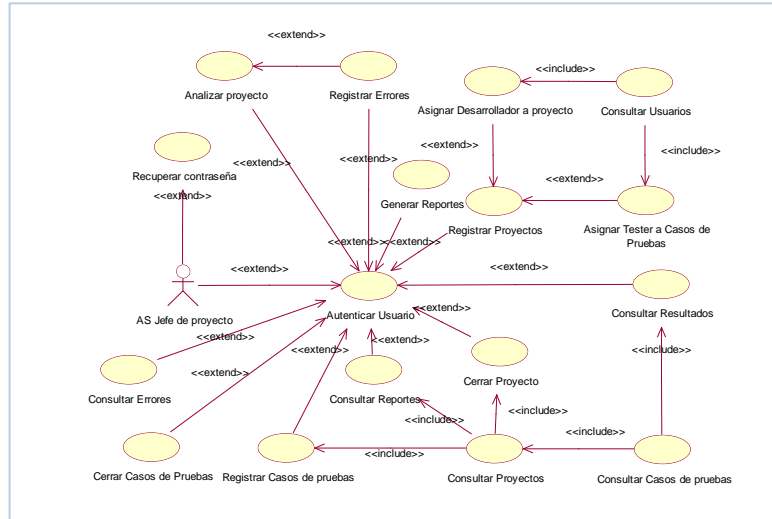


Figura 21 Diagrama de Casos de Uso de Sistema del Jefe de Proyecto

Explicación: En la Figura N° 4, se puede observar la relación entre el actor Jefe de Proyecto y los casos de uso del sistema.

Fuente: Elaboración propia

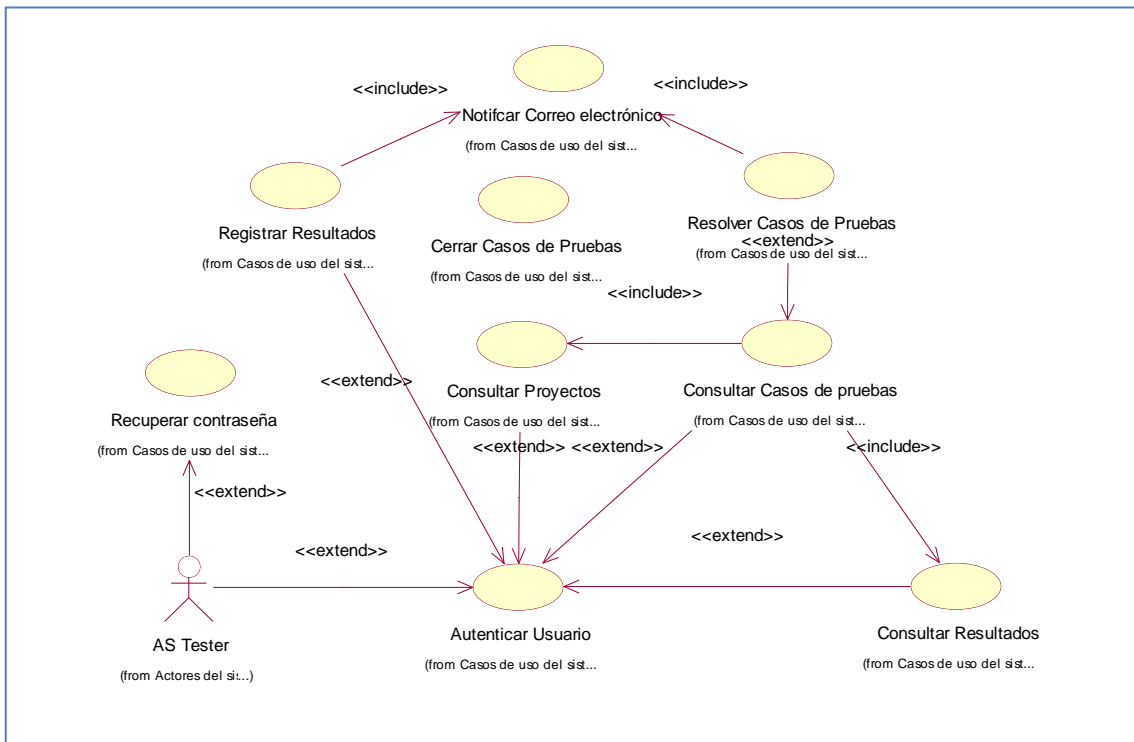


Figura 22 Diagrama de Casos de Uso de Sistema del Tester

Explicación: En la Figura N° 5, se puede observar la relación entre el actor Tester y los casos de uso del sistema.

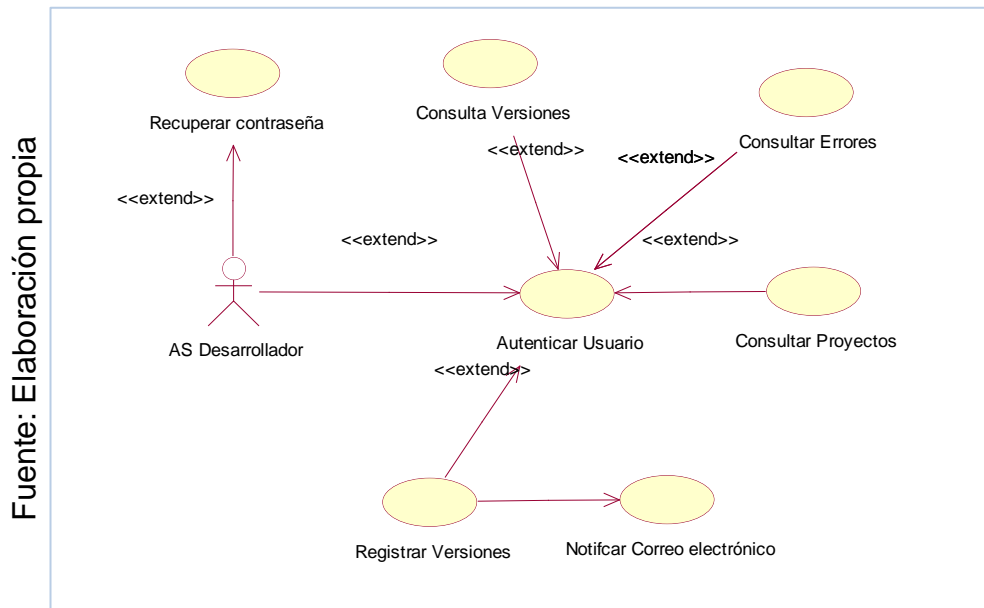


Figura 23 Diagrama de Casos de Uso de Sistema del Desarrollador

Explicación: En la Figura N° 6, se puede observar la relación entre el actor Desarrollador y los casos de uso del sistema.

G. Especificaciones de caso de uso

En la siguiente tabla, se muestra las especificaciones del caso de uso “Autenticar Usuario”.

Tabla 20 Especificaciones del caso de uso "Autenticar Usuario"

Caso de uso:	CUS01 - Autenticar Usuario
Actor(es):	Administrador Jefe de Proyecto Tester Desarrollador
Descripción:	El sistema permite al usuario acceder a su contenido mediante una cuenta de usuario y su respectiva contraseña
Precondiciones:	Los usuarios deben estar registrados en el sistema

Postcondiciones:	El usuario ha iniciado sesión y el sistema dirige a la pantalla de dashboard y el menú principal.
-------------------------	---

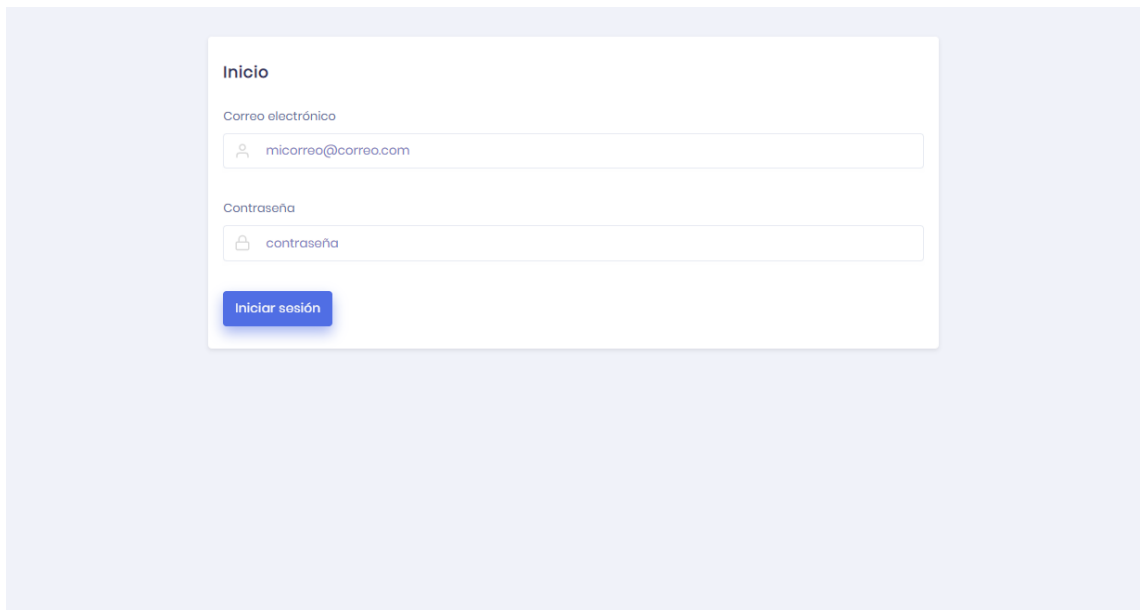
FLUJO DE EVENTOS

Acción del Actor	Acción del Sistema
2. El usuario ingresa los datos solicitados: usuario y contraseña. Posteriormente pulsa el botón "Ingresar".	1. El sistema muestra la interfaz principal "Login". 3. El sistema valida y verifica los datos ingresar. 4. El sistema muestra la interfaz de "Dashboard" y un mensaje de bienvenida al usuario. 5. El caso de uso finalizar.

FLUJOS ALTERNATIVOS

Si el usuario ingresa un nombre de usuario y contraseña incorrecta, el sistema mostrará el mensaje "El usuario o contraseña ingresado es incorrecto."

PROTOTIPO



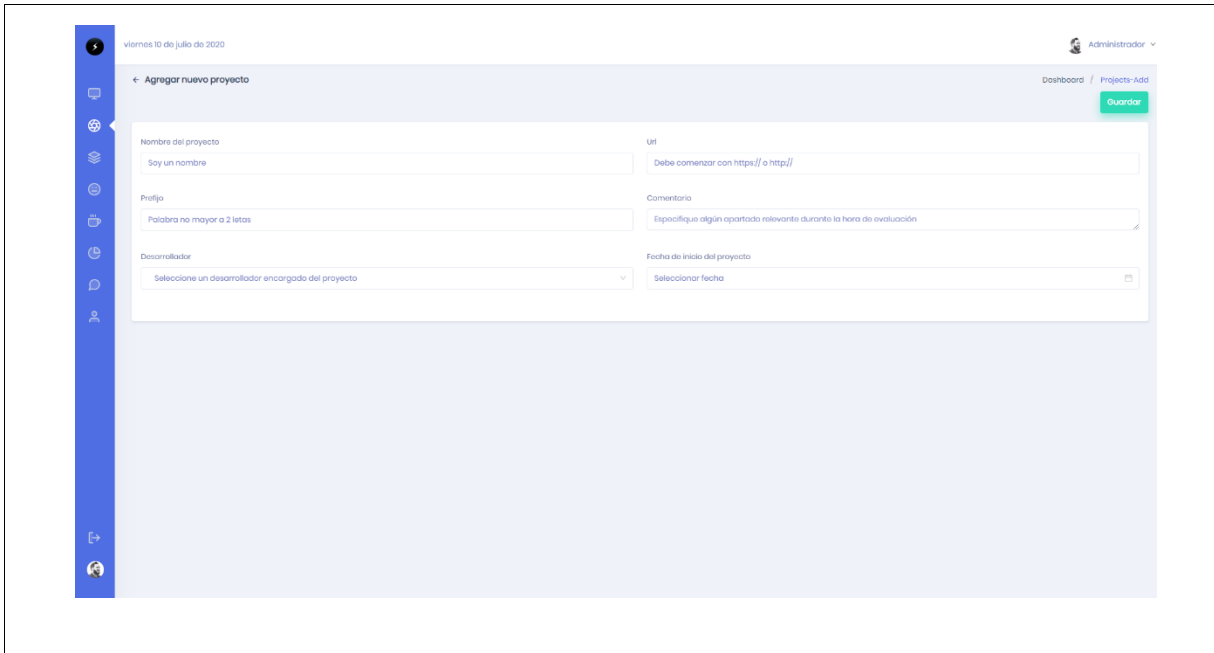
Fuente: Elaboración propia

Explicación: Se muestra la descripción, actores participantes y flujos de eventos del caso de uso de sistema "Autenticar Usuario".

En la siguiente tabla, se muestra las especificaciones del caso de uso "Registro Proyecto".

Tabla 21 Especificaciones del caso de uso "Registro Proyecto"

Caso de uso:	CUS07 - Registro Proyecto	
Actor(es):	Jefe de proyecto	
Descripción:	El sistema permite al usuario "jefe de proyecto" registrar proyectos.	
Precondiciones:	El Jefe de Proyecto del sistema debe haber iniciado sesión El usuario debe contar con el rol Jefe de Proyecto	
Postcondiciones:	El Jefe de Proyecto ha registrado el proyecto.	
FLUJO DE EVENTOS		
Acción del Actor	Acción del Sistema	
2. El Jefe de Proyecto ingresa el nombre del proyecto, ingresa una descripción, ingresa la url del proyecto, el prefijo del proyecto, si lo desea ingresa un comentario del proyecto, selecciona la fecha que comenzó el proyecto y selecciona el desarrollador encargado del proyecto. 4. El Jefe de Proyecto da click en guardar	1. El sistema muestra la interfaz "Registrar Proyecto", conformado por la sección de datos del proyecto, detalle del proyecto, fecha inicio del proyecto, url del proyecto. 3. El sistema valida los datos válidos. 5. El sistema envía los datos al backend para registrar el proyecto. 8. El sistema muestra mensaje: "Proyecto registrado correctamente". 9. El caso de uso finaliza	
FLUJOS ALTERNATIVOS		
FA01: "Datos Obligatorios"		
Si el Jefe de Proyecto ingresa algún dato no valido o deja algún campo vacío el sistema mostrará el mensaje "Ingresar campos obligatorios" y marcará los campos que deben ser ingresados		
PROTOTIPO		



Fuente: Elaboración propia

Explicación: Se muestra la descripción, actores participantes y flujos de eventos del caso de uso de sistema “Registro Proyecto”.

En la siguiente tabla, se muestra las especificaciones del caso de uso “Registrar Casos de Pruebas”.

Tabla 22 Especificaciones del caso de uso "Registrar Casos de Pruebas"

Caso de uso:	CUS09 - Registrar Casos de Pruebas	
Actor(es):	Jefe de proyecto	
Descripción:	El sistema permite al usuario “jefe de proyecto” registrar casos de prueba.	
Precondiciones:	El Jefe de Proyecto del sistema debe haber iniciado sesión El usuario debe contar con el rol Jefe de Proyecto	
Postcondiciones:	El Jefe de Proyecto ha registrado el caso de prueba	
FLUJO DE EVENTOS		
Acción del Actor	Acción del Sistema	
2. El Jefe de Proyecto selecciona el proyecto correspondiente al caso de prueba, ingresa el nombre del caso de	1. El sistema muestra la interfaz “Registrar Caso de prueba”.	

<p>prueba, ingresa un identificador del caso de prueba, selecciona la complejidad del caso de prueba, selecciona el diseño de la prueba, si lo desea el usuario puede agregar una descripción.</p> <p>2. El usuario da clic en guarda.</p> <p>8. El usuario si lo desea ingresa las pre condiciones en las que se ejecutara el caso de prueba, ingresa las post condiciones , condiciones especiales si las haya, y los pasos del caso de prueba, así como los resultados esperados del caso de prueba</p>	<p>4. El sistema valida los datos ingresados.</p> <p>5. El sistema muestra el mensaje “Caso de prueba registrado”</p> <p>6. El sistema redirecciona al usuario a la ventana del caso de prueba.</p> <p>7. El sistema redirige a la ventana de editar caso de prueba, donde el usuario ingresa datos adicionales como condiciones previas, esperadas, posteriores, especiales y los pasos a seguir para el caso de prueba.</p> <p>9. El usuario asigna al tester, el cual es otro caso de uso.</p> <p>10. El caso de uso finaliza.</p>
--	---

FLUJOS ALTERNATIVOS

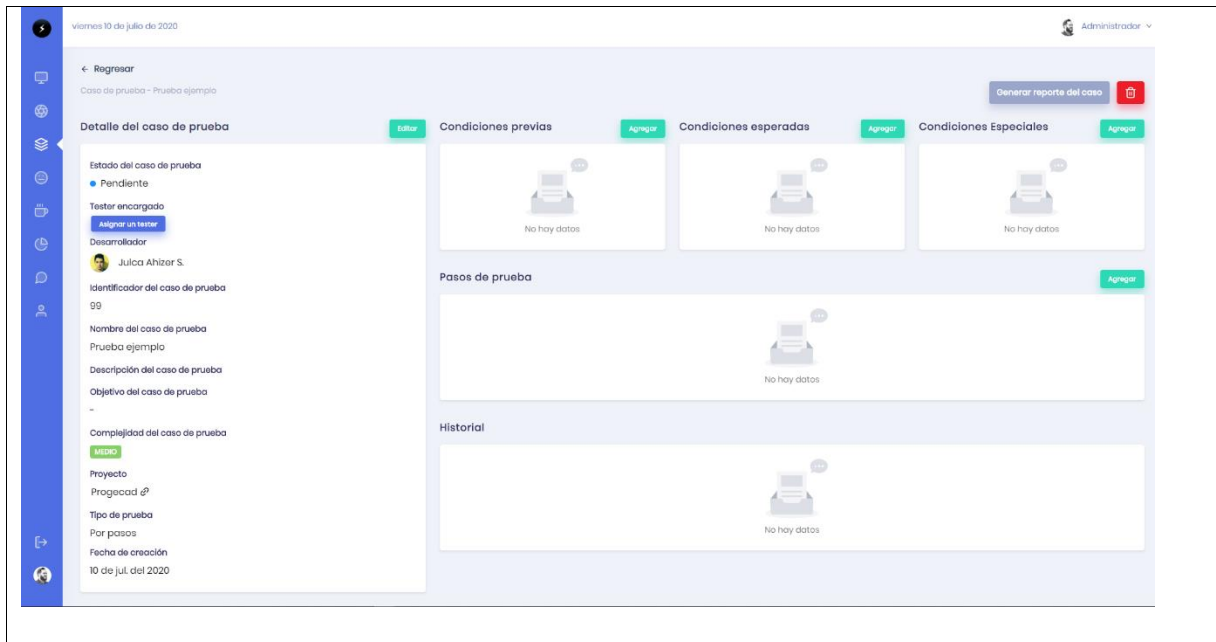
FA01: “Datos Obligatorios”

Si el Jefe de Proyecto ingresa algún dato no válido o deja algún campo vacío el sistema mostrará el mensaje “Ingresar campos obligatorios” y marcará los campos que deben ser ingresados

PROTOTIPO

The screenshot shows a web application interface for creating a new test case. The main content area is titled "Nuevo caso de prueba" and contains several input fields and a text area. On the left, there is a vertical sidebar with various navigation icons. At the top, there is a navigation bar with a "Regresar" button and a user profile dropdown menu. The form fields include:

- Proyecto:** A dropdown menu with the option "Seleccionar el proyecto".
- Nombre del caso de prueba:** A text input field with the placeholder "Nombre del caso de prueba".
- Fecha de creado:** A date input field showing "2020-07-10".
- Descripción:** A rich text editor area with the placeholder "Especifique alguna descripción".
- Complejidad:** A section with three radio buttons labeled "COMPLEJIDAD BAJA", "COMPLEJIDAD MEDIO", and "COMPLEJIDAD ALTA".
- Código de identificación:** A text input field with the placeholder "Código de identificación (Máximo 2 caracteres)".
- Diseño de prueba:** A dropdown menu with the selected option "Prueba por Pasos" and a "Ver diseño" button.



Fuente: Elaboración propia

Explicación: Se muestra la descripción, actores participantes y flujos de eventos del caso de uso de sistema “Registrar Casos de Pruebas”.

En la siguiente tabla, se muestra las especificaciones del caso de uso “Registrar Resultados”.

Tabla 23 Especificaciones del caso de uso "Registrar Resultados"

Caso de uso:	CUS12 - Resolver Casos de prueba	
Actor(es):	Tester	
Descripción:	El sistema permite al usuario “Tester” resolver los casos de prueba.	
Precondiciones:	El Tester del sistema debe haber iniciado sesión El usuario debe contar con el rol Tester El caso de prueba debe estar asignado al Tester	
Postcondiciones:	El Tester ha registrado los resultados	
FLUJO DE EVENTOS		
Acción del Actor	Acción del Sistema	

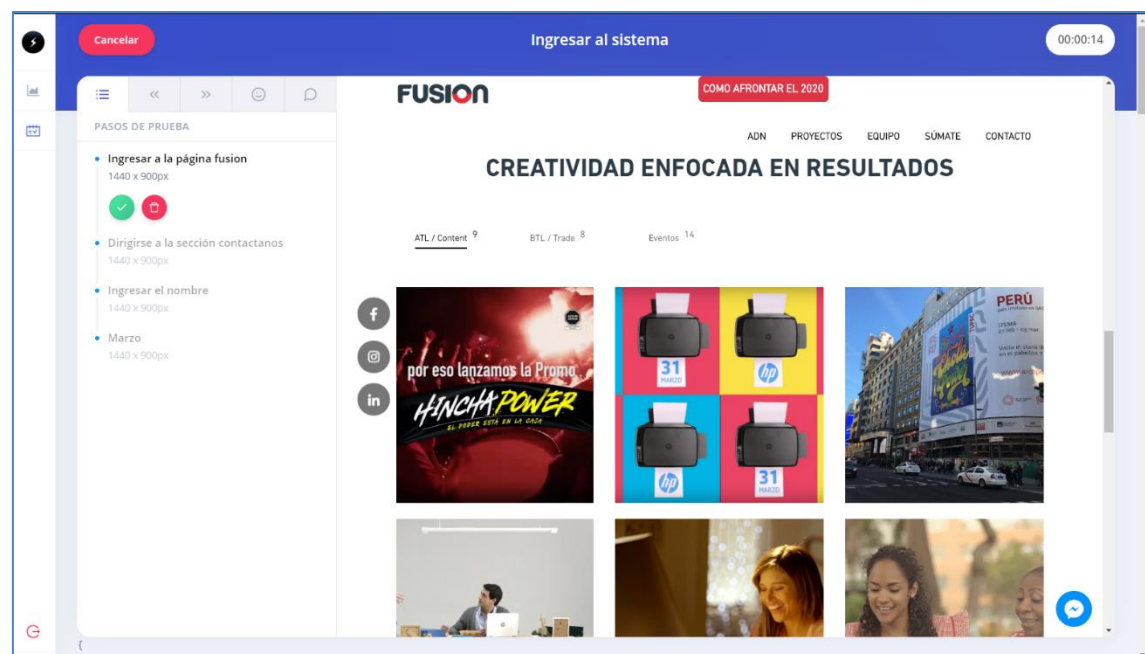
<p>2. El Tester selecciona que casos de prueba desea resolver.</p> <p>5. El usuario prueba la página según los pasos que están registrados.</p> <p>7. El Tester hace click en el botón “Enviar resultados”.</p>	<p>1. El sistema muestra la interfaz de “Casos asignados” donde se enlista los casos de prueba asignados al tester y pendientes.</p> <p>3. El sistema enviará al tester a la interfaz de “Resolver caso de prueba”.</p> <p>6. El usuario evaluará el caso de prueba y resolverá según su criterio si el caso de prueba cumple o no para resolver el caso de prueba.</p> <p>8. El sistema enviará un correo electrónico al jefe de proyecto indicándole los resultados del Tester.</p> <p>9. El caso de uso finaliza</p>
---	---

FLUJOS ALTERNATIVOS

FA01: “Seleccionar al menos un caso de prueba”

Si el Tester no selecciona ningún caso de prueba para registrar el resultado, el sistema mostrara el mensaje “Debe seleccionar al menos un mínimo de 1 caso de prueba para poder grabar los resultados”.

PROTOTIPO



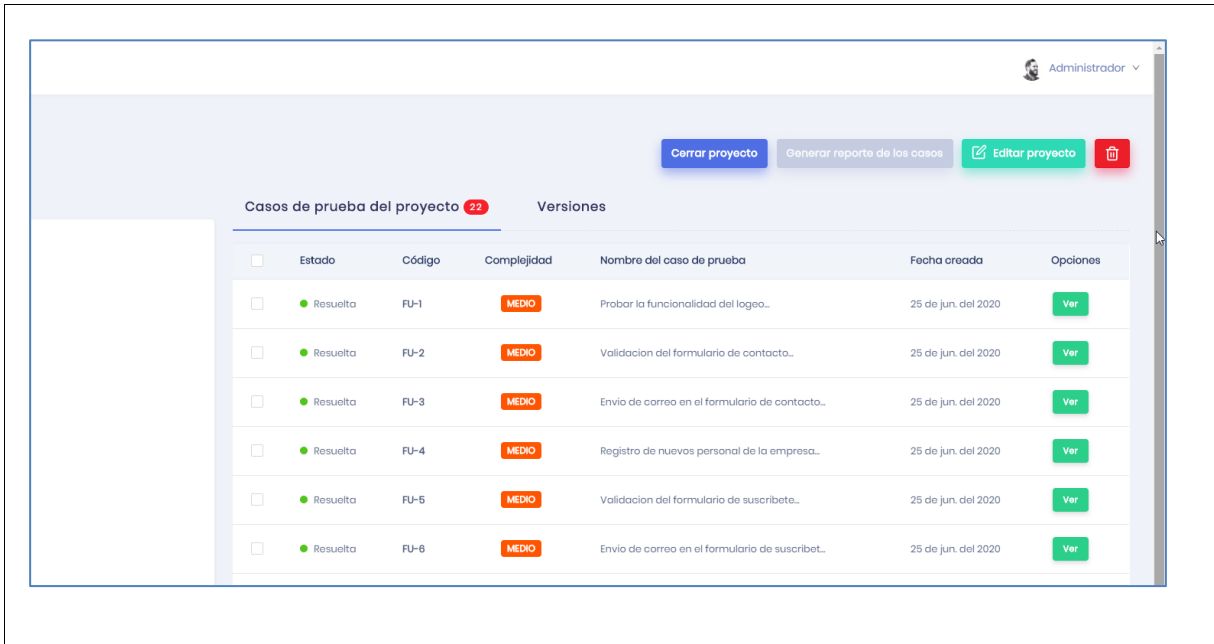
Fuente: Elaboración propia

Explicación: Se muestra la descripción, actores participantes y flujos de eventos del caso de uso de sistema “Registrar Resultados”.

En la siguiente tabla, se muestra las especificaciones del caso de uso "Analizar Proyecto".

Tabla 24 Especificaciones del caso de uso " Cerrar Proyecto"

Caso de uso:	CUS13 – Cerrar Proyecto	
Actor(es):	Jefe de Proyecto	
Descripción:	El sistema permite al usuario "Jefe de Proyecto" cerrar proyecto.	
Precondiciones:	El Jefe de Proyecto del sistema debe haber iniciado sesión El usuario debe contar con el rol Jefe de Proyecto	
Postcondiciones:	El Sistema cambia de estado al proyecto	
FLUJO DE EVENTOS		
Acción del Actor	Acción del Sistema	
2. El Jefe de proyecto selecciona que proyecto desea darle por cerrado. 4. El Jefe de Proyecto hace click en el botón de Aceptar para cerrar el proyecto.	1. El sistema muestra la interfaz de "Proyectos", donde se enlista todos los proyectos activos. 3. El sistema mostrará un modal con un mensaje "Desea cerrar el proyecto". 5. El sistema mostrará un mensaje de confirmación. 6. El sistema cambia de estado al proyecto. 7. El caso de uso finaliza	
FLUJOS ALTERNATIVOS		
Ninguno		
PROTOTIPO		



Fuente: Elaboración propia

Explicación: Se muestra la descripción, actores participantes y flujos de eventos del caso de uso de sistema “Analizar Proyecto”.

En la siguiente tabla, se muestra las especificaciones del caso de uso “Registrar Errores”.

Tabla 25 Especificaciones del caso de uso “Registrar Errores”

Caso de uso:	CUS15 - Registrar Errores	
Actor(es):	Jefe de Proyecto	
Descripción:	El sistema permite al usuario “Jefe de Proyecto” registrar errores por proyecto	
Precondiciones:	El Jefe de Proyecto del sistema debe haber iniciado sesión El usuario debe contar con el rol Jefe de Proyecto	
Postcondiciones:	El Jefe de Proyecto ha registrado el caso de prueba	
FLUJO DE EVENTOS		
Acción del Actor	Acción del Sistema	

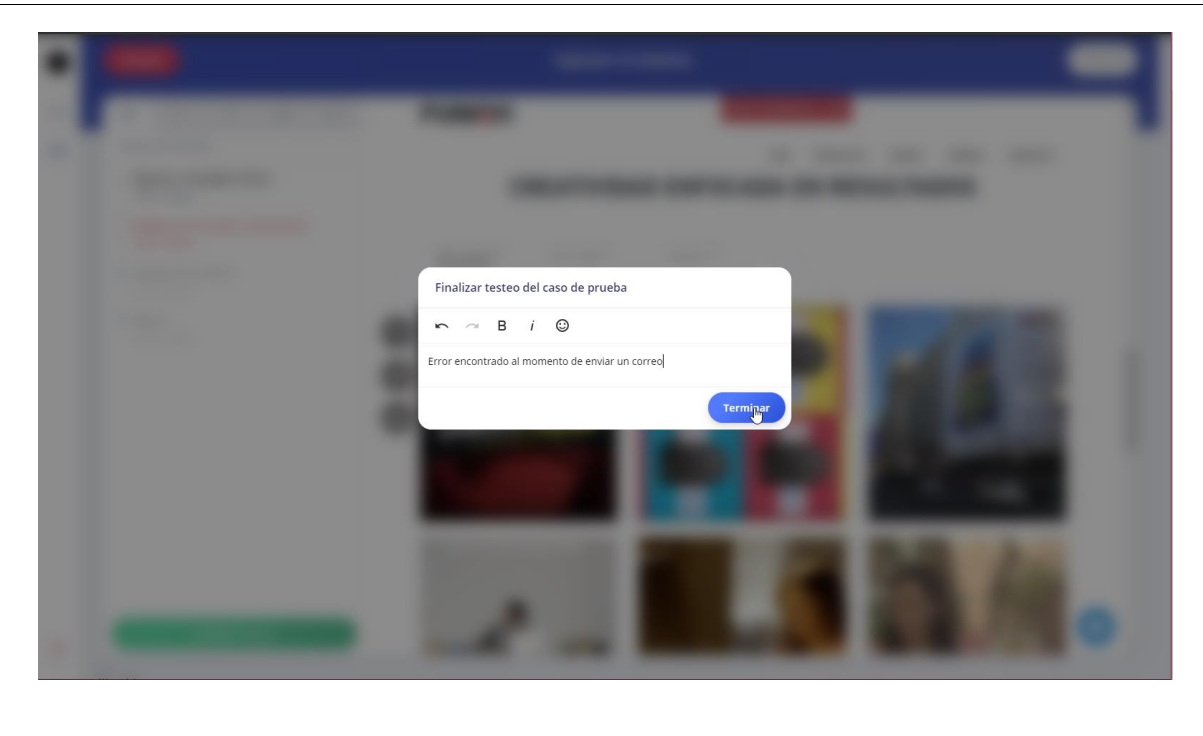
<p>2. El Jefe de Proyecto ingresa el detalle del error, ingresa la fecha del error, la fecha en la que se detectó el error y selecciona la complejidad del error.</p> <p>4. El Jefe de Proyecto hace click en el botón de “Registrar Error”.</p>	<p>1. El sistema muestra la interfaz “Registrar errores”, conformado por la sección de datos del error por proyecto: generar número de error, detalle del error, fecha del error y complejidad.</p> <p>3. El sistema genera automáticamente el número del error.</p> <p>5. El sistema muestra el mensaje “El error se ha registrado correctamente, y se ha notificado al desarrollador”.</p> <p>6. El sistema notifica al desarrollador del error por correo electrónico.</p> <p>7. El caso de uso finaliza</p>
--	---

FLUJOS ALTERNATIVOS

FA01: “Datos Obligatorios”

Si el Jefe de Proyecto ingresa algún dato no valido o deja algún campo vacío el sistema mostrará el mensaje “Ingresar campos obligatorios” y marcará los campos que deben ser ingresados

PROTOTIPO



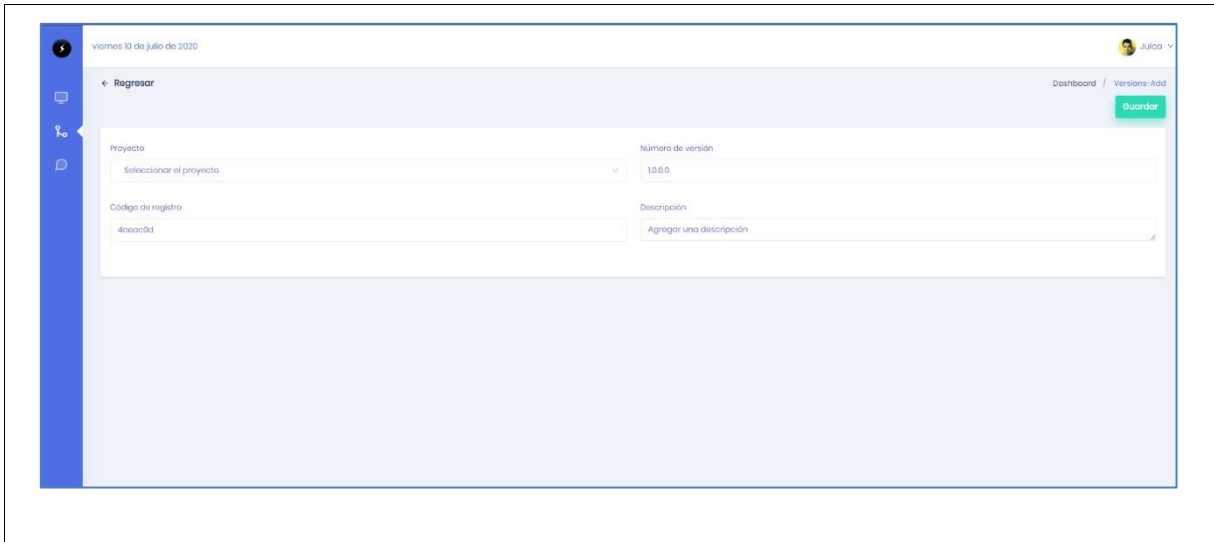
Fuente: Elaboración propia

Explicación: Se muestra la descripción, actores participantes y flujos de eventos del caso de uso de sistema “Registrar Errores”.

En la siguiente tabla, se muestra las especificaciones del caso de uso “Registrar Versiones”.

Tabla 26 Especificaciones del caso de uso “Registrar Versiones”

Caso de uso:	CUS17 – Registrar Versiones	
Actor(es):	Desarrollador	
Descripción:	El sistema permite al “Desarrollador” registrar versiones de los proyectos	
Precondiciones:	El Desarrollador del sistema debe haber iniciado sesión El usuario debe contar con el rol Desarrollador El Desarrollador debe estar asignado al proyecto	
Postcondiciones:	El Desarrollador ha registrado una nueva versión del proyecto	
FLUJO DE EVENTOS		
Acción del Actor	Acción del Sistema	
2. El usuario hace click en el menú “Versiones” 3. El usuario da click en registrar versión. 5. El Desarrollador los datos de la nueva versión. 6. El Desarrollador hace click en el botón de “Registrar versión”.	1. El sistema muestra la interfaz “Dashboard”, conformado por la sección de menú del sistema. 2. El sistema muestra la interfaz “Versiones” 4. El sistema muestra la interfaz “Registrar versiones”, conformado por los datos de la nueva versión del proyecto: proyecto, número de versión, código de registro y descripción de la versión. 7. El sistema muestra el mensaje: “La versión se ha registrado correctamente” 9. El caso de uso finaliza.	
FLUJOS ALTERNATIVOS		
FA01: “Datos Obligatorios”		
Si el Desarrollador ingresa algún dato no valido o deja algún campo vacío el sistema mostrará el mensaje “Ingresar campos obligatorios” y marcará los campos que deben ser ingresados		
PROTOTIPO		



Fuente: Elaboración propia

Explicación: Se muestra la descripción, actores participantes y flujos de eventos del caso de uso de sistema “Registrar Versiones”

H. Realización de Casos de Uso del Sistema

- **CUS01 - Autenticar Usuario**

Fuente: Elaboración propia

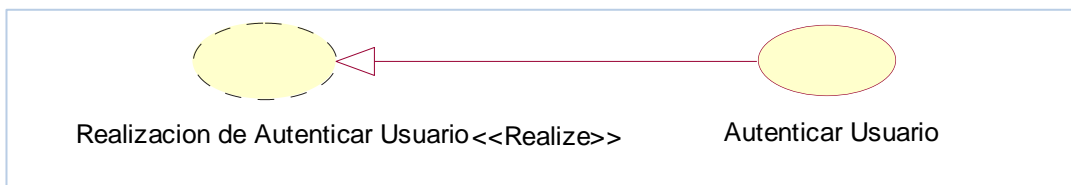


Figura 24 Realización de Caso de Uso del sistema: Autenticar Usuario

En la Figura 7, se observa la realización del caso de uso del sistema Autenticar Usuario.

- **CUS02 - Recuperar contraseña**

Fuente: Elaboración propia

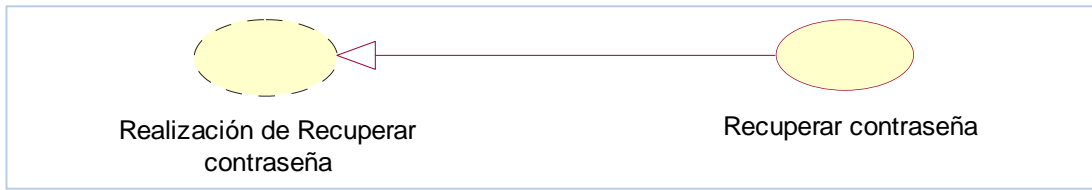


Figura 25 Realización de Caso de Uso del sistema: Recuperar Contraseña

En la Figura 8, se observa la realización del caso de uso del sistema Recuperar Contraseña.

- **CUS03 - Consultar Usuarios**

Fuente: Elaboración propia

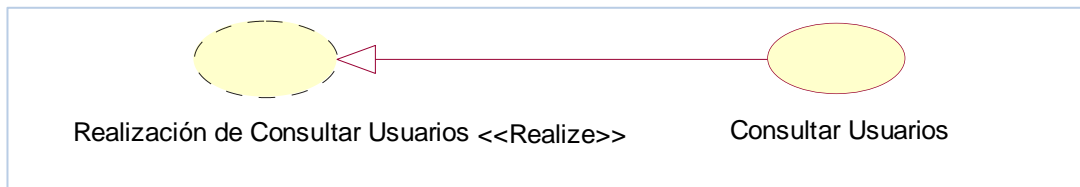


Figura 26 Realización de Caso de Uso del sistema: Consultar Usuarios

En la Figura 9, se observa la realización del caso de uso del sistema Consultar Usuarios.

- **CUS04 - Registrar Usuarios**

Fuente: Elaboración propia

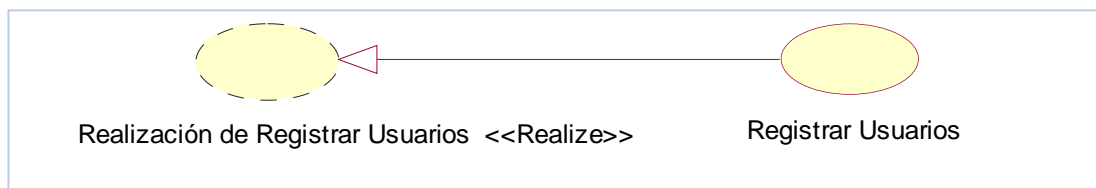


Figura 27 Realización de Caso de Uso del sistema: Registrar Usuarios

En la Figura 10, se observa la realización del caso de uso del sistema Registrar Usuarios.

- **CUS05 - Consultar Roles**

Fuente: Elaboración propia

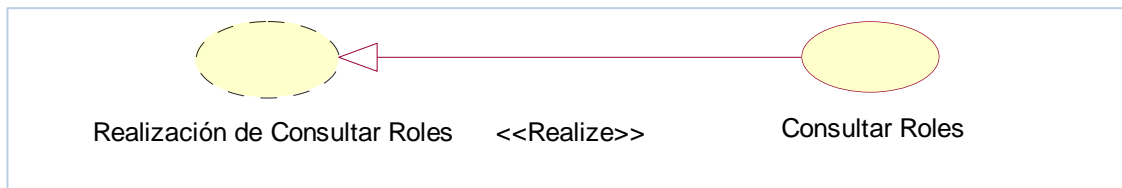


Figura 28 Realización de Caso de Uso del sistema: Consultar Roles

En la Figura 11, se observa la realización del caso de uso del sistema Consultar Roles.

- **CUS06 - Consultar Proyectos**

Fuente: Elaboración propia

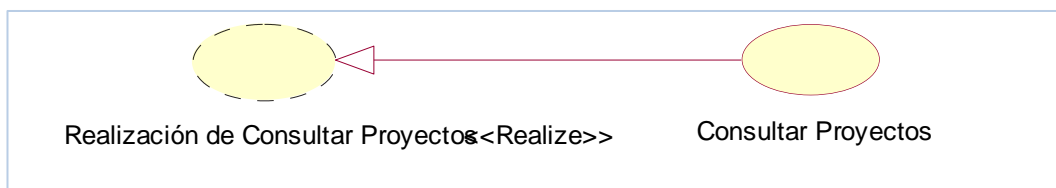


Figura 29 Realización de Caso de Uso del sistema: Consultar Proyectos

En la Figura 12, se observa la realización del caso de uso del sistema Consultar Proyectos.

- **CUS07 - Registro Proyecto**

Fuente: Elaboración propia

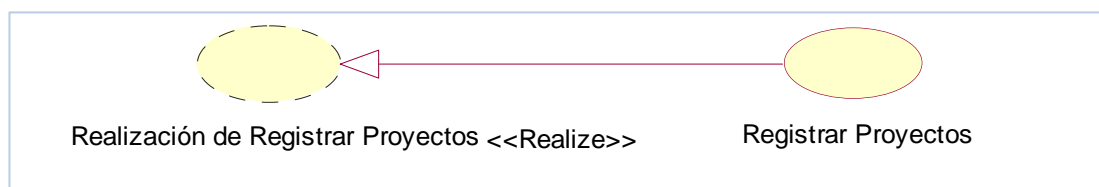


Figura 30 Realización de Caso de Uso del sistema: Registrar Proyectos

En la Figura 13, se observa la realización del caso de uso del sistema Registrar Proyectos.

- **CUS08 - Consultar Casos de Pruebas**

Fuente: Elaboración propia

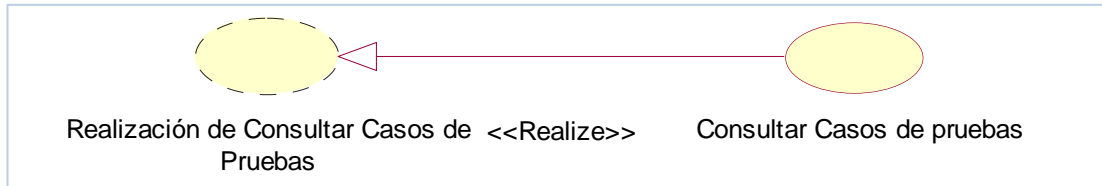


Figura 31 Realización de Caso de Uso del sistema: Consultar Casos de Pruebas

En la Figura 14, se observa la realización del caso de uso del sistema Consultar Casos de Pruebas.

- **CUS09 - Registrar Casos de Pruebas**

Fuente: Elaboración propia

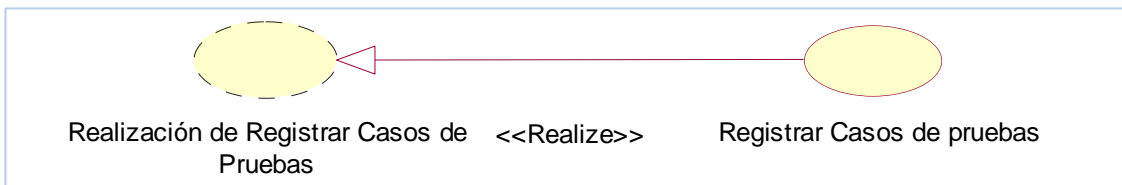


Figura 32 Realización de Caso de Uso del sistema: Registrar Casos de Pruebas

En la Figura 15, se observa la realización del caso de uso del sistema Registrar Casos de Pruebas.

- **CUS10 - Consultar Reportes**

Fuente: Elaboración propia

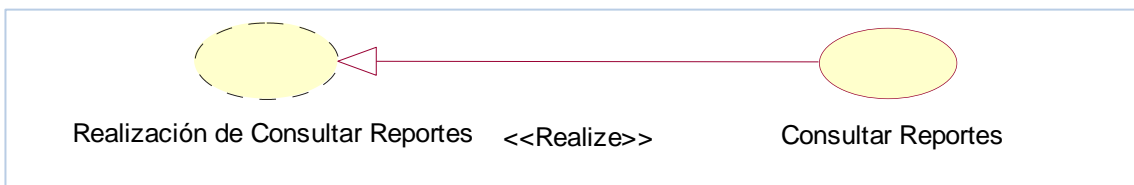


Figura 33 Realización de Caso de Uso del sistema: Consultar Reportes

En la Figura 16, se observa la realización del caso de uso del sistema Consultar Reportes.

- **CUS11 – Generar Reportes**

Fuente: Elaboración propia

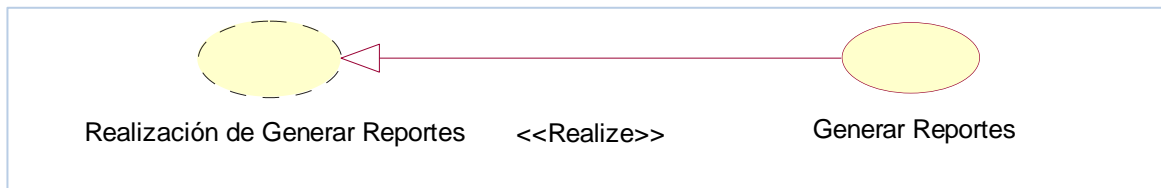


Figura 34 Realización de Caso de Uso del sistema: Generar Reportes

En la Figura 17, se observa la realización del caso de uso del sistema Generar Reportes.

- **CUS12 – Resolver Casos de Pruebas**

Fuente: Elaboración propia

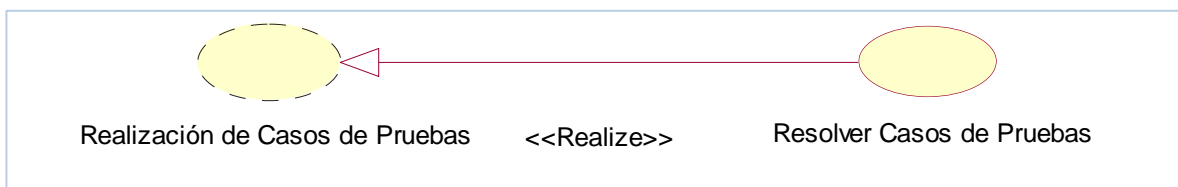


Figura 35 Realización de Caso de Uso del sistema: Resolver Casos de Pruebas

En la Figura 18, se observa la realización del caso de uso del Resolver Casos de Pruebas.

- **CUS13 – Cerrar Proyecto**

Fuente: Elaboración propia

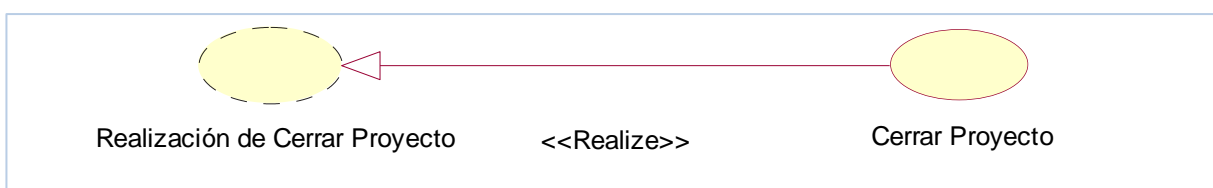


Figura 36 Realización de Caso de Uso del sistema: Cerrar Proyecto

En la Figura 19, se observa la realización del caso de uso del sistema Cerrar Proyecto.

- **CUS14 – Consultar Errores**

Fuente: Elaboración propia

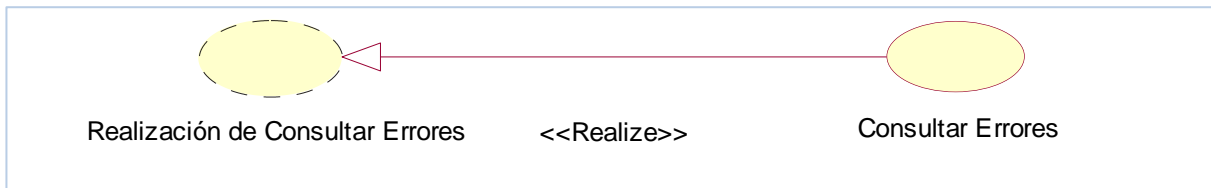


Figura 37 Realización de Caso de Uso del sistema: Consultar Errores

En la Figura 20, se observa la realización del caso de uso del sistema Consultar Errores.

- **CUS15 – Registrar Errores**

Fuente: Elaboración propia

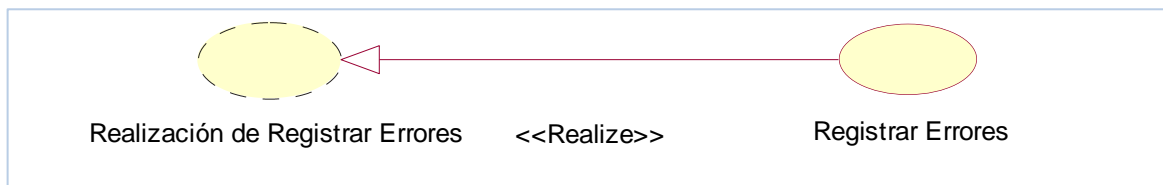


Figura 38 Realización de Caso de Uso del sistema: Registrar Errores

En la Figura 21, se observa la realización del caso de uso del sistema Registrar Errores.

- **CUS16 – Consultar Versiones**

Fuente: Elaboración propia

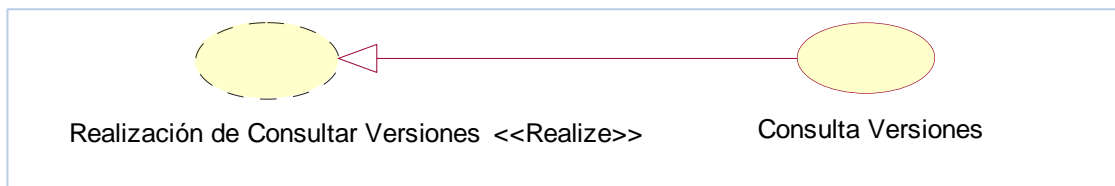


Figura 39 Realización de Caso de Uso del sistema: Consultar Versiones

En la Figura 22, se observa la realización del caso de uso del sistema Consultar Versiones.

- **CUS17 – Registrar Versiones**

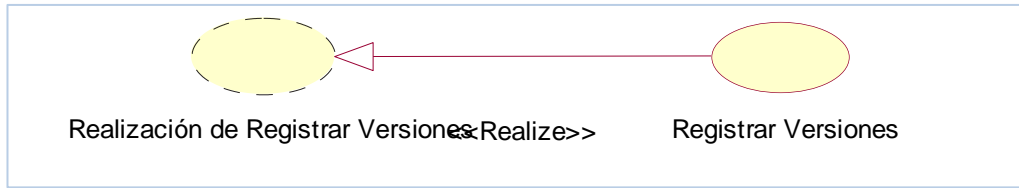


Figura 40 Realización de Caso de Uso del sistema: Registrar Versiones

En la Figura 23, se observa la realización del caso de uso del Registrar Versiones.

- **CUS18 – Asignar Desarrollador a Proyecto**

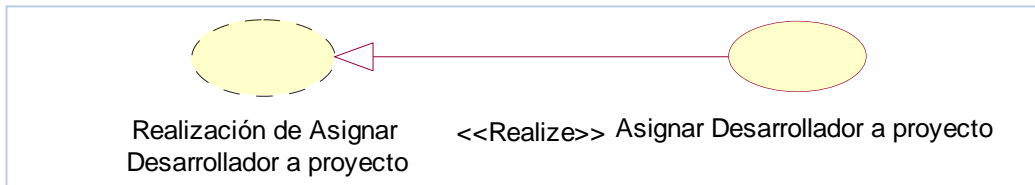


Figura 41 Realización de Caso de Uso del sistema: Asignar Desarrollador a proyecto

En la Figura 24, se observa la realización del caso de uso del Asignar Desarrollador a proyecto.

- **CUS19 - Asignar Tester a Casos de Pruebas**

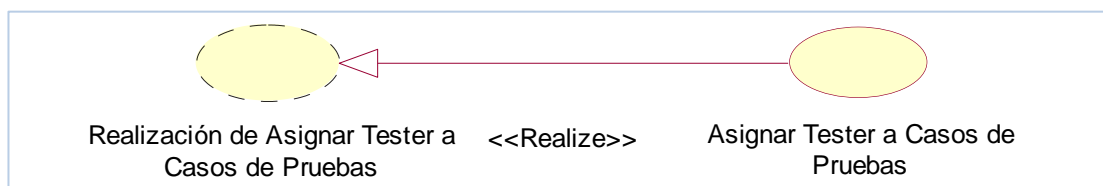


Figura 42 Realización de Caso de Uso del sistema: Asignar Tester a Casos de Pruebas

En la Figura 25, se observa la realización del caso de uso del Asignar Tester a Casos de Pruebas.

Fuente: Elaboración propia

Fuente: Elaboración propia

Fuente: Elaboración propia

- **CUS20 – Notificar Correo Electrónico**

Fuente: Elaboración propia

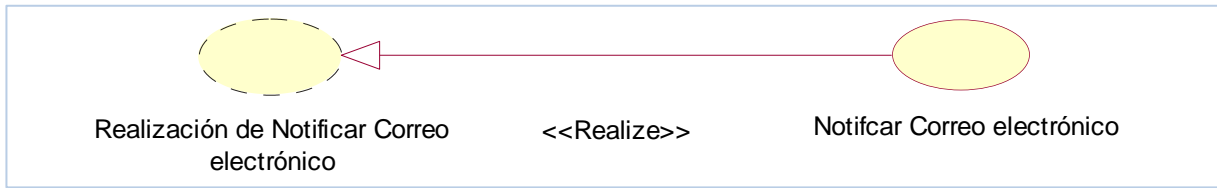


Figura 43 Realización de Caso de Uso del sistema: Notificar Correo electrónico

En la Figura 26, se observa la realización del caso de uso del Notificar Correo electrónico.

I. Diagramas de clases de análisis

- **Caso de Uso 01: Autenticar Usuario**

En la siguiente figura se muestra el diagrama de clases del caso de uso de sistema “Autenticar Usuario”, plasmando la relación entre usuarios, interfaces, routes, controladores y entidades.

Fuente: Elaboración propia

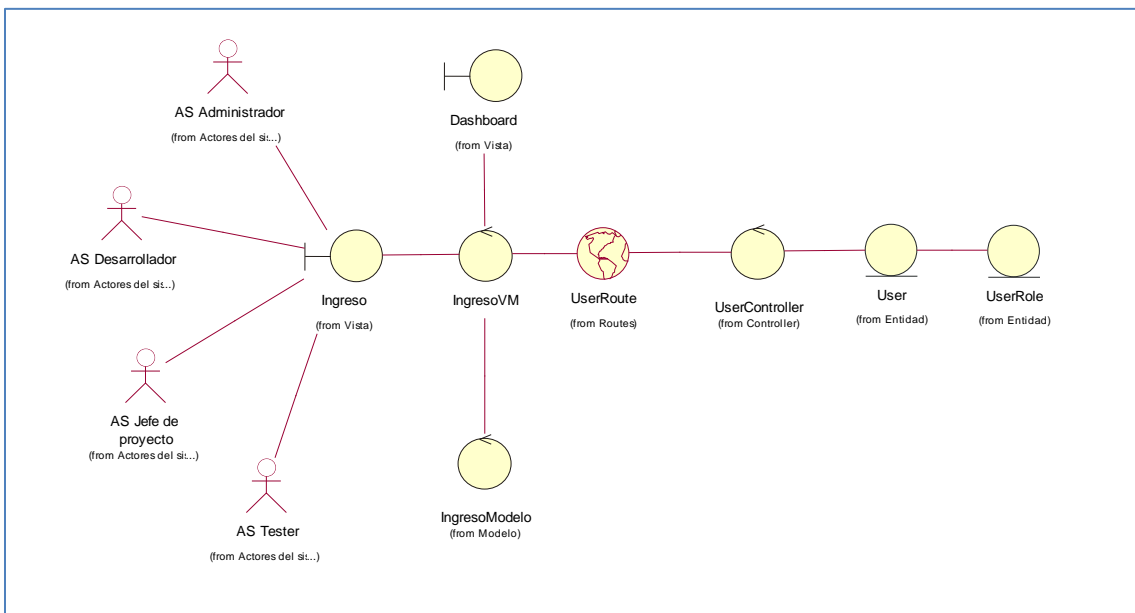


Figura 44 Diagrama de Clases de análisis “Autenticar Usuario”

Explicación: En la Figura N° 27, se muestra el caso de uso del sistema Autenticar Usuario contiene 1 route, que se encarga de recibir los datos de la vista Ingreso por medio del IngresoVM encargada de recoger los datos, 1 control, el UserController, que se encarga de recibir los datos que le son enviados.

- **Caso de Uso 08: Registro Proyecto**

En la siguiente figura se muestra el diagrama de clases del caso de uso de sistema “Registrar Proyecto”, plasmando la relación entre usuarios, interfaces, controladores y entidades.

Fuente: Elaboración propia

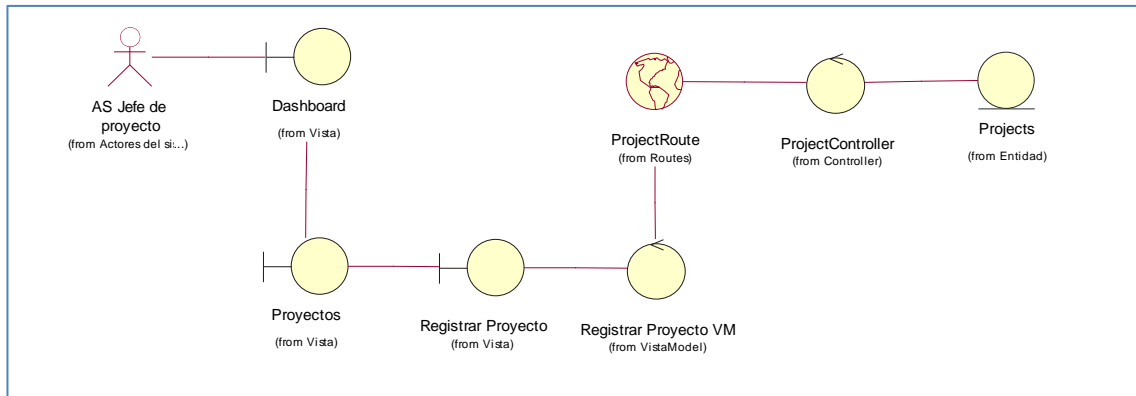


Figura 45 Diagrama de Clases de análisis “Registro Proyecto”

Explicación: En la Figura N° 28, se muestra el caso de uso del sistema Registro Proyecto que contiene 1 route, que se encarga de recibir los datos de la vista Registrar Proyecto por medio del Registrar Proyecto VM encargada de recoger los datos, 1 control, el ProjectController, que se encarga de recibir los datos que le son enviados.

- **Caso de Uso 10: Registrar Casos de Pruebas**

En la siguiente figura se muestra el diagrama de clases del caso de uso de sistema “Registrar Casos de Pruebas”, plasmando la relación entre usuarios, interfaces, routes, controladores y entidades.

Fuente: Elaboración propia

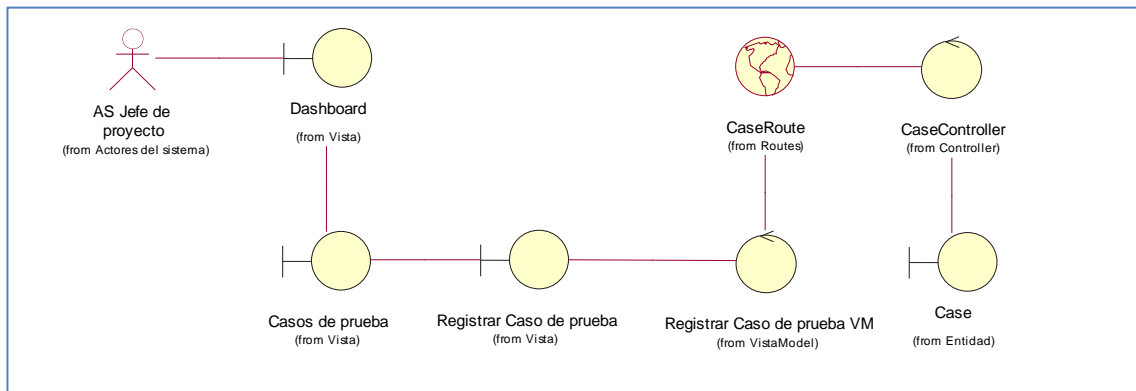


Figura 46 Diagrama de Clases de análisis “Registrar Casos de pruebas”

Explicación: En la Figura N° 29, se muestra el caso de uso del sistema Registrar Casos de Pruebas que contiene 1 route, que se encarga de recibir los datos de la vista Registrar Casos de Pruebas por medio de la vista-modelo Registrar Casos de Pruebas VM encargada de recoger los datos, 1 control, el CaseController, que se encarga de recibir los datos que le son enviados.

- **Caso de Uso 13: Resolver Casos de prueba**

En la siguiente figura se muestra el diagrama de clases del caso de uso de sistema “Resolver Casos de prueba”, plasmando la relación entre usuarios, interfaces, routes, controladores y entidades

Fuente: Elaboración propia

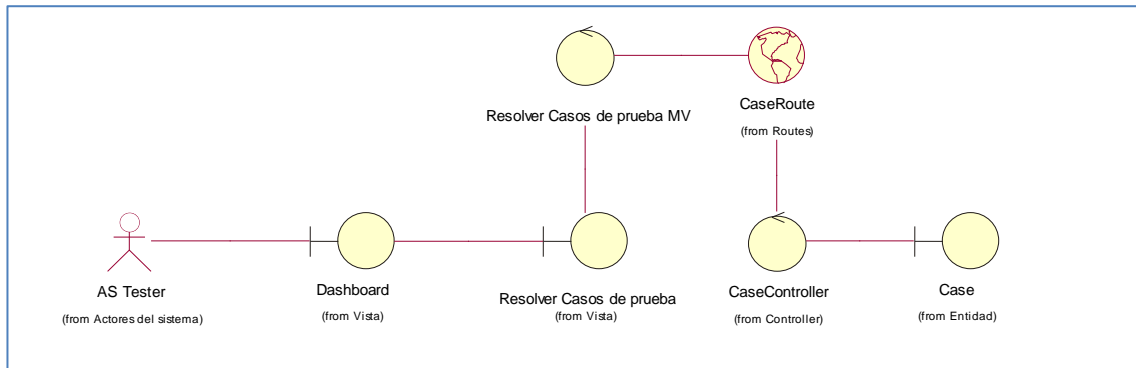


Figura 47 Diagrama de Clases de análisis “Resolver Casos de prueba”

Explicación: En la Figura N° 30, se muestra el caso de uso del sistema Resolver Casos de prueba que contiene 1 route, que se encarga de recibir los datos de la vista Resolver Casos de prueba por medio de la vista-modelo Resolver Casos de prueba VM encargada de recoger los datos, 1 control, el CaseController, que se encarga de recibir los datos que le son enviados.

- **Caso de Uso 14: Cerrar Proyecto**

En la siguiente figura se muestra el diagrama de clases del caso de uso de sistema "Cerrar Proyecto", plasmando la relación entre usuarios, interfaces, routes, vista-modelo, controladores y entidades

Fuente: Elaboración propia .

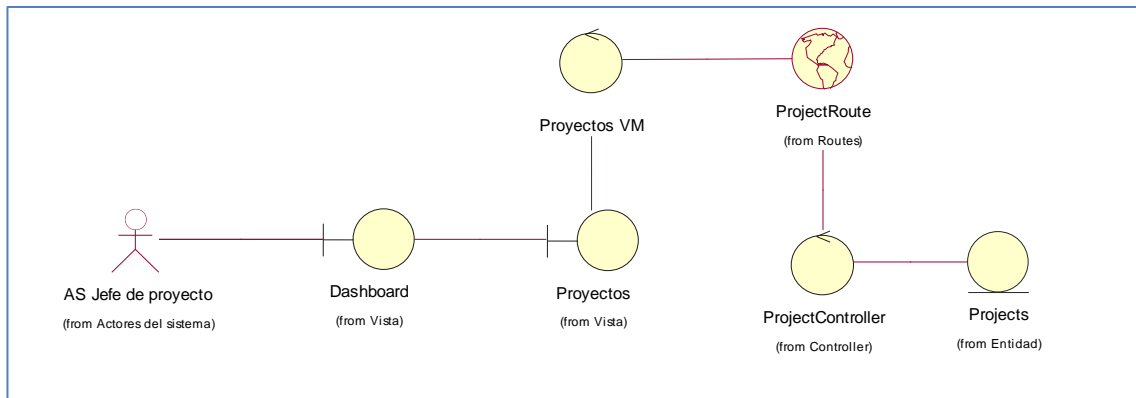


Figura 48 Diagrama de Clases de análisis "Cerrar Proyecto"

Explicación: En la Figura N° 31, se muestra el caso de uso del sistema Cerrar Proyecto que contiene 1 route, que se encarga de recibir los datos de la vista Cerrar Proyecto por medio de la vista-modelo Cerrar Proyecto VM encargada de recoger los datos, 1 control, el ProjectController, que se encarga de recibir los datos que le son enviados.

- **Caso de Uso 16: Registrar Errores**

En la siguiente figura se muestra el diagrama de clases del caso de uso de sistema "Registrar Errores", plasmando la relación entre usuarios, interfaces, routes, vista-modelo, controladores y entidades

Fuente: Elaboración propia

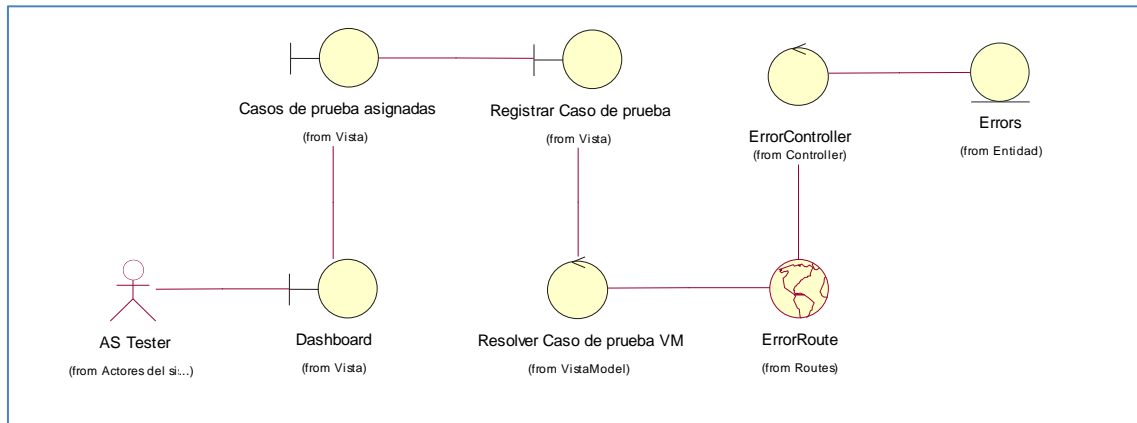


Figura 49 Diagrama de Clases de análisis "Registrar Errores"

Explicación: En la Figura N° 32, se muestra el caso de uso del sistema Registrar Errores que contiene 1 route, que se encarga de recibir los datos de la vista Registrar Errores por medio de la vista-modelo Registrar Errores VM encargada de recoger los datos, 1 control, el ErrorController, que se encarga de recibir los datos que le son enviados.

- **Caso de Uso 18: Registrar Versiones**

En la siguiente figura se muestra el diagrama de clases del caso de uso de sistema "Registrar Versiones", plasmando la relación entre usuarios, interfaces, routes, vista-modelo, controladores y entidades

Fuente: Elaboración propia

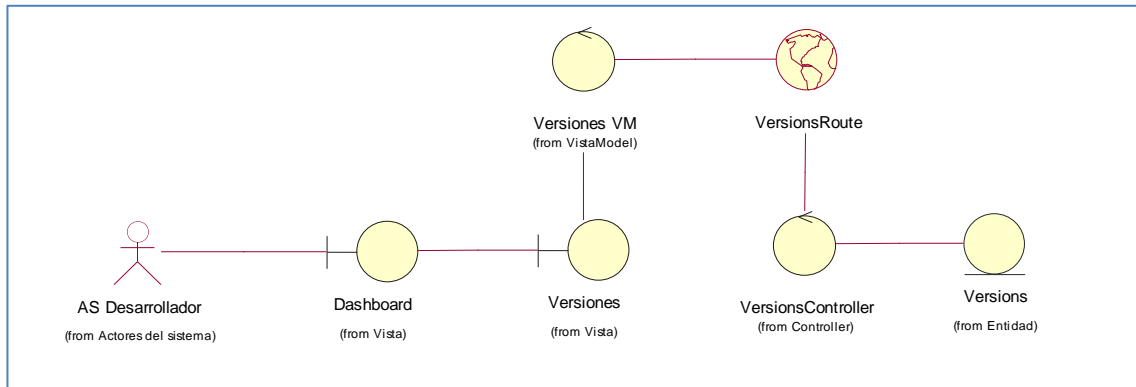


Figura 50 Diagrama de Clases de análisis "Registrar Versiones"

Explicación: En la Figura N° 33, se muestra el caso de uso del sistema Registrar Versiones que contiene 1 route, que se encarga de recibir los datos de la vista Registrar Versiones por medio de la vista-modelo Registrar Versiones VM encargada de recoger los datos, 1 control, el VersionsController, que se encarga de recibir los datos que le son enviados.

J. Listado de Interfaces

Fuente: Elaboración propia

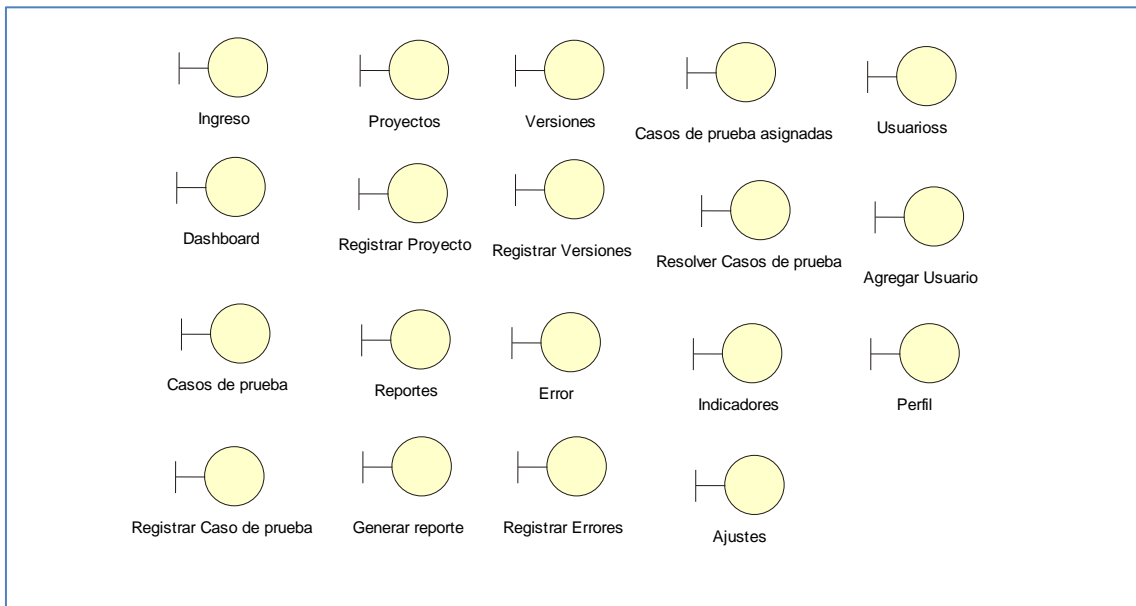


Figura 51 Lista de interfaces

En la Figura N° 24, se observa la lista de interfaces que conforman el sistema.

FRONTEND

K. Diseño de Interfaces (Vistas)

- **Interfaz 01: Ingreso**

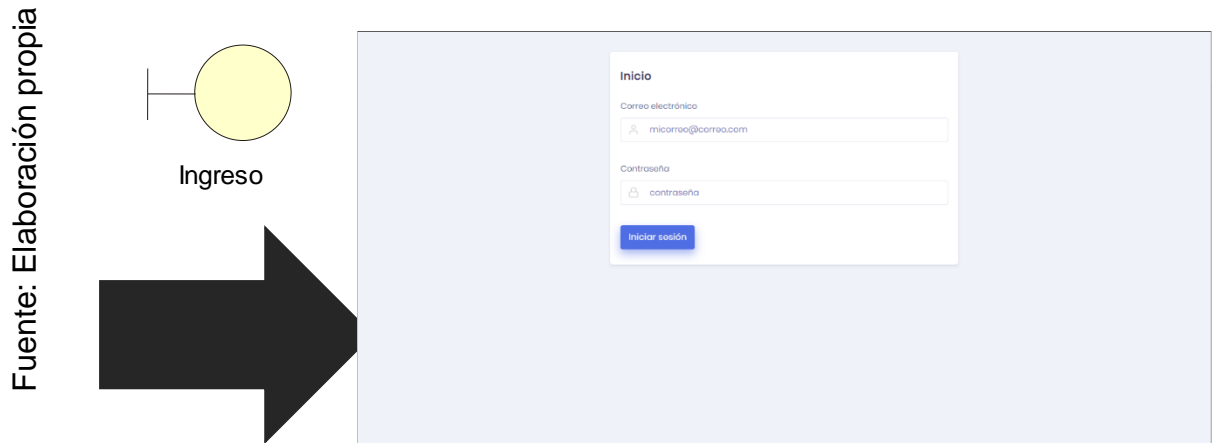


Figura 52 Interfaz: Ingreso

Explicación: El usuario Administrador, Jefe de Proyecto, Tester y Developer, podrán iniciar sesión para ingresar al sistema. Así como se muestra en la Figura N° 35.

- **Interfaz 02: Dashboard**

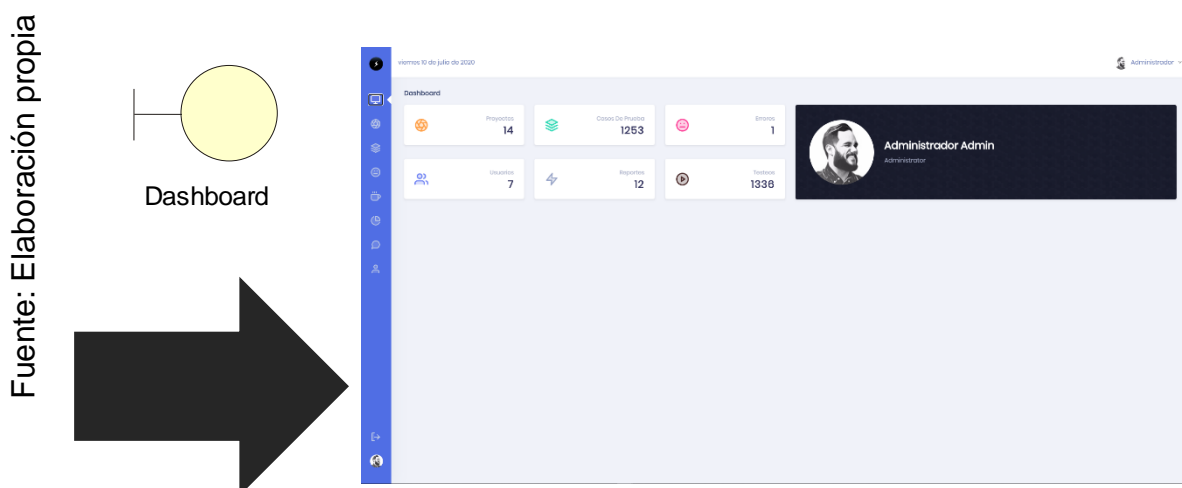


Figura 53 Interfaz Dashboard

Explicación: El usuario Administrador, Jefe de Proyecto, Tester y Developer, podrán ver el resumen del sistema. Así como se muestra en la Figura N° 36.

- **Interfaz 03: Casos de prueba**

Fuente: Elaboración propia

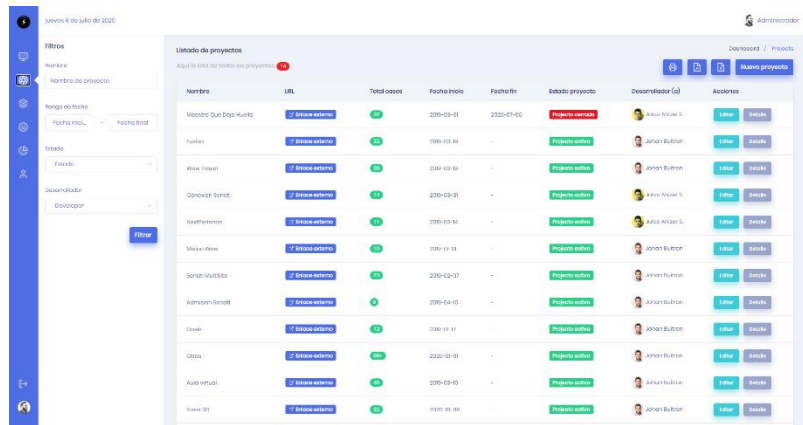
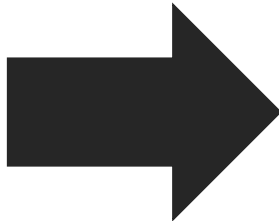
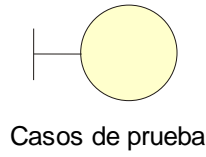


Figura 54 Interfaz Caso de prueba

Explicación: El usuario Jefe de Proyecto, podrá visualizar el estado de todos los casos de prueba. Así como se muestra en la Figura N° 37.

- **Interfaz 04: Registrar Caso de prueba**

Fuente: Elaboración propia

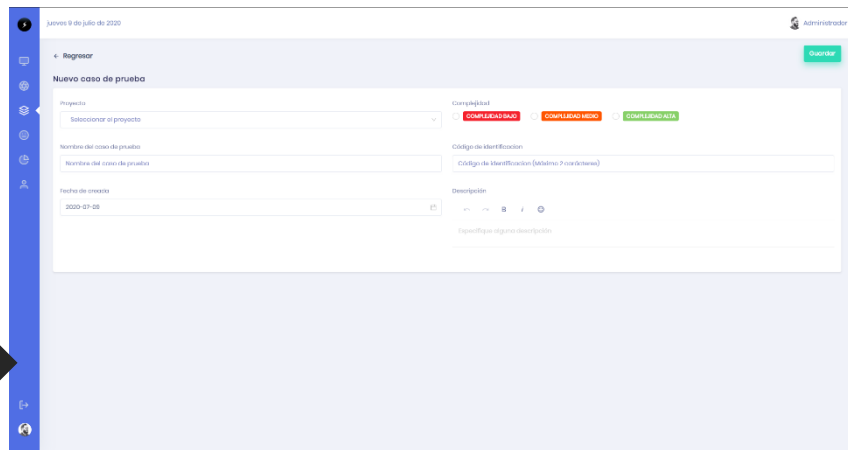
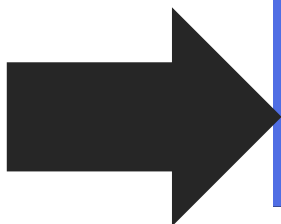
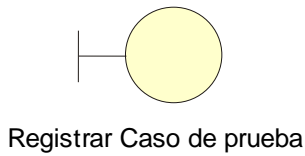


Figura 55 Interfaz Registrar Caso de prueba

Explicación: El usuario Jefe de Proyecto, podrá agregar nuevos casos de prueba. Así como se muestra en la Figura N° 38.

- **Interfaz 05: Proyectos**

Fuente: Elaboración propia

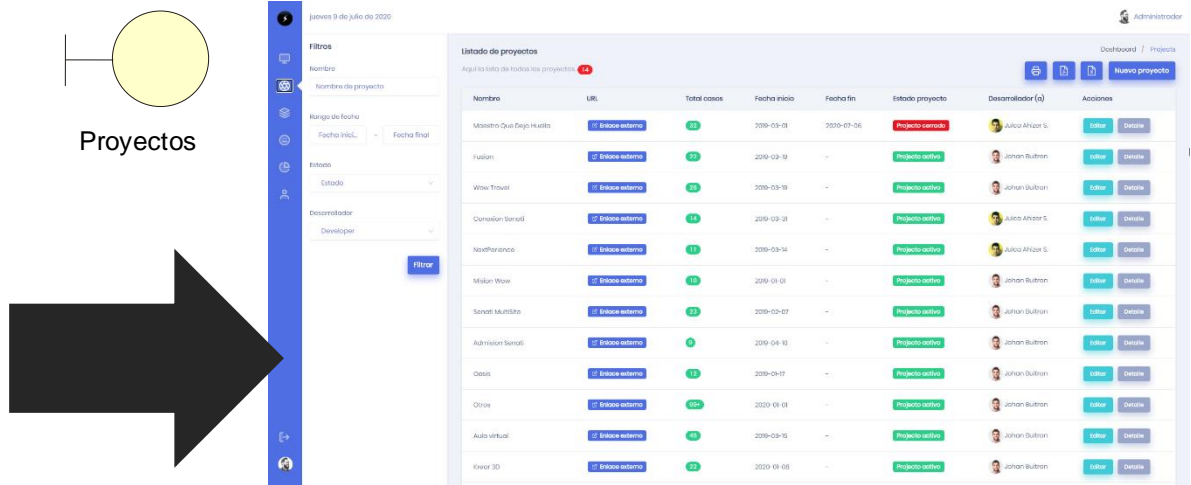


Figura 56 Interfaz Proyectos

Explicación: El usuario Jefe de Proyecto, podrá visualizar el estado de todos los Proyectos. Así como se muestra en la Figura N° 39.

- **Interfaz 06: Registrar Proyecto**

Fuente: Elaboración propia

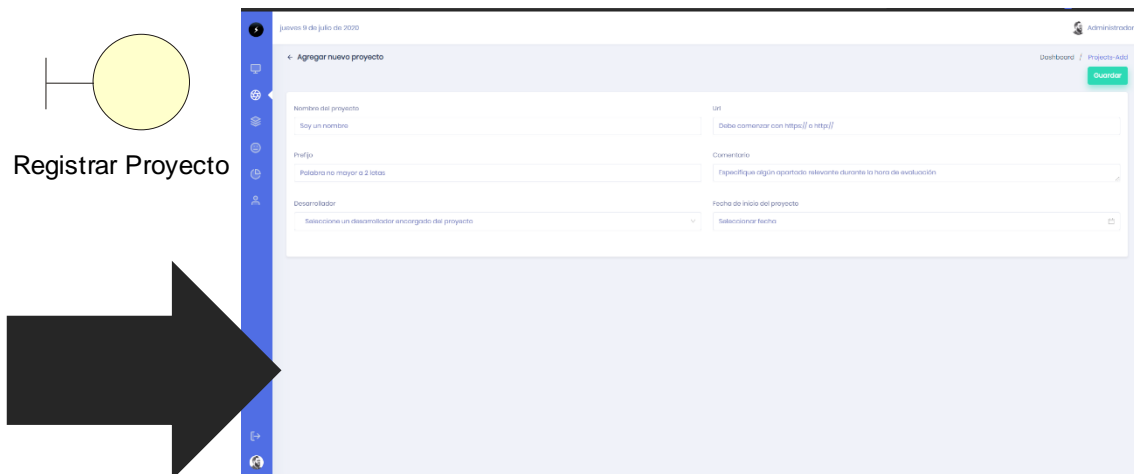


Figura 57 Interfaz Registrar Proyecto

Explicación: El usuario Jefe de Proyecto, podrá agregar nuevos proyectos. Así como se muestra en la Figura N° 40.

- **Interfaz 07: Reportes**

Fuente: Elaboración propia

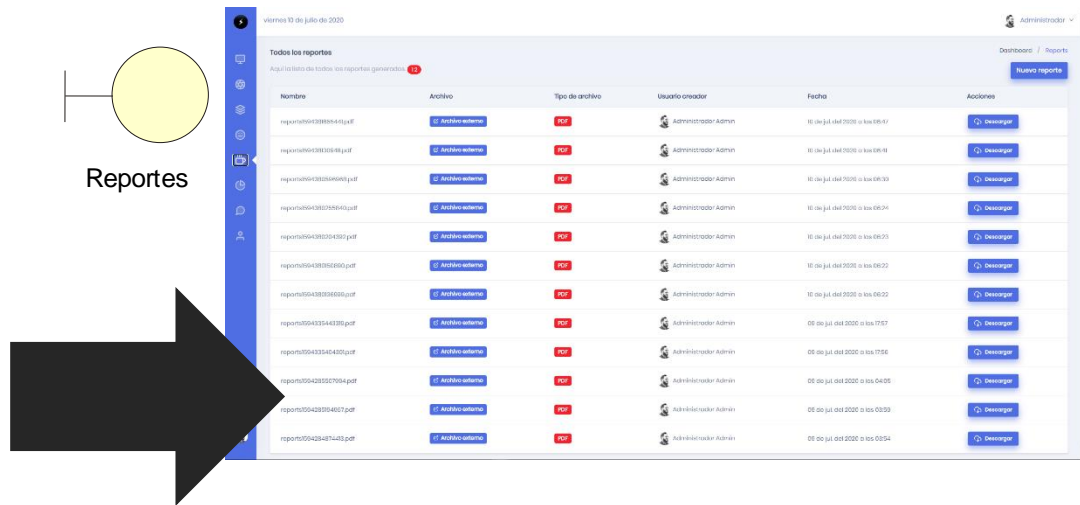


Figura 58 Interfaz Reportes

Explicación: El usuario Jefe de Proyecto, podrá visualizar los reportes generados previamente. Así como se muestra en la Figura N° 41.

- **Interfaz 08: Generar Reportes**

Fuente: Elaboración propia

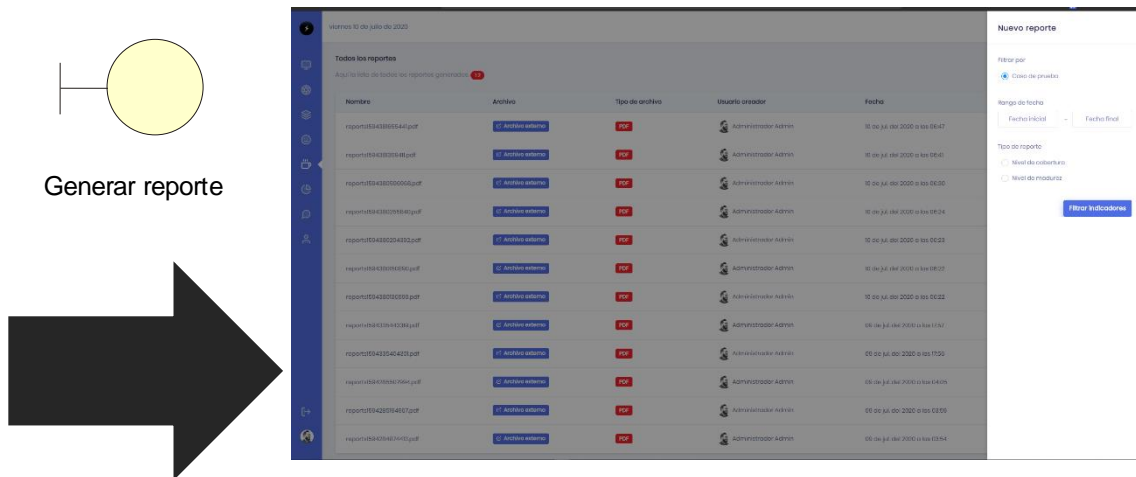


Figura 59 Interfaz Generar reportes

Explicación: El usuario Jefe de Proyecto, podrá generar nuevos reportes. Así como se muestra en la Figura N° 42.

- **Interfaz 09: Versiones**

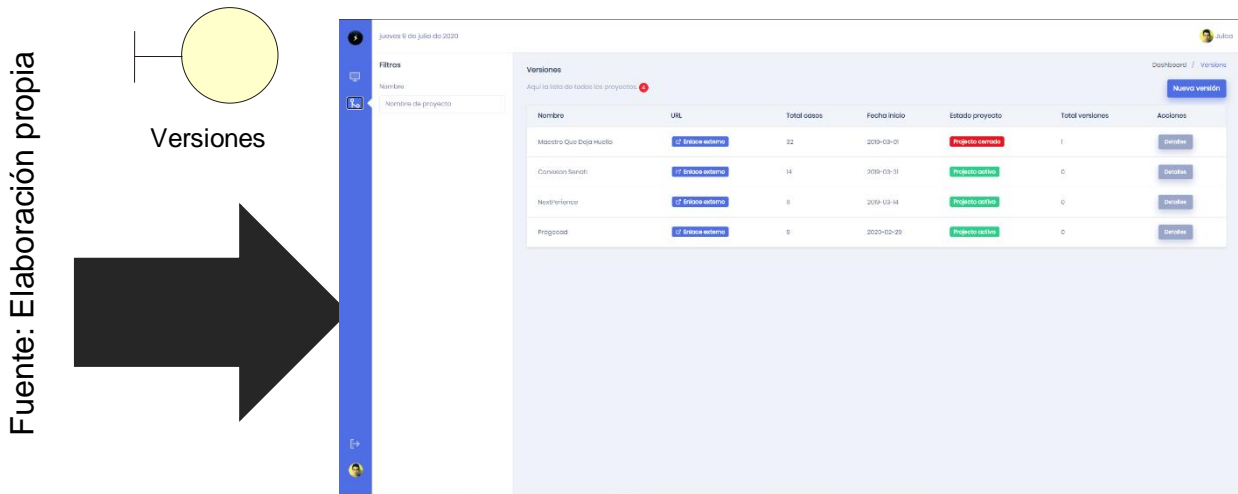


Figura 60 Interfaz Versiones

Explicación: El usuario Developer y Jefe de Proyecto, podrán visualizar versiones por proyecto. Así como se muestra en la Figura N° 43.

- **Interfaz 10: Registrar Versiones**

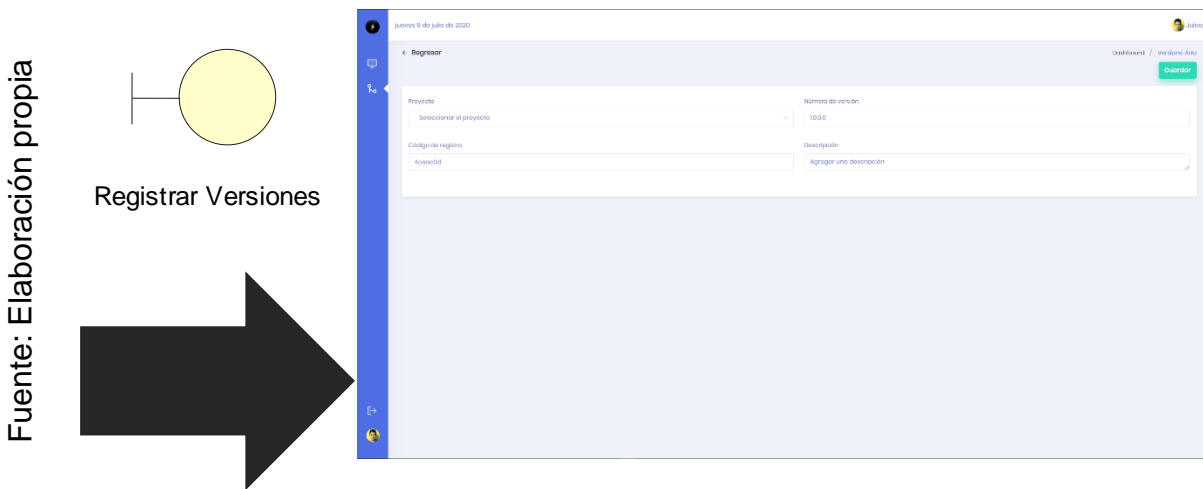


Figura 61 Interfaz Registrar Versiones

Explicación: El usuario Developer, podrá agregar nuevas versiones por proyecto. Así como se muestra en la Figura N° 44.

- **Interfaz 11: Errores**

Fuente: Elaboración propia

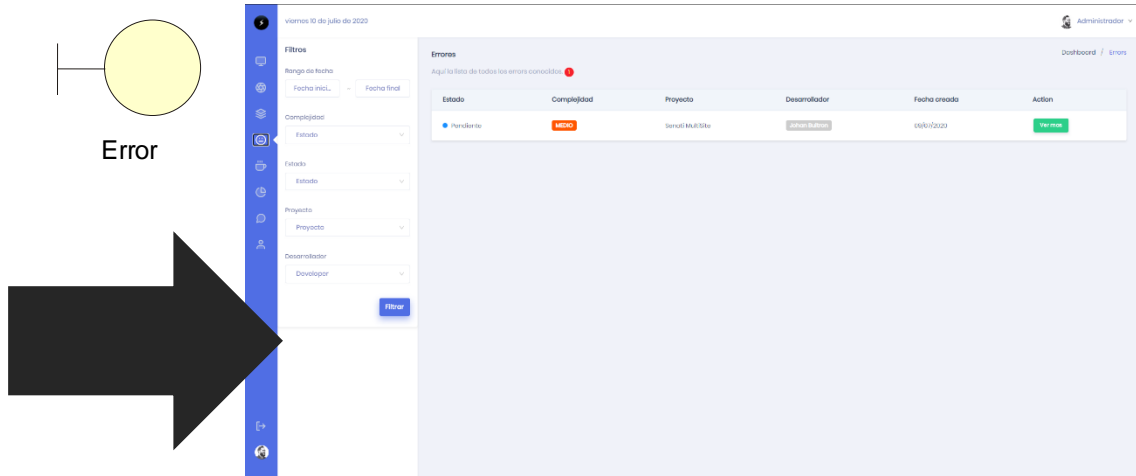


Figura 62 Interfaz Errores

Explicación: El usuario Tester y Jefe de Proyecto, podrán visualizar los errores por proyecto encontrados al momento de resolver el caso de prueba. Así como se muestra en la Figura N° 45.

- **Interfaz 12: Registrar Errores**

Fuente: Elaboración propia

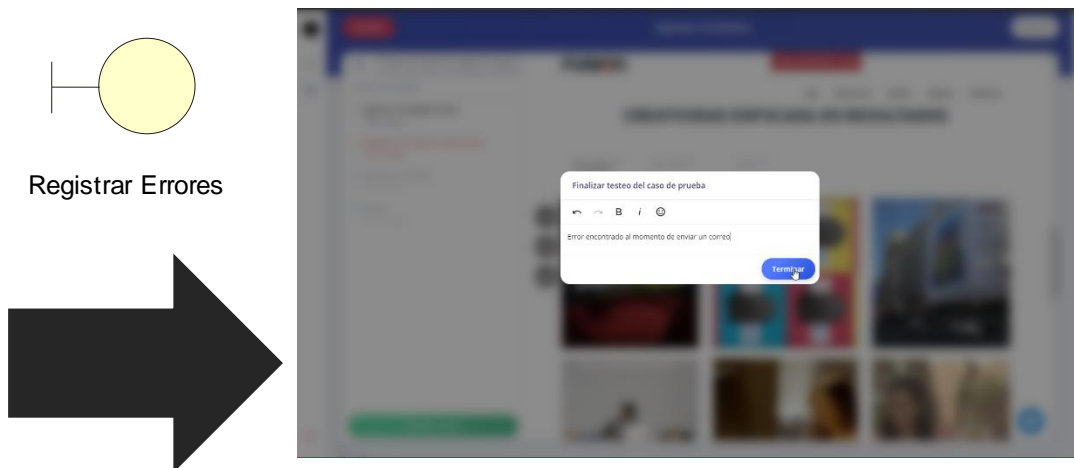


Figura 63 Registrar Errores

Explicación: El usuario Developer, podrá agregar nuevos errores por proyecto. Así como se muestra en la Figura N° 46.

- **Interfaz 13: Casos de prueba asignadas**

Fuente: Elaboración propia

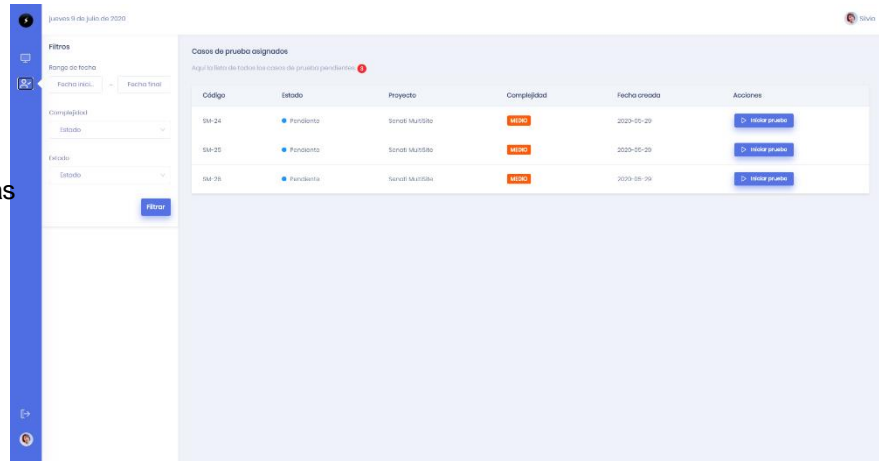
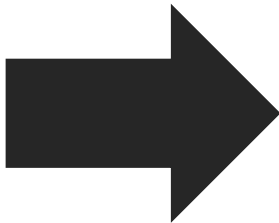
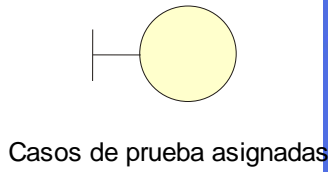


Figura 64 Interfaz Casos de prueba asignadas

Explicación: El usuario Tester, podrá visualizar los casos de prueba asignadas para su revisión. Así como se muestra en la Figura N° 47.

- **Interfaz 14: Resolver Casos de pruebas**

Fuente: Elaboración propia

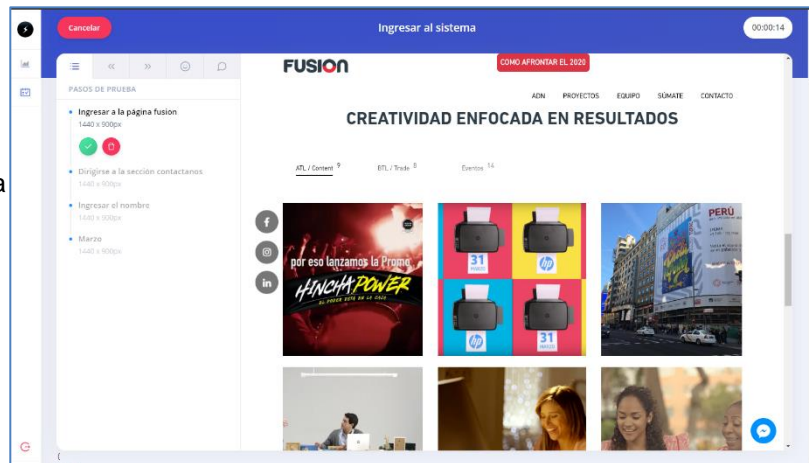
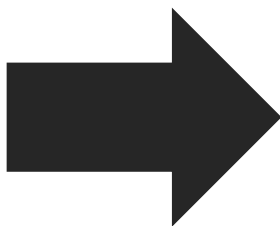
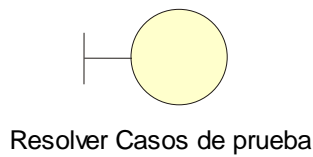


Figura 65 Interfaz Resolver Casos de pruebas

Explicación: El usuario Tester, podrá resolver o revisar los casos de prueba. Así como se muestra en la Figura N° 48.

- **Interfaz 15: Indicadores**

Fuente: Elaboración propia

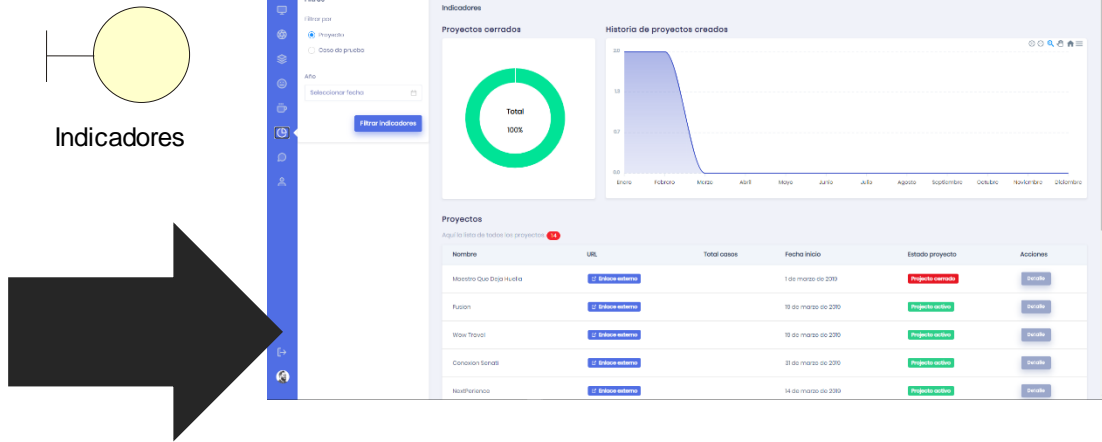


Figura 66 Interfaz Indicadores

Explicación: El usuario Jefe de Proyecto y Administrador, podrá visualizar indicadores por casos de prueba, proyecto, errores y versiones del sistema. Así como se muestra en la Figura N° 49.

- **Interfaz 16: Usuarios**

Fuente: Elaboración propia

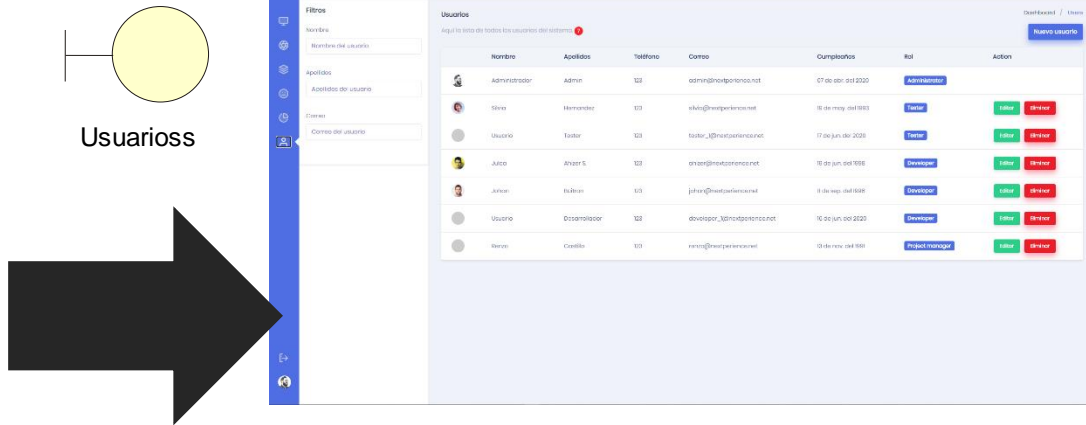


Figura 67 Interfaz Usuarios

Explicación: El usuario Jefe de Proyecto, podrá ver todos los usuarios activos del sistema. Así como se muestra en la Figura N° 50.

- **Interfaz 17: Agregar Usuario**

Fuente: Elaboración propia

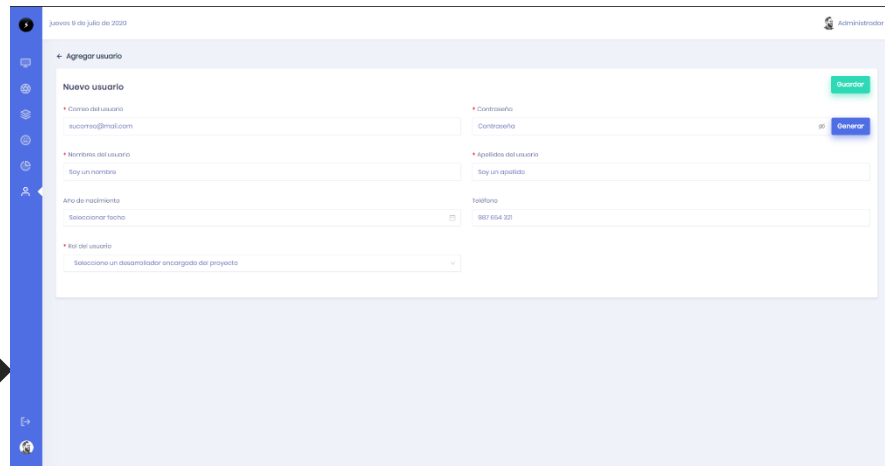
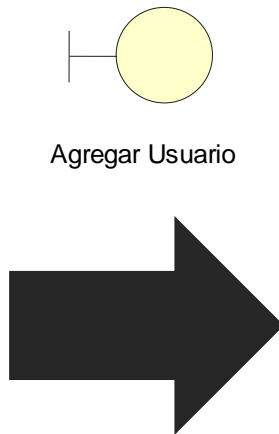


Figura 68 Interfaz Agregar Usuario

Explicación: El usuario Jefe de Proyecto, podrá agregar nuevos usuarios al sistema. Así como se muestra en la Figura N° 51.

- **Interfaz 18: Perfil**

Fuente: Elaboración propia

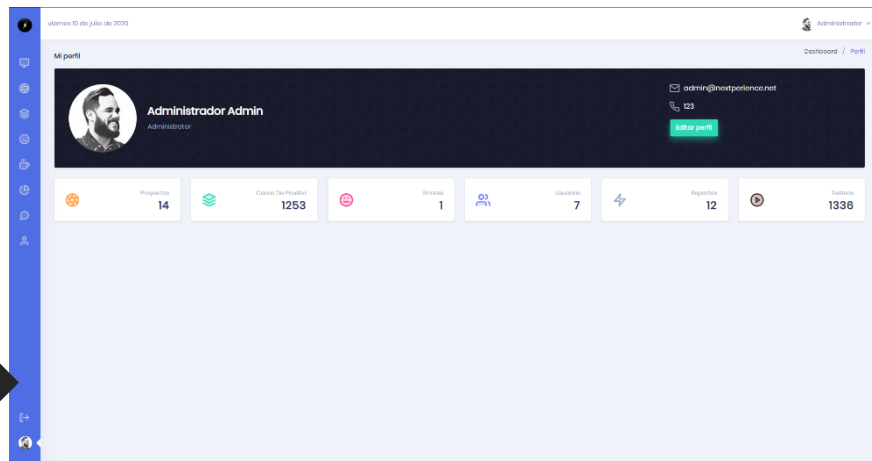
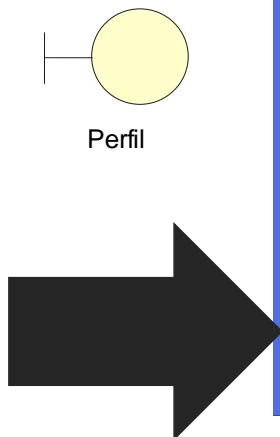


Figura 69 Interfaz Perfil

Explicación: El usuario Jefe de Proyecto, Tester, Developer, podrá información de su perfil. Así como se muestra en la Figura N° 52.

L. Lista de Modelos (Model)

La lista de modelos se encarga de inicializar los datos para el renderizado de la vista. Estos elementos guardan datos que son reactivos a la hora de cambiar el estado de alguna de ellas

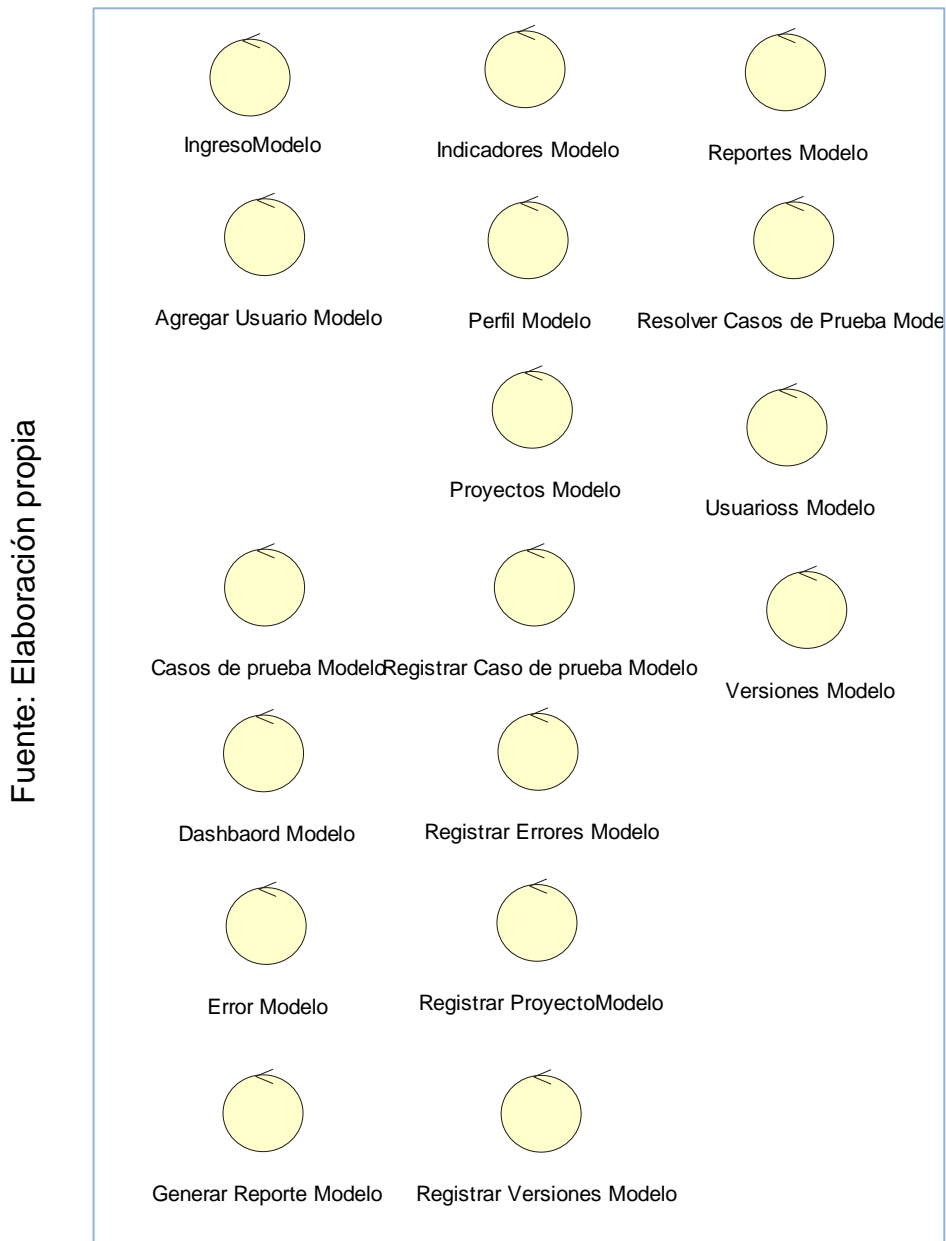


Figura 70 Lista de Modelos

Explicación: La figura 53 se muestra cada una de los modelos correspondiente en cada interfaz, son elementos JSON encaradas de guardar datos iniciales de las vistas.

M. Lista de Vista-Modelo (ViewModel)

El listado de vista-modelo, contiene los métodos encargados de la comunicación entre el modelo y la vista, así como la comunicación del sistema con el api.

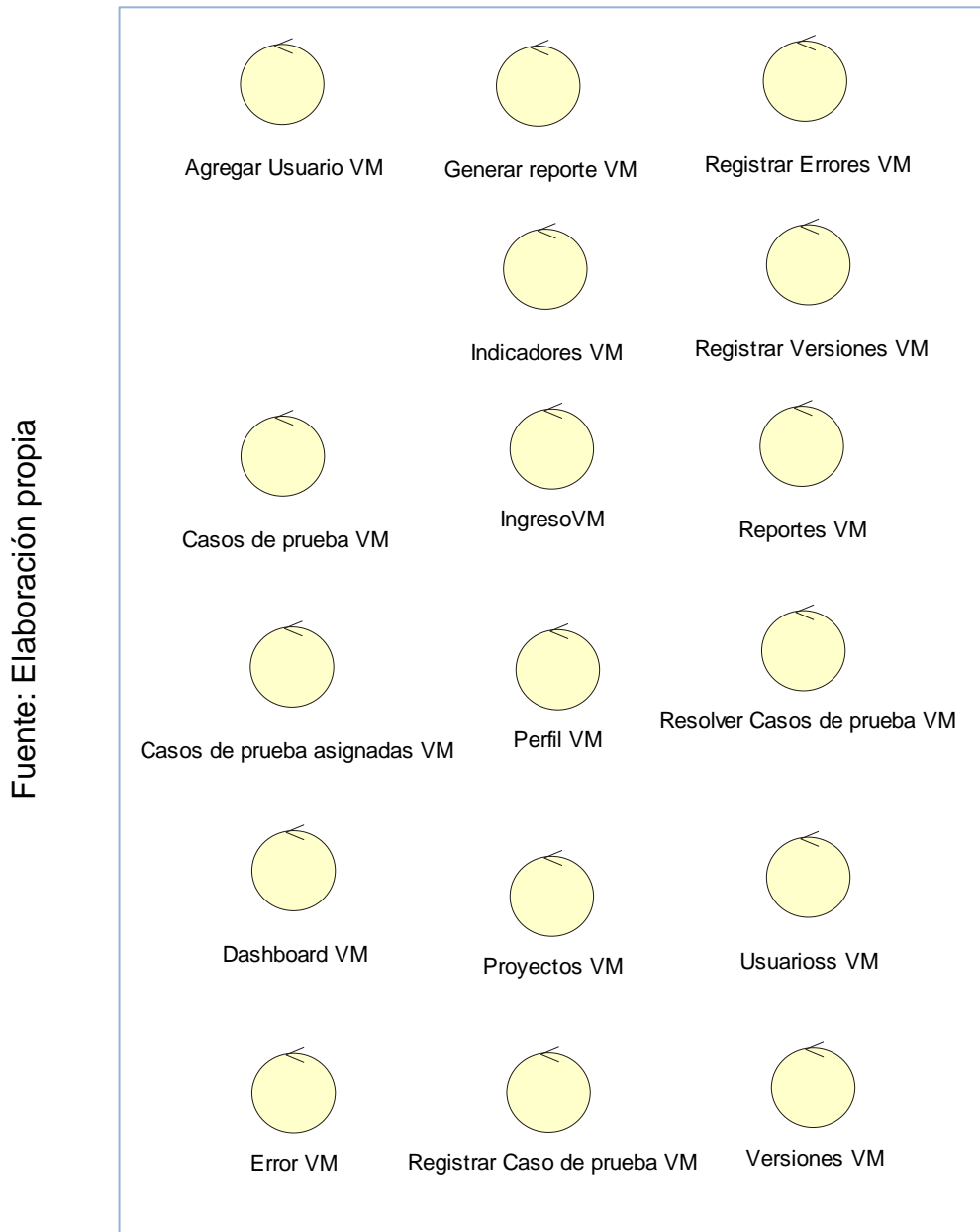


Figura 71 Lista de Vista-Modelo

Explicación: La Figura 54 se muestra cada una de los visa-modelo, encargados de la comunicación entre la interfaz y los modelos, así como la comunicación de los Apis con la interfaz.

BACKEND

N. Lista de Routes

La lista de routes son las rutas disponibles para enviar datos y conectarlos con los controladores y modelos del sistema backend.

Fuente: Elaboración propia



Figura 72 Lista de Routes

Explicación: La Figura 55, se muestra las rutas se encargarán de recibir información para luego procesarlo en los controladores y según el método HTTP esta consultara, actualizaría eliminara o creara datos según el método.

O. Lista de Controladores

La lista de controladores son las encargadas de la conexión de la base de datos con las rutas, conteniendo operación y dependencias que facilitan la programación.

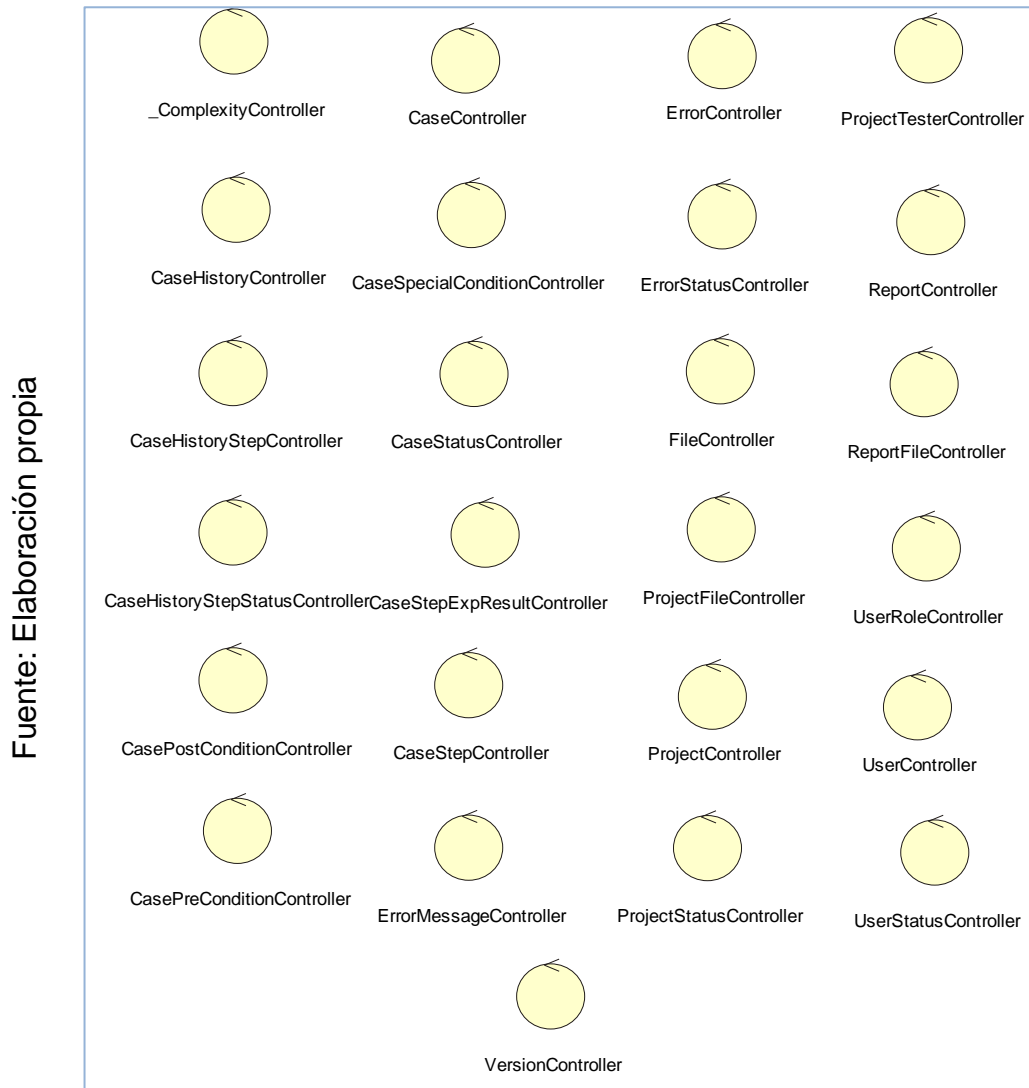


Figura 73 Lista de Controladores

Explicación: La Figura 56 muestra la lista de controladores de cada una de las operaciones existentes, son elementos que determinan la conexión entre las rutas y las entidades.

P. Lista de Entidades

La lista de entidades del diagrama de clases de análisis es aquellas que permiten el desarrollo del modelo lógico y físico de la base de datos.

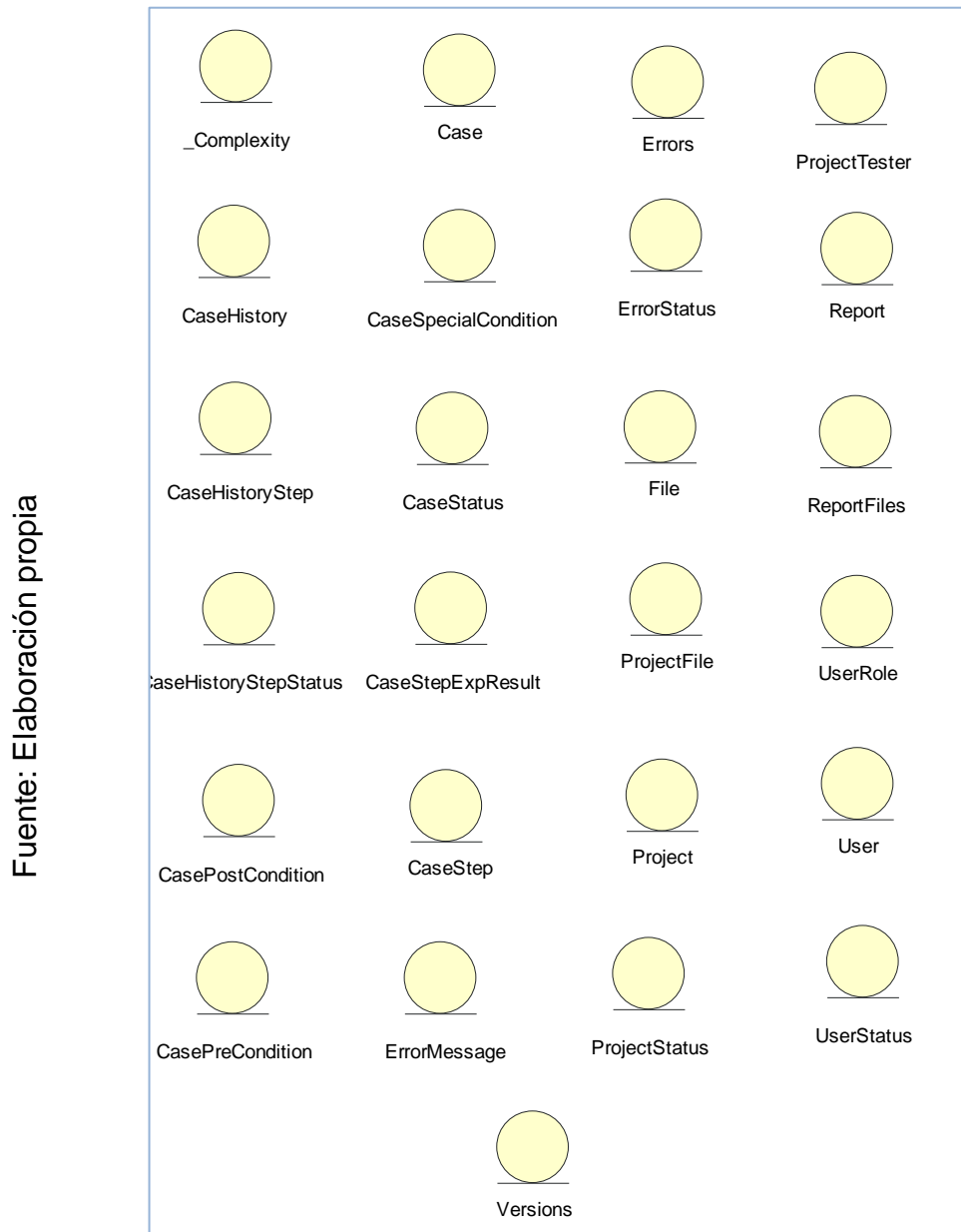


Figura 74 Lista de Entidades

Explicación: : La Figura 57, se muestra la lista de entidades encontradas en los diagramas de clases de análisis. Estas entidades, junto con el modelo conceptual, permitirán desarrollar el modelo lógico y físico de la base de datos.

Q. Diagramas de secuencia de casos de uso del sistema

Para cada caso de uso de sistema identificado se hace su respectivo diagrama de secuencia de sistema.

En la siguiente figura se visualiza el diagrama de secuencia del caso de uso “Autenticar Usuario”.

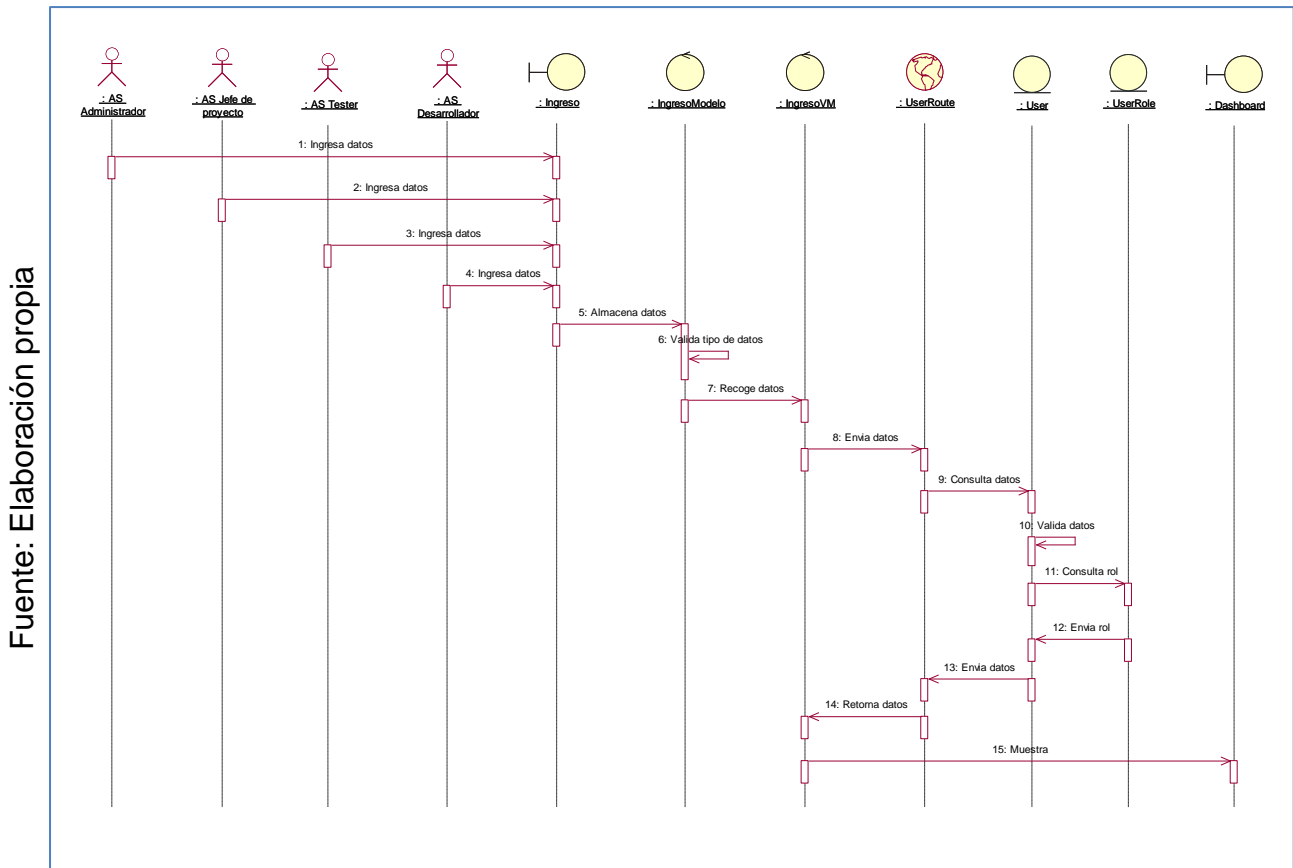


Figura 75 Diagrama de Secuencia del caso de uso del Sistema “Autenticar Usuario”

Explicación: La figura 58 se muestra la secuencia de actividades en la interacción del usuario con las interfaces, modelos, vista-modelo, controladores y entidades del caso de uso “Autenticar Usuario”.

En la siguiente figura se visualiza el diagrama de secuencia del caso de uso "Registrar proyecto".

Fuente: Elaboración propia

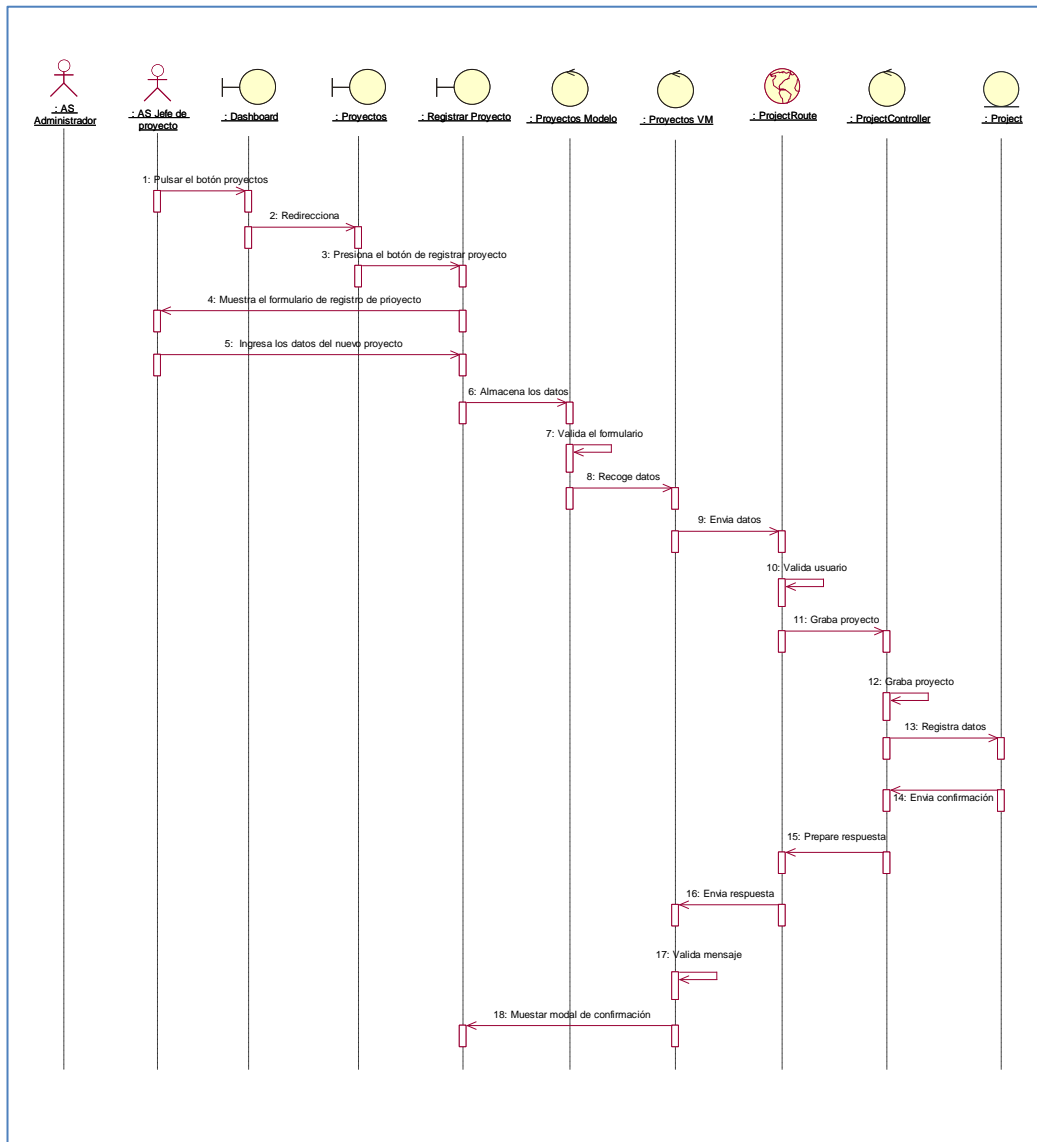


Figura 76 Diagrama de Secuencia del caso de uso del Sistema "Registrar proyecto"

Explicación: La figura 59 se muestra la secuencia de actividades en la interacción del usuario con las interfaces, modelos, vista-modelo, controladores y entidades del caso de uso "Registrar proyecto".

En la siguiente figura se visualiza el diagrama de secuencia del caso de uso “Registrar Casos de Pruebas”.

Fuente: Elaboración propia

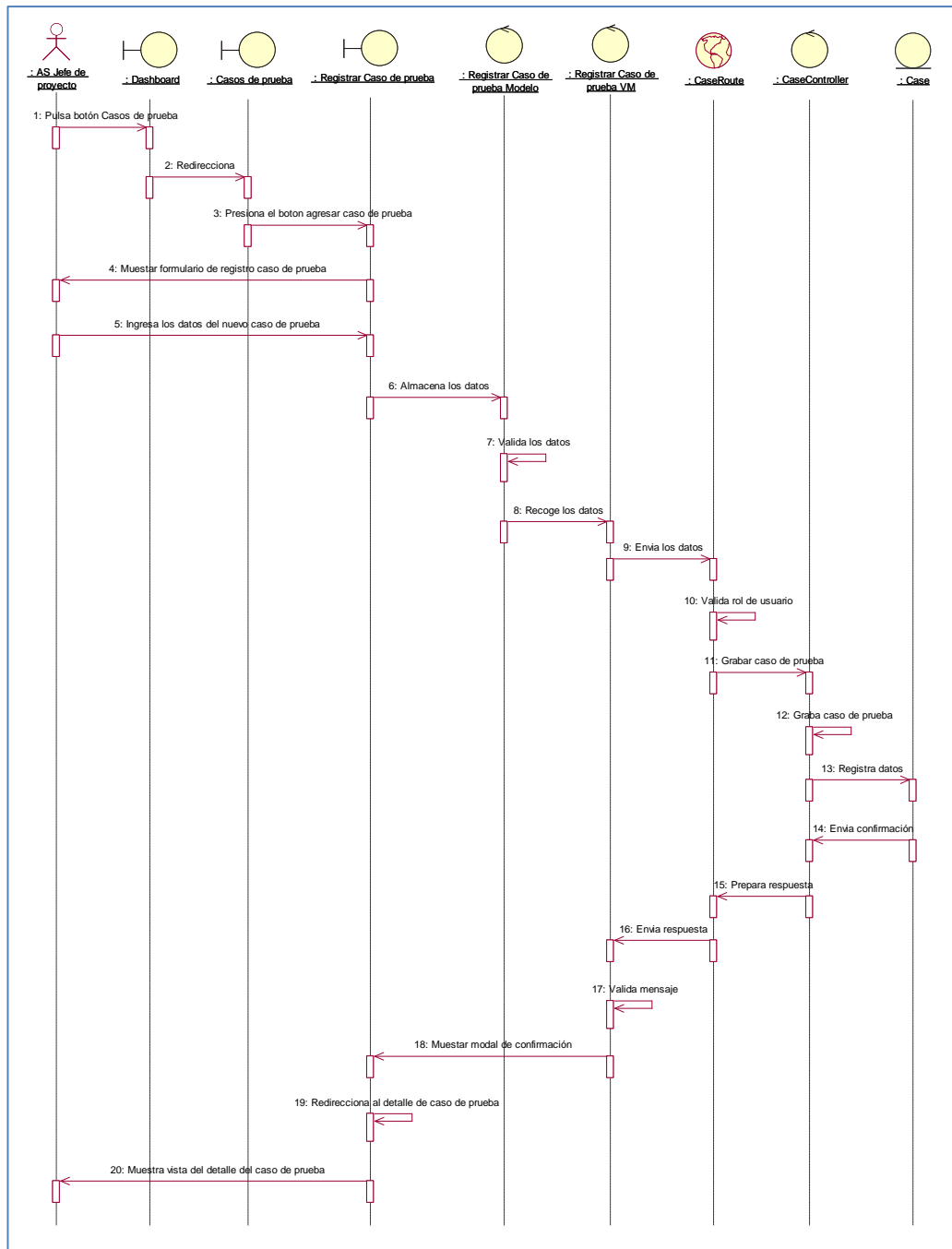


Figura 77 Diagrama de Secuencia del caso de uso del Sistema "Registrar Casos de pruebas"

Explicación: La figura 60 se muestra la secuencia de actividades en la interacción del usuario con las interfaces, modelos, vista-modelo, controladores y entidades del caso de uso “Registrar Casos de pruebas”.

En la siguiente figura se visualiza el diagrama de secuencia del caso de uso "Resolver Casos de prueba".

Fuente: Elaboración propia

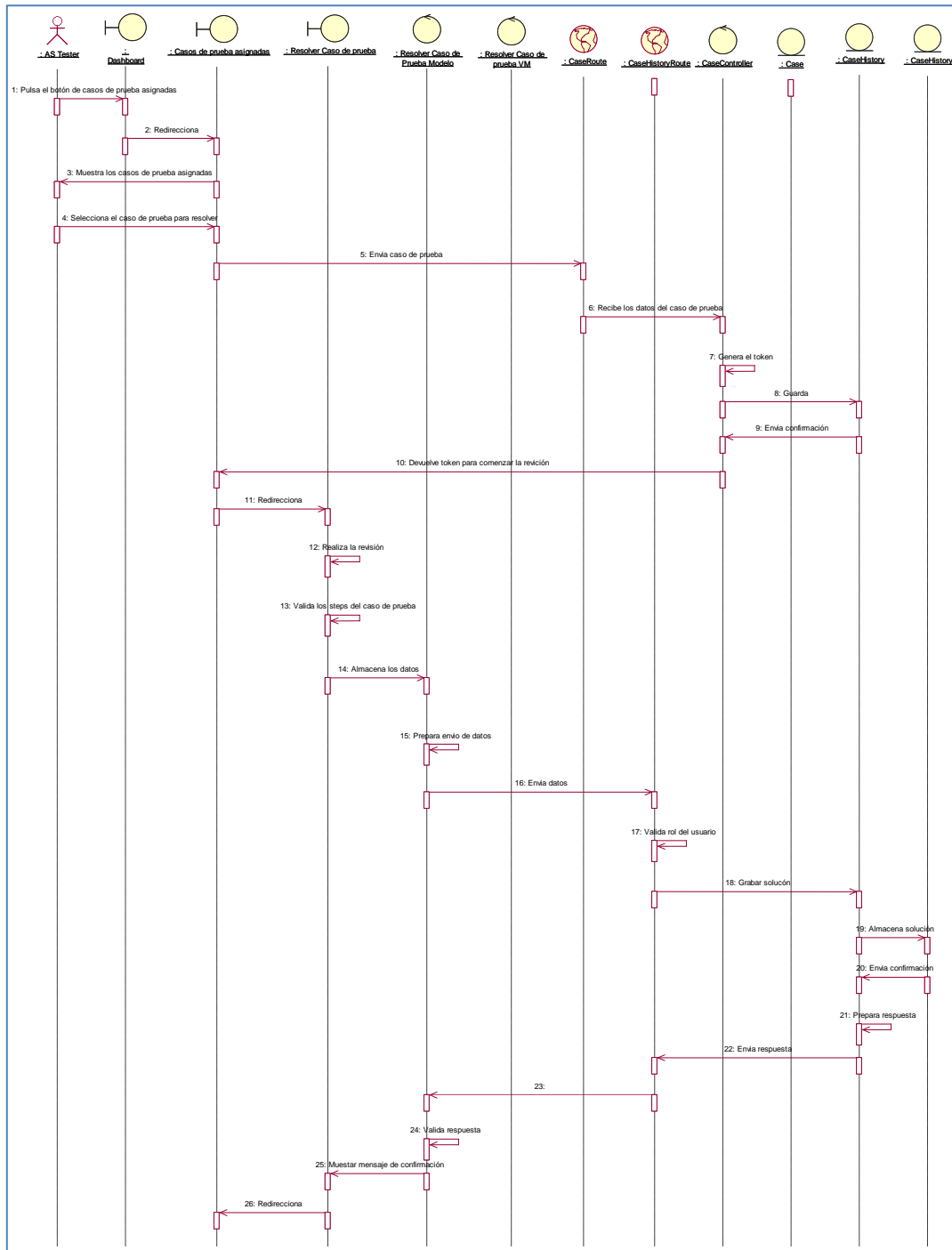


Figura 78 Diagrama de Secuencia del caso de uso del Sistema "Resolver casos de prueba"

Explicación: La figura 61 se muestra la secuencia de actividades en la interacción del usuario con las interfaces, modelos, vista-modelo, controladores y entidades del caso de uso "Resolver casos de prueba".

En la siguiente figura se visualiza el diagrama de secuencia del caso de uso "Cerrar proyecto".

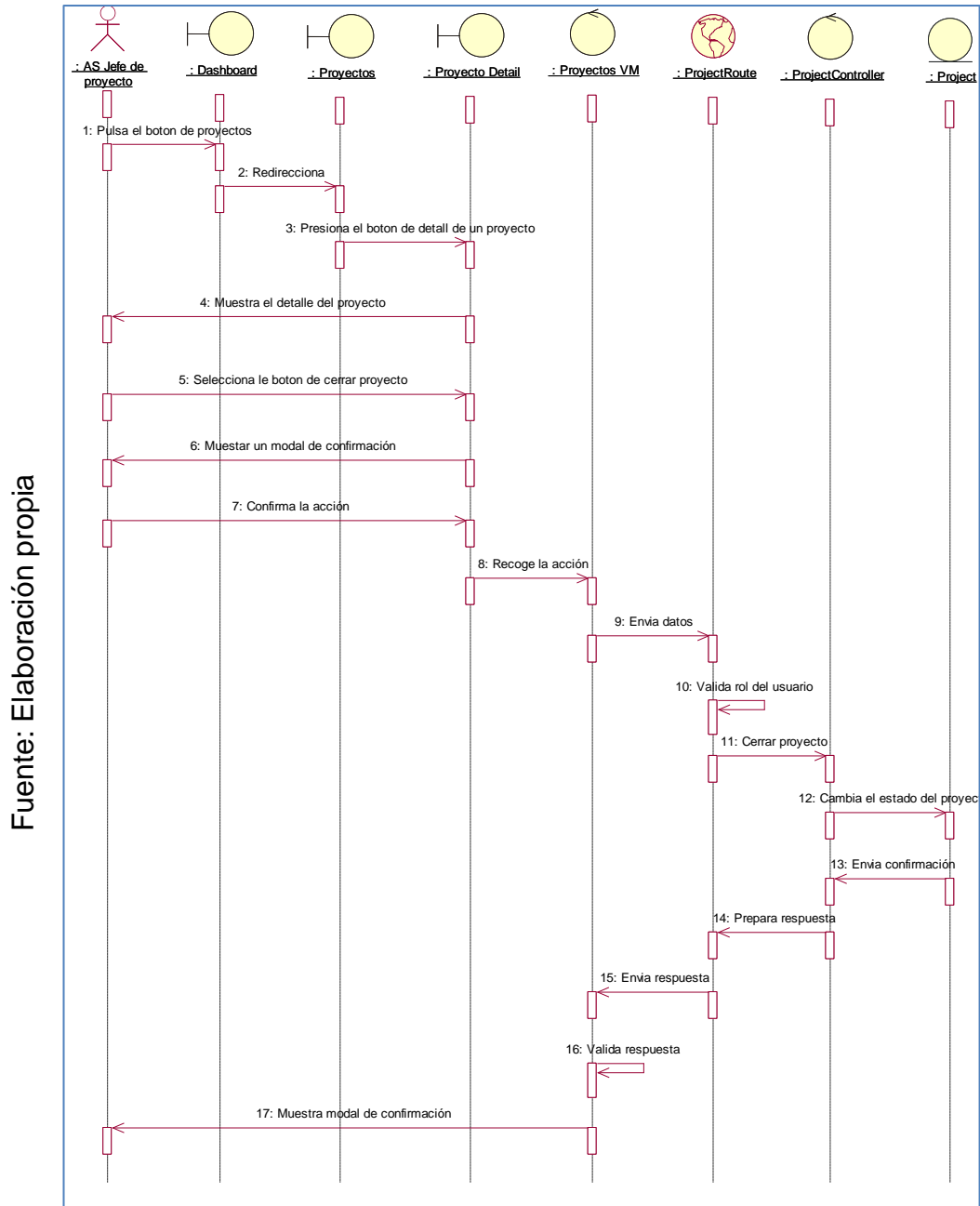


Figura 79 Diagrama de Secuencia del caso de uso del Sistema "Cerrar proyecto"

Explicación: La figura 62 Se muestra la secuencia de actividades en la interacción del usuario con las interfaces, modelos, vista-modelo, controladores y entidades del caso de uso "Cerrar proyecto".

En la siguiente figura se visualiza el diagrama de secuencia del caso de uso "Registrar error".

Fuente: Elaboración propia

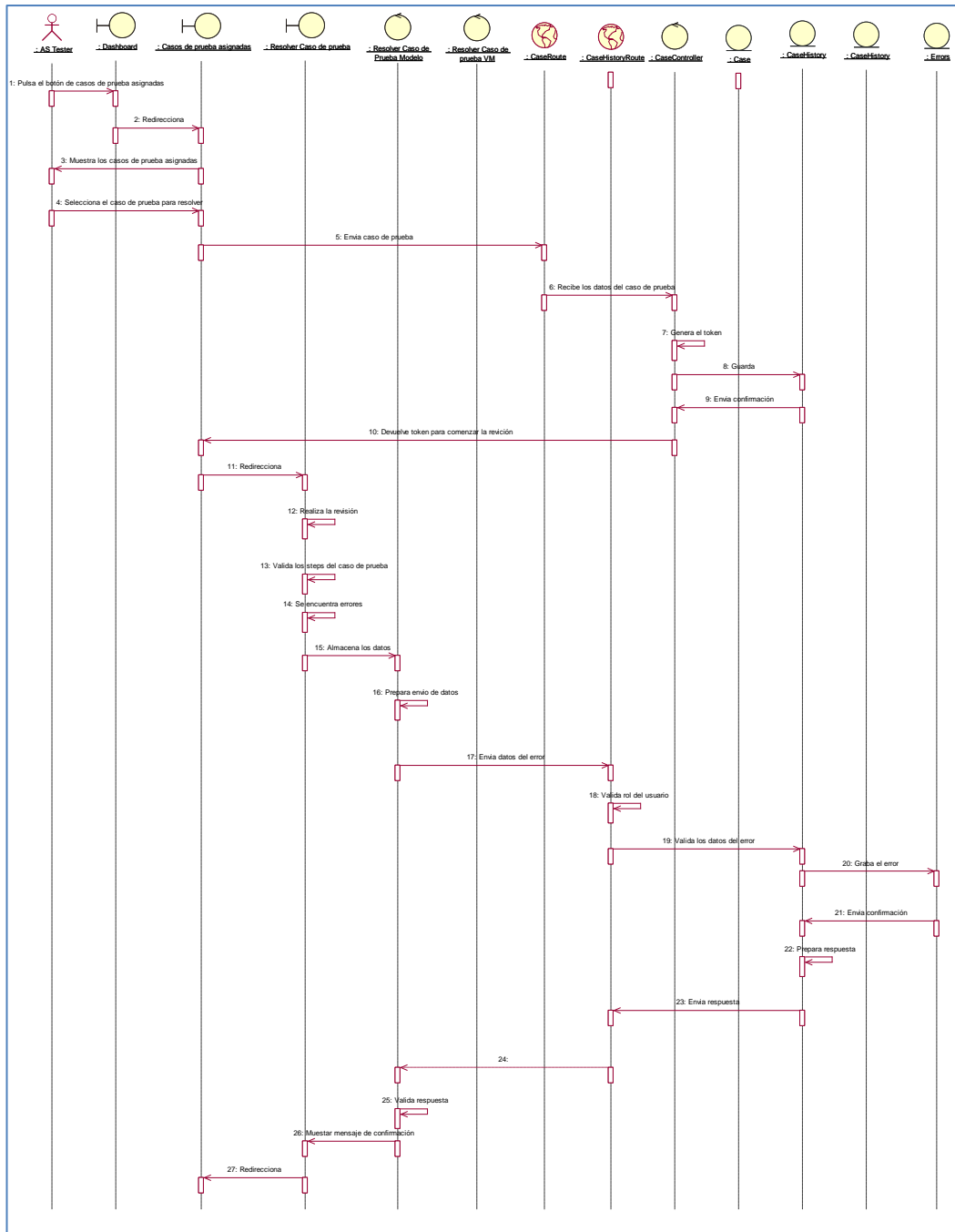


Figura 80 Diagrama de Secuencia del caso de uso del Sistema "Registrar errores"

Explicación: La figura 63 Se muestra la secuencia de actividades en la interacción del usuario con las interfaces, modelos, vista-modelo, controladores y entidades del caso de uso "Registrar errores".

En la siguiente figura se visualiza el diagrama de secuencia del caso de uso "Registrar versiones".

Fuente: Elaboración propia

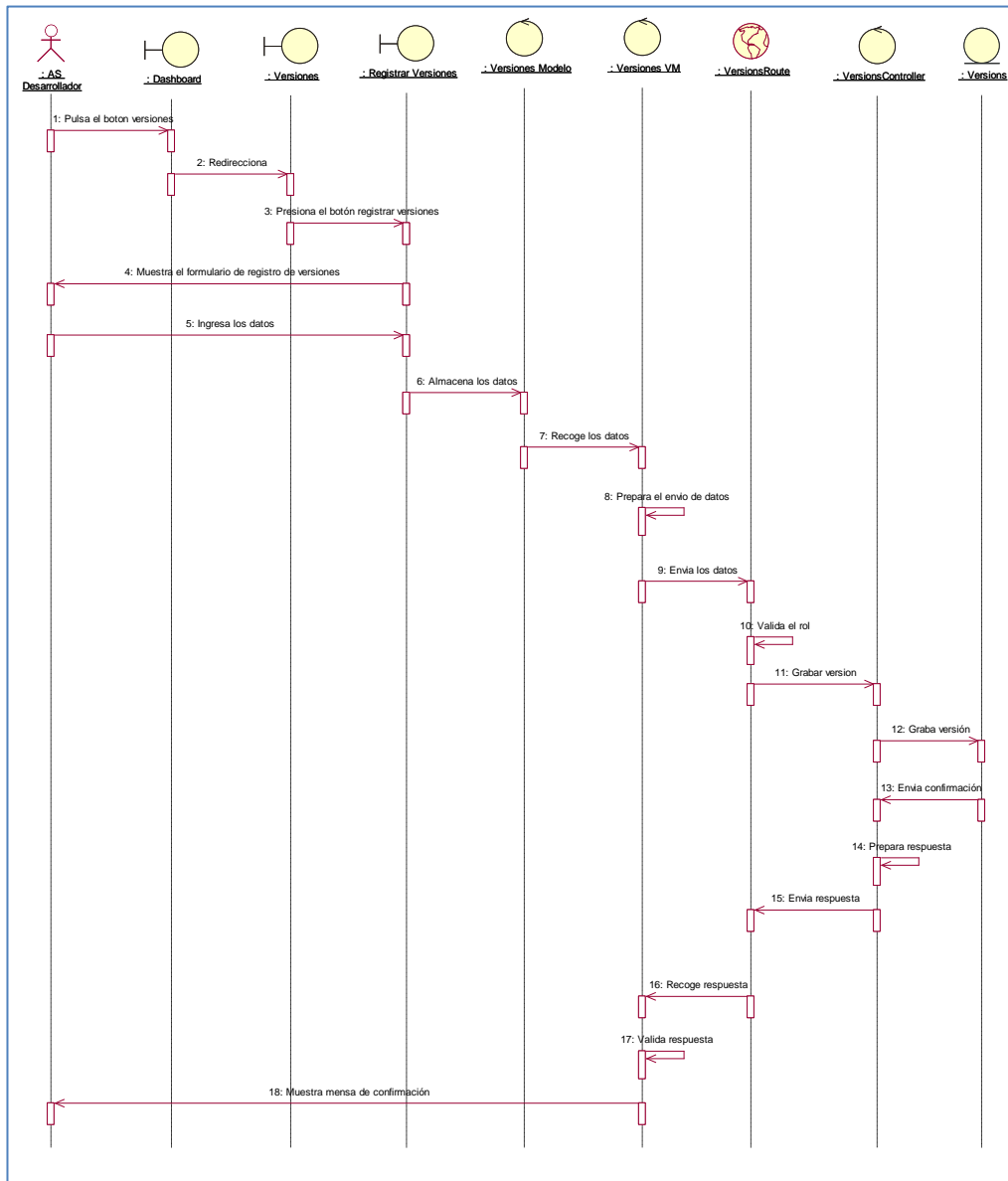


Figura 81 Diagrama de Secuencia del caso de uso del Sistema "Registrar versión"

Explicación: La figura 64 se muestra la secuencia de actividades en la interacción del usuario con las interfaces, modelos, vista-modelo, controladores y entidades del caso de uso "Registrar versión".

R. Diagrama de Colaboración de Casos de Uso del Sistema

Para cada caso de uso de sistema identificado se hace su respectivo diagrama de colaboración de sistema.

En la siguiente figura se visualiza el diagrama de colaboración del caso de uso de sistema "Autenticar Usuario".

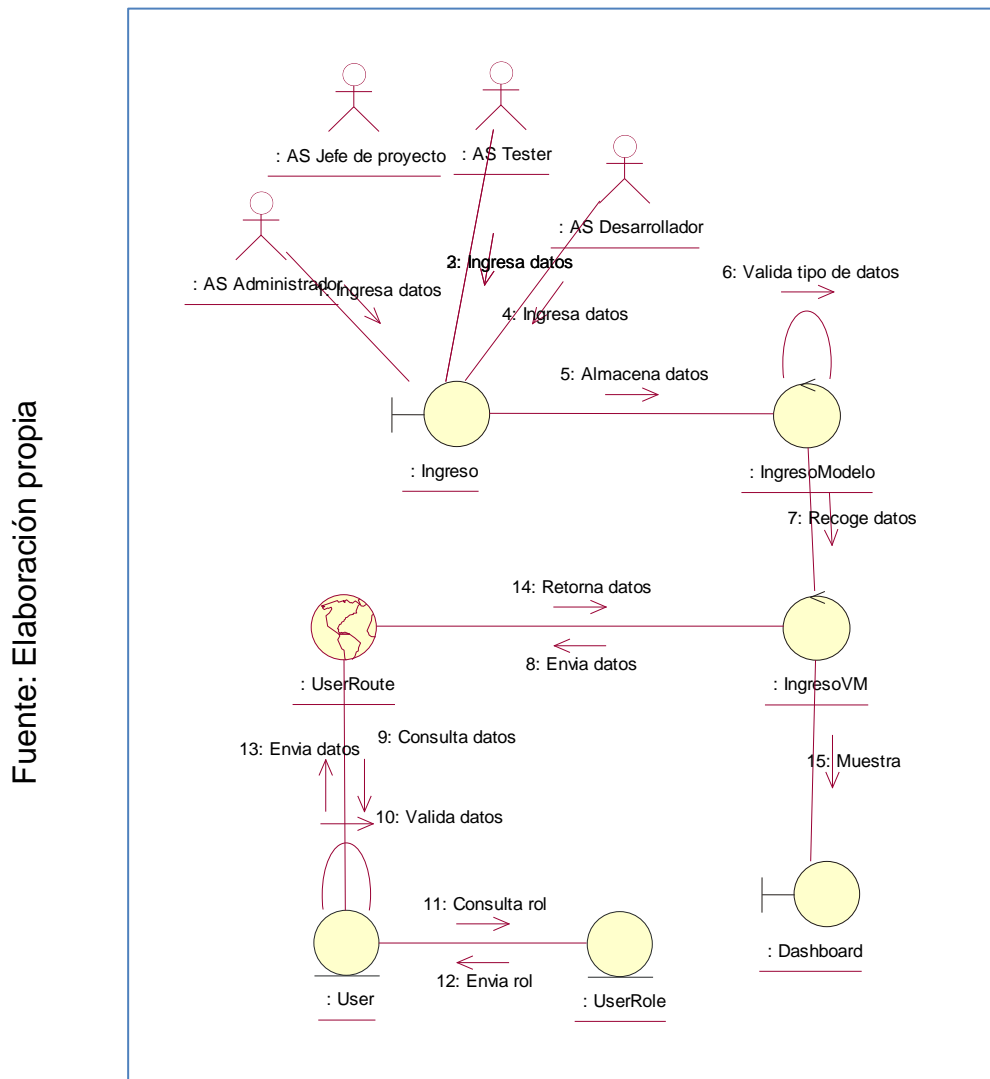


Figura 82 Diagrama de Colaboración del caso de uso de sistema "Autenticar usuario"

Explicación: La figura 65 se muestra el diagrama de colaboración donde se observa la interacción del usuario con las vistas, vista-modelo, rutas, controladores y entidades del caso de uso "Autenticar Usuario".

En la siguiente figura se visualiza el diagrama de colaboración del caso de uso de sistema “Registrar proyecto”.

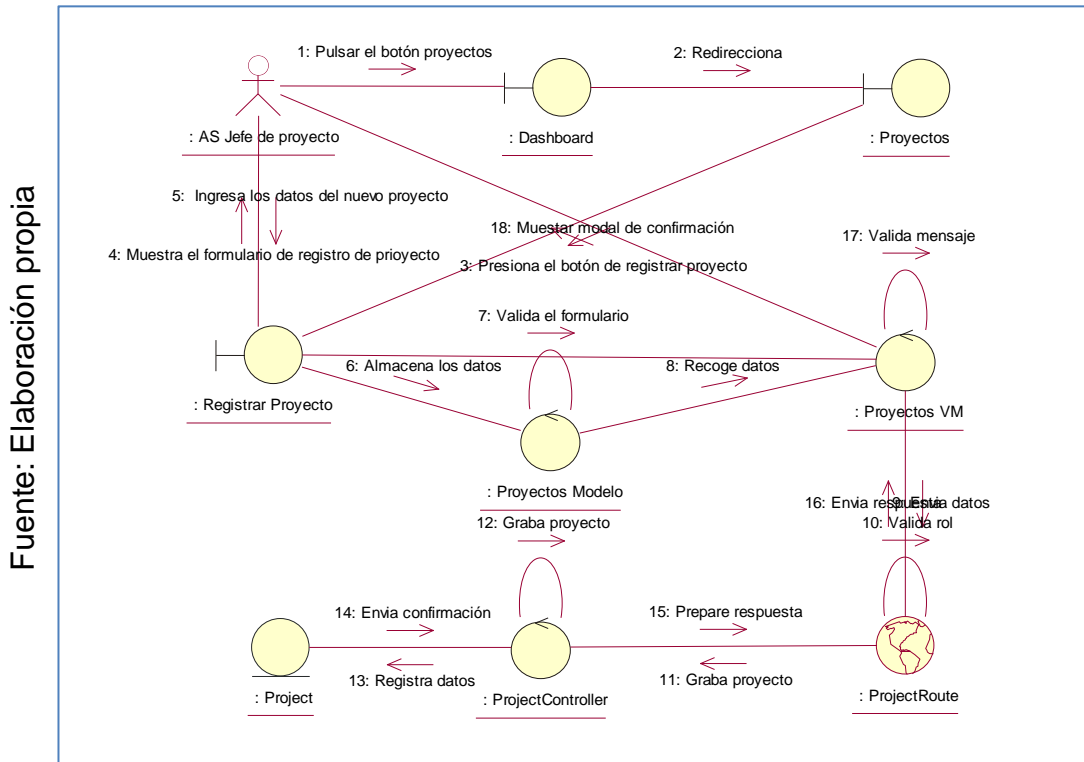


Figura 83 Diagrama de Colaboración del caso de uso de sistema "Registrar proyecto"

Explicación: La figura 66 se muestra el diagrama de colaboración donde se observa la interacción del usuario con las vistas, vista-modelo, rutas, controladores y entidades del caso de uso “Registrar proyecto”.

En la siguiente figura se visualiza el diagrama de colaboración del caso de uso de sistema "Registrar casos de prueba".

Fuente: Elaboración propia

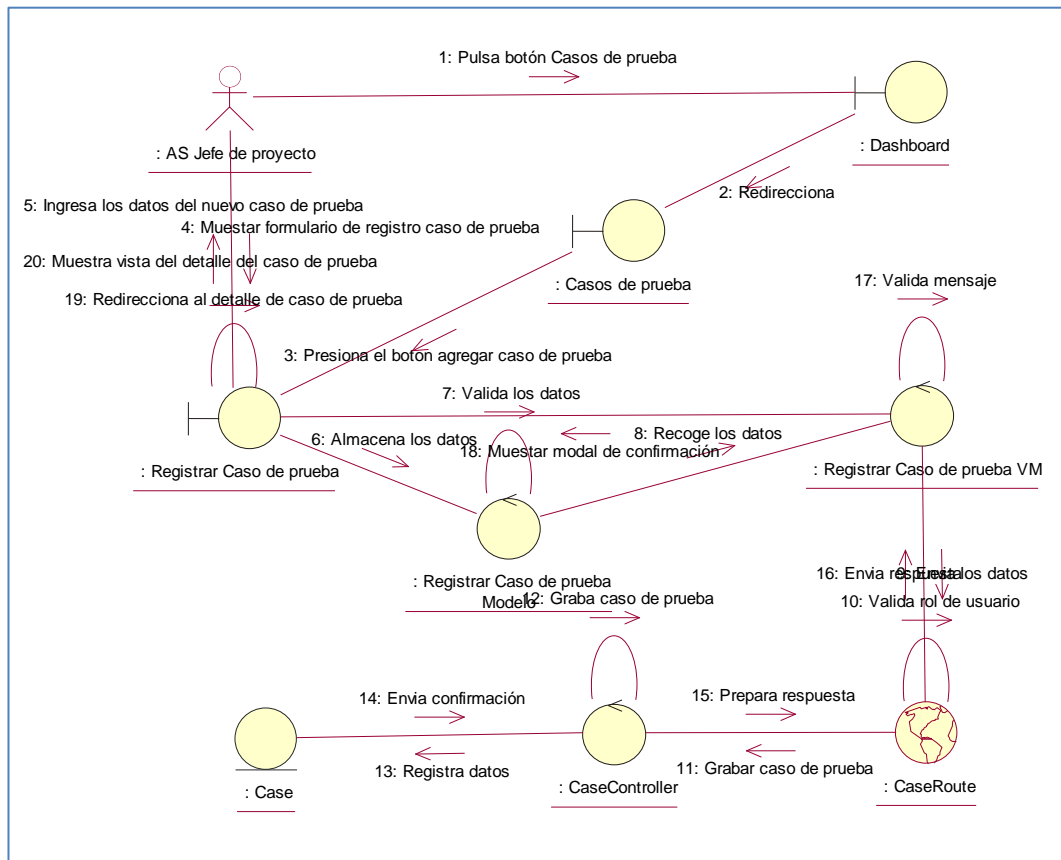


Figura 84 Diagrama de Colaboración del caso de uso de sistema "Registrar casos de prueba"

Explicación: la figura 67 se muestra el diagrama de colaboración donde se observa la interacción del usuario con las vistas, vista-modelo, rutas, controladores y entidades del caso de uso "Registrar casos de prueba".

En la siguiente figura se visualiza el diagrama de colaboración del caso de uso de sistema “Resolver casos de prueba”.

Fuente: Elaboración propia.

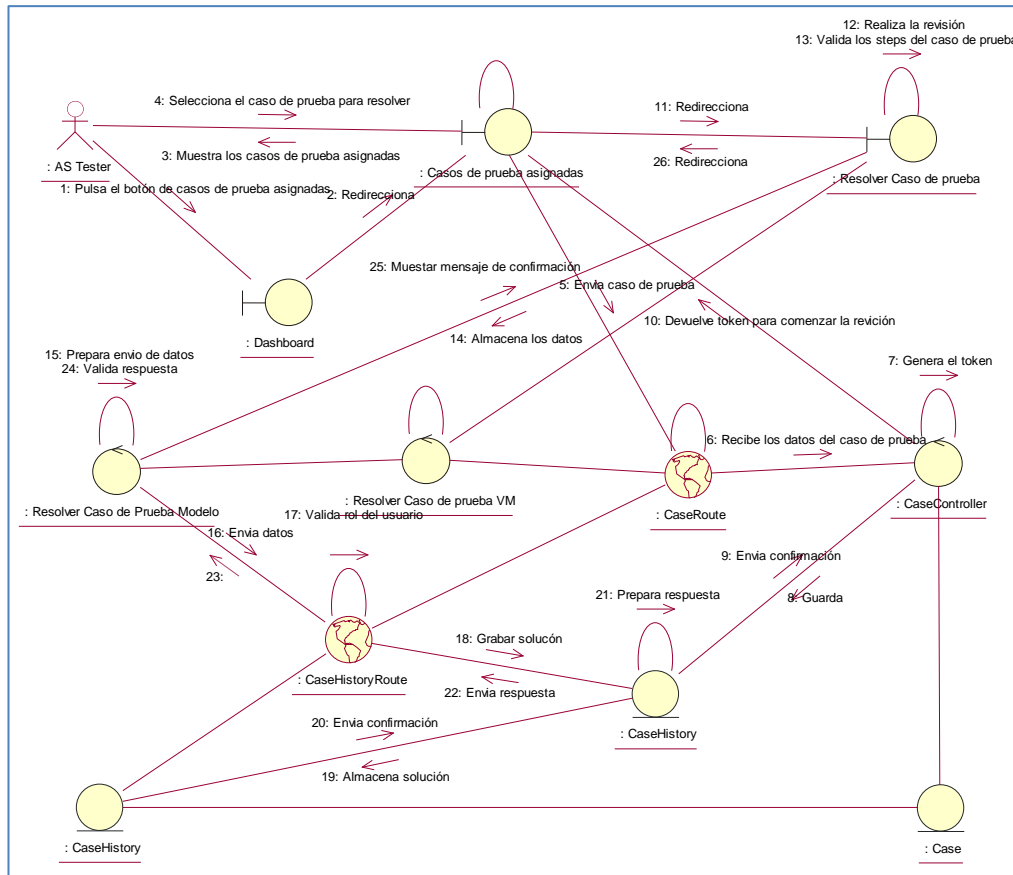


Figura 85 Diagrama de Colaboración del caso de uso de sistema "Resolver caso de prueba"

Explicación: La figura 68 se muestra el diagrama de colaboración donde se observa la interacción del usuario con las vistas, vista-modelo, rutas, controladores y entidades del caso de uso “Resolver caso de prueba”.

En la siguiente figura se visualiza el diagrama de colaboración del caso de uso de sistema "Cerrar proyecto".

Fuente: Elaboración propia

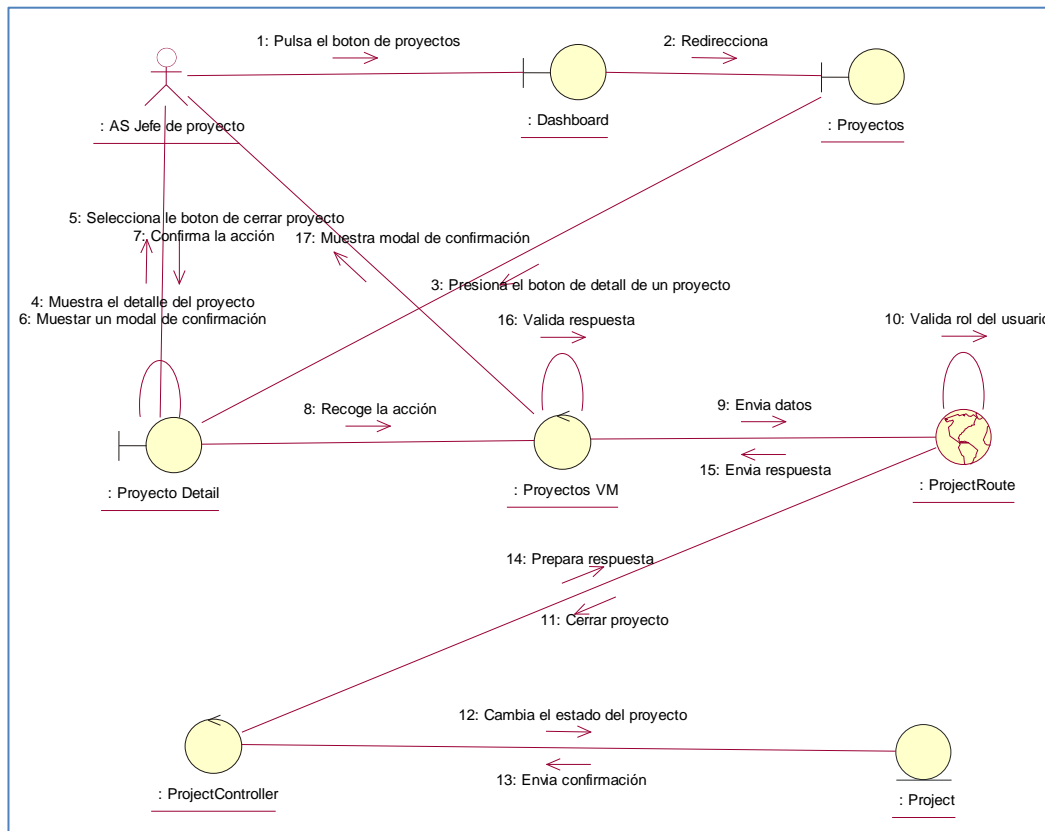


Figura 86 Diagrama de Colaboración del caso de uso de sistema "Cerrar proyecto"

Explicación: La figura 69 se muestra el diagrama de colaboración donde se observa la interacción del usuario con las vistas, vista-modelo, rutas, controladores y entidades del caso de uso "Cerrar proyecto".

En la siguiente figura se visualiza el diagrama de colaboración del caso de uso de sistema "Registrar error".

Fuente: Elaboración propia

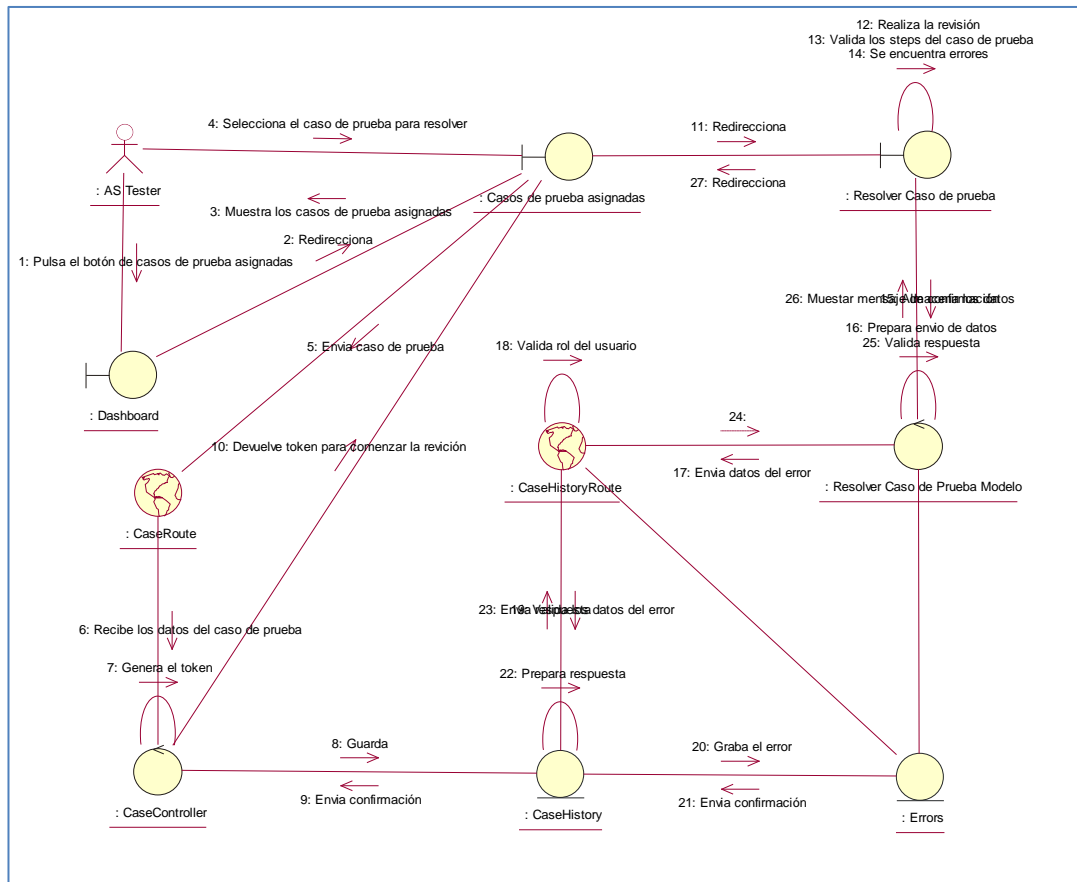


Figura 87 Diagrama de Colaboración del caso de uso de sistema "Registrar error"

Explicación: La figura 70 se muestra el diagrama de colaboración donde se observa la interacción del usuario con las vistas, vista-modelo, rutas, controladores y entidades del caso de uso "Registrar error".

En la siguiente figura se visualiza el diagrama de colaboración del caso de uso de sistema “Registrar versión”.

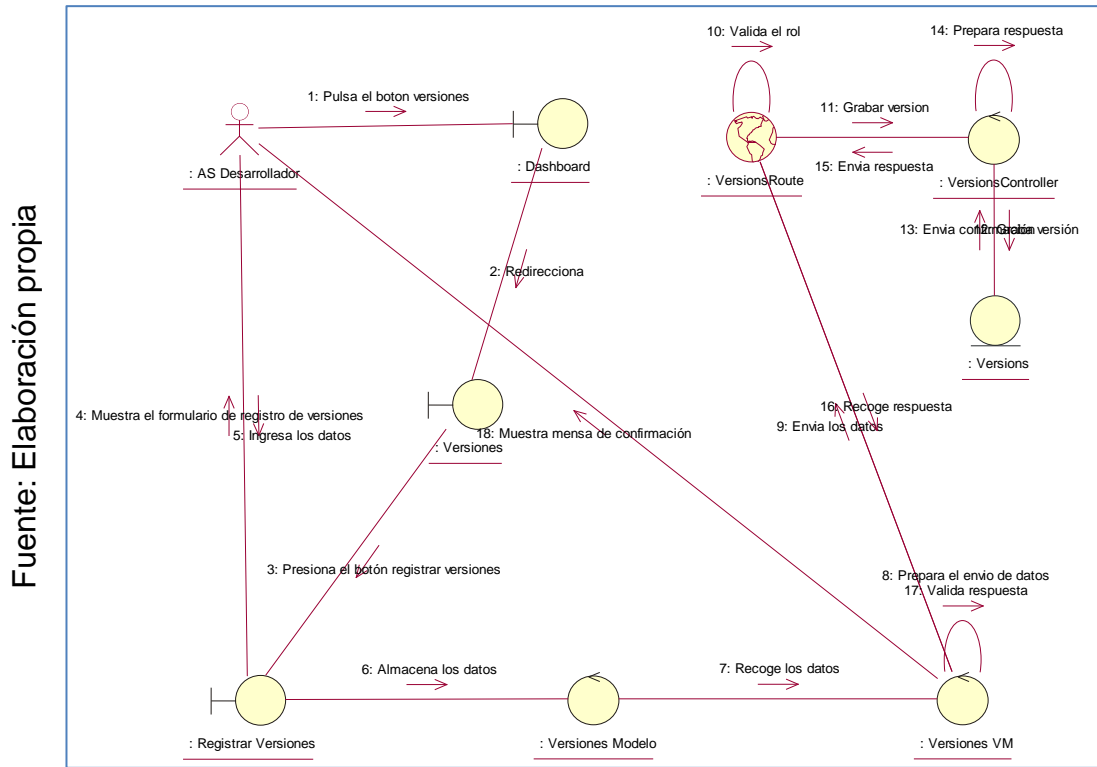


Figura 88 Diagrama de Colaboración del caso de uso de sistema "Registrar versiones"

Explicación: La figura 71 se muestra el diagrama de colaboración donde se observa la interacción del usuario con las vistas, vista-modelo, rutas, controladores y entidades del caso de uso “Registrar versiones”.

S. Diagrama de Actividades de Casos de Uso del Sistema

Para las realizaciones de cada caso de uso de sistema identificado se realiza su respectivo diagrama de actividades del sistema, que muestra el orden en que las clases de análisis ejecutan las acciones para realizar el caso de uso correspondiente.

En la siguiente figura se visualiza el diagrama de actividades de la realización del caso de uso “Autenticar Usuario”.

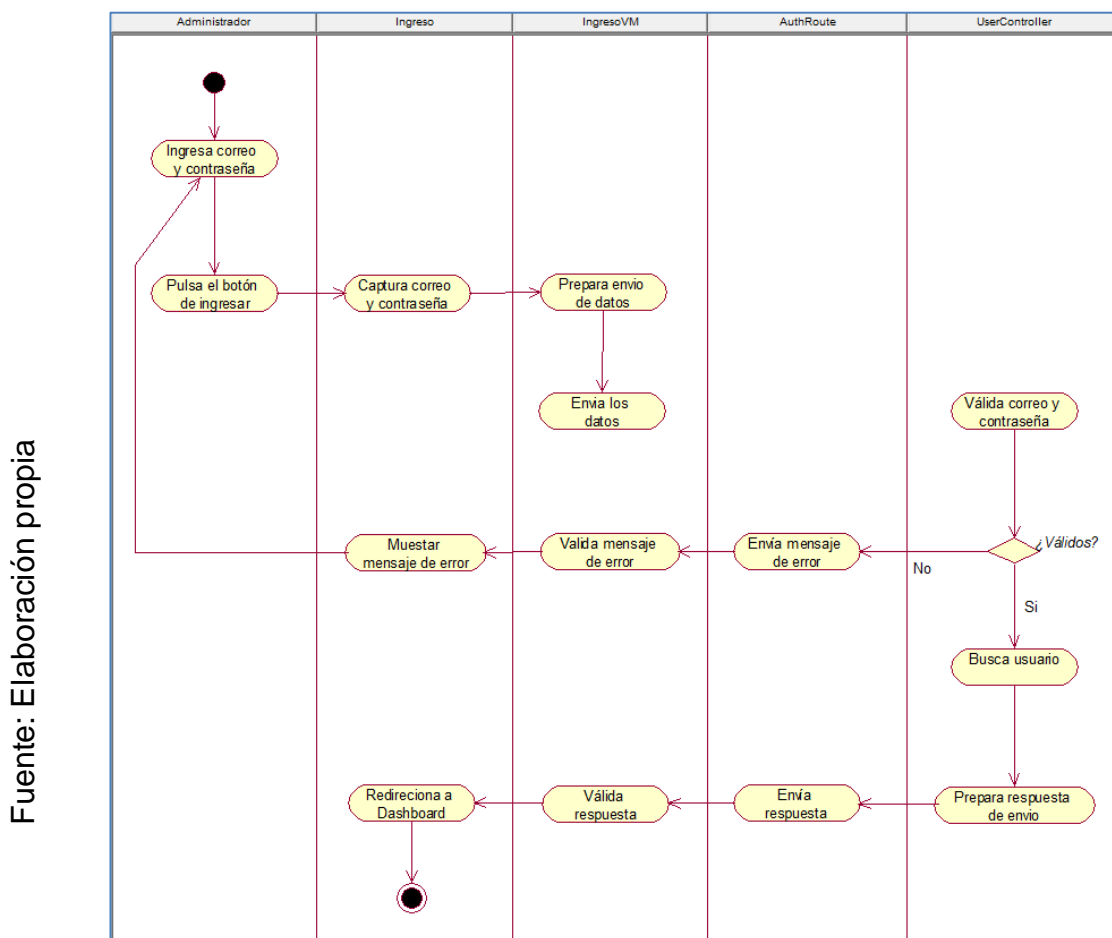


Figura 89 Diagrama de Actividades del caso de uso de sistema “Autenticar Usuario”

Explicación: La figura 72 se muestra el diagrama actividades donde se muestra como el usuario inicia sesión y accede al sistema.

En la siguiente figura se visualiza el diagrama de actividades de la realización del caso de uso "Registrar proyecto".

Fuente: Elaboración propia

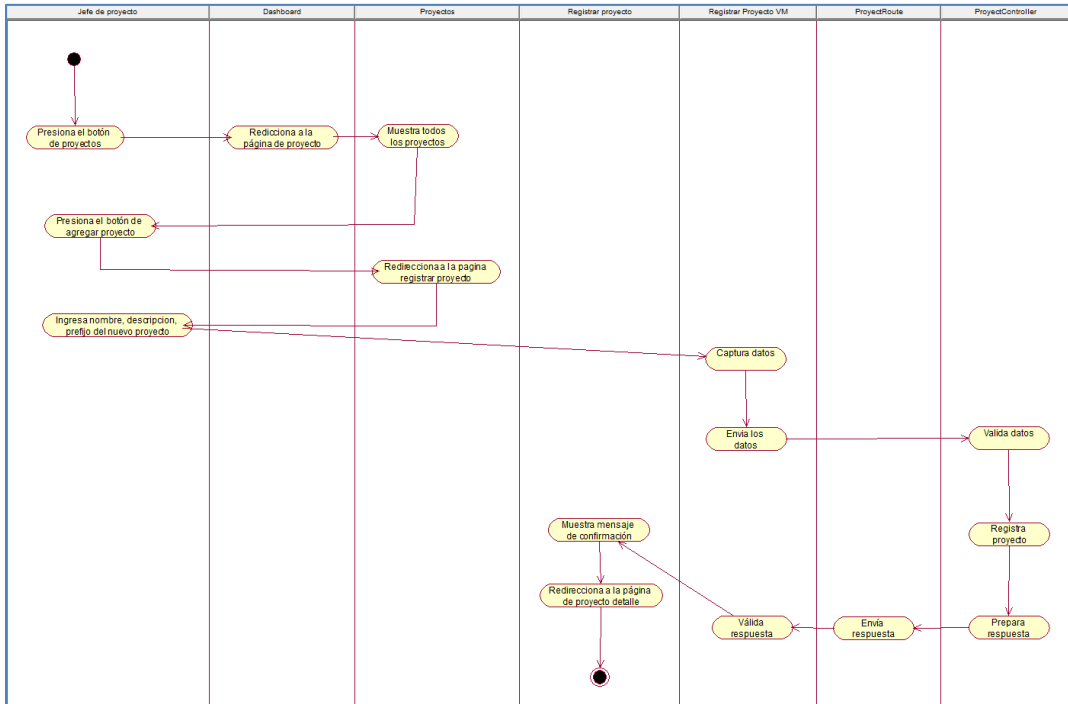


Figura 90 Diagrama de Actividades del caso de uso de sistema "Registrar proyecto"

Explicación: La figura 73 se muestra el diagrama actividades donde se muestra como el usuario registra un nuevo proyecto.

En la siguiente figura se visualiza el diagrama de actividades de la realización del caso de uso "Registrar caso de prueba".

Fuente: Elaboración propia

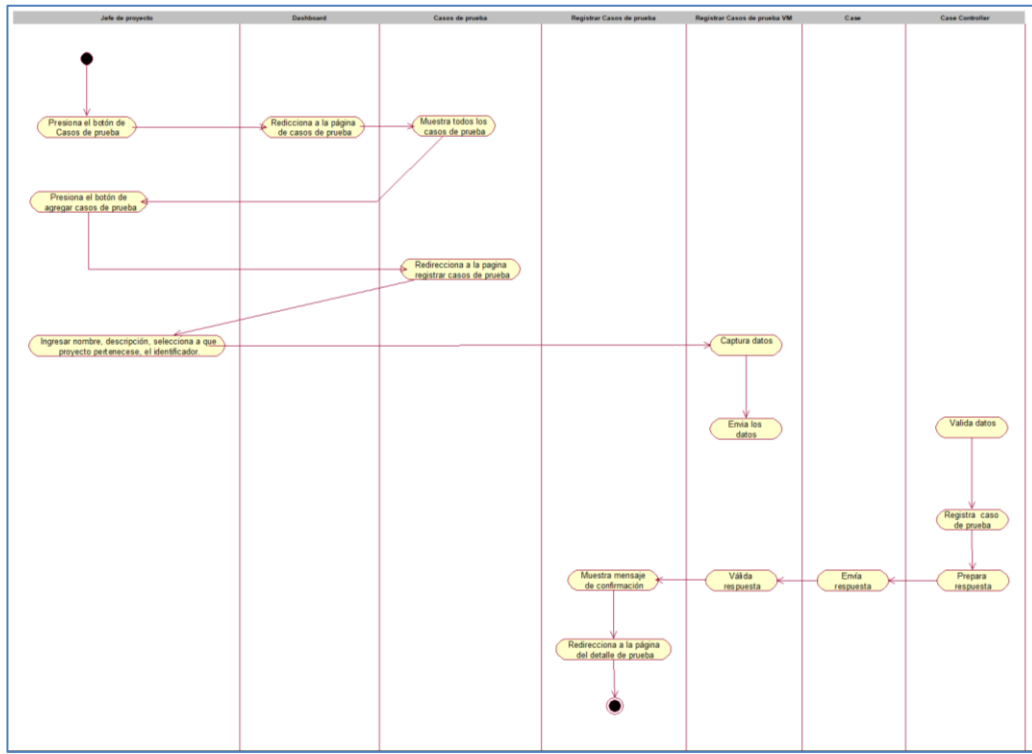


Figura 91 Diagrama de Actividades del caso de uso de sistema "Registrar caso de prueba"

Explicación: La figura 74 se muestra el diagrama actividades donde se muestra como el usuario registra un nuevo caso de prueba.

En la siguiente figura se visualiza el diagrama de actividades de la realización del caso de uso “Resolver casos de prueba”.

Fuente: Elaboración propia

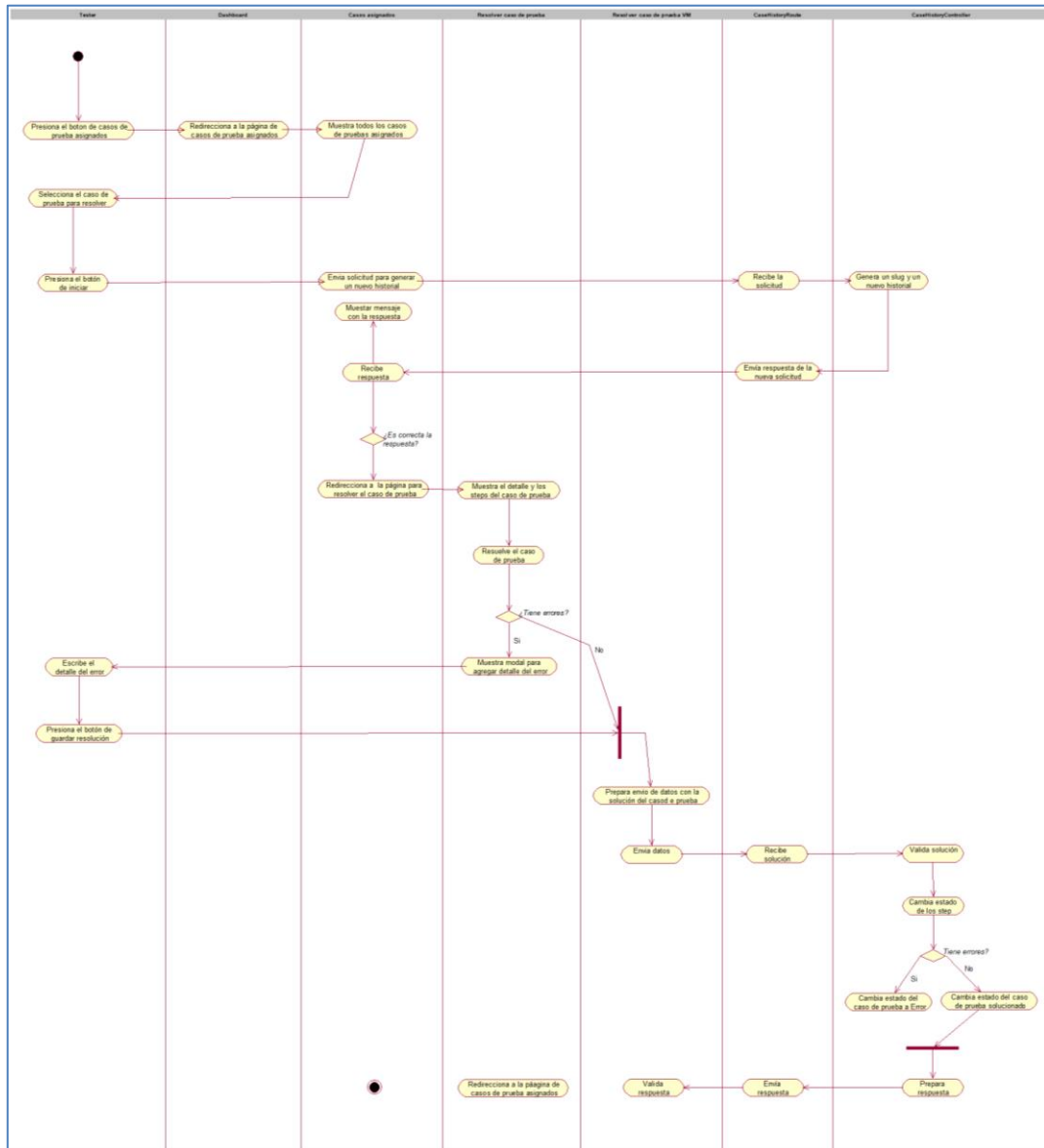


Figura 92 Diagrama de Actividades del caso de uso de sistema "Resolver caso de prueba"

Explicación: La figura 75 se muestra el diagrama actividades donde se muestra como el usuario tester resuelve un caso de prueba.

En la siguiente figura se visualiza el diagrama de actividades de la realización del caso de uso "Cerrar proyecto".

Fuente: Elaboración propia

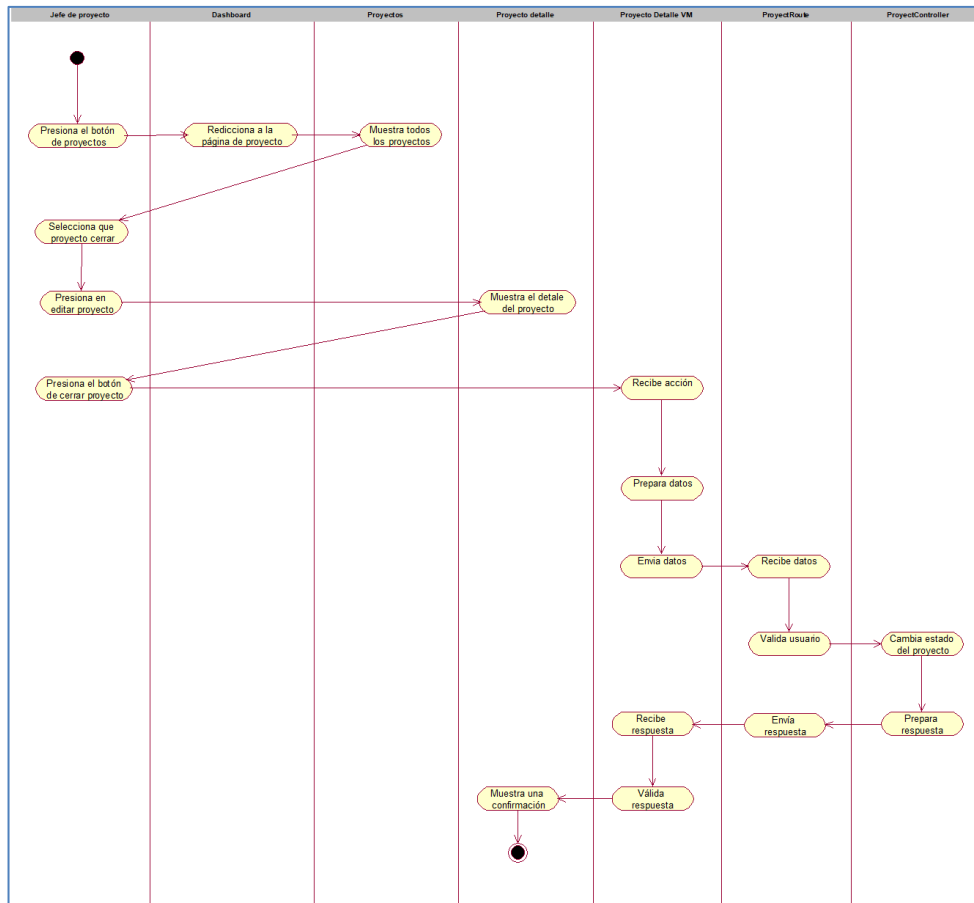


Figura 93 Diagrama de Actividades del caso de uso de sistema "Cerrar proyecto"

Explicación: La figura 76 se muestra el diagrama actividades donde se muestra como el usuario jefe de proyecto cierra un proyecto.

En la siguiente figura se visualiza el diagrama de actividades de la realización del caso de uso "Registrar versión".

Fuente: Elaboración propia

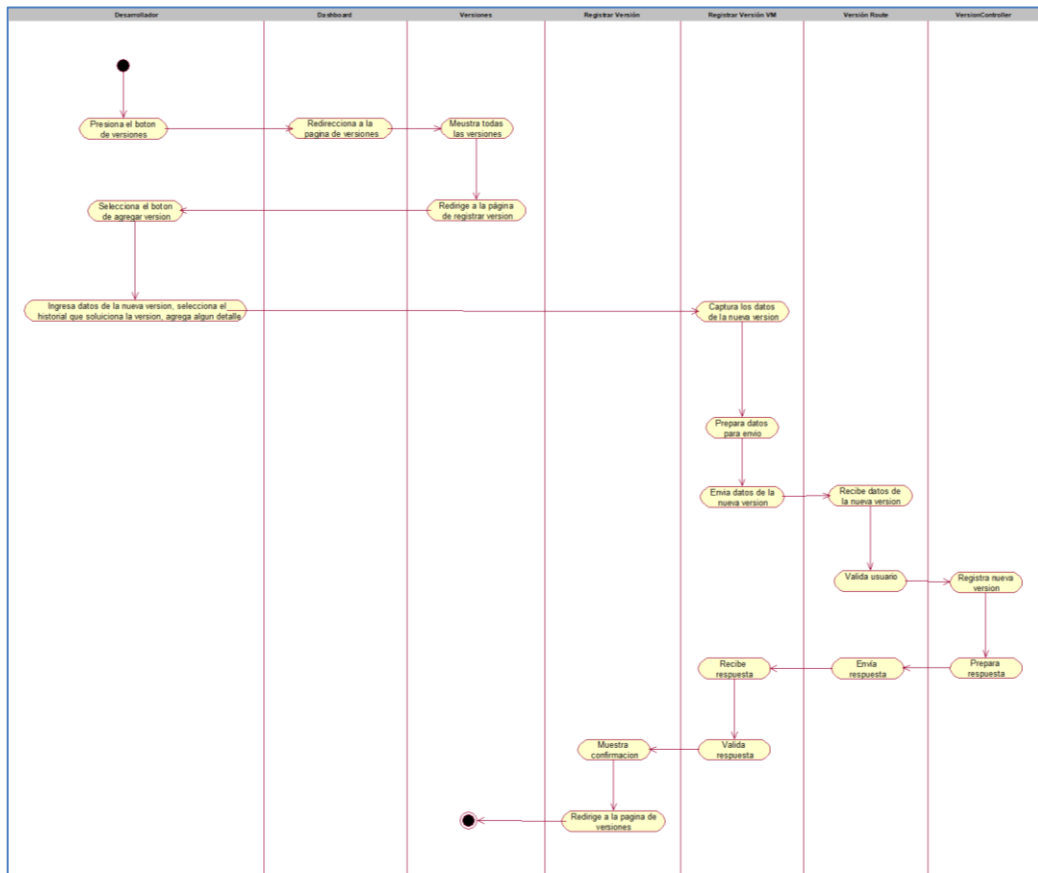


Figura 95 Diagrama de Actividades del caso de uso de sistema "Registrar Versión"

Explicación: La figura 78 muestra el diagrama actividades donde se muestra como el usuario desarrollador agregar una nueva versión.

U. Modelo lógico de base de datos

Fuente: Elaboración propia

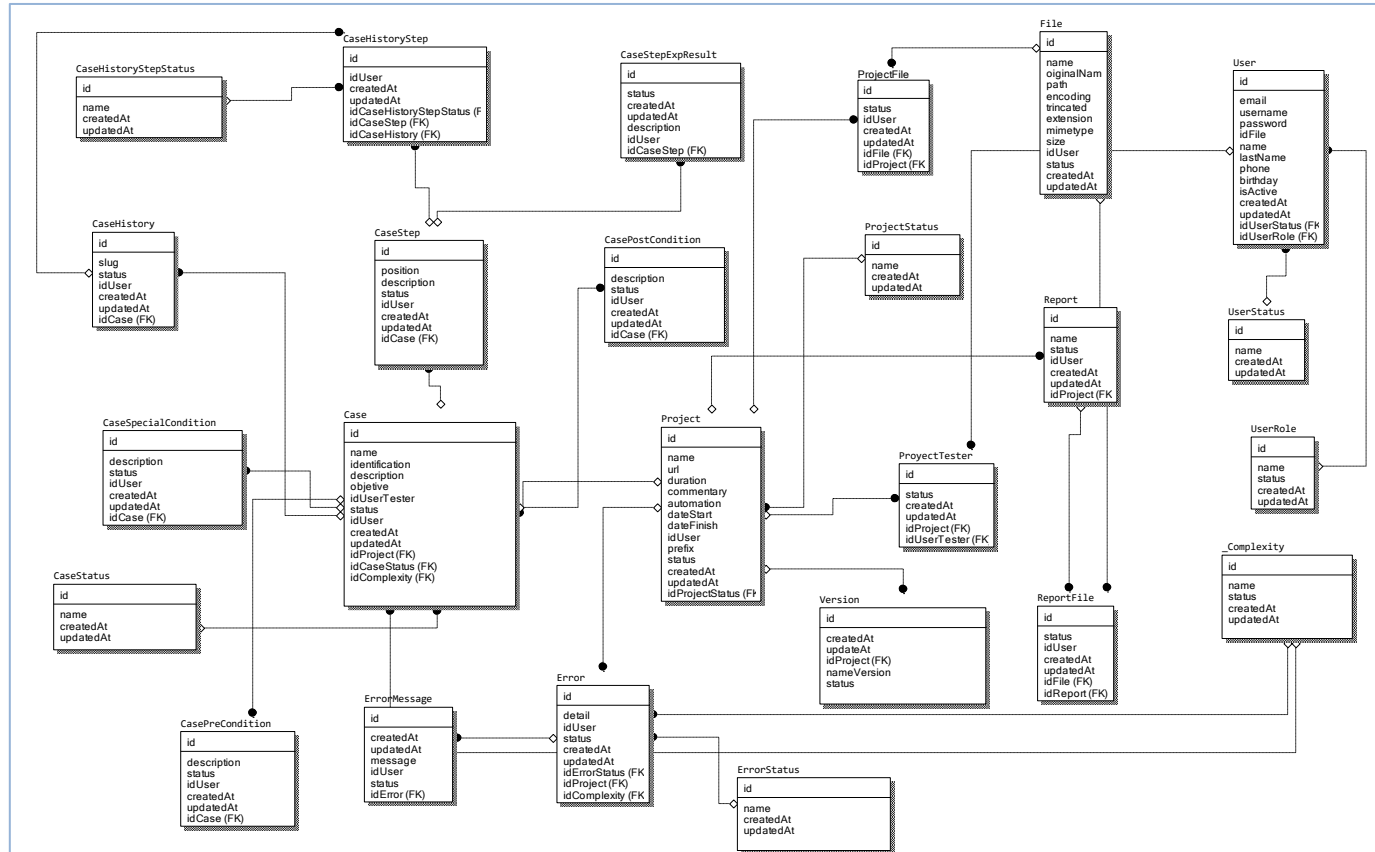


Figura 97 Modelo lógico de base de datos

Explicación: En la Figura N° 80, se muestra la estructura interna del sistema, la información que necesita, para cada uno de las clases ya establecidas, con ello ya tenemos una visión más clara para diseñar la base de datos del sistema.

W. Diccionario de datos

El diccionario de base de datos, permite conocer la estructura de las tablas creadas en el modelo lógico, debido a que la construcción de la base de datos será realizada en el servidor de MySQL, este diccionario de datos muestra el tipo de variable de cada tabla, así como las llaves primarias y foráneas.

Tabla 27 Detalle tabla Case

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	text	Sí	'NULL'	Nombre del caso
idProject	int(11)	Sí	NULL	ID del Proyecto a la que pertenece
identification	varchar(100)	Sí	NULL	Identificador del caso de prueba
description	text	Sí	NULL	Descripción del caso de prueba
objective	text	Sí	NULL	Objetivo del caso de prueba
idComplexity	int(11)	Sí	NULL	Complejidad del caso de prueba
idUserTester	int(11)	Sí	NULL	Usuario Tester
idCaseStatus	int(11)	Sí	1	Estado del caso de prueba
status	int(11)	No	1	Flag de estado, activo o inactivo
idUser	int(11)	Sí	NULL	ID Usuario que creo el caso de prueba
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información del caso de prueba.

Tabla 28 Detalle tabla CaseHistory

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
slug	text	Sí	NULL	Slug del Historial de Caso
timeFinish	varchar(50)	Sí	NULL	Tiempo de finalización
idCase	int(11)	Sí	NULL	ID del caso de prueba
idCaseHistoryStatus	int(11)	Sí	1	Estado del Historial de caso
status	int(11)	No	1	Flag activo del registro
idUser	int(11)	Sí	NULL	Usuario creador
createdAt	timestamp	Sí	current_timestamp()	Fecha creada
updatedAt	timestamp	Sí	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información del historial de caso.

Tabla 29 Detalle tabla CaseHistoryMessage

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
message	text	Sí	NULL	Mensaje
idUser	int(11)	Sí	NULL	Usuario creador
idCaseHistory	int(11)	Sí	NULL	ID del historial del caso
status	int(11)	No	1	Flag activo del registro
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los mensajes del historial de caso.

Tabla 30 Detalle tabla CaseHistoryStatus

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	text	Sí	NULL	Nombre
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información del estado del historial de caso.

Tabla 31 Detalle tabla CaseHistoryStep

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
idCaseHistory	int(11)	No		ID Historial del caso
idCaseStep	int(11)	No		ID Paso del caso
timeStep	varchar(50)	Sí	00:00:00	Tiempo que se realizo
idCaseHistoryStepStatus	int(11)	No	1	ID Estado del Paso
idUser	int(11)	Sí	NULL	Usuario creador
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los pasos del historial de caso.

Tabla 32 Detalle tabla CaseHistoryStepStatus

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	varchar(255)	No		Nombre
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los estados de los pasos del historial de caso.

Tabla 33 Detalle tabla CasePostCondition

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
description	text	Sí	NULL	Descripción
idCase	int(11)	Sí	NULL	ID del caso de prueba
status	int(11)	No	1	Flag activo del registro
idUser	int(11)	Sí	NULL	Usuario creado
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de la condición posterior del caso de prueba.

Tabla 34 Detalle tabla CasePreCondition

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
description	text	Sí	NULL	Descripción
idCase	int(11)	Sí	NULL	ID del caso de prueba
status	int(11)	No	1	Flag activo del registro
idUser	int(11)	Sí	NULL	Usuario creado
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de la condición previa del caso de prueba.

Tabla 35 Detalle tabla CaseSpecialCondition

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
description	text	No		Descripción
idCase	int(11)	No		ID del caso de prueba
status	int(11)	No	1	Flag activo del registro
idUser	int(11)	Sí	NULL	Usuario creado
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de la condición especial del caso de prueba.

Tabla 36 Detalle tabla CaseStatus

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	text	Sí	NULL	Nombre
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los estados del caso de prueba.

Tabla 37 Detalle tabla CaseStep

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
position	int(11)	Sí	NULL	Posición del paso del caso de prueba
description	text	Sí	NULL	Descripción del paso de caso de prueba
expectedResult	text	Sí	NULL	Resultado esperado
idCase	int(11)	Sí	NULL	ID del caso de prueba
status	int(11)	No	1	Flag activo del registro
idUser	int(11)	Sí	NULL	Usuario creador
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los pasos del caso de prueba.

Tabla 38 Detalle tabla Complexity

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	varchar(255)	Sí	NULL	Nombre
status	int(11)	Sí	1	Flag activo del registro
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de la complejidad.

Tabla 39 Detalle tabla Error

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
detail	text	Sí	NULL	Detalle del error
idComplexity	int(11)	Sí	NULL	ID de Complejidad
idErrorStatus	int(11)	Sí	1	ID Estado del error
idCaseHistory	int(11)	Sí	NULL	ID del caso de prueba
idUser	int(11)	Sí	NULL	ID usuario creador
status	int(11)	No	1	Flag activo del registro
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los errores.

Tabla 40 Detalle tabla ErrorMessage

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
message	text	Sí	NULL	Mensaje del error
idError	int(11)	Sí	NULL	ID del error
idUser	int(11)	Sí	NULL	ID del usuario creador
status	int(11)	No	1	Flag activo del registro
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los mensajes de los errores.

Tabla 41 Detalle tabla ErrorStatus

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	varchar(255)	Sí	NULL	Nombre
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los estados de los errores.

Tabla 42 Detalle tabla File

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	varchar(255)	Sí	NULL	Nombre del archivo
originalname	varchar(255)	Sí	NULL	Nombre original
path	varchar(255)	Sí	NULL	Ruta del archivo
encoding	varchar(255)	Sí	NULL	Encoding
truncated	varchar(255)	Sí	NULL	Truncado
extension	varchar(255)	Sí	NULL	Extensión
mimetype	varchar(255)	Sí	NULL	Mimetype
size	varchar(255)	Sí	NULL	Tamaño del archivo
idUser	int(11)	No		ID usuario creador
status	int(11)	No	1	Flag activo del registro
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los archivos.

Tabla 43 Detalle tabla Project

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	varchar(255)	Sí	NULL	Nombre del proyecto
url	varchar(255)	Sí	NULL	Url del proyecto
prefix	varchar(255)	Sí	NULL	Prefijo del proyecto
duration	int(11)	Sí	NULL	Duración del proyecto
comentary	text	Sí	NULL	Comentario
idUserDev	int(11)	No		ID usuario tester
dateStart	date	Sí	NULL	Fecha inicio
dateFinish	date	Sí	NULL	Fecha fin
idProjectStatus	int(11)	Sí	1	ID estado del proyecto
status	int(11)	No	1	Flag activo del registro
idUser	int(11)	Sí	NULL	ID usuario creador
createdAt	timestamp	No	current_timestamp()	Fecha creada

updatedAt	timestamp	No	current_timestamp()	Fecha actualizada
-----------	-----------	----	---------------------	-------------------

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los proyectos.

Tabla 44 Detalle tabla ProjectFile

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
idFile	int(11)	Sí	NULL	ID archivo
idProject	int(11)	Sí	NULL	ID proyecto
status	int(11)	No	1	Flag activo del registro
idUser	int(11)	Sí	NULL	ID usuario creado
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los archivos de los proyectos.

Tabla 45 Detalle tabla ProjectStatus

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	varchar(255)	Sí	NULL	Nombre
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los estados de los proyectos.

Tabla 46 Detalle tabla Report

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	text	No		Nombre del reporte
idProject	int(11)	Sí	NULL	ID del proyecto
status	int(11)	No	1	Flag activo del registro
idUser	int(11)	Sí	NULL	ID usuario creado
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los reportes.

Tabla 47 Detalle tabla ReportFiles

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
idFile	int(11)	Sí	NULL	ID archivo
idReport	int(11)	Sí	NULL	ID reporte
status	int(11)	Sí	1	Flag activo del registro
idUser	int(11)	Sí	NULL	ID usuario creado
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los archivos de los reportes.

Tabla 48 Detalle tabla User

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
email	varchar(255)	Sí	NULL	Correo del usuario
username	varchar(255)	Sí	NULL	Apodo de usuario
password	varchar(255)	Sí	NULL	Contraseña
idFile	int(11)	Sí	NULL	ID archivo foto
name	varchar(255)	No		Nomrbe del usuario
lastName	varchar(255)	Sí	NULL	Apellido del usuario
phone	int(11)	Sí	NULL	Teléfono
birthday	date	Sí	NULL	Cumpleaños
idUserRole	int(11)	Sí	2	ID rol de usuario
idUserStatus	int(11)	No	1	ID estado del usuario
isActive	int(11)	Sí	0	Flag correo confirmado
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los usuarios.

Tabla 49 Detalle tabla UserRole

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	varchar(255)	No		Nombre
status	int(11)	No	1	Flag activo del registro
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los roles de los usuarios.

Tabla 50 Detalle tabla UserStatus

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
name	varchar(255)	No		Nombre
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de los estados de los usuarios.

Tabla 51 Detalle tabla Version

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		Código de identificación de la tabla
idProject	int(11)	Sí	NULL	ID proyecto
status	int(11)	No	1	Flag activo del registro
idUser	int(11)	Sí	NULL	ID usuario creador
createdAt	timestamp	No	current_timestamp()	Fecha creada
updatedAt	timestamp	No	current_timestamp()	Fecha actualizada

Fuente: Elaboración propia

Descripción: Esta tabla almacena información de las versiones.

X. Diagrama de Componentes

En la Figura, se muestra el Diagrama de Componentes, en el que se representan los archivos físicos de la implementación del sistema web open source Vue js para el proceso de pruebas de calidad de software en la empresa NextPerience

Fuente: Elaboración propia

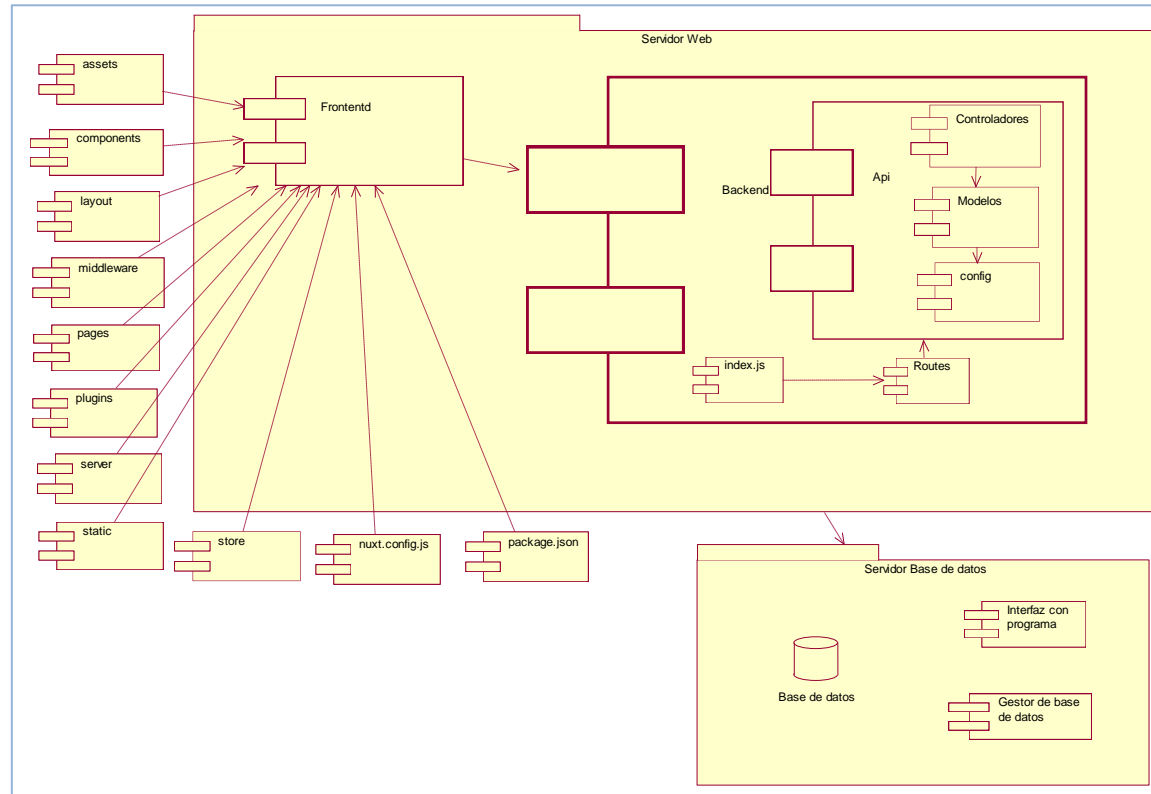


Figura 99 Diagrama de Componentes

Explicación: En la figura se muestran los componentes que implementan las clases de diseño para el sistema web.

Y. Diagrama de despliegue

Representa una vista estática de la configuración de tiempo de ejecución de nodos de procesamiento interconectados. En otras palabras, estos nodos son elementos de hardware sobre los cuales se ejecuta el software.

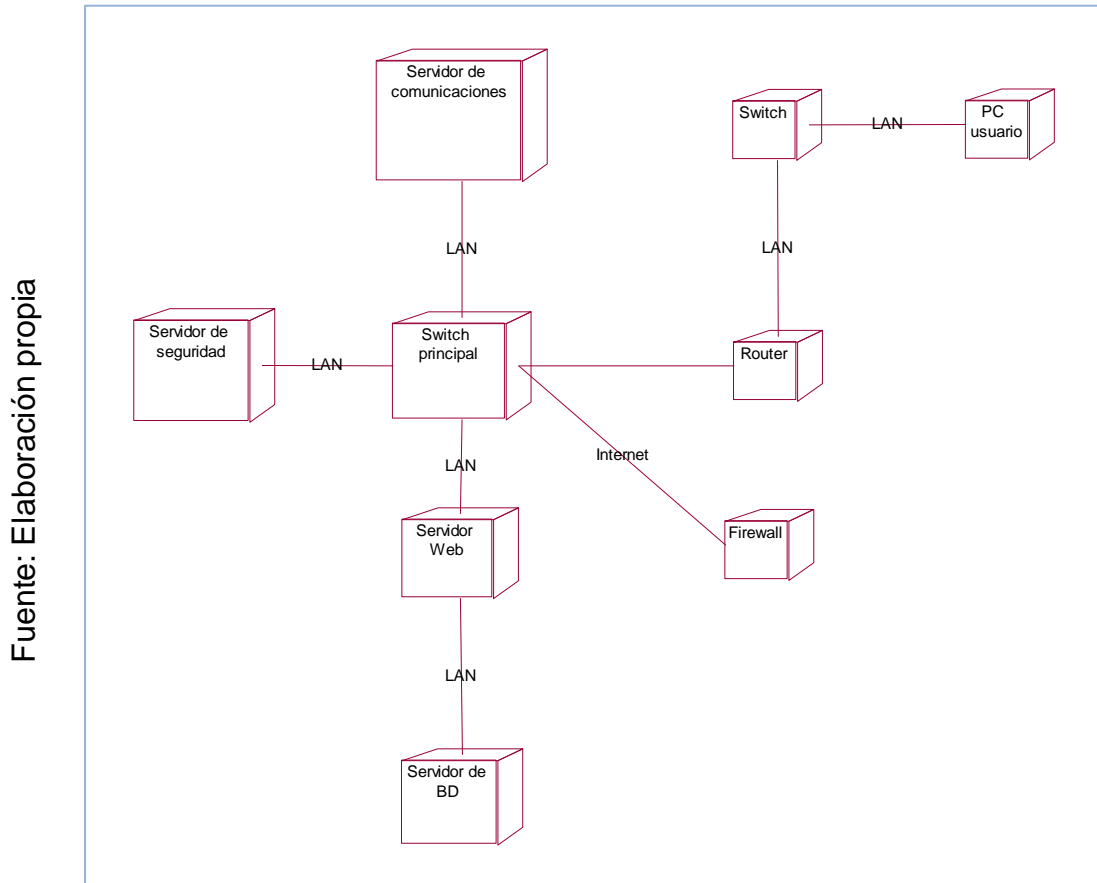


Figura 100 Diagrama de despliegue

Explicación: En la figura se puede observar la vista arquitectónica del diagrama de despliegue del sistema web y como se distribuye a través de los nodos.