



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Metodología de evaluación de la calidad del producto de software en  
base a las normas ISO/IEC 25000 e ISO/IEC 14598 y la metodología  
RUP

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

**AUTOR:**

Romualdo Molina, Jesús Antonio (ORCID:0000-0002-1652-9465)

**ASESOR:**

Dr. Hilario Falcon, Francisco Manuel (ORCID:0000-0003-3153-9343)

**LÍNEA DE INVESTIGACIÓN:**

Auditoría de Sistemas y Seguridad de la Información

LIMA – PERÚ

2021

### **Dedicatoria**

Dedico con todo mi corazón mi proyecto de investigación a las personas que me acompañaron en todo este proceso, en especial a mi madre y hermana.

### **Agradecimiento**

Quiero agradecer en primer lugar a mis abuelos, a mi madre y a mi hermana por mostrar su apoyo durante mi etapa universitaria.

A mi asesor, Dr. Manuel Hilario Falcon por ser guía constante en esta investigación.

Finalmente, al Dr. Emigdio Alfaro Paredes por su tiempo y todas las pautas que me brindo para la realización de este trabajo.

## Índice de contenidos

I. INTRODUCCIÓN .....	1
II. MARCO TEÓRICO .....	6
III. METODOLOGÍA.....	15
<b>3.1 Tipo y diseño de investigación .....</b>	<b>16</b>
<b>3.2 Variables y operacionalización .....</b>	<b>16</b>
<b>3.3 Población, muestra y muestreo .....</b>	<b>17</b>
<b>3.4 Técnicas e instrumentos de recolección de datos.....</b>	<b>17</b>
<b>3.5 Evaluaciones .....</b>	<b>18</b>
<b>3.6 Método de análisis de datos.....</b>	<b>18</b>
<b>3.7 Aspectos éticos .....</b>	<b>19</b>
<b>3.8. Desarrollo de la Metodología .....</b>	<b>19</b>
<b>3.9 Arquitectura de la metodología .....</b>	<b>19</b>
IV. RESULTADOS.....	39
A. Resultados del diagnóstico .....	40
B. Evaluación y formatos de registros elaborados para la empresa según lo indica la metodología.....	70
C. Medición de indicadores .....	113
D. Prueba de hipótesis.....	122
V. DISCUSIÓN.....	128
VI. CONCLUSIONES.....	132
VII. RECOMENDACIONES.....	134
REFERENCIAS .....	136
ANEXOS .....	142

## Índice de tablas

Tabla 1. Análisis de la Norma ISO/IEC 25000.....	24
Tabla 2. Análisis de la Norma ISO/IEC 14598.....	25
Tabla 3. Metodología RUP .....	27
Tabla 4. Metodología Propuesta .....	28
Tabla 5 Comparación metodológica. ....	29
Tabla 6 Elección de Evaluación .....	31
Tabla 7 Lista de evaluaciones propuestos de la metodología .....	37
Tabla 8. Capacidad de respuesta .....	113
Tabla 9. Prueba de normalidad.....	114
Tabla 10. Disponibilidad .....	114
Tabla 11. Prueba de normalidad.....	115
Tabla 12. Número de vulnerabilidades .....	115
Tabla 13. Prueba de normalidad.....	115
Tabla 14.Porcentaje de detección de defectos.....	116
Tabla 15. Prueba de normalidad.....	116
Tabla 16. Tiempo de instalación .....	117
Tabla 17. Prueba de normalidad.....	117
Tabla 18. Tiempo de actualización .....	118
Tabla 19. Prueba de normalidad.....	118
Tabla 20. Tiempo de entrega de cambios .....	119
Tabla 21. Prueba de normalidad.....	119
Tabla 22. Cantidad de código .....	120
Tabla 23. Prueba de normalidad.....	120
Tabla 24. Tiempo medio de reparación .....	120
Tabla 25. Prueba de normalidad.....	121
Tabla 26. Tiempo medio entre fallos.....	121
Tabla 27. Prueba de normalidad.....	122
Tabla 28. Prueba T de la capacidad de respuesta.....	122
Tabla 29. Prueba T del número de vulnerabilidades.....	123
Tabla 30. Prueba T del porcentaje de detección de defectos .....	123
Tabla 31. Prueba T del tiempo de entrega de cambios.....	123
Tabla 32.Prueba T de la cantidad de código .....	124
Tabla 33. Prueba T de tiempo medio de reparación .....	124
Tabla 34. Prueba T de tiempo medio entre fallos.....	124
Tabla 35. Prueba de Wicolxon de la disponibilidad.....	125

Tabla 36. Prueba de Wicolxon del tiempo de instalación.....	125
Tabla 37. Prueba de Wicolxon del tiempo de actualización.....	125
Tabla 38. Resumen de aceptación de hipótesis.....	126
Tabla 39 Matriz de operacionalización de variables.....	145

## Índice de figuras

Figura 1: Estructura de la metodología de evaluación de la calidad de software....	20
Figura 2: Metodología de evaluación de la calidad de software. ....	22
Figura 3: Esquema de Evaluaciones .....	23
Figura 4. Metodología RUP y comparación de pauta Iso 14598, 25000.....	156

## Índice de abreviaturas

<b>Sigla</b>	<b>Significado</b>	<b>Pagina</b>
I/E 25000	Estándar para calificar la calidad de programa informático	12,16,18,20,21,30
SFTW	Software	31,32,33,34,35,36, 37,38
I/E 14598	Estándar para calificar la calidad de un programa informático que tiene peticiones y métricas	12,16,20,30
RUP	Proceso unificado de Rational	3,4,5,13,14,20,24 ,27,28,29,34,35,36,37
MARES	Metodología para la evaluación de procesos de softwares	183



## Resumen

El problema de la investigación fue que no se han encontrado modelos de evaluación de calidad de software con requisitos de evaluación, métricas de evaluación de calidad y un proceso de evaluación en base a los estándares ISO/IEC 25000 e ISO/IEC 14598 y la metodología RUP. El objetivo de la investigación fue desarrollar una metodología de calificación de la calidad de programas en las cuales incluya peticiones de calificación, métricas y un proceso del mismo en base a las pautas I/E 25000 e I/E 14598 y la metodología RUP. La investigación es del tipo descriptiva, que tiene como variable principal a la calificación de calidad de programas, y como muestra se tiene al conjunto de métricas que presentan los modelos y la metodología antes mencionada; se logró desarrollar la metodología de calificación de calidad de programas, con sus métricas, requerimientos y modelos correspondientes; lo que será de gran apoyo a las organizaciones para evaluar sus programas según las necesidades actuales de las TI. Como resultado se evidenció la mejora en la calidad del producto software una vez implementada la metodología desarrollada, se recomienda para futuras investigaciones ampliar la cantidad de indicadores, criterios de análisis y profundizar el análisis de evoluciones históricas.

**Palabras clave:** calificación de calidad de programas, I/E 25000, I/E 14598, metodología RUP

## **Abstract**

The research problem was that no software quality assessment models have been found with assessment requirements, quality assessment metrics and an assessment process based on the ISO/IEC 25000 and ISO/IEC 14598 standards and the RUP methodology. The objective of the research was to develop a software quality qualification methodology that includes qualification requests, metrics and a qualification process based on the I/E 25000 and I/E 14598 guidelines and the RUP methodology. The research is of the descriptive type, which has as main variable the qualification of program quality, and as a sample we have the set of metrics that present the models and the methodology mentioned above; it was possible to develop the methodology of qualification of program quality, with its metrics, requirements and corresponding models; which will be of great support to the organizations to evaluate their programs according to the current needs of the IT. As a result, it was evidenced the improvement in the quality of the software product once the developed methodology was implemented. It is recommended for future researches to expand the number of indicators, analysis criteria and to deepen the analysis of historical evolutions.

**Keywords:** program quality qualification, I/E 25000, I/E 14598, RUP methodology.

# **I. INTRODUCCIÓN**

En el presente capítulo se describe la realidad problemática a la cual afronta el estudio, planteando el problema de investigación y problemas específicos a resolver, así como los objetivos generales y específicos; de la misma manera se presenta la justificación del estudio a nivel teórico, metodológico, tecnológico, social y económico.

El constructo de calidad comprende todas las características generales y significativas de un producto o una actividad que se relaciona con la satisfacción de requisitos dados. Es así que, existen diversos puntos clave para determinar la calidad de los programas: (a) el grado en que el producto posee un atributo deseado o una combinación de atributos. (b) El grado en que un usuario percibe que su producto cumple con sus expectativas, (c) las características del sistema que determinan el grado en que su uso satisfará sus expectativas. (Álvaro y Baseem, 2006)

Como es de conocimiento, los programas se utilizan cada vez más en una amplia variedad de áreas, por lo que su correcto funcionamiento es muy importante para el éxito de las organizaciones. Siendo así que una de las principales preocupaciones de la industria de programas sea producir programas de calidad; por lo tanto, la evaluación de la calidad de programas siempre ha sido de primordial importancia y una tarea de gran prioridad para los profesionales de las industrias de programas. Koh (2017) mencionó que la evaluación de programas es un factor importante para garantizar la calidad y esto puede ser logrado empleando características de calidad adecuadas, teniendo en cuenta el propósito del uso del producto. Por lo tanto, es muy importante que la calidad de programas se evalúe utilizando métricas ampliamente aceptadas y reconocidas.

Para Koh (2017) los procesos y metodologías que se crean para medir la calidad de programas carecen de uniformidad, por lo que se pueden encontrar metodologías muy específicas que miden un conjunto particular de dimensiones de calidad con una determinada métrica. Por lo tanto, es necesario proporcionar un sistema consistente para medir la calidad de programas a un nivel amplio y diverso.

Por otro lado, la investigación se justifica por su conveniencia, ya que el diseño de una nueva metodología de evaluación de calidad de programas permitirá a las organizaciones tener una herramienta con la cual adaptar su producto a los entornos actuales, de acuerdo con Ferro, Martínez y Otero (2009) el uso de nuevas metodologías puede ser extremadamente beneficioso para el desarrollo de las personas las cuales las utilizan.

Desde el punto de vista práctico, el estudio permite aplicar los conocimientos aprendidos en la carrera, de esa forma verificar que dichos conocimientos generan un impacto en la organización donde se aplica, Suárez Téllez et al. (2018) nos dice que la aplicación de conocimientos obtenidos puede aumentar significativamente el interés de las personas que lo aplican por el tema o área en la cual están centrados.

Respecto a la relevancia social del proyecto, está abarcada bajo el concepto de que, si las empresas ganan, todos ganan, ya que al tener una metodología que permite tener un mejor producto de programas, genera más utilidades a la organización y eso se traduce en mayor utilidad para los colaboradores. Lopez (2018) señaló que los beneficios sociales son percepciones que se entregan al trabajador para promover un mayor bienestar a él y a su familia

Finalmente se justifica por su metodología, ya que este estudio pretende contribuir con aportes al diseñar una nueva metodología de evaluación de programas, siendo así que la investigación sirve también como un punto de referencia para futuros estudios en temas relacionados a la evaluación de calidad de programas, ya que aporta en maneras de creación de una nueva metodología. Gamboa (2018) señaló que entre los años 80 y 90 metodologías con una carga pesada de trabajo en planificación, diseño y documentación, absorbían gran parte del tiempo destinado al desarrollo del sistema.

En base a lo descrito anteriormente, el problema de investigación fue que no se han encontrado modelos de evaluación de calidad de software con requisitos de evaluación, métricas de evaluación de calidad y un proceso de

evaluación en base a los estándares ISO/IEC 25000 e ISO/IEC 14598 y la metodología RUP. Los problemas específicos fueron los siguientes:

- **PE1:** No se han encontrado metodologías de evaluación de calidad de software con requisitos de evaluación según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.
- **PE2:** No se han encontrado metodologías de evaluación de calidad de software con métricas de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.
- **PE3:** No se han encontrado metodologías de evaluación de calidad de software con un proceso de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.

El objetivo general fue desarrollar una metodología de evaluación de la calidad de software que incluya requisitos de evaluación, métricas de evaluación de calidad y un proceso de evaluación en base a los estándares ISO/IEC 25000 e ISO/IEC 14598 y la metodología RUP.

Los objetivos específicos son los siguientes:

- **OE1:** Desarrollar una metodología de evaluación de calidad de software con requisitos de evaluación según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP
- **OE2:** Desarrollar una metodología de evaluación de calidad de software con métricas de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP
- **OE3:** Desarrollar una metodología de evaluación de calidad de software con un proceso de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP

La hipótesis general de la investigación fue: Se desarrolló una metodología de evaluación de la calidad de software que incluya requisitos de evaluación, métricas de evaluación de calidad y un proceso de evaluación en base a los estándares ISO/IEC 25000 e ISO/IEC 14598 y la metodología RUP:

Las hipótesis específicas fueron las siguientes:

- **HE1:** Se desarrolló una metodología de evaluación de calidad de software con requisitos de evaluación según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.  
De acuerdo con («Norma ISO/IEC 25000 | Tecnología Investigación y Academia» 2017) las normas ISO 25000 nos proveen de una lista de requisitos para el desarrollo de programas informáticos de calidad.
- **HE2:** Se desarrolló una metodología de evaluación de calidad de software con métricas de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.  
Según (Torrealba-Peña y Mendoza Mejia 2018) las normas ISO 9126 e ISO 14598 se especializan en brindar normas y estándares para la evaluación de programas informáticos de calidad.
- **HE3:** Se desarrolló una metodología de evaluación de calidad de software con un proceso de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.  
Según (Diaz-Antón et al. 2003) la metodología RUP puede utilizarse en conjunto con otras metodologías para poder elaborar procesos de evaluación.

## **II. MARCO TEÓRICO**



En este capítulo encontramos los antecedentes nacionales e internacionales relacionados con el presente estudio; así también se presentan las teorías relacionadas a las pautas de calidad de programas.

Dalla et al (2020) en su estudio propone un catálogo que consta de 46 métricas para identificar las propiedades de la infraestructura como código (IaC) centrándose en Ansible. Primero se buscó métricas de código fuente tradicionales e independientes del lenguaje que sean potencialmente aplicables a IaC. En este paso, el autor consideró dos criterios de selección: (i) la métrica debe ser obtenida de la fuente analizada; (ii) el significado de la métrica debe ser claro para permitir discusiones sobre su impacto potencial en la calidad del código de infraestructura. Posteriormente, se realizó una discusión abierta y un ejercicio de clasificación de tarjetas con los autores restantes para refinar el catálogo. El estudio concluye con la propuesta de un amplio catálogo de 46 medidas de código basadas en estructuras para evaluar los diferentes aspectos de IaC,

Atoum (2020) en su investigación titulada propone un nuevo marco QinU (QinUF) para medir las revisiones de software que consumen QinU de manera competente. Se propuso un marco QinU global (QinUF) que es capaz de predecir y puntuar QinU a partir de revisiones de software. El marco utiliza una medida de similitud semántica para clasificar las oraciones de revisión de software en temas de QinU. A continuación, se estiman los valores de orientación del sentimiento de las oraciones de revisión de QinU. Finalmente, los valores de orientación se agrupan y resumen por tema de QinU para calcular el QinU general. Los experimentos en casos de uso de la vida real mostraron que QinUF logró el mayor rendimiento en comparación con los métodos conocidos seleccionados con solo las 10 características principales de QinU. Los resultados de la investigación son consistentes y contribuyen a esta área raramente investigada.

Xin (2019) en su investigación proporciona un resumen organizado y sintetizado de los modelos de calificación de calidad (QAM) actuales. Se obtuvo un total de 716 artículos, y finalmente se seleccionan 31. El trabajo se centra en los QAM desde los siguientes aspectos: métricas de software, factores de calidad, métodos de agregación, métodos de evaluación y soporte de

herramientas. De acuerdo con los resultados del análisis, el trabajo descubre cinco necesidades que los investigadores en esta área deben continuar abordando: (1) un nuevo método y criterios para adaptar un marco de calidad (2) investigaciones sistemáticas sobre la efectividad, fortaleza y debilidad de diferentes métodos de agregación, (3) más investigaciones sobre la evaluación de QAM en el contexto de casos industriales, (4) más investigaciones o estudios de casos del mundo real sobre las herramientas relacionadas con QAM, y (5) la construcción de un software de referencia público y diverso que se puede adoptar en diferentes contextos de aplicación.

Hynninen, Kasurinen y Taipale (2018) en su estudio presenta un marco para recopilar métricas de tiempo de ejecución y medir la calidad en uso del software con una interfaz sistemática; para realizar el marco, se siguió el proceso descrito en la norma ISO 15939. En la prueba de concepto, se implementó las métricas como una biblioteca de medición en una aplicación de código abierto. De los resultados del estudio, el marco de medición y el proyecto de prueba de concepto se evaluaron mediante el uso de escenarios descriptivos para el software en la fase de mantenimiento de su ciclo de vida. El marco de medición se implementó como una biblioteca de métricas y las mediciones se vincularon al software como sondas durante el desarrollo. Este trabajo asigna las métricas del software en tiempo de ejecución a las características de calidad.

Vargas et al (2018) en su investigación revisa MECRAD propone su introducción en los programas de formación de ingenieros que es una metodología para la técnica de evaluación de las herramientas de entornos visuales para generar sistemas de información de tipo comerciales. Los productos se evaluaron utilizando como referencia elementos básicos de las normas internacionales. Con esta metodología se pudo evaluar y seleccionar de forma eficaz y sencilla aquellas herramientas y plataformas de desarrollo más adecuadas para crear aplicaciones en entornos visuales, con el fin de generar sistemas de información con calidad y sostenibilidad; siendo útil para instituciones de educación superior, organizaciones, empresas y usuarios finales del sistema, entre otros.

Sacha (2015) en su tesis describe un método que se usó para calificar la calidad esperada del programa que fue desarrollado para un enorme sistema gubernamental. El enfoque se basó en una modificación de GQM y se centró en la evaluación de la calidad de los métodos, herramientas y entregables del proyecto. Se describe un método práctico que se puede utilizar para evaluar la calidad del programa en diseño. El método se utilizó con éxito para evaluar la calidad de los productos de software; durante el desarrollo del sistema IACS. La evaluación se realizó en nombre del cliente y no el fabricante del sistema. Los criterios y preguntas que guió el proceso de evaluación permitió un análisis sistemático y en profundidad de los entregables de las actividades de desarrollo particulares. Como resultado, se revelaron varios riesgos, las recomendaciones ayudaron a evitar los riesgos en el producto final. El sistema IACS fue construido y certificado para su uso dentro del plazo.

Wen-Hong y Xin (2012) en su artículo busca mejorar la efectividad, visibilidad y especificación de las pruebas del software TT&C, este trabajo realizó un estudio en profundidad de acuerdo a sus características. El artículo presenta un modelo de evaluación de la calidad con alta confiabilidad y demandas en tiempo real obteniendo una idea del proceso de jerarquía analítica (AHP). Luego, se presenta un entorno de prueba de simulación dedicado y un método de generación de casos de prueba de software basado en el análisis de árbol de fallas (FTA). Los resultados muestran que el modelo dado de este documento puede representar la calidad del software de manera objetiva, el método de generación puede mejorar la suficiencia de los casos de prueba de software de manera efectiva y la evaluación definida puede garantizar la especificación de las pruebas de software disponibles.

Kanellopoulos et al (2010) en su tesis titulada propone una metodología para la evaluación de la calidad del código fuente y el comportamiento estático de un software basado en la norma I/E 9126. Utiliza elementos derivados automáticamente del código fuente mejorado con conocimiento experto en forma de clasificaciones de características de calidad, lo que permite que el software asigne pesos a los atributos del código fuente. Es flexible en términos del conjunto de métricas y atributos del código fuente empleados, incluso en términos de las características I/E 9126 a evaluar. Se aplicó la metodología a

dos estudios de caso, que involucran cinco códigos abiertos y un sistema propietario. Los resultados demostraron que la metodología puede capturar las tendencias de calidad del software y expresar percepciones sobre la calidad del sistema de manera cuantitativa y sistemática.

Respecto al marco teórico, se comenzará entendiendo la importancia de los métodos en la ingeniería de programas, la cual es importante para desarrollar productos de programas de alta calidad a un costo razonable. A medida que las computadoras se utilizan en áreas cada vez más críticas de la industria, la calidad del programa se convierte en una clave para el éxito empresarial y la seguridad humana. La calificación y aseguramiento de la calidad del programa eficaz requiere modelos que describan cuál es la calidad del programa y cómo se puede rastrear hasta la evaluación de desarrollo. Se pueden seguir dos enfoques para garantizar la calidad del programa. Uno se centra en una especificación y calificación directa de la calidad del producto de programa, mientras que el otro se centra en asegurar la alta calidad de la evaluación mediante el cual se desarrolla el producto.

La industria del programa está entrando actualmente en un período de madurez, en el que los enfoques informales particulares se especifican con mayor precisión. Las particularidades de calidad de los productos de programa se definen en ISO / IEC 9126, para cada característica, se determina un conjunto de atributos que se pueden medir. Esta definición ayuda a evaluar la calidad del programa, pero no ofrece una guía sobre cómo construir un producto de programa de alta calidad. Todas las peticiones están pensadas para su aplicación dentro de una evaluación de programa con el fin de mejorar la satisfacción del cliente, que se considera la medida principal de la calidad del producto. El sistema de gestión de la calidad, tal como lo define la pauta, puede estar sujeto a una certificación. (I/E 9126)

Definimos la calidad de programas, qué según la Junta Internacional de Cualificaciones de Pruebas de Programas (ISTQB), la descripción de la garantía de calidad es cumplir con lo solicitado por el cliente. (ISTQB 2017). Se describe con más detalle la garantía de calidad, el cual es un conjunto de diferentes labores para garantizar calidad en los Evaluaciones de ingeniería de programas que, en última instancia, resultan en calidad de programas. El aseguramiento de

la calidad incluye labores como la definición y puesta en marcha de evaluaciones, auditoría y formación. La garantía de calidad puede ser utilizada por varias evaluaciones como programas, metodología de desarrollo, administración de esquemas, administración de la configuración, administración de peticiones, estimación, diseño y pruebas de programas.

Según Pressman (2005) la calidad del programa no aparece simplemente. La calidad del programa es el resultado de una buena gestión de esquemas e ingeniería de programa. Con el uso del aseguramiento de la calidad es posible crear la infraestructura para apoyar los métodos de ingeniería de programa, la gestión de esquemas y las acciones de control de calidad. El fin de la garantía de calidad es proporcionar al equipo técnico y de gestión los datos necesarios para lograr la calidad del producto. Owens y Khazanchi (2009) definen el aseguramiento de la calidad del programa (SQA) como una evaluación bien definida y repetible que forma parte de la gestión de esquemas y del lapso de vida del desarrollo del programa. El fin es asegurar el cumplimiento de las peticiones, reducir el riesgo y mejorar la calidad.

Las responsabilidades de aseguramiento de la calidad son identificar debilidades en las evaluaciones y corregir esas debilidades para mejorar continuamente. El sistema puntualmente se basa en uno o más de las siguientes pautas más populares, como CMMI, Six Sigma o ISO 9000. (Fundamentos de pruebas de programas 2017). El aseguramiento de la calidad de programas abarca todo el lapso de vida del desarrollo de programas y el fin es asegurar que las evaluaciones de desarrollo o mantenimiento sean continuos, mejorados para producir productos que cumplan con especificaciones o peticiones. (Pruebas de programas fundamentos 2017).

La administración de calidad se utiliza para garantizar la calidad del producto de programas como parte de la evaluación de desarrollo. La prueba es una parte de la garantía de calidad y es la evaluación de analizar un elemento de programas para detectar las diferencias entre las condiciones existentes y requeridas. Tradicionalmente, las pruebas han incluido un conjunto de labores al final del esquema de desarrollo, sin embargo, esto tiene a menudo peticiones de duración y esfuerzo impredecibles.

Por lo tanto, los esquemas ágiles en el tiempo requieren un enfoque diferente para el aseguramiento de la calidad. (Petkov et al 2008). De tal manera,

se propone un esquema de calidad de programas compuesto por una jerarquía de vistas y tres esquemas de particularidades de calidad de programas. La jerarquía de la vista de programas se compone de dos niveles: vista final y vista de medios en el primer nivel, vista de contingencia y vista intrínseca como subvistas de la vista de medios.

Por otro lado, la seguridad de un sistema depende principalmente de programas y las evaluaciones de seguridad más que de los productos de programas de aplicación individual. (Muñoz et al., 2019)

Dentro de las principales pautas de calificación de la calidad de programas encontramos la I/E 25000, la cual es un conjunto de pautas que buscan medir la calidad de programas a nivel interno o externo, basada en 6 sub particularidades principales: seguridad, fiabilidad, funcionalidad, interoperabilidad, eficiencia, usabilidad, portabilidad y mantenibilidad; y cada una con sus sub particularidades y fórmulas específicas para evaluar la calidad de programas. (anexo 8)

For developers, debe ser utilizado por empresas que están planificando el desarrollo de un nuevo producto o actualizar un producto existente. La evaluación se centra en indicadores que miden productos intermedios durante el lapso de vida. For acquires, debe ser utilizado por empresas que están planeando adquirir un nuevo producto o actualizar un producto existente. El Evaluación se utiliza para aceptación o selección del producto.

For evaluators, debe ser utilizado por evaluadores que llevan a cabo una calificación independiente del producto. Esta evaluación generalmente se realiza desde la solicitud de un desarrollador o de alguna otra parte involucrada.

For planning and managemnt, contiene peticiones y orientación para particularidades de soporte para la calificación de productos de programas. El soporte es para planificar y gestionar una evaluación y labores de calificación de programas, incluido en él. Esta parte de la pauta suele ser utilizada por el gerente para producir plan de calificación (ISO 14598-1, 2012).

Los métodos de calificación de la calidad descritos, representan el punto de vista de la organización de desarrollo de programa. Sin embargo, existe una gran diferencia entre la calificación de la calidad realizada por un fabricante de programa y la calificación que se realiza para el cliente. Un fabricante puede definir un conjunto de métricas que describen particularidades de calidad

particulares, medir y recopilar datos históricos de un conjunto de esquemas similares y comparar los datos actuales con los tomados de la base de datos histórica.

Este enfoque se centra en una calificación directa de la calidad de un producto de programa y se puede poner en marcha utilizando el método GQM (Meta - Pregunta - Métrica), descrito por primera vez y desarrollado desde entonces por la NASA. El conjunto de fines o particularidades de calidad puede ser el mismo o similar al definido en ISO / IEC 9126 (Basili et al, 1994). Es muy difícil para un cliente seguir este enfoque. Un problema es que el cliente solo tiene acceso limitado a los datos del esquema, y la calificación de la calidad debe basarse en una calificación de los entregables de la evaluación de programa que se enumeraron en el contrato.

Otro problema es que el cliente no tiene datos históricos relacionados con un conjunto de esquemas similares y no puede comparar los datos reales con los históricos. Por lo tanto, el cliente no tiene fines comerciales específicos, como mejorar la evaluación del programa, utilizar menos recursos o mejorar la (aún desconocida) eficiencia del programa. Después de firmar un contrato, es el fabricante el responsable de desarrollar el programa, mientras que el cliente solo quiere estar seguro de que todo está bien hecho. (Basili et al, 1984)

El Proceso Unificado Racional (RUP) es un proceso de desarrollo de programas que consiste en un conjunto más o menos completo de elementos de evaluación para esquemas de desarrollo de programas. RUP define un esquema de desarrollo de programas como un conjunto de disciplinas (p. ej. gestión de peticiones, puesta en marcha, etc.), de principio a fin a lo largo de todo el lapso del esquema dividido en un conjunto de periodos del esquema. Un esquema es realizado por un grupo de actores, cada uno con uno o más roles bien definidos. Cada rol participa en una o más labores que producen uno o más artefactos. Las labores, roles y artefactos son los elementos básicos de la evaluación de RUP.

Como RUP es un marco integral que comprende la mayoría de los puntos de la evaluación de desarrollo de programas significa que de alguna manera debe adaptarse a la situación de uso, ya sea ad-hoc para cada esquema o anticipadamente para producir una pauta amplia. (Krutchen, 2000).

La RUP se entrega por medio de una base de conocimientos de revisión permitida para la web. La evaluación es destinada a mejorar el desempeño del

grupo y ofrecer las prácticas optimizadas de programa a través de pasos, formatos y asesores de herramientas para todas las labores críticas del lapso de vida del programa. El conocimiento inicial permite que los grupos de desarrollo obtengan todos los beneficios de la pauta de la industria Unified Lenguaje de bosquejado (UML) (Rumbaugh, 1999). RUP se puede redactar en dos dimensiones: (1) el fundamento horizontal es el tiempo y releva el aspecto dinámico de la evaluación, (2) el fundamento vertical representa el aspecto estático de la evaluación - labores, artefactos, roles y disciplinas.



### **III. METODOLOGÍA**

### 3.1 Tipo y diseño de investigación

De acuerdo con el enfoque, la investigación es cuantitativa. Este tipo de investigaciones obtienen y procesan los datos de manera estadística y siguen un sistema predecible y estructurado. (Hernández et al., 2014).

Tipo de investigación: básica

El diseño de la investigación es descriptivo, ya que busca describir las particularidades de una situación o suceso por medio de una población sometida al análisis. (Hernández et al 2014). Cuya gráfica es la siguiente:

$$M \rightarrow O$$

Dónde:

M: Calidad del programa informático

O: Programa de la empresa

### 3.2 Variables y operacionalización

#### Variable 1: Calidad del programa informático

Definición conceptual: son los requisitos del cliente que el programa informático debe satisfacer y las características que este debe tener según estándar. (ISTQB 2017).

#### Indicadores:

- Capacidad de respuesta (I/E 25000 e I/E 14598)
- Disponibilidad (I/E 25000 e I/E 14598)
- Número de vulnerabilidades (I/E 25000 e I/E 14598)
- Porcentaje de detección de defectos (I/E 25000 e I/E 14598)
- Tiempo de instalación (I/E 25000 e I/E 14598)
- Tiempo de actualización (I/E 25000 e I/E 14598)
- Tiempo de entrega de cambios (I/E 25000 e I/E 14598)
- Cantidad de código (I/E 25000 e I/E 14598)
- Tiempo medio de reparación (I/E 25000 e I/E 14598)
- Tiempo medio entre fallos (I/E 25000 e I/E 14598)

### **3.3 Población, muestra y muestreo**

La población de esta investigación son los programas elaborados por la empresa con la cual se trabajará, de esta forma por conveniencia es que se tomará como muestra uno de los mismos.

La técnica de muestreo es tipo no probabilístico; el cual se caracteriza por ser una muestra seleccionada por el investigador del estudio bajo ciertas particularidades únicas de la muestra o limitaciones del estudio (Hernández et al, 2014). Esto por conveniencia para la investigación, ya que de esta forma es posible afirmar que todos los miembros de la población tienen la misma probabilidad de ser escogidos.

Teniendo en cuenta ello y dadas las condiciones de la población, la muestra está conformada por un esquema de desarrollo de programa de la empresa enfocado a sistemas de ventas con facturación electrónica, se selecciona solo un esquema debido al tiempo que toma desarrollarlo, que es aproximadamente 3 meses y el tiempo aproximado para el desarrollo de la investigación es de 4 meses, por lo tanto, se aborda sólo un esquema.

### **3.4 Técnicas e instrumentos de recolección de datos**

La técnica que se usó para la recolección de datos fue la ficha de registro puesto que la ficha de registro es un instrumento para la investigación en él se puede almacenar información coherente y ordenada. Este tipo de fichas pueden ser en formato físico o digital dependiendo de las necesidades y conveniencia del investigador (Hernández, 2014).

En el caso de esta investigación, se construyeron tres diferentes fichas de registro, una por cada uno de los indicadores, las cuales fueron revisadas por un ingeniero cuyo campo de acción es el desarrollo puesto que así se puede asegurar que los campos a llenar podrán satisfacer las necesidades de la investigación (Hernández, 2014).

Finalmente, por la misma naturaleza de las fichas de registro, la prueba de confiabilidad no es necesaria, puesto que la data la cual será almacenada en ellas es un reflejo de la realidad y por lo tanto los datos son confiables en su totalidad.

### **3.5 Evaluaciones**

El proceso de elaboración de la metodología propuesta como parte de la investigación, consta de las siguientes actividades:

- a. Revisión bibliográfica sobre los estándares de I/E 25000, 14598 y la metodología RUP.
- b. Selección de los requisitos, métricas y procesos de la I/E 25000, 14598 y la metodología RUP.
- c. Elaboración de la metodología para la evaluación de calidad de software.
- d. Presentación de la metodología a 3 expertos para su evaluación
- e. Aplicación de la metodología en un programa informático de gestión de ventas de una empresa.
- f. Mejora de la metodología
- g. Aplicación de la metodología mejorada.
- h. Evaluación de resultados de la aplicación de la metodología en un programa informático de gestión de ventas de una empresa.

### **3.6 Método de análisis de datos**

Los resultados obtenidos de la aplicación de la metodología, se analizan en base al impacto en los indicadores de calidad, tiempo de aplicación y costos de aplicación.

Para el tiempo de aplicación se utilizará una ficha de registro en el cual se detalla cada requisito de la metodología y el tiempo que le tomó a la empresa implementar cada uno.

Para calcular el costo de aplicación se utilizará una ficha de registro en el cual se detalla cada requisito de la metodología y el costo que le tomó a la empresa implementar cada uno.

### **3.7 Aspectos éticos**

A lo largo del desarrollo de este trabajo de investigación se mantuvo el respeto por la autoría de las fuentes las cuales fueron usadas, asimismo se respetaron los valores inculcados como profesional de ingeniería como lo pueden ser: la responsabilidad, la puntualidad, la honestidad, la tolerancia y la lealtad. La elaboración de esta investigación en ningún momento de su desarrollo fue perjudicial para ninguna institución, empresa o sociedad. Además, se cumplió con todo lo estipulado dentro del Código de Ética de Investigación de la Universidad Cesar Vallejo (Universidad Cesar Vallejo, 2020) con relación a lo detallado a continuación:

- Esta investigación fue desarrollada por el autor y no es copia de ningún otro documento de ninguna forma.
- Esta investigación respeto los máximos estándares de rigor científico, por lo que esta investigación se puede afirmar que cumplió con justicia, libertad, responsabilidad, autonomía y beneficencia. Adicionalmente, el investigador pasara por un sistema anti-plagio.
- Este trabajo fue realizado con la única finalidad de aportar conocimiento y originalidad.

### **3.8. Desarrollo de la Metodología**

El proceso para el desarrollo de la metodología propuesta es el siguiente:

#### **3.8.1 Arquitectura de la metodología**

En el siguiente gráfico, se expresa la estructura de la metodología propuesta en esta investigación

- a. Para elaborar los requisitos, métricas y proceso de evaluación se revisan los requisitos, métricas y procesos propios de cada estándar I/E 25000, I/E 14598 y la metodología RUP.
- b. Desarrollado lo anterior, se elabora el Evaluación general de aplicación de la metodología bajo el ciclo de mejora continua PHVA.
- c. Finalmente, se elaboran las evaluaciones enlazadas a la evaluación general de la metodología de evaluación de calidad de software propuestos.

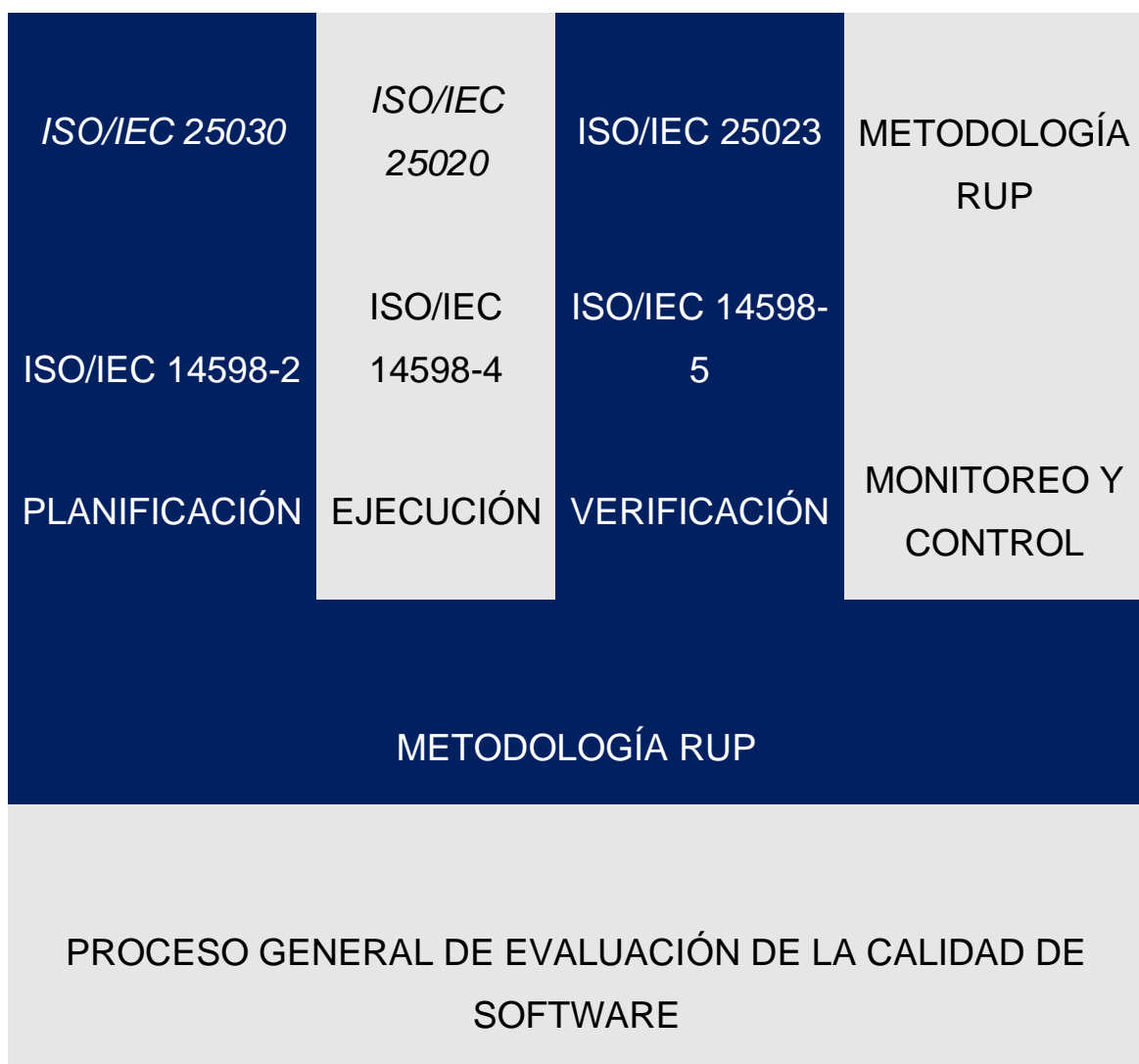


Figura 1: Estructura de la metodología de evaluación de la calidad de software

La figura 1 representa la estructura de objetivos de control que plantea la metodología propuesta, la cual a su vez está enmarcada en el modelo de procesos y como se puede apreciar todo está basado en la familia de I/E 25000 y 14598.

### **3.8.2 Estructura de la metodología**

La metodología resultante en la figura 2, comprende los siguientes elementos: alcance de la evaluación del software, elaboración del plan de trabajo, entradas (información necesaria), proceso de la metodología de evaluación y salidas (informe). De la misma forma, cada evaluación es necesario para aplicar la metodología en donde tiene la siguiente estructura: objetivo, alcance, entradas, proceso y las salidas; en la tabla 1 se proponen algunas evaluaciones de la metodología basado en los estándares analizados.

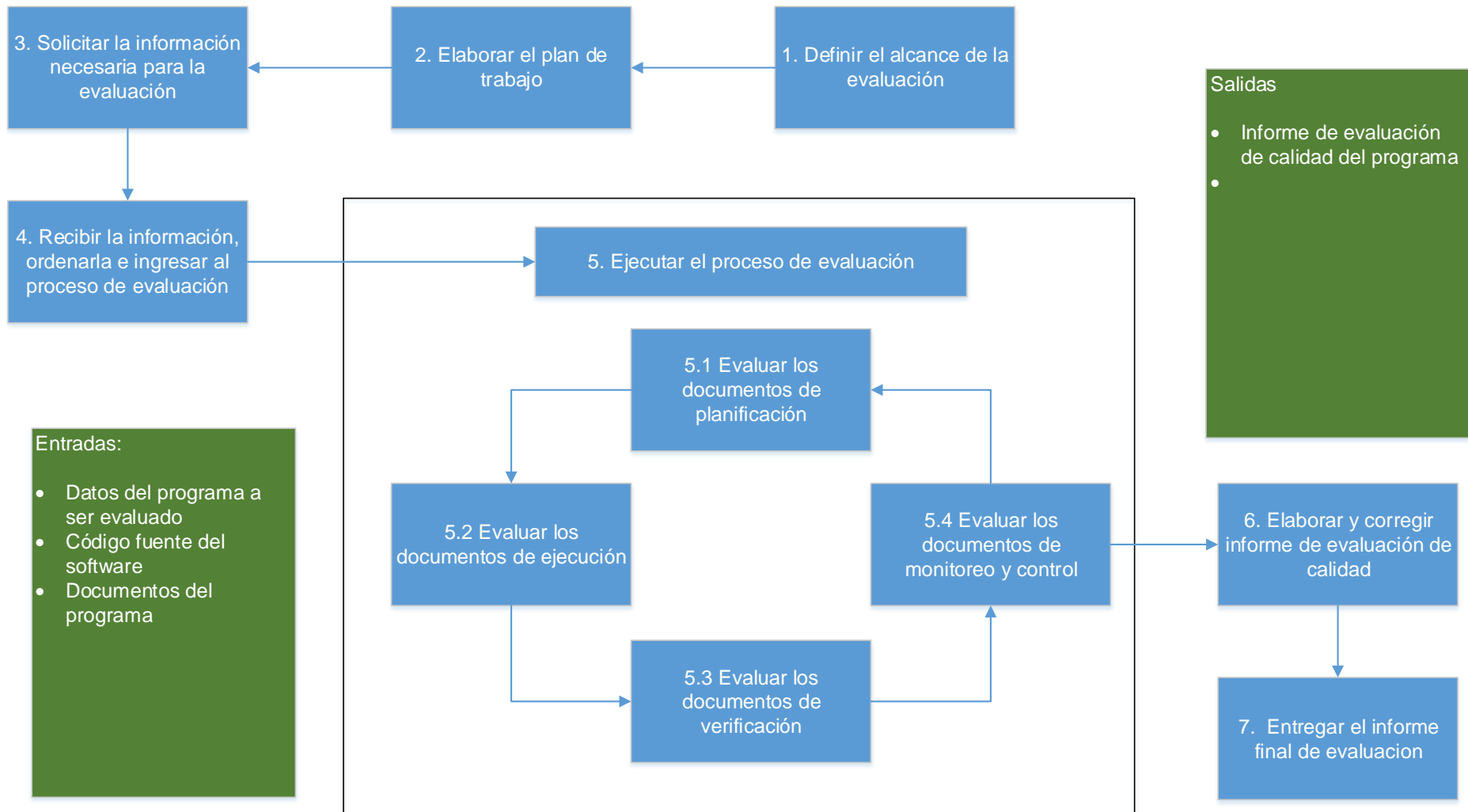


Figura 2: Metodología de evaluación de la calidad de software.



### 3.8.3 Tecnologías

- SonarQube: Es un programa el cual está desarrollado pensando en la realización de pruebas y el análisis de código en más de 28 lenguajes de programación en sus versiones más actuales, el cual cuenta con diferentes consideraciones en sus pruebas, los cuales están alineados a las métricas consideradas por los diferentes ISOS centrados en la calidad y el desarrollo de programas. (Lenarduzzi et al. 2020)
- NetData: Es una herramienta la cual se centra en la visualización y monitoreo de métricas en tiempo real, como lo pueden ser uso de CPU, labor de disco, consultas SQL, visitas y uso de un sitio web, entre otras. Cuenta con diferentes vistas para los datos siempre enfocándose en proveer la mayor cantidad de detalle posible. (Netdata - Monitor everything in real time for free with Netdata)

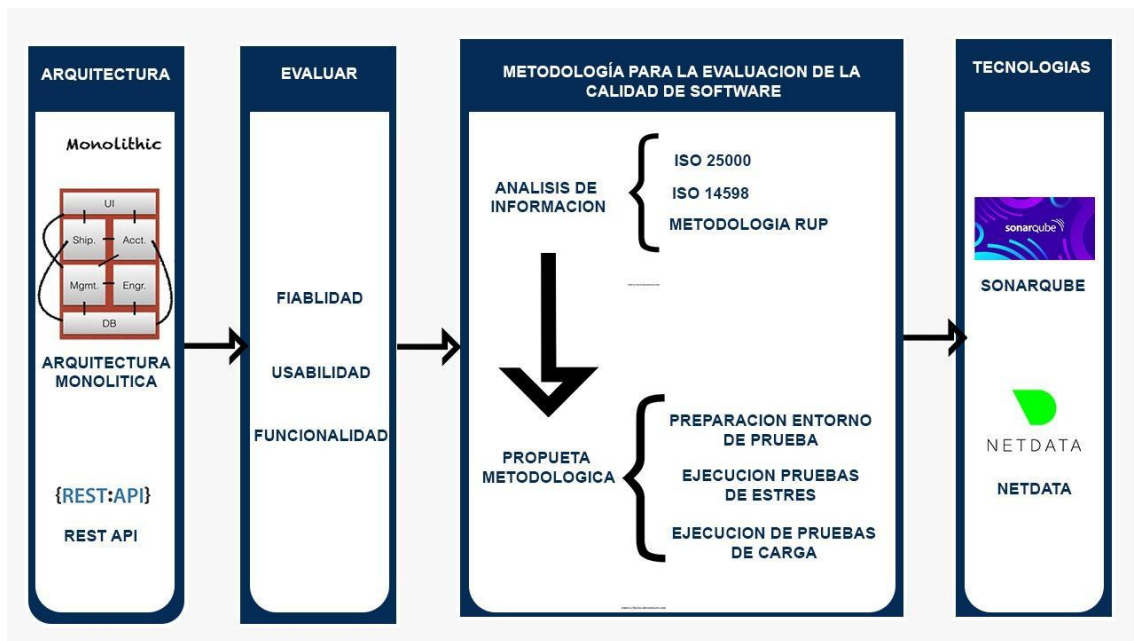


Figura 3: Esquema de Evaluaciones

### 3.8.4 Análisis de normas

A través del análisis de las Normas ISO/IEC 25000, ISO/IEC 14958 y la Metodología RUP, se tiene como objetivo comparar y determinar el aporte y enfoque que tiene la metodología propuesta del presente estudio.

Tabla 1. Análisis de la Norma ISO/IEC 25000

Norma	ISO/IEC 25000
Antecedentes	La Norma ISO/IEC 25000, está basada en las normas: ISO/IEC 9126 y ISO/IEC 14598
Objetivo	La norma tiene como fin organizar, enriquecer y unificar las series que cubren los procesos a través de requisitos de calidad y su respectiva evaluación para llegar a evaluar la calidad del software como producto.
Enfoque	Compañías y empresas que buscan desarrollar sus propias aplicaciones o programas.
Etapas	<ul style="list-style-type: none"><li>• ISO/IEC 2500n: División para gestión de la calidad.</li><li>• ISO/IEC 2501n: División para el modelo de calidad.</li><li>• ISO/IEC 2502n: División para la medición de calidad.</li><li>• ISO/IEC 2503n: División para los requisitos de calidad.</li><li>• ISO/IEC 2504n: División para la evaluación de calidad.</li></ul>
Características	<ul style="list-style-type: none"><li>• La norma ISO 25000 también tiene el nombre de SQuaRE (System and Software Quality Requirements and Evaluation) (Evaluación de Calidad de Productos Software).</li><li>• Se aplica específicamente en la creación o desarrollo de softwares tanto para ellos como para clientes externos.</li></ul>

Ventajas	<ul style="list-style-type: none"> <li>• Generan ahorro de costos ya que se controla operaciones y se reduce desperdicios y errores.</li> <li>• Satisfacción del cliente, ya que a través de las normas se puede obtener producto o servicio de mejor calidad.</li> <li>• Acceso a nuevos mercados en el extranjero ya que las Iso son reconocidas internacionalmente, otorgando prestigio a la empresa.</li> <li>• Genera una ventaja comparativa.</li> <li>• Permite a la empresa trabajar como unidad con objetivos unificados.</li> </ul>
Desventajas	<p>El soporte prestado a las empresas mayormente no concuerda con el modelo de evaluación de ISO/IEC 25000, por ello se deben adecuar.</p> <p>En caso la evaluación del software no pase, debe comenzar de nuevo todo hasta encontrar el error y repararlo.</p> <p>La refactorización del producto es muy costosa.</p>

Tabla 2. Análisis de la Norma ISO/IEC 14598

Norma	ISO/IEC 14598
Objetivo	Plantear un marco de trabajo para realizar evaluaciones a la calidad del software desarrollado como producto.
Enfoque	Se enfoca en la aplicación de administradores, desarrolladores y personas del mantenimiento dentro de una organización.
Etapas	<ul style="list-style-type: none"> <li>•ISO/IEC 14598-1: Visión general de la norma.</li> <li>•ISO/IEC 14598-2: Planeamiento y gestión.</li> </ul>

	<ul style="list-style-type: none"> <li>•ISO/IEC 14598-3: Procesos de evaluación.</li> <li>•ISO/IEC 14598-4: Procesos para compradores y proveedores de requerimientos.</li> <li>•ISO/IEC 14598-5: Procesos para avaladores.</li> <li>•ISO/IEC 14598-6: Evaluación de documentación.</li> </ul>
Características	<ul style="list-style-type: none"> <li>•Repetitividad</li> <li>•Reproducibilidad</li> <li>•Imparcialidad</li> <li>•Objetividad</li> </ul> <p><b>Medidas concretas que participan:</b></p> <ul style="list-style-type: none"> <li>•Análisis de requisitos de evaluación</li> <li>•Evaluación de especificaciones</li> <li>•Evaluación de diseño y definición del plan de evaluación</li> <li>•Ejecución del plan de evaluación</li> <li>•Evaluación de la conclusión</li> </ul>
Ventajas	<ul style="list-style-type: none"> <li>•Provee métodos y evaluaciones de calidad por producto.</li> <li>•Proporciona métricas y requisitos de evaluación certificados.</li> <li>•Realiza una evaluación de forma integral, dando una calificación del resultado final.</li> </ul>
Desventajas	<ul style="list-style-type: none"> <li>• No profundizan en los procesos, solo dan un vistazo a los procesos de forma relativa.</li> </ul>

Tabla 3. Metodología RUP

Metodología	RUP
Objetivo	Desarrollar y ordenar estructura de software, donde se tiene un conjunto de actividades para cumplir con los requisitos del usuario que requiere.
Enfoque	Administra procesos de organizaciones, control de variables para el análisis, implementación y documentación de sistemas orientados a objetos.
Fases	<ul style="list-style-type: none"> <li>• Fase de inicio: Se define el alcance del proyecto</li> <li>• Fase de elaboración: Se diseña una solución de arquitectura base del sistema.</li> <li>• Fase de desarrollo o construcción: Se completa la funcionalidad del sistema, se administra cambios de evaluaciones y se plantea mejoras.</li> <li>• Fase de transición: Se asegura la disponibilidad del software para los usuarios finales, se ajustan errores y defectos.</li> </ul>
Características	<ul style="list-style-type: none"> <li>• Incluye asignación de tiempo, hitos principales, iteraciones por fases y plan de proyecto.</li> <li>• Brinda soporte a la gestión de cambio, configuraciones, proyecto y entorno.</li> </ul>
Ventajas	<ul style="list-style-type: none"> <li>• Puede ser adaptada y ampliada para satisfacer las necesidades de cualquiera organización.</li> <li>• Describe las dimensiones, el aspecto dinámico, fases, ciclos e iteraciones de los procesos del software.</li> <li>• Facilitan la construcción de prototipos.</li> </ul>
Desventajas	<ul style="list-style-type: none"> <li>• Tiene grado de complejidad</li> <li>• En proyectos pequeños no es dable para cubrir los costos</li> <li>• Es un método pesado</li> <li>• Necesita equipos totalmente profesionales</li> </ul>

Tabla 4. Metodología Propuesta

Después de realizar el análisis a la metodología norma ISO/IEC 25000, norma ISO 14598 y la metodología RUP, se pueden describir sus limitaciones y el aporte que realiza la metodología presente respecto a ellas.

METODOLOGÍA	PROPUESTA
Objetivo	Metodología de evaluación de calidad de software como producto y sus procesos a través de análisis de requisitos de evaluación, métricas de evaluación de calidad en base a estándares internacionales.
Enfoque	Se enfoca en la aplicación de evaluación de calidad como producto y procesos de software en las MYPES Y PYMES
Etapas	Consta de 21 procesos donde se busca detallar todo tipo de pruebas de calidad y sus procesos del proyecto del software.
Características	<ul style="list-style-type: none"> <li>• Cuenta con identificación de usabilidad, seguridad, portabilidad, eficiencia y mantenibilidad.</li> <li>• Evalúa el desarrollo de procesos en base a sus actividades, planificación y medidas de control.</li> <li>• Evalúa tomando en cuenta riesgos y normas de seguridad.</li> </ul>
Ventajas	<ul style="list-style-type: none"> <li>• Realiza una evaluación más específica y profundizada de los procesos de desarrollo de software.</li> <li>• Se pueden aplicar tanto para proyectos pequeños como grandes.</li> <li>• No solo se enfocan en la calidad del proyecto sino también en la contratación y, elección de sus evaluadores de gestión, supervisión y control de calidad.</li> <li>• Busca evaluar e identificar los problemas del desarrollo y control de calidad software.</li> <li>• Se basa en desarrollar, controlar y evaluar tanto el desarrollo del software desde cero hasta la gestión y seguridad del proyecto.</li> </ul>

Tabla 5 Comparación metodológica.

ISO/IEC 25000	ISO/IEC 14598	Metodología RUP	METODOLOGÍA PROPUESTA
Tienen como base a la Norma ISO/IEC 14498	No tienen una Norma como base	No tienen una Norma como base	Busca ser una metodología más específica, basándose en las Norma ISO/IEC 25000, 14598 y Metodología RUP
Es costosa	Es costosa	Es costosa	Necesita poca inversión
La refactorización del producto es muy costosa	La refactorización del producto es muy costosa	La refactorización del producto es muy costosa	La refactorización no es muy costosa pero laboriosa.
Se aplican mayormente en grandes compañías	Se aplican mayormente en grandes compañías	Se aplican mayormente en grandes compañías	Su aplicación se centraliza en las MYPES, pero también es aplicable a grandes organizaciones.
Son difíciles de adecuar a cualquier organización	Son difíciles de adecuar a cualquier organización	Se adecuan a cualquier tipo de organización.	Se adecuan a cualquier tipo de organización.
Tienen poco grado de complejidad	Tienen poco grado de complejidad	Tienen grado de complejidad porque necesitan personal especializado.	Tienen poco grado de complejidad
Evalúa la calidad del software como producto y de forma ligera evalúa sus procesos.	Evalúa la calidad del software como producto y de forma ligera evalúa sus procesos.	Desarrolla y ordena la estructura de sus procesos.	Evalúa la calidad del software como producto y también se centraliza en analizar cada proceso.

A través del análisis y la tabla comparativa metodológica se puede determinar que la metodología propuesta; al tomar como base las normas internacionales ISO/IEC 25000, la norma ISO/IEC 14598 y la Metodología RUP; pudo establecer criterios de

evaluación de calidad , gestión, desarrollo de proyectos adecuables y accesible para empresas que no cuentan con mucha inversión y de menor alcance, generándole una ventaja competitiva dentro de su posición del mercado , el cual sería las MYPES , además que cuenta con procesos detallados, con poca complejidad, centralizados en la calidad de procesos y la calidad del software como producto.

### **3.8.5 Evaluación**

Para realizar la elección y definición de cada evaluación; se analizó los requisitos, métricas y procesos de evaluación de calidad de cada estándar I/E 25000, I/E 14598 y la metodología RUP para luego a través del ciclo de mejora continua PVHA fortalecer y crear una nueva metodología en base a los evaluaciones generales que se adecuen mejor a las PYMES y MYPES, por ello se pudo plantear las siguientes evaluaciones



Tabla 6 Elección de Evaluación

<b>N°</b>	<b>DOMINIO</b>	<b>CRITERIO</b>	<b>Métodos</b>	<b>Evaluación</b>
P001	Rendimiento	Buscar y analizar las propiedades de forma específica e íntegra del software.	ISO 25000	Evaluación de análisis de la calidad del programa mediante la calificación de las propiedades del SFTW.
P002	Rendimiento	Lograr controlar y monitorear el rendimiento del software.	ISO 25000	Evaluación de medición del rendimiento del programa por medio del análisis del comportamiento del software en producción.
P003	Procesos	Evaluar cada proceso que se desarrolla en el programa software.	ISO 25000, ISO 14598 Metodología RUP	Evaluación de calificación del proceso de elaboración del programa SFTW.
P004	Análisis externo	Buscar realizar el análisis externo e interno del contexto que maneja el programa software	ISO 25000, Metodología RUP	Evaluación del análisis del contexto de uso del programa SFTW.

P005	Rendimiento	Lograr controlar y monitorear el rendimiento del software	ISO 25000, ISO 14598	Evaluación del análisis de medidas de la efectividad del SFTW.
P006	Rendimiento	Buscar analizar las características de calidad de forma específica e íntegra del software.	ISO 25000	Evaluación de identificación de las características de la calidad del programa en función de la eficiencia.
P007	Rendimiento	Buscar analizar las características de calidad de forma específica e íntegra del software en función a la seguridad.	ISO 25000	Evaluación de identificación de las características de la calidad del programa en función de la seguridad
P008	Rendimiento	Buscar analizar las características de calidad de forma específica e íntegra del software en función a la usabilidad.	ISO 25000	Evaluación de identificación de las características de la calidad del programa en función de la usabilidad.

P009	Rendimiento	Buscar analizar las características de calidad de forma específica e íntegra del software en función a la portabilidad.	ISO 25000	Evaluación de identificación de las características de la calidad del programa en función de la portabilidad.
P010	Rendimiento	Buscar analizar las características de calidad de forma específica e íntegra del software en función al mantenimiento.	ISO 25000	Evaluación de identificación de las características de la calidad del programa en función del mantenimiento.
P011	Gestión de riesgos	Buscar identificar peligros y riesgos durante el proceso del desarrollo del programa de software.	ISO 25000	Evaluación de identificación de la libertad de riesgo del SFTW.
P012	Evaluación	Buscar el personal adecuado para llevar a cabo evaluaciones de calidad de software.	ISO 25000	Evaluación de contratación de calificador independiente.

P013	Procesos	Buscar analizar e identificar toda la estructura y esquema del programa software.	ISO 25000, ISO 14598, Metodología RUP	Evaluación de identificación del esquema de calidad del producto SFTW.
P014	Procesos	Buscar analizar e identificar toda la estructura y esquema del programa software.	ISO 14598, Metodología RUP	Evaluación de identificación del modelo de diseño del producto SFTW.
P015	Evaluación	Plantear Evaluación de estrategia y gestión de calidad del software.	ISO 14598, Metodología RUP	Evaluación de planeación y gestión de desarrollo del producto SFTW.
P016	Evaluación	Buscar formatos, guías de calificación y observaciones estándar para los procesos de evaluación del programa software	ISO 14598, Metodología RUP	Evaluación guía de calificación de producto Software para calificadores.

P017	Evaluación	Buscar formatos, guías de calificación y observaciones estándar para los procesos de evaluación del programa software	ISO 14598	Evaluación guía de codificación para desarrolladores.
P018	Evaluación	Buscar formatos, guías de calificación y observaciones estándar para los procesos de evaluación del programa software	ISO 14598	Evaluación guía de calificación de producto SFTW para compradores.
P019	Procesos	Buscar analizar e identificar toda la estructura y esquema del programa software.	Metodología RUP	Evaluación de diseño de arquitectura basada en factores.

P020	Evaluación	Plantear Evaluación de estrategia y gestión de calidad del software.	Metodología RUP	Evaluación de gestión del cambio.
P021	Evaluación	Plantear Evaluación de estrategia y gestión de calidad del software.	Metodología RUP	Evaluación de estudio de los problemas del producto SFTW.

Tabla 7 Lista de evaluaciones propuestos de la metodología

N°	Evaluación	ISO 25000	ISO 14598	Metodología RUP
P001	Evaluación de análisis de la calidad del programa mediante la calificación de las propiedades del SFTW.	x		
P002	Evaluación de medición del rendimiento del programa por medio del análisis del comportamiento del software en producción.	x		
P003	Evaluación de calificación del proceso de elaboración del programa SFTW.	x	x	x
P004	Evaluación del análisis del contexto de uso del programa SFTW.	x		x
P005	Evaluación del análisis de medidas de la efectividad del SFTW.	x	x	
P006	Evaluación de identificación de las características de la calidad del programa en función de la eficiencia.	x		
P007	Evaluación de identificación de las características de la calidad del programa en función de la seguridad	x		
P008	Evaluación de identificación de las características de la calidad del programa en función de la usabilidad.	x		
P009	Evaluación de identificación de las características de la calidad del programa en función de la portabilidad.	x		
P010	Evaluación de identificación de las características de la calidad del programa en función del mantenimiento.	x		
P011	Evaluación de identificación de la libertad de riesgo del SFTW.	x		
P012	Evaluación de contratación de calificador independiente.	x		
P013	Evaluación de identificación del esquema de calidad del producto SFTW.	x	x	
P014	Evaluación de identificación del modelo de diseño del producto SFTW.		x	x
P015	Evaluación de planeación y gestión de desarrollo del producto SFTW.		x	x
P016	Evaluación guía de calificación de producto Software para calificadores.		x	x
P017	Evaluación guía de codificación para desarrolladores.		x	

P018	Evaluación guía de calificación de producto SFTW para compradores.		x	
P019	Evaluación de diseño de arquitectura basada en factores.			x
P020	Evaluación de gestión del cambio.			x
P021	Evaluación de estudio de los problemas del producto SFTW.			x



## **IV. RESULTADOS**

El presente capítulo está dividido en 3 secciones: resultados de diagnóstico de cumplimiento de la metodología planteada, elaboración de evaluación de los requisitos planteados en la metodología que la empresa no ha implementado; y medición de los indicadores de calidad del programa como: capacidad de respuesta, disponibilidad, número de vulnerabilidades, porcentaje de detección de defectos, tiempo de instalación, tiempo de actualización, tiempo de entrega de cambios, cantidad de código, tiempo medio de reparación y tiempo medio entre fallo.

### A. Resultados del diagnóstico

A continuación, se presentan los resultados aplicando un checklist de cumplimiento de las evaluaciones de la metodología planteada.

#### P01. Evaluación de análisis de la calidad del programa mediante la calificación de las propiedades del Software.

Pregunta	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se encuentran definidos los elementos del sistema de gestión de calidad?				X
¿Existe orden en documentación del sistema de calidad?				X
¿Existe soporte del control de calidad?				X
¿El sistema cuenta con mecanismos de revisión por propiedad?				X

¿El sistema cuenta con mecanismo de control de calidad?				X
---	--	--	--	---

**P02. Evaluación de medición del rendimiento del programa por medio del análisis del comportamiento del software en producción.**

Pregunta	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se establece y define el comportamiento del software respecto a la producción de la organización?	X			
¿Se realiza análisis y diagnóstico al comportamiento del software en producción?	X			
¿Se desarrollan métricas de rendimiento de producción a partir del diagnóstico realizado?				X
¿Hay un perfil de rendimiento del sistema?				X
¿Existe un sistema de métricas establecidas según el comportamiento				X

del software en producción?				
-----------------------------	--	--	--	--

**P03. Evaluación de calificación del proceso de elaboración del programa Software.**

ETAPA: EXPLORACIÓN	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se realizan observaciones de cada uno de los procesos que se lleva a cabo?	<b>X</b>			
¿Se determinan entradas y salidas de información?				X
¿Se evalúa las vulnerabilidades y amenazas que está expuesto el sistema?				x
¿Existe análisis y evaluación de riesgos de forma preliminar?				X
<b>ETAPA 2: IMPLEMENTACIÓN</b>	Cumple			No Cumple
¿El plan de calificación tiene los objetivos, alcances, metodología, recursos y cronograma de				X

calificación bien establecidos?				
¿El programa de calificaciones cuenta con responsabilidades y actividades a desarrollar por el evaluador?				X
¿Existe método y evaluaciones de calificación del desarrollo del software?				X
¿Cuenta con herramientas de calificación como listas de chequeo, cuestionarios, entre otros?				X
¿Existe un plan de pruebas del desarrollo del software?				X
<b>ETAPA 3: EVALUACIÓN</b>	<b>Cumple</b>			<b>No Cumple</b>
¿Hay un diagnóstico final del proceso de calificación del desarrollo del software, presentando				X

informe final con las posibles mejoras o hallazgos presentados en la ejecución del proceso?				
---	--	--	--	--

**P04. Evaluación del análisis del contexto de uso del programa Software.**

Preguntas del proceso	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se elabora un inventario de agentes involucrados en el sistema?				X
¿Se Realizan diagnóstico entre las iteraciones de agentes y necesidades del contexto de uso del programa?	X			
¿Se determina la relación entre agentes y necesidades, así como los problemas con que cuenta estos y afectan al programa?				X

**P05. Evaluación del análisis de medidas de la efectividad del Software.**

ETAPA 01: EVALUAR LA EFECTIVIDAD	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se encuentran los requisitos de evaluación que se deben tener en cuenta para controlar la efectividad?				X
¿Hay métricas y criterios de evaluación de la efectividad del software?				X
¿Hay un plan y cronograma de actividades de medición?				X
¿Hay periodos de evaluación de la efectividad del software?	X			
ETAPA 02: REALIZAR EL DIAGNÓSTICO DE LA EFECTIVIDAD				

¿Los resultados de evaluación se someten a un análisis?				X
¿Se identifican los puntos débiles de la efectividad del software?				X
¿El sistema cuenta con un reporte de análisis de las medidas de efectividad del software tomando en cuenta los criterios anteriores analizados?				X

**P06. Evaluación de identificación de las características de la calidad del programa en función de la eficiencia.**

<b>ETAPA: ANÁLISIS DE LA CALIDAD DE LA INFORMACIÓN</b>	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
Cumple con los siguientes indicadores:				



Durabilidad del proceso				X
Efectividad y flexibilidad				X
Cantidad de información				X
Confiabilidad de utilidad				X
Utilización				x
Disponibilidad de reusabilidad				X
Usabilidad				X
<b>ETAPA 2: DEFINIR REQUISITOS DE CONTROL DE CALIDAD</b>	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se identifica el propósito y requisitos del software respecto a su efectividad?	X			
<b>ETAPA 3: MEDIDAS DE UTILIZACIÓN</b>	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿El proceso cuenta con herramientas que permitan evidenciar la				X

efectividad del sistema?				
ETAPA 4: SELECCIONAR Y DEFINIR CARACTERÍSTICAS	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Existe herramientas, indicadores y requisitos de control de calidad del software?				X

**P07. Evaluación de identificación de características de calidad del programa en función de la seguridad**

ID	PREGUNTA	SI	NO
S1	¿La contraseña requiere poseer 8 caracteres de identificación?		<input checked="" type="radio"/>
S2	¿La contraseña debe contener letras mayúsculas y minúsculas?		<input checked="" type="radio"/>
S3	¿La contraseña debe contener número y letras?		<input checked="" type="radio"/>
S4	¿La contraseña requiere características especiales?		<input checked="" type="radio"/>
S5	¿El sistema utiliza como medida de seguridad la conexión mediante HTTPS?	<input checked="" type="radio"/>	

S6	¿La base de datos contiene datos encriptados?		<input type="radio"/>
S7	¿Se puede acceder a las diferentes funcionalidades del sistema sin permiso?	<input type="radio"/>	
S8	¿Cualquier persona puede acceder a la base de datos?		<input type="radio"/>
S9	¿Cualquier persona puede tener acceso al código del sistema?	<input type="radio"/>	
S10	¿Cualquier persona puede acceder al servidor físico?		<input type="radio"/>
S11	¿Cualquier persona puede acceder al servidor remoto?	<input type="radio"/>	
S12	¿El sistema cuenta con redireccionamientos a sitios seguros?		<input type="radio"/>
S13	¿El sistema necesita una confirmación de registro a través del uso de correo electrónico?		<input type="radio"/>
S14	¿El sistema permite que se pueda modificar la base de datos por cualquier persona?	<input type="radio"/>	
S15	¿El sistema puede modificarse por cualquier persona a través del código del servidor de la aplicación?	<input type="radio"/>	
S16	¿El sistema permite inyecciones SQL?	<input type="radio"/>	
S17	¿El sistema cuenta con un historial de las acciones y actividades realizadas?		<input type="radio"/>
S18	¿El sistema posee algoritmos descifrados de datos?		<input type="radio"/>
S19	¿El sistema cuenta con un mecanismo de criptografía, como firma digital?		<input type="radio"/>
S20	¿El sistema requiere confirmación en el momento de realizar una acción?	<input type="radio"/>	
S21	¿El sistema posee una protección a la hora de realizar una acción?		<input type="radio"/>
S22	¿El sistema de aviso se puede acceder desde una ubicación desconocida?	<input type="radio"/>	

S23	¿El sistema informa vía mail las operaciones realizadas?		<input type="radio"/>
S24	¿El sistema tiene un registro de fecha y hora de ingreso?		<input type="radio"/>
S25	¿El sistema registra tanto el tipo de navegador como el sistema de operación que se realizan?		<input type="radio"/>
S26	¿El sistema registra la dirección de IP del navegador que ingresa al sitio?	<input type="radio"/>	
S27	¿El sistema realiza una verificación de identidad a través de un certificado digital?		<input type="radio"/>
S28	¿El sistema cuenta con un sistema de inspección y verificación los cuales consta de dos pasos?		<input type="radio"/>
S29	¿Se requiere una clave de segundo nivel para lograr ingresar al sistema?		<input type="radio"/>
S30	¿Se comprueba la identidad del usuario mediante datos biométricos?		<input type="radio"/>
S31	¿El sistema realiza una comprobación de identidad a través de tarjeta de coordenadas?		<input type="radio"/>
S32	¿Se realiza la comprobación de identidad mediante el uso de credenciales?		<input type="radio"/>
S33	¿El sistema realiza una comprobación de identidad a través de la identificación y revisión de una firma electrónica?		<input type="radio"/>

**P08. Evaluación de identificación de características de calidad del programa en función de la usabilidad**

ID	PREGUNTA	SI	NO
U1	¿El sistema se cambia de colores y se adecua según las necesidades de los usuarios?		<input type="radio"/>
U2	¿Se puede cambiar el tamaño de letra de los textos del sistema?		<input type="radio"/>
U3	¿El sistema puede realizar lectura de pantalla a través de la voz?		<input type="radio"/>

U4	¿El sistema presenta textos difíciles de comprender?		<input type="radio"/>
U5	¿El sistema tiene textos que no tengan trascendencia y sean irrelevantes dentro de sus Evaluaciones?	<input type="radio"/>	
U6	¿El sistema posee textos con faltas ortográficas?		<input type="radio"/>
U7	¿El sistema permite eliminar acciones realizadas?	<input type="radio"/>	
U8	¿El texto tiene codificación y texto en diferentes idiomas?		<input type="radio"/>
U9	¿El sistema tiene la opción de configurar el lenguaje del sistema de un idioma a otro?	<input type="radio"/>	
U10	¿La interfaz del sistema es amigable e interactúo de fácil entendimiento?		<input type="radio"/>
U11	¿El sistema puede indicar las secciones que el usuario está accediendo?	<input type="radio"/>	
U12	¿El sistema puede mostrar las áreas que se han accedido hasta el momento?		<input type="radio"/>
U13	¿Se puede realizar una acción con más de un término?		<input type="radio"/>
U14	¿El sistema cuenta con una organización de páginas en un listado?	<input type="radio"/>	
U15	¿El sistema presenta consistencia de colores en todas sus secciones?	<input type="radio"/>	
U16	¿El sistema posee errores visuales? (Como menús solapados, desplegables sin funcionar, textos en lugares no destinados a ello, etc.)	<input type="radio"/>	
U17	¿El sistema realiza operaciones y avisa mediante mensajes si se llevó con éxito/ sin éxito?	<input type="radio"/>	
U18	¿El sistema permite salir de cada sección utilizando los comandos atrás, cancelar, salir, volver, entre otros?		<input type="radio"/>
U19	¿El sistema cuenta con atajos de teclado para poder utilizar accesos para diferentes funcionalidades?		<input type="radio"/>

U20	¿El sistema cuenta con íconos de acceso a las diferentes funcionalidades?		<input type="radio"/>
U21	¿El sistema tiene medidas de ayuda, donde detalla el error ocurrido y la solución?		<input type="radio"/>
U22	¿El sistema detalla a través de un comentario si se comete un error?		<input type="radio"/>
U23	¿El sistema muestra una alarma de error para prevenir que vuelva a ocurrir el mismo acontecimiento?		<input type="radio"/>
U24	¿El sistema incide en mostrar mensajes de advertencia de error en el interfaz?		<input type="radio"/>
U25	¿El sistema muestra los requerimientos y requisitos que se necesitan en cada uno de los campos?	<input type="radio"/>	
U26	¿El sistema indica cuales son los campos que se deben llenar de forma obligatoria en el formulario?	<input type="radio"/>	
U27	¿El sistema permite ingresar caracteres o datos que difieran con las indicaciones? (Ej.: El sistema permite ingresar letras en un campo de DNI)	<input type="radio"/>	
U28	¿En el momento de completar un formulario, existe información pre cargada en algunos de sus campos? (Ej.: El campo país posee una lista desplegable con diferentes países)	<input type="radio"/>	
U29	¿Las secciones del sistema contienen indicaciones de ayuda sobre las acciones que puedan realizar en el campo?		<input type="radio"/>
U30	¿El sistema posee una sección de ayuda y descripción de las funciones?		<input type="radio"/>
U31	¿El sistema posee una sección de preguntas frecuentes?		<input type="radio"/>
U32	¿El sistema realiza comentario de pregunta, donde se detalla la pregunta si se logró resolver la inquietud de las dudas que encuentra como usuario?		<input type="radio"/>
U33	¿El sistema provee un acceso rápido de ayuda e instrucciones dirigido a los usuarios?		<input type="radio"/>

**P09. Evaluación de identificación de características de calidad del programa en función de la portabilidad**

ID	PREGUNTA	SI	NO
P1	¿El sistema cuenta con una aplicación móvil?	<input checked="" type="radio"/>	<input type="radio"/>
P2	¿El sistema logra funcionar correctamente desde cualquier ordenador con diferentes características?	<input checked="" type="radio"/>	<input type="radio"/>
P3	¿El sistema se puede utilizar adecuadamente desde un navegador como una Tablet?	<input type="radio"/>	<input checked="" type="radio"/>
P4	¿El sistema puede utilizarse desde cualquier dispositivo con cualquier sistema operativo?	<input checked="" type="radio"/>	<input type="radio"/>
P5	¿El sistema funciona correctamente desde cualquier navegador que tenga acceso a internet?	<input checked="" type="radio"/>	<input type="radio"/>
P6	¿El sistema funciona correctamente desde cualquier navegador de un dispositivo móvil?	<input checked="" type="radio"/>	<input type="radio"/>

**P010. Evaluación de identificación de las características de la calidad del programa en función del mantenimiento.**

Pregunta	Cumple	No Cumple
¿Se analiza las actividades de planificación, diseño, desarrollo, producción de toda la documentación necesaria para distintos actores del proyecto tales como director, ingeniero y usuario?	<input type="checkbox"/>	<input checked="" type="checkbox"/>

¿Se identifica el tipo de configuración, se controla y se pone como meta las fechas de calendario de entrega?	X	
¿Se verifica que se cumplan los requisitos especificados y que sigan el plan establecido?	X	
¿Se comprueba que el producto sirve para el uso proyectado?	X	
¿Se lleva a cabo revisión y dar seguimiento al sistema en función del mantenimiento?		X
¿Hay un control externo de la calidad y mantenimiento como actividades complementarias?		X

**P011. Evaluación de identificación de la libertad de riesgo del software**

<b>Pregunta</b>	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se revisa la lista de requerimientos y sus características funcionales que debe tener el software en función a su seguridad?				X
¿Se analizan los factores más determinantes y las consecuencias que pueden tener estos en el software?				X
¿Se simula todas las situaciones que puedan ocurrir a				X



partir de identificación de un posible riesgo?				
¿Se fragmenta y analiza tanto las posibilidades de riesgos, así como las decisiones que pueden acarrear a un problema que ponga en peligro la seguridad del sistema?				X

**P012. Evaluación de contratación de calificador independiente.**

<b>Pregunta</b>	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Hay una lista definida de requerimientos y conocimientos que debe contar el calificador externo?				X
¿El proceso tiene un perfil competitivo del calificador externo?				X
¿Hay un proceso de reclutamiento de personal calificado para evaluar los procesos y				X

desarrollo del software de la organización?				
¿Existe un proceso de evaluación para contratación de un calificador externo?				x

**P013. Evaluación de identificación del esquema de calidad del producto Software.**

Pregunta	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿El software cuenta con políticas de calidad identificadas?				x
¿Se conoce los requerimientos empleados en la gestión de calidad?				x
¿Se tienen definidos los criterios de control de calidad de los procesos del sistema?				x
¿Se realiza una inspección de calidad del sistema tomando como referencia las				x

normas de la ISO 9004-2?				
¿Se realiza la revisión y control de procesos de calidad?				x
¿Los indicadores de calidad del sistema están bien identificados?				x

**P014. Evaluación de identificación del modelo de diseño del producto Software.**

Proceso	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se identifican las funciones del modelo?	X			
¿Se analizan las funciones sobre el diseño del sistema?	X			
¿Se revisan los requerimientos, objetos y especificaciones técnicas de los procesos?	X			
¿Se analiza la estructura de las funciones y los modificadores funcionales??	X			

¿Se definen los diferentes estados por los que pasa el sistema?				X
¿Se define el diseño conceptual a partir de los resultados de los pasos anteriores ?	X			
¿Se selecciona y define el modelo del diseño del proceso?	X			

**P015. Evaluación de planeación y gestión de desarrollo del producto Software.**

Proceso	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se definen criterios de evaluación?				X
¿Los desarrolladores del diseño, contestan el cuestionario y producen un informe de evaluación?				X

¿Se estudia la precisión de los hallazgos?				X
¿Hay una guía de calificación a partir del método implementado en el proceso de evaluación?				X

**P016. Evaluación guía de calificación de producto Software para calificadores.**

Proceso	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se define criterios de evaluación?				X
¿Los desarrolladores del diseño, contentan el cuestionario y producen un informe de evaluación?				X
¿Se estudia la precisión de los hallazgos?				X
¿Hay una guía de calificación a				X

partir del método implementado en el proceso de evaluación?				
---	--	--	--	--

**P017. Evaluación guía de codificación para desarrolladores**

Proceso	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
Diseño: la organización debe establecer un diseño consistente de desarrollo de código de software que evite errores innecesarios				X
El código fuente debe utilizar sangría con espacios en lugar de tabulaciones, se recomienda que la sangría debe ser de 4 espacios para evitar problemas de lectura.	X			
Limite la longitud de las líneas de código fuente debe ser de 55 caracteres				X

Limitar la longitud (número de líneas) de las cláusulas a 24 líneas.				X
Ser consistente en el uso del espacio entre las comas, cada coma tiene un espacio				X
Poner cada subobjetivo en una línea separada				X
El espacio vertical está forma coherente para mejorar la legibilidad				X
Comente los archivos de origen, no solo los predicados dentro de ellos				X
Utilice el diseño para hacer que los comentarios sean más legibles	X			
Anular los comentarios a la derecha del código, a menos que sean inseparables de las líneas en las que aparecen	X			

Considere usar los comentarios de manera constante como recordatorios				X
Hacer que las cláusulas sean comprensibles de forma aislada				X
Sangría de un nivel adicional entre la repetición y el corte correspondiente	X			
Decida cómo quiere romper las cláusulas largas y los subobjetivos				X
Decidir cómo formatear disyunciones y if-then-elses	X			
Considere el uso de herramientas automatizadas para mejorar la legibilidad				X

**P018. Evaluación guía de calificación de producto software para compradores**

Proceso	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
---------	-------------------------	------------------------------	------------------	----------------



¿Los criterios de calificación son proporcionados y adecuados al tipo, naturaleza, condiciones del mercado, complejidad, riesgo, valor y objetivo de la contratación?	X			
¿Los criterios de calificación son cuantificables?				X
¿Los criterios de calificación completos y la forma específica se aplican?				X
¿Los criterios de calificación se aplicarán de manera coherente a todas las Ofertas / Propuestas presentadas?				X
¿La calificación del costo utilizan una metodología adecuada a la naturaleza de la contratación?				X

<p>¿Existe una metodología para determinar el grado en que los bienes, obras, servicios distintos de consultoría o servicios de consultoría cumplen o superan los requisitos?</p>				<p>x</p>
<p>¿Hay criterios que mitigan el riesgo evaluado relevante?</p>				<p>x</p>
<p>¿El sistema cuenta con criterios que permitan evaluar la innovación en el diseño y / o entrega de los bienes, obras, servicios de no consultoría o servicios de consultoría y que brinden a los licitantes / proponentes la oportunidad de incluir, cuando corresponda, en sus Ofertas / Propuestas.?</p>				<p>x</p>

<p>¿Los criterios de calificación (aprobado / reprobado) se establecen de tal manera que una evaluación pueda determinar si la Oferta / Propuesta responde sustancialmente a los requisitos técnicos y comerciales?</p>				<p>x</p>
<p>¿La evaluación de las calificaciones de un solicitante lo evalúa la empresa que presentó la solicitud de precalificación / selección inicial?</p>				<p>x</p>
<p>¿Las puntuaciones que se otorgarán a cada criterio y subcriterio se especifican en la hoja de datos? criterios calificados se puede utilizar una matriz de priorización simple?</p>				<p>x</p>

**P019. Evaluación de diseño de arquitectura basada en factores.**

Proceso	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
Extraer las inquietudes de las partes interesadas por medio de entrevistas, la lectura de documentos y las experiencias personales				X
Las inquietudes pueden estar muy relacionadas con las partes interesadas y puede ser beneficioso filtrar las inquietudes para que solo los hechos de las inquietudes entren en el análisis y no la relación con las partes interesadas				X
Para cada cambio en las preocupaciones, el arquitecto de software debe satisfacer nuevos atributos de calidad del software que debe lograr en el sistema con respecto a la	X			

arquitectura del software				
Los factores de influencia son parte de las preocupaciones de las partes interesadas e incluyen tendencias, entorno técnico, experiencias previas y demandas del mercado, etc				X
Para ser proactivo, el arquitecto debe analizar continuamente las preocupaciones cambiantes y no solo esperar a que las preocupaciones se transformen en requisitos funcionales y no funcionales				X
priorizar los factores que influyen en la que se identifican los factores de influencia y sus influencias en los objetivos comerciales y los atributos de calidad del software se documentaron. En lo que se extraen aquellos	X			

factores que influyen que tienen un impacto positivo en el objetivo comercial priorizado				
Los factores de influencia deben agruparse en torno a un impacto positivo en un objetivo empresarial específico. En este caso, el arquitecto puede intentar verificar con la dirección que este objetivo empresarial específico es el priorizado en la organización				X

**P020. Evaluación de gestión del cambio.**

Proceso	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se evalúa el cambio, incluida la determinación del nivel de prioridad del servicio y el riesgo del cambio propuesto?				x
¿La gestión cuenta con plan de comunicación, plan de prueba y el plan de reversión?				x

¿Hay comunicación y el plan de cambio con sus compañeros y / o la Junta Asesora de Cambio con respecto a su éxito o fracaso y si el cambio resultó en una falla en el servicio?				x
¿El cambio se realiza correctamente y el cambio se cierra respecto a su gestión?				x

**P021. Evaluación de estudio de los problemas del producto software**

Proceso	Implementado totalmente	En proceso de implementación	En planteamiento	Sin planificar
¿Se tiene identificado el tipo y sus elementos del dominio?	X			
¿El comportamiento siempre esta evaluado según la respuesta de información, piezas de trabajo y pantalla de información?				x
¿Hay un diagnóstico de problemas del producto?	X			
¿Cuenta con un enfoque realista que identifica los problemas de	X			

visualización de información?				
-------------------------------	--	--	--	--

## B. Evaluación y formatos de registros elaborados para la empresa según lo indica la metodología

En este punto se presentan las evaluaciones y formatos que se ha implementado en la organización para que esta pueda cumplir con lo indicado en la metodología.

	<b>Metodología de evaluación de la calidad del producto de software</b>			
	<b>N°:</b>	<b>01</b>		
	<b>EVALUACIÓN:</b>	<b>Evaluación de análisis de la calidad del programa mediante la calificación de las propiedades del Software.</b>		
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b>	<input type="checkbox"/>
<b>Evaluación de análisis de la calidad del programa mediante la calificación de las propiedades del Software.</b>	<b>Fecha:</b>			
	<b>Versión: 1</b>			
	<b>Responsable:</b> <b>Jefe de control de calidad del software</b>			

### 1. Objetivo:

Este Evaluación busca analizar la calidad del sistema de la empresa a partir de la calificación debida de las propiedades del software.

### 2. Alcance:

La evaluación pretende alcanzar el seguimiento permanente de la calidad del programa a través de la calificación de sus propiedades.

### 3. Responsables:

Jefe de control de calidad de los procesos del software, equipo de supervisión de calidad del software.

### 4. Actividades



N°	Actividad	Descripción	Responsable
1	Diagnóstico de la calidad del software	Se realiza un análisis de calidad del software, para ello se define sus objetivos, propiedades, así como sus mecanismos actuales y diferentes problemas que tenga.	Equipo de supervisión de calidad del software.
2	Calificación de propiedades	A través del diagnóstico se determina las propiedades que rigen el sistema de la empresa, para proceder a calificarlas	Equipo de supervisión de calidad del software.
3	Reporte de calidad del software	Se realiza el reporte de la calidad mediante las calificaciones de las propiedades del sistema y su impacto en el software.	Jefe de procesos de calidad del software.

### 5. Documentación:

- Formato de análisis de calidad del programa

<b>Form 01</b>	<b>Evaluación de análisis de la calidad del programa mediante la calificación de las propiedades del Software.</b>
<b>Formato de análisis de calidad</b>	
Generalidades	Instrucción 1: Se debe definir los objetivos y propiedades del sistema.  <div style="border: 1px solid black; width: 100%; height: 30px;"></div>
Estándares	Instrucción 2: Comparar los sistemas de control de calidad actuales y propiedades con la ISO 9001  <div style="border: 1px solid black; width: 100%; height: 30px;"></div>

Diagnóstico de propiedades	Instrucción 3: Realizar el diagnóstico de las propiedades del sistema.  <input type="text"/>
Mecanismo de control de calidad	Instrucción 4: Se debe incorporar métricas de calidad al sistema
Evaluación	Instrucción 5: Se debe calificar las propiedades, métricas, sistemas de control y el impacto que repercuten en la calidad del software.  <input type="text"/>

- Formato de calificación de propiedad respecto a la calidad del sistema.

Propiedad:		
Descripción general:	Imagen de la herramienta:	
<input type="text"/>	<input type="text"/>	
Seguimiento al proceso	Fase: <input type="text"/>	Apoyo:
	Impacto en la calidad: <input type="text"/>	Métrica de control: <input type="text"/>
	Calificación:	

	<div style="border: 1px solid black; width: 100px; height: 30px; margin: 0 auto;"></div>
--	--

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>02</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación de medición del rendimiento del programa por medio del análisis del comportamiento del software en producción.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b>
<b>Evaluación de medición del rendimiento del programa por medio del análisis del comportamiento del software en producción</b>	<b>Fecha:</b>		
	<b>Versión: 1</b>		
	<b>Responsable: Supervisor de control del desarrollo del software.</b>		

**1. Objetivo:**

Esta evaluación busca determinar el rendimiento del programa a través del análisis del comportamiento del software en producción.

**2. Alcance:**

La evaluación pretende medir el rendimiento del programa a partir de métricas de control del comportamiento del software en producción.

**3. Responsables:**

Equipo de control de producción, supervisor del desarrollo del software.

**4. Actividades**

N°	Actividad	Descripción	Responsable

1	Diagnóstico del rendimiento del programa	del del	A través de políticas de control, métricas y el análisis del rendimiento se determina la situación actual del programa respecto a su comportamiento en producción.	Supervisor del desarrollo del software.
2	Análisis del comportamiento del software en producción.	del	Se debe identificar el comportamiento del software respecto a producción, sus problemáticas de rendimiento, así como su nivel.	Equipo de control de producción
3	Realización de métricas de rendimiento en producción.	de de de	Después del diagnóstico del rendimiento del programa y el comportamiento del software en producción se realiza las métricas que permita medir el rendimiento respecto al comportamiento en producción.	Supervisor del desarrollo del software.
4	Realización de documentación de control	de de	Realizar una lista de check de las métricas que debe cumplir según el perfil establecido con relación al comportamiento del software en producción, así como también formatos de rendimiento.	Supervisor del desarrollo del software.

## 5. Documentación

- Formatos de seguimiento al rendimiento del software con relación del comportamiento del sistema en producción.

Form 02:	Evaluación de medición del rendimiento del programa por medio del análisis del comportamiento del software en producción.
Seguimiento al rendimiento del software	

Fase	Completo	Proceso- Etapa actual	Incompleto	Fecha
Delimitación de fechas de medición del rendimiento del software:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis al rendimiento de los procesos del software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Calificación de métricas actuales del rendimiento:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diagnóstico final del rendimiento del software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Formato de calificación de métricas de rendimiento del software respecto a su nivel de producción.			Número de Evaluación: 2	
Registro de calificación				
Métrica a calificar				
Características identificadas:				
Nivel de impacto que general al sistema de producción del software:				
Calificación total:				

	<b>Metodología de evaluación de la calidad del producto de software</b>
--	---

	N°:	03		
	EVALUACIÓN:	Evaluación de calificación del proceso de elaboración del programa Software.		
	COMPLETO	<input type="checkbox"/>	INCOMPLETO	<input type="checkbox"/>
Evaluación de calificación del proceso de elaboración del programa Software.		Fecha:		
		Versión: 1		
		Responsable: Equipo de control de procesos del software		

**1. Objetivo:**

El objetivo de este proceso es mostrar el nivel con el que se desarrolla el software a través de la calificación del proceso.

**2. Alcance:**

Este proceso pretende dar seguimiento y evaluación de los procesos durante el desarrollo de un programa.

**3. Responsables:**

Supervisión de procesos del desarrollo de software.

**4. Actividades**

N°	Actividad	Descripción	Responsable
1	Definición y detalle de procesos	Se define todos los procesos y sus respectivas actividades durante todo el desarrollo de software.	Equipo de control de producción
2	Diagnóstico de procesos.	Se realiza el análisis de cada proceso, elaborando un diagnóstico de cada uno de ellos.	Equipo de control de producción
2	Identificación de riesgos y problemas de los procesos.	A través del diagnóstico se identifica los diferentes riesgos y problemas de los procesos.	Equipo de control de producción

3	Desarrollar plan de trabajo y registro	Se debe desarrollar un plan y registro de trabajo del proceso para calificar los Evaluaciones respectivos durante el desarrollo del software.	Equipo de control de producción
4	Diagnóstico de calificación	Se analiza la calificación del registro y plan de trabajo para poder determinar un diagnóstico de la evaluación	Supervisor del desarrollo del software.

### 5. Documentación

- Plan de trabajo
- Registro de calificación de procesos.

Registro de actividades de plan de trabajo				
Fase	Completo	Proceso- Etapa actual	Incompleto	Fecha
Delimitación de fechas del desarrollo del software.				
Realización de actividades preliminares al desarrollo del software				
Actividades del desarrollo del software				
Disposición de recursos				

Form 03	Evaluación: Evaluación de calificación del proceso de elaboración del programa software.
Plan de trabajo	
1. Fechas inicio-fin	Se plantea las fechas de inicio y fin del proyecto según los promedios con el historial que cuentan de desarrollo de softwares de la empresa.
2. Descripción y actividades preliminares del plan de trabajo	Elaborar el plan de trabajo de la implementación del software, donde se defina el proveedor y equipo de trabajo, además debe contar con las siguientes actividades preliminares: <ul style="list-style-type: none"> <li>- Realizar las especificaciones del diseño físico y tipo de datos</li> <li>- Especificación y requerimientos de procesos.</li> <li>- Especificación y requerimientos del servicio.</li> <li>- Especificación y requerimientos de modelo de negocio del sistema</li> <li>- Definir políticas del modelo de negocio del software a desarrollar.</li> </ul>
3. Actividades a calificar durante el proceso.	<ul style="list-style-type: none"> <li>- Diseño del modelo del software</li> <li>- Programación de base de datos</li> <li>- Desarrollo de prototipo</li> <li>- Programación de actividades en sistema real</li> <li>- Programación de aplicaciones del cliente</li> <li>- Etapas de pruebas</li> <li>- Etapas de documentación</li> <li>- Implantación y prueba en marcha</li> </ul>
4. Recursos	Jefe del proyecto a desarrollar y equipo de trabajo
5. Restricciones	Definir limitaciones que pueda tener el proyecto, como el tiempo para el desarrollo del desarrollo del software.



	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>04</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación del análisis del contexto de uso del programa Software.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación del análisis del contexto de uso del programa Software.</b>		<b>Fecha:</b>	
		<b>Versión: 1</b>	
		<b>Responsable: Analista de diseño del sistema</b>	

**1. Objetivo:**

El objetivo de este proceso es realizar el análisis correspondiente al contexto de uso del programa software.

**2. Alcance:**

Este proceso pretende identificar el contexto de uso del sistema e identificar todos los criterios de análisis que sean necesarios.

**3. Responsables:**

Analista de diseño del sistema.

**4. Actividades**

N°	Actividad	Descripción	Responsable
1	Describir el tipo de contexto de uso.	Identificar las características del contexto de uso, a través de inventario de agentes involucrados.	Analista de diseño del sistema.
2	Definir el modelo de contexto de uso	Se identifica el tipo de modelo de contexto de uso, las características y propiedades con la cual cuenta.	Analista de diseño del sistema.

3	Desarrollar formato de análisis de contexto de uso	Se elabora formato de análisis donde se definirá tantos agentes, necesidades, número de iteraciones, entre otros.	Analista de diseño del sistema.
4	Diagnosticar caso de uso del contexto:	A partir de los análisis e identificación del modelo de caso de uso del contexto se debe determinar la relación entre agentes y necesidades, así como los problemas con que cuenta estos y afectan al programa.	Analista de diseño del sistema.

### 5. Documentación

- Formato de análisis del contexto de caso de uso del sistema.

<b>Formato de análisis</b>			
<b>Evaluar</b>	<b>Descripción</b>	<b>Aplicación</b>	<b>Diagnóstico</b>
Modelo de contexto de uso	Se define el modelo de contexto de uso:		
Agentes del sistema	Se identifica los agentes involucrados y el número de iteraciones		
Necesidades del contexto de uso	Se identifica las necesidades del contexto de uso del sistema		
Problemas del contexto de uso	Se identifica los problemas y determina posibles causas.		

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>05</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación del análisis de medidas de la efectividad del Software.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación del análisis de medidas de la efectividad del Software.</b>		<b>Fecha:</b>	
		<b>Versión: 1</b>	
		<b>Responsable: Analista de diseño del sistema</b>	

**1. Objetivo:**

El Evaluación tiene como objetivo determinar la efectividad del software a través del análisis de sus medidas.

**2. Alcance:**

Este proceso pretende analizar todas las medidas de efectividad con la que cuente el sistema.

**3. Responsables:**

Supervisor de control de calidad del software, equipo de control de calidad.

**4. Actividades**

N°	Actividad	Descripción	Responsable
1	Realizar plan de trabajo de la evaluación de efectividad	Se debe evaluar los requisitos necesarios para medir la efectividad, las métricos y criterios, así como también el cronograma de actividades.	Equipo de control de calidad.
2	Diagnóstico de la efectividad actual del software	Se debe analizar los resultados de efectividad encontrados a partir del plan de aplicación y detallar la situación actual de su rendimiento, así	Supervisor de control de calidad del software.

		como también sus diferentes problemas.	
3	Elaborar formato de análisis de la efectividad del software.	Se debe crear un formato de evaluación de la efectividad donde se realizará el análisis de la efectividad según el diagnóstico preeliminar realizado.	Equipo de control de calidad.
4	Elaborar diagnóstico final del análisis de la efectividad.	Después de analizar y calificar la efectividad según las métricas y diferentes criterios de medición se detallará la situación actual de efectividad del sistema.	Supervisor de control de calidad del software.

## 5. Documentación

- Plan de trabajo

<b>Form 05</b>	<b>Evaluación del análisis de medidas de la efectividad del Software.</b>
<b>Plan de trabajo</b>	
Definir fechas	Se debe definir las fechas de inicio y fin de la aplicación.
Planificar actividades	Se debe realizar el cronograma de evaluaciones de la efectividad del software
Definir requisitos	Se debe determinar las métricas y criterios empleadas para medir la efectividad.
Realizar formato de análisis de efectividad	Se debe realizar un formato de análisis de la efectividad para evaluar la efectividad del software a través de las métricas y criterios identificados anteriormente.

Diagnóstico del análisis de la efectividad	A partir de evaluación de efectividad del software, se puede determinar la situación actual y sus diferentes problemáticas que puedan tener que afecte al sistema.
--	--

- Formato de análisis de efectividad

<b>Form 05</b>	<b>Evaluación del análisis de medidas de la efectividad del Software.</b>
<b>Formato de análisis de efectividad</b>	
01	Determinar el Nivel de efectividad actual: <input type="text"/>
02	Evaluar la efectividad según su métrica y criterio de control: <input type="text"/>
03	Revisar los resultados y determinar sus diferentes problemáticas y causas <input type="text"/>
04	Determinar la situación actual de efectividad según sus herramientas de control: <input type="text"/>

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>06</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación de identificación de las características de la calidad del programa en función de la eficiencia.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación de identificación de las características de la calidad del programa en función de la eficiencia.</b>	<b>Fecha:</b>		
	<b>Versión: 1</b>		
	<b>Responsable: jefe de control de calidad de los procesos del software.</b>		

**1. Objetivo:**

Este Evaluación tiene como finalidad identificar las características de la calidad de la eficiencia.

**2. Alcance:**

El Evaluación pretende alcanzar a realizar las actividades para identificar las características de la calidad del programa que están en función de la eficiencia.

**3. Responsables:**

Jefe de control de calidad de los procesos del software, equipo de control de calidad del software.

**4. Actividades**

N°	Actividad	Descripción	Responsable
1	Revisar el diagnóstico de efectividad del sistema.	Revisar el diagnóstico de la efectividad del sistema del proceso N°5 e identificar las características de calidad que cuenta.	Equipo de control de calidad.
2	Analizar las características identificadas	Se debe evaluar las características identificadas y determinar el impacto que generan en la eficiencia del sistema.	Supervisor de control de calidad del software.

3	Elaborar formato de indicadores	Se debe crear un formato de indicadores para emplearlos como mecanismo de control de la repercusión de las características referente a la eficiencia del sistema.	Equipo de control de calidad.
---	---------------------------------	---	-------------------------------

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	N°:	07	
	<b>EVALUACIÓN:</b>	Evaluación de identificación de características de calidad del programa en función de la seguridad.	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
Evaluación de identificación de características de calidad del programa en función de la seguridad.	Fecha:		
	Versión: 1		
	Responsable: jefe de control de calidad de los procesos del software.		

### 1. OBJETIVO

El presente Evaluación busca determinar las características de calidad de seguridad del programa software.

### 2. ALCANCE

Pretende alcanzar analizar e identificar las características de calidad en función a la seguridad que este cuenta.

### 3. RESPONSABLE

Jefe de control de calidad de los procesos del software y equipo de control de procesos.

### 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Realizar evaluación de la seguridad	Se debe evaluar el nivel de seguridad con que cuenta el software a través de la metodología del Anexo 20 y evaluar la aplicación del software respecto a su seguridad.	Equipo de control de calidad.
2	Diagnóstico del nivel de seguridad	El resultado de dicha evaluación se calcula en porcentaje, considerando el total de aspectos a cumplir dividido entre el total de aspectos, determinando así la situación actual del nivel de seguridad	Supervisor de control de calidad del software.
3	Identificar las características de calidad	Después de elaborar la evaluación y realizar el diagnóstico se procede a identificar sus características más notables respecto a su calidad de seguridad que cuenta.	Equipo de control de calidad.

## 5. Documentación

- Evaluación del nivel de seguridad del sistema (AnexoN°20)
- Formato de Diagnóstico de la calidad del software respecto a su seguridad.

Formato de Diagnóstico de la calidad del software respecto a su seguridad.	Número de Evaluación: 7
Registro de actividades de plan de trabajo	
Puntaje de evaluación:	
Criterios que cumple el proceso:	



Características identificadas:	
Nivel de impacto que general al sistema de seguridad del software:	

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>08</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación de identificación de características de calidad del programa en función de la usabilidad.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación de identificación de características de calidad del programa en función de la usabilidad.</b>	<b>Fecha:</b>		
	<b>Versión: 1</b>		
	<b>Responsable: jefe de control de calidad de los procesos del software.</b>		

### 1. OBJETIVO

El presente Evaluación busca identificar las características de calidad del sistema, pero a través de función de usabilidad.

### 2. ALCANCE

Pretende alcanzar analizar la calidad del programa en función de su usabilidad y poder identificar todas sus características, para ello desea evaluar su usabilidad del sistema.

### 3. RESPONSABLE

Jefe de control de calidad de los procesos del software y equipo de control de procesos.

### 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Realizar evaluación de usabilidad del sistema.	Se debe evaluar el nivel de usabilidad con que cuenta el sistema a través de la metodología del Anexo 21.	Equipo de control de calidad.
2	Diagnóstico del nivel de usabilidad	El resultado de dicha evaluación se calcula en porcentaje, considerando el total de aspectos a cumplir dividido entre el total de aspectos, determinando así la situación actual del nivel de usabilidad.	Supervisor de control de calidad del software.
3	Identificar las características de calidad	Después de elaborar la evaluación y realizar el diagnóstico se procede a identificar sus características más notables respecto a su calidad de usabilidad que cuenta.	Equipo de control de calidad.

## 5. Documentación

- Evaluación del nivel de usabilidad del sistema (AnexoN°21)
- Formato de Diagnóstico de la calidad del software respecto a su usabilidad.

Formato de Diagnóstico de la calidad del software respecto a su seguridad.	Número de Evaluación: 8
Registro de actividades de plan de trabajo	
Puntaje de evaluación:	
Criterios que cumple el proceso:	
Características identificadas:	
Nivel de impacto que general al sistema de usabilidad del software:	

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>09</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación de identificación de características de calidad del programa en función de la portabilidad.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación de identificación de características de calidad del programa en función de la portabilidad.</b>	<b>Fecha:</b>		
	<b>Versión: 1</b>		
	<b>Responsable:</b> <b>Jefe de control de calidad de los procesos del software.</b>		

### 1. OBJETIVO

El presente Evaluación busca identificar las características de calidad del sistema, pero a través de función de portabilidad.

### 2. ALCANCE

Pretende alcanzar analizar la calidad del programa en función de su portabilidad y poder identificar todas sus características, para ello desea evaluar su portabilidad del sistema.

### 3. RESPONSABLE

Jefe de control de calidad de los procesos del software y equipo de control de procesos.

### 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Realizar evaluación de portabilidad del sistema.	Se debe evaluar el nivel de portabilidad con que cuenta el sistema a través de la metodología del Anexo 22.	Equipo de control de calidad.

2	Diagnóstico del nivel de portabilidad	El resultado de dicha evaluación se calcula en porcentaje, considerando el total de aspectos a cumplir dividido entre el total de aspectos, determinando así la situación actual del nivel de portabilidad.	Supervisor de control de calidad del software.
3	Identificar las características de calidad	Después de elaborar la evaluación y realizar el diagnóstico se procede a identificar sus características más notables respecto a su calidad de portabilidad que cuenta.	Equipo de control de calidad.

### 5. Documentación

- Evaluación del nivel de usabilidad del sistema (Anexo N°22)
- Pruebas de la aplicación del software respecto a su usabilidad.
- Formato de Diagnóstico de la calidad del software respecto a su portabilidad.

Formato de Diagnóstico de la calidad del software respecto a su seguridad.		Número de Evaluación: 9	
Registro de actividades de plan de trabajo			
Puntaje de evaluación:			
Criterios que cumple el proceso:			
Características identificadas:			
Nivel de impacto que general al sistema de portabilidad del software:			
		<b>Metodología de evaluación de la calidad del producto de software</b>	
		<b>N°:</b>	<b>10</b>
		<b>EVALUACIÓN:</b>	<b>Evaluación de identificación de las características de la calidad del</b>

		<b>programa en función del mantenimiento.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación de identificación de las características de la calidad del programa en función del mantenimiento.</b>		<b>Fecha:</b>	
		<b>Versión: 1</b>	
		<b>Responsable: jefe de control de calidad de los procesos del software y equipo de control de mantenimiento de los procesos.</b>	

### 1. OBJETIVO

El presente Evaluación busca identificar las características de calidad del sistema, pero a través de función de su mantenimiento.

### 2. ALCANCE

Pretende alcanzar analizar la calidad del programa en función de su mantenimiento y poder identificar todas sus características, para ello desea evaluar el proceso de mantenimiento del sistema.

### 3. RESPONSABLE

Jefe de control de calidad de los procesos del software y equipo de control de mantenimiento de los procesos.

### 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Realizar plan de trabajo de análisis de mantenimiento del sistema.	Se debe elaborar un plan de trabajo de análisis del mantenimiento de los procesos, donde se debe enfocar tanto su sistema de planificación como de control.	Equipo de control de mantenimiento de los procesos.

2	Diagnóstico del nivel de mantenimiento	Se debe determinar el nivel de calidad de los procesos de mantenimiento del sistema.	Supervisor de control de calidad del software.
3	Identificar las características de calidad	Después de elaborar el diagnóstico se procede a identificar sus características más notables respecto a su calidad de mantenimiento que cuenta.	Equipo de control de calidad.

## 5. Documentación

- Formato de plan de trabajo del análisis de mantenimiento de los procesos del sistema.

PLAN DE TRABAJO	
Delimitar fechas inicio-fin	El proceso debe contar con cronograma de fechas y un parámetro de duración del mantenimiento
Programación de actividades	<ul style="list-style-type: none"> <li>- Análisis del sistema de planificación de mantenimiento.</li> <li>- Se analiza y estudia el diseño y como este va cambiando.</li> <li>- Se realiza la documentación correspondiente</li> <li>- Se da seguimiento al control de configuración y fechas de entrega de mantenimiento del software.</li> </ul>
Revisión de seguridad de calidad	-Inspección del programa de actividades e identificación de sus características más notables
Diagnóstico de calidad	<ul style="list-style-type: none"> <li>- Detalla las características de calidad de las actividades de mantenimiento, así como sus posibles problemáticas.</li> </ul>

## 1. OBJETIVO

El presente Evaluación busca identificar los riesgos que tienen el sistema informático y cuál es su impacto.

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>11</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación de identificación de la libertad de riesgo del software.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación de identificación de la libertad de riesgo del software.</b>	<b>Fecha:</b>		
	<b>Versión: 1</b>		
	<b>Responsable:</b> <b>Supervisor de seguridad y riesgos del sistema</b>		

## 2. ALCANCE

El alcance pretende diagnosticar los riesgos que está teniendo a cabo dentro de su ciclo de vida el software diseñado.

## 3. RESPONSABLE

Supervisor de seguridad y riesgos del sistema.

## 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Supervisión del sistema actual de riesgos del sistema.	Se identifica o analiza si es que el software cuenta con un sistema de riesgos definidos, en caso contrario se analiza los factores determinantes, características que ponen en peligro la seguridad del programa informático.	Supervisor de seguridad y riesgos del sistema.

2	Análisis del sistema a su nivel de riesgos.	Se debe fragmentar y analizar todas las posibilidades de riesgos, así como las decisiones que pueden acarrear a un problema que ponga en peligro la seguridad del sistema.	Supervisor de seguridad y riesgos del sistema.
3	Identificar los riesgos.	De los pasos anteriores se selecciona los riesgos encontrados a partir de cada parámetro que se tomó en cada proceso.	Supervisor de seguridad y riesgos del sistema.

### 5. Documentación

- Registro de análisis de trabajo.

N°	Actividad	Descripción	Responsable
1			
2			
3			

	<b>Metodología de evaluación de la calidad del producto de software</b>			
	<b>N°:</b>	<b>12</b>		
	<b>EVALUACIÓN:</b>	<b>Evaluación de contratación de calificador independiente.</b>		
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b>	<input type="checkbox"/>
<b>Evaluación de contratación de calificador independiente.</b>	<b>Fecha:</b>			
	<b>Versión: 1</b>			



	<b>Responsable:</b>  <b>Jefe de recursos humanos y jefe de área de informática de la empresa.</b>
--	---

### 1. OBJETIVO

El objetivo de esta Evaluación es definir la secuencia de pasos para contratar a un personal calificado.

### 2. ALCANCE

Este proceso pretende determinar la mejor elección del personal contratado que ocupará el puesto de calificador.

### 3. RESPONSABLE

Jefe de recursos humanos y jefe de área de informática de la empresa.

### 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Definir el perfil competitivo	Definir y detallar las funciones de las competencias profesionales, responsabilidades y decisiones que va a tomar el calificador externo de software a contratar	Jefe de área de informática de la empresa.
2	Reclutamiento y preselección	Se anuncia la oferta de trabajo y se filtra los curriculum según el perfil competitivo definido.	Jefe de Recursos Humanos
3	Entrevista y evaluación.	Todos los postulantes pasan por una entrevista y evaluación de conocimientos previos tanto respecto a la empresa como su campo.	Jefe de Recursos Humanos

4	Contratación	Se califica a los postulantes y selecciona el que cuente con la más alta puntuación y cumpla con el perfil competitivo del puesto para la contratación de este, de caso contrario, se volverá a publicar el requerimiento del calificador externo.	Jefe de Recursos Humanos
---	--------------	--	--------------------------

## 5. Documentación

- Guía de calificación de perfil competitivo de calificador externo.

Nombre de evaluador:		
Nombre de postulante:		
Cumple con:	Requisitos competitivos <input type="checkbox"/>	Requisitos de nivel de experiencia <input type="checkbox"/>
Calificación de la entrevista:		
Actitud, aptitud y desenvolvimiento	Alta <input type="checkbox"/>	Bajo <input type="checkbox"/>
Calificación de conocimientos previos		
Puntaje total de calificaciones		
Nivel de cumplimiento y comportamiento		
Estado final	Aprobado <input type="checkbox"/>	Desaprobado <input type="checkbox"/>

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>13</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación de identificación del esquema de calidad del producto Software.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>

<b>Evaluación de identificación del esquema de calidad del producto Software.</b>	<b>Fecha:</b>
	<b>Versión: 1</b>
	<b>Responsable: jefe de control de calidad de los procesos del software.</b>

## 1. OBJETIVO

La presente evaluación busca identificar todo el esquema de calidad que maneja el sistema como producto.

## 2. ALCANCE

Pretende alcanzar a dar seguimiento e identificar el esquema de calidad del programa en función del software como producto.

## 3. RESPONSABLE

Jefe de control de calidad del sistema.

## 4. ACTIVIDAD

Nº	Actividad	Descripción	Responsable
1	Realizar evaluación de calidad.	Identificar las políticas de calidad y estructura que maneja el sistema.	Jefe de control de calidad del sistema.
2	Evaluación de la gestión de calidad.	El resultado de dicha evaluación se calcula en porcentaje, considerando el total de aspectos a cumplir dividido entre el total de aspectos, determinando así de la gestión actual del sistema.	Jefe de control de calidad del sistema.
3	Identificar las características de calidad como esquema.	Identificar los indicadores de control, características y las características de calidad del sistema de forma global en	Jefe de control de calidad del sistema.

		esquema tomando en cuenta los diagnósticos anteriores.	
--	--	--	--

## 5. Documentación

- Esquema de evaluación de calidad del sistema.

Formato de Diagnóstico de la calidad del software respecto a su seguridad.	Número de Evaluación: 13
Diagnóstico de calidad del sistema.	
Puntaje de evaluación:	
Criterios que cumple el proceso:	
Características identificadas:	
Nivel de impacto que general al sistema de portabilidad del software:	

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>14</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación de identificación del modelo de diseño del producto Software.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación de identificación del modelo de diseño del producto Software.</b>	<b>Fecha:</b>		
	<b>Versión: 1</b>		
	<b>Responsable:</b> <b>Jefe de control de calidad del sistema.</b>		

## 1. OBJETIVO

El presente Evaluación busca identificar el modelo del diseño a través de sus procesos y principales características del conjunto de actividades realizadas para obtener el software como producto.

## 2. ALCANCE

El alcance pretende identificar todo el modelo del producto del sistema.

## 3. RESPONSABLE

Jefe de control de calidad del sistema.

## 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Realizar el diagnóstico del modelo del sistema	Identificar las funciones del modelo, requerimientos, objetos y especificaciones técnicas que siguen los procesos.	Jefe de control de calidad del sistema.
2	Analizar la estructura del modelo	Analizar la estructura de las funciones. Estados de los procesos y repercusiones que general en sistema.	Jefe de control de calidad del sistema.
3	Elaborar diagnóstico final del modelo del sistema.	Definir el diseño conceptual del modelo y su estructura a partir los resultados de las actividades anteriores.	Jefe de control de calidad del sistema.

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>15</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación de planeación y gestión de desarrollo del producto Software.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación de planeación y gestión de desarrollo del producto Software.</b>	<b>Fecha:</b>		
	<b>Versión: 1</b>		
	<b>Responsable:</b>		

## 1. OBJETIVO

El presente Evaluación busca elaborar un plan de trabajo y gestión de desarrollo del producto.

## 2. ALCANCE

El alcance pretende generar actividades sistematizadas de planificación y gestión para seguir una adecuada secuencia de pasos para el desarrollo de un software.

## 3. RESPONSABLE

Jefe de sistemas e informática y equipo de trabajo.

## 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Elaborar lista de requerimientos.	A partir de las necesidades del cliente, se elabora lista de requerimientos para el desarrollo del software y las herramientas que se deba emplear.	Jefe de sistemas e informática y equipo de trabajo.
2	Elaborar plan de trabajo del desarrollo del software.	A partir de las necesidades identificadas, los requerimientos y herramientas necesarias se elabora un plan de trabajo donde se defina tanto el inicio y fin del proyecto como las actividades del desarrollo.	Jefe de sistemas e informática y equipo de trabajo.
3	Elaborar normas de gestión del desarrollo del producto.	Se debe elaborar un sistema de gestión para controlar y llevar a cabo de la mejor manera el desarrollo del producto.	Jefe de sistemas e informática y equipo de trabajo.

## 5. Documentación

- Plan de trabajo del desarrollo del producto software.

PLAN DE TRABAJO	
Delimitar fechas inicio-fin	El proceso debe contar con cronograma de fechas y un parámetro de duración
Programación de actividades	<ul style="list-style-type: none"> <li>- Detallar especificaciones y características según la necesidad del comprador.</li> <li>- Realizar estructura y esquema del modelo de software.</li> <li>- Normas de seguridad y niveles de riesgos a controlar.</li> <li>- Elaborar plan de control y gestión de procesos y calidad del software.</li> <li>- Realizar prototipo y pruebas del software.</li> <li>- Realizar la documentación necesaria.</li> </ul>
Revisión final	-Inspección del programa del desarrollo de sus procesos

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>Nº:</b>	<b>16</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación guía de calificación de producto Software para calificadores.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación guía de calificación de producto Software para calificadores.</b>	<b>Fecha:</b>		
	<b>Versión: 1</b>		
	<b>Responsable:</b> <b>Supervisor de gestión y desarrollo de software.</b>		

## 1. OBJETIVO

El presente Evaluación busca elaborar una guía y evaluación sistematizada para la aplicación de calificación del producto.

## 2. ALCANCE

El alcance pretende desarrollar una correcta guía de calificación de todo el software de forma integral.

## 3. RESPONSABLE

Supervisor de gestión y desarrollo de software.

## 4. ACTIVIDAD

Nº	Actividad	Descripción	Responsable
1	Definir criterios a evaluar	Detallar los criterios y características que se evaluarán del software.	Supervisor de gestión y desarrollo de software.
2	Elaborar plan de cronogramas de calificaciones.	Definir el inicio y fin de una calificación, elaborar cronogramas y fechas de calificación establecidas.	Supervisor de gestión y desarrollo de software.
3	Estudiar la precisión de los hallazgos	Los datos recopilados deben consolidarse en hallazgos precisos de acuerdo con criterios definidos.	Supervisor de gestión y desarrollo de software.



4	Se evalúa el nivel de confiabilidad.	Asegurar la confiabilidad de la calificación dará a la organización veracidad para determinar a la calificación como verídica.	Supervisor de gestión y desarrollo de software.
5	Armar guía de calificación.	Se debe integrar tanto el cronograma, fechas de inicio-fin, los criterios a evaluar y la veracidad de los resultados	Supervisor de gestión y desarrollo de software.

## 5. Documentación

- Guía de calificación de software.

Datos del software	
Nombre:	
Tipo:	
Rubro:	
Procesos:	
Proceso 1:	
Impacto:	
Proceso 2:	
Impacto:	
Proceso 3:	
Impacto:	
Proceso N:	
Impacto:	
N° de Riesgos:	

Fuerte	
Regular	
Leve	
Problemas del software	
Calidad	
Seguridad	
Gestión	
Eficiencia	
Funcionamiento	

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>17</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación guía de codificación para desarrolladores.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación guía de codificación para desarrolladores.</b>	<b>Fecha:</b>		
	<b>Versión: 1</b>		
	<b>Responsable: Programador de software</b>		

### 1. OBJETIVO

El propósito de esta Evaluación es elaborar una guía de codificación para desarrolladores de forma sistemática según lo que necesite como producto.

### 2. ALCANCE

Esta Evaluación es aplicable para todo equipo o unidad de trabajo que quiera implementar a una organización una metodología de codificación.

### 3. RESPONSABLE

#### 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Elaborar el modelo del diseño del software.	La organización debe definir y establecer el diseño del software en función a las necesidades del cliente.	Programador de software.
2	Elaborar normativas y herramientas de ayuda de codificación.	Establecer reglas y consejos que aporten y faciliten a la codificación del software.	Programador de software.
3	Diseñar metodología de codificación	Teniendo en cuenta el modelo del diseño del software, sus requerimientos, herramientas y normativas de codificación se genera una metodología global de codificación que abarque los puntos anteriores mencionados.	Programador de software.
4	Elaborar guía de codificación	La guía debe contener detalle del tipo de modelo de diseño, la metodología, la normativa y requerimientos a evaluar respecto al impacto que generan en el desarrollo del software.	Programador de software.

<b>Metodología de evaluación de la calidad del producto de software</b>			
<b>N°:</b>		<b>18</b>	
<b>EVALUACIÓN:</b>		<b>Evaluación guía de calificación de producto software para compradores.</b>	
<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b>	<input type="checkbox"/>
<b>Fecha:</b>			

<b>Evaluación guía de calificación de producto software para compradores.</b>	<b>Versión: 1</b>
	<b>Responsable: jefe de informática de la empresa y equipo de trabajo.</b>

## 1. OBJETIVO

El propósito de este Evaluación es elaborar una guía de calificación de software de forma entendible a los compradores según lo que necesiten.

## 2. ALCANCE

Este Evaluación es aplicable para todo cliente que desee evaluar el software que va adquirir.

## 3. RESPONSABLE

Jefe de informática de la empresa y equipo de trabajo.

## 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Criterios de calificación	Se subdivide tipo de criterios de calificación según la segmentación del mercado.	Jefe de informática de la empresa y equipo de trabajo.
2	Calificación por proceso	Se divide la calificación por procesos y especificaciones a evaluar.	Jefe de informática de la empresa y equipo de trabajo.
3	Calificación de calidad al sistema	Se elabora criterios de calidad a evaluar según el interés del cliente.	Jefe de informática de la empresa y

			equipo de trabajo.
4	Elaborar formato de calificación	Elaborar el formato de evaluación donde se segmenta en tipo de software, según los procesos, según la calidad del software.	Jefe de informática de la empresa y equipo de trabajo.

## 5. Documentación

- Guía de calificación de software para compradores.

Datos del software	
Nombre:	
Tipo:	
Rubro:	
Procesos:	
Proceso 1:	
Impacto:	
Proceso 2:	
Impacto:	
Proceso 3:	
Impacto:	
Proceso N:	
Impacto:	
N° de Riesgos:	

Fuerte	
Regular	
Leve	
Problemas del software	
Calidad	
Seguridad	
Gestión	
Eficiencia	
Funcionamiento	

	<b>Metodología de evaluación de la calidad del producto de software</b>		
	<b>N°:</b>	<b>19</b>	
	<b>EVALUACIÓN:</b>	<b>Evaluación de diseño de arquitectura basada en factores.</b>	
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b> <input type="checkbox"/>
<b>Evaluación de diseño de arquitectura basada en factores.</b>	<b>Fecha:</b>		
	<b>Versión: 1</b>		
	<b>Responsable: Programador de software</b>		

### 1. OBJETIVO

El propósito de este Evaluación es elaborar método de proceso para diseñar la arquitectura del sistema.

### 2. ALCANCE

Este Evaluación es aplicable para toda organización que desee utilizar el método de diseño de arquitectura de su programa informático.

### 3. RESPONSABLE

Programador de software

#### 4. ACTIVIDAD

N°	Actividad	Descripción	Responsable
1	Definir las necesidades e inquietudes del cliente	Extraer las inquietudes de las partes interesadas y los beneficios que desea obtener en el diseño de la arquitectura del software.	Programador de software
2	Definir los factores que influyen en la arquitectura del sistema.	Se debe determinar los factores y el impacto que generen en la arquitectura del sistema.	Programador de software
3	Elaborar un plan de trabajo de diseño	Donde se deba definir el fin y inicio de la elaboración del diseño, sus actividades, cronograma y medidas de control.	Programador de software

		<b>Metodología de evaluación de la calidad del producto de software</b>			
		<b>N°:</b>	<b>20</b>		
		<b>EVALUACIÓN:</b>	<b>Evaluación de gestión del cambio.</b>		
		<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b>	<input type="checkbox"/>
<b>Evaluación de gestión del cambio.</b>		<b>Fecha:</b>			
		<b>Versión: 1</b>			
		<b>Responsable: Jefe de control de calidad de software</b>			

#### 1. OBJETIVO

El propósito de esta Evaluación es elaborar una gestión que logre contribuir al cambio y mejora del sistema.

#### 2. ALCANCE

Este Evaluación es aplicable para toda organización que deseen renovar o mejorar el sistema de gestión actual que maneje.

### 3. RESPONSABLE

Jefe de control de calidad de software

### 4. ACTIVIDAD

Nº	Actividad	Descripción	Responsable
1	Elaborar diagnóstico actual de la gestión.	Se analiza el sistema de gestión, donde se evalúa la determinación de riesgos, nivel de calidad y el sistema de planificación y control del software.	Jefe de control de calidad de software
2	Elaborar plan de trabajo	Se realiza el plan de trabajo a partir del diagnóstico donde se identifica los problemas y se plantea un plan de mejora, así como su aplicación en el sistema.	Jefe de control de calidad de software
3	Diagnóstico de la nueva gestión de cambio.	Se evalúa la nueva gestión de cambio y si de ella denota un cambio favorable a comparación del diagnóstico inicial, se considera aprobado de caso contrario se vuelve a plantear un nuevo plan de mejora.	Jefe de control de calidad de software

### 5. Documentación

- Registro de análisis de trabajo

Nº	Actividad	Descripción	Responsable
1	Análisis de la evaluación de la gestión actual	Se debe elaborar un análisis a todas las evaluaciones, tantos de la gestión actual de forma integral, así como también sus criterios,	Jefe de control de calidad de software



		métricas, medidas de control, entre otros.	
2	Análisis de fiabilidad de los resultados.	Las calificaciones deben ser evaluadas según la veracidad debida, así como de forma objetiva por eso se debe realizar un test de fiabilidad.	Jefe de control de calidad de software
3	Diagnóstico de análisis de gestión actual	Se debe analizar los resultados anteriores y realizar diagnóstico de forma integral la gestión actual y los problemas que puedan tener.	Jefe de control de calidad de software
4	Diagnóstico de plan de mejora	Se debe analizar los métodos, medidas y procesos entre otras herramientas que se emplee para mejorar la gestión del sistema.	Jefe de control de calidad de software
5	Diagnóstico final y análisis de la gestión de cambio.	Se compara el diagnóstico actual con las consecuencias de la aplicación de la mejora con la repercusión de la nueva gestión y como han logrado a mejorar el software	Jefe de control de calidad de software

	<b>Metodología de evaluación de la calidad del producto de software</b>			
	<b>Nº:</b>	<b>21</b>		
	<b>EVALUACIÓN:</b>	<b>Evaluación de estudio de los problemas del producto software</b>		
	<b>COMPLETO</b>	<input type="checkbox"/>	<b>INCOMPLETO</b>	<input type="checkbox"/>
<b>Evaluación de gestión del cambio.</b>	<b>Fecha:</b>			
	<b>Versión: 1</b>			
	<b>Responsable: jefe de control de calidad de los procesos del software.</b>			

## 1. OBJETIVO

El propósito de esta Evaluación es diseñar un formato de identificación de problemas que pueda tener el software de forma sistematizada.

## 2. ALCANCE

Esta Evaluación pretende ayudar a las empresas u organizaciones a identificar de una manera más rápida los problemas que este tenga,

## 3. RESPONSABLE

Supervisor de software

## 4. ACTIVIDAD

Nº	Actividad	Descripción	Responsable
1	Elaborar formato de evaluación	Se debe realizar evaluaciones a todos los procesos, niveles de seguridad, gestión, control, efectividad y todos los criterios que impacten en el desarrollo del software.	Supervisor de software.
2	Se debe elaborar pruebas de respuestas y eficiencia	Se debe descomponer los criterios de a calificar como las piezas de trabajo, problemas generales y problemas específicos.	Supervisor de software.
3	Plan de diagnóstico	Se debe parametrar los problemas por tipo, importancia e impacto que generan en el software	Supervisor de software.

## 5. Documentación

- Registro de análisis de trabajo

Nº	Actividad	Descripción	Responsable
----	-----------	-------------	-------------

1	Análisis de la evaluación del proceso.	Se debe elaborar un análisis a todas las evaluaciones, tantos procesos, criterios, métricas que maneja, entre otros.	Equipo de control de mantenimiento de los procesos.
2	Análisis de fiabilidad de los resultados.	Las calificaciones deben ser evaluadas según la veracidad debida, así como de forma objetiva por eso se debe realizar un test de fiabilidad.	Supervisor de control de calidad del software.
3	Diagnóstico de análisis	Se debe analizar los resultados anteriores y realizar diagnóstico de forma integral de los problemas e impactos que han generado al sistema.	Equipo de control de calidad.

### C. Medición de indicadores

Se presentan el procesamiento de datos con el programa SPSS, de los siguientes indicadores: capacidad de respuesta, disponibilidad, número de vulnerabilidades, porcentaje de detección de defectos, tiempo de instalación, tiempo de actualización, tiempo de entrega de cambios, cantidad de código, tiempo medio de reparación y tiempo medio entre fallo.

- **Capacidad de respuesta**

Tabla 8. *Capacidad de respuesta*

	N	Media	Desviación estándar
Capacidad de respuesta sin metodología	21	17,4414	16,08503
Capacidad de respuesta con metodología	21	14,4495	13,67166
N válido (por lista)	21		

Fuente: Elaboración propia

La distribución de los datos para el indicador de capacidad de respuesta fue de 1, 10 y 100 usuarios; cuya media sin aplicación de la metodología fue de 17.44 segundos de respuesta con una desviación de 16.09 segundos; y luego de aplicar la metodología planteada la media de respuesta mejoro a 14.45 segundos como distribución general, con una desviación de 13.67 segundos.

### Prueba de normalidad

Para este indicador, dado que los datos son menores que 50 se utiliza el estadístico de Shapiro Wilk

Tabla 9. *Prueba de normalidad*

	Shapiro-Wilk		
	Estadístico	Gl	Sig.
Capacidad de respuesta sin metodología	,880	21	,014
Capacidad de respuesta con metodología	,864	21	,007

Fuente: Tabla 8

Como el nivel de significancia es menor a 0.05, los datos siguen una distribución no normal.

- **Disponibilidad**

Tabla 10. *Disponibilidad*

	N	Media	Desviación estándar
Disponibilidad sin metodología	31	,0645	,21377
Disponibilidad con metodología	31	,0187	,05948
N válido (por lista)	31		

Fuente: Elaboración propia

De la tabla se observa que el tiempo promedio de disponibilidad de sistema sin aplicar la metodología fue de 23.35 horas durante el mes de evaluación; siendo el promedio de tiempo no disponible de 0.0645 horas. Y luego de aplicar la metodología el tiempo de disponibilidad fue de 23.81 horas; con un promedio de tiempo no disponible de 0.187 horas.

### Prueba de normalidad

Para este indicador, dado que los datos son menores que 50 se utiliza el estadístico de Shapiro Wilk.

Tabla 11. *Prueba de normalidad*

	Shapiro-Wilk		
	Estadístico	GI	Sig.
Disponibilidad sin metodología	,343	31	,000
Disponibilidad con metodología	,353	31	,000

Fuente: Tabla 10

Como el nivel de significancia es menor a 0.05, los datos siguen una distribución no normal.

- **Número de vulnerabilidades**

Tabla 12. *Número de vulnerabilidades*

	N	Media	Desviación estándar
Número de vulnerabilidades sin metodología	9	4,2222	3,38296
Número de vulnerabilidades con metodología	9	1,1111	1,36423
N válido (por lista)	9		

Fuente: Elaboración propia

De la tabla se observa que la cantidad de vulnerabilidades encontradas fue de 4 en promedio; teniendo un máximo de 10 y un mínimo de 0 durante el periodo de prueba del sistema. Luego de la aplicación de la metodología el número promedio de vulnerabilidades encontradas fue de 1.

### Prueba de normalidad

Para este indicador, dado que los datos son menores que 50 se utiliza el estadístico de Shapiro Wilk

Tabla 13. *Prueba de normalidad*

	Shapiro-Wilk		
	Estadístico	GI	Sig.
Número de vulnerabilidades sin metodología	,938	9	,558

Número de vulnerabilidades con metodología	,820	9	,035
--	------	---	------

Fuente: Tabla 12

Como el nivel de significancia es mayor a 0.05, los datos siguen una distribución normal.

- **Porcentaje de detección de defectos**

Tabla 14. *Porcentaje de detección de defectos*

	N	Mí- nimo	Má- ximo	Media	Desviación es- tándar
Porcentaje de detección de defectos sin metodología	11	,00	10,00	5,1818	2,96034
Porcentaje de detección de defectos con metodología	11	,00	3,00	1,0909	1,13618
N válido (por lista)	11				

Fuente: Elaboración propia

El porcentaje de promedio de defectos fue del 5.18% de todas las pruebas realizadas en el sistema sin la aplicación de la metodología; luego de aplicar la metodología se redujo a 1.09%.

### Prueba de normalidad

Para este indicador, dado que los datos son menores que 50 se utiliza el estadístico de Shapiro Wilk

Tabla 15. *Prueba de normalidad*

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Porcentaje de detección de defectos sin metodología	,935	11	,461
Porcentaje de detección de defectos con metodología	,828	11	,022

Fuente: Tabla 15

Como el nivel de significancia es mayor a 0.05, los datos siguen una distribución normal.

- **Tiempo de instalación**

Tabla 16. *Tiempo de instalación*

	N	Media	Desviación estándar
Tiempo de instalación sin metodología	31	12,4839	2,23414
Tiempo de instalación con metodología	31	9,5161	1,82338
N válido (por lista)	31		

*Fuente: Elaboración propia*

El tiempo medio de instalación del sistema es de 12.48 minutos; luego de aplicar la metodología este redujo a 9. 51 minutos.

### **Prueba de normalidad**

Para este indicador, dado que los datos son menores que 50 se utiliza el estadístico de Shapiro Wilk

Tabla 17. *Prueba de normalidad*

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Tiempo de instalación sin metodología	,758	31	,000
Tiempo de instalación con metodología	,869	31	,001

*Fuente: Tabla 16*

Como el nivel de significancia es mayor a 0.05, los datos siguen una distribución normal, llegando a aumentar a 0.869.

- **Tiempo de actualización**

Tabla 18. *Tiempo de actualización*

	N	Media	Desviación estándar
Tiempo de actualización sin metodología	31	3,4194	1,43235
Tiempo de actualización con metodología	31	2,6774	1,13687
N válido (por lista)	31		

*Fuente:* Elaboración propia

El tiempo medio de actualización del sistema fue de 3.41 minutos durante las pruebas realizadas en el periodo de estudio, luego de aplicar la metodología el nuevo tiempo de actualización es 2.67.

### **Prueba de normalidad**

Para este indicador, dado que los datos son menores que 50 se utiliza el estadístico de Shapiro Wilk

Tabla 19. *Prueba de normalidad*

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Tiempo de actualización sin metodología	,856	31	,001
Tiempo de actualización con metodología	,827	31	,000

*Fuente:* Tabla 18

Como el nivel de significancia es mayor a 0.05, los datos siguen una distribución normal llevando a esto que el tiempo de actualización de 0.856 pase a disminuir a un tiempo de 0.827 con la aplicación de la metodología.



- **Tiempo de entrega de cambios**

Tabla 20. *Tiempo de entrega de cambios*

	N	Media	Desviación estándar
Tiempo de entrega de cambios sin metodología	11	4,0909	2,70017
Tiempo de entrega de cambios con metodología	11	2,5345	1,63067
N válido (por lista)	11		

*Fuente:* Elaboración propia

El tiempo promedio de entrega cambios solicitados por el cliente fue de 4.09 horas; luego de aplicar la metodología este redujo a 2. 53 minutos.

### **Prueba de normalidad**

Para este indicador, dado que los datos son menores que 50 se utiliza el estadístico de Shapiro Wilk.

Tabla 21. *Prueba de normalidad*

	Shapiro-Wilk		
	Estadístico	Gl	Sig.
Tiempo de entrega de cambios sin metodología	,903	11	,202
Tiempo de entrega de cambios con metodología	,940	11	,526

*Fuente:* Tabla 20

Como el nivel de significancia es mayor a 0.05, los datos siguen una distribución normal.

- **Cantidad de código**

Tabla 22. *Cantidad de código*

	N	Media	Varianza
Cantidad de código sin metodología	11	844,1818	250004,564
Cantidad de código con metodología	11	668,3636	194344,055
N válido (por lista)	11		

*Fuente:* Elaboración propia

De la tabla la cantidad promedio de líneas de código en módulo fueron de 844 líneas, pero gracias a la aplicación de la metodología este se redujo a 668 líneas, manejando la codificación de una manera más óptima.

### **Prueba de normalidad**

Para este indicador, dado que los datos son menores que 50 se utiliza el estadístico de Shapiro Wilk

Tabla 23. *Prueba de normalidad*

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Cantidad de código sin metodología	,844	11	,035
Cantidad de código con metodología	,786	11	,006

*Fuente:* Elaboración propia

Como el nivel de significancia es menor a 0.05, los datos siguen una distribución no normal.

- **Tiempo medio de reparación**

Tabla 24. *Tiempo medio de reparación*

	N	Media	Desviación estándar	Varianza
Tiempo medio de reparación sin metodología	6	23,0000	15,23155	232,000
Tiempo medio de reparación con metodología	6	13,4083	9,82814	96,592
N válido (por lista)	6			

*Fuente:* Elaboración propia

El tiempo medio de reparación de errores del sistema que no involucran el desempeño total del programa fue de 23 minutos; luego de la aplicación de la metodología el tiempo se redujo a 13 minutos.

### Prueba de normalidad

Para este indicador, dado que los datos son menores que 50 se utiliza el estadístico de Shapiro Wilk

Tabla 25. *Prueba de normalidad*

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Tiempo medio de reparación sin metodología	,887	6	,302
Tiempo medio de reparación con metodología	,857	6	,179

Fuente: Tabla 24

Como el nivel de significancia es mayor a 0.05, los datos siguen una distribución normal.

- **Tiempo medio entre fallos**

Tabla 26. *Tiempo medio entre fallos*

	N	Media	Desviación estándar	Varianza
Tiempo medio entre fallos sin metodología	5	138,2000	105,71045	11174,700
Tiempo medio entre fallos con metodología	5	117,0520	133,45732	17810,856
N válido (por lista)	5			

Fuente: Elaboración propia

El tiempo entre fallos promedio durante el periodo de prueba fue de 138.2 horas; luego de aplicar la metodología este se redujo a 117.05 minutos.

### Prueba de normalidad

Para este indicador, dado que los datos son menores que 50 se utiliza el estadístico de Shapiro Wilk.

Tabla 27. Prueba de normalidad

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Tiempo de entrega de cambios sin metodología	,903	11	,202
Tiempo de entrega de cambios con metodología	,940	11	,526

Fuente: Tabla 26

Como el nivel de significancia es mayor a 0.05, los datos siguen una distribución normal.

#### D. Prueba de hipótesis

Para la prueba de hipótesis se tiene que una parte de los datos son mayores que 30, por lo que se utilizará el estadístico de prueba de wilcoxon para probar la hipótesis; y en caso de que los datos sean menores que 30 se utiliza el estadístico de t de student.

- **Prueba T**

Se realizó la prueba T para los indicadores de capacidad de respuesta, número vulnerabilidades, porcentaje de detección de defectos, tiempo de entrega de cambios, cantidad de código, tiempo medio de reparación, tiempo medio entre fallos.

Tabla 28. Prueba T de la capacidad de respuesta

		t	gl	Sig. (bilateral)
Par 1	Capacidad de respuesta sin metodología - Capacidad de respuesta con metodología	5,296	20	,000

Fuente: Elaboración propia

Lo que se determina que la aplicación de la metodología planteada no mejora la capacidad de respuesta del programa, rechazándose así la hipótesis nula.

Tabla 29. *Prueba T del número de vulnerabilidades*

		t	gl	Sig. (bilateral)
Par 2	Número de vulnerabilidades sin metodología - Número de vulnerabilidades con metodología	4,233	8	,003

*Fuente:* Elaboración propia

Lo que se determina que la aplicación de la metodología, planteada no mejora la capacidad de respuesta del programa, rechazándose así la hipótesis nula.

Tabla 30. *Prueba T del porcentaje de detección de defectos*

		t	GI	Sig. (bilateral)
Par 3	Porcentaje de detección de defectos sin metodología - Porcentaje de detección de defectos con metodología	6,878	10	,000

*Fuente:* Elaboración propia

Lo que se determina que la aplicación de la metodología planteada no mejora la capacidad de respuesta del programa, rechazándose así la hipótesis nula.

Tabla 31. *Prueba T del tiempo de entrega de cambios*

		T	GI	Sig. (bilateral)
Par 4	Tiempo de entrega de cambios sin metodología - Tiempo de entrega de cambios con metodología	4,612	10	,001

*Fuente:* Elaboración propia

Lo que se determina que la aplicación de la metodología planteada no mejora la capacidad de respuesta del programa, rechazándose así la hipótesis nula.

Tabla 32. Prueba T de la cantidad de código

		T	gl	Sig. (bi-lateral)
Par 5	Cantidad de código sin metodología – Cantidad de código con metodología	6,477	10	,000

Fuente: Elaboración propia

Lo que se determina que la aplicación de la metodología planteada no mejora la capacidad de respuesta del programa, rechazándose así la hipótesis nula.

Tabla 33. Prueba T de tiempo medio de reparación

		t	gl	Sig. (bi-lateral)
Par 6	Tiempo medio de reparación sin metodología - Tiempo medio de reparación con metodología	4,180	5	,009

Fuente: Elaboración propia

Tabla 34. Prueba T de tiempo medio entre fallos

		T	gl	Sig. (bi-lateral)
Par 7	Tiempo medio entre fallos sin metodología - Tiempo medio entre fallos con metodología	- 3,027	4	,039

Fuente: Elaboración propia

Lo que se determina que la aplicación de la metodología planteada mejora la disponibilidad de del programa, aceptándose así la hipótesis nula, por lo que concluyo que la aplicación de la metodología planteada mejora la capacidad de respuesta del programa

- **Prueba de Wicolxon**

Tabla 35. *Prueba de Wicolxon de la disponibilidad*

	Z	Sig. asintótica (bilateral)
Disponibilidad con metodología - Disponibilidad sin metodología	- ,946 <sup>b</sup>	,344

*Fuente:* Elaboración propia

Lo que se determina que la aplicación de la metodología planteada mejora la disponibilidad de del programa, aceptándose así la hipótesis nula por lo que concluyo que la aplicación de la metodología planteada mejora la capacidad de respuesta del programa.

Tabla 36. *Prueba de Wicolxon del tiempo de instalación*

	Z	Sig. asintótica (bilateral)
Tiempo de instalación con metodología - Tiempo de instalación sin metodología	- 4,925 <sup>b</sup>	,000

*Fuente:* Elaboración propia

Lo que se determina que la aplicación de la metodología planteada no mejora la capacidad de respuesta del programa, rechazándose así la hipótesis nula.

Tabla 37. *Prueba de Wicolxon del tiempo de actualización*

	Z	Sig. asintótica (bilateral)
Tiempo de actualización con metodología - Tiempo de actualización sin metodología	- 4,796 <sup>b</sup>	,000

*Fuente:* Elaboración propia

Lo que se determina que la aplicación de la metodología planteada no mejora la capacidad de respuesta del programa, rechazándose así la hipótesis nula.

Dado que a través de los resultados del diagnóstico , la evaluación de los formatos, la medición de los indicadores y la prueba de hipótesis de datos, se pudo concluir que se llegó a aceptar las condiciones de las hipótesis específicas 1, 2 y 3; entonces, se aceptó la hipótesis general: “Se desarrolló una metodología de evaluación de la calidad de software que incluya requisitos de evaluación, métricas de evaluación de calidad y un proceso de evaluación en base a los estándares ISO/IEC 25000 e ISO/IEC 14598 y la metodología RUP”

#### Resumen

A continuación, se muestra un resumen de los resultados de aceptación o rechazo de las hipótesis planteadas en la investigación:

Tabla 38. *Resumen de aceptación de hipótesis*

ÍTEM	HIPÓTESIS	RESULTADO (Aceptada o Rechazada)
1	<p>Se desarrolló una metodología de evaluación de calidad de software con requisitos de evaluación según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</p> <p>De acuerdo con («Norma ISO/IEC 25000   Tecnología Investigación y Academia» 2017) las normas ISO 25000 nos proveen de una lista de requisitos para el desarrollo de programas informáticos de calidad.</p>	ACEPTADA
2	<p>Se desarrolló una metodología de evaluación de calidad de software con métricas de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</p> <p>Según (Torrealba-Peña y Mendoza Mejia 2018) las normas ISO 9126 e ISO 14598 se especializan en brindar normas y estándares para la evaluación de programas informáticos de calidad.</p>	ACEPTADA



3	<p>Se desarrolló una metodología de evaluación de calidad de software con un proceso de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</p> <p>Según (Diez Antón et al.2003) la metodología RUP puede utilizarse en conjunto con otras metodologías para poder elaborar procesos de evaluación.</p>	ACEPTADA
---	--	----------

## **V. DISCUSIÓN**

Los resultados del presente estudio demostraron la mejora en la calidad del producto software; esto se pudo evidenciar en los resultados de control y seguimiento de calidad que se utilizaron, tomando como herramienta los indicadores que evaluaron tanto el comportamiento del software sin la metodología como también con la aplicación de esta, dando resultados favorables de forma notable tales como:

La capacidad de respuesta ya que sin aplicación de la metodología fue de 17.44 segundos; y luego de aplicar esta mejoró a 14.45 segundos mientras que la disponibilidad de sistema sin aplicar la metodología fue de 23.35 horas; pero luego de aplicar la metodología aumentó a 23.81 horas por otro lado la cantidad de vulnerabilidades del sistema tuvo inicialmente un máximo de 10 para luego después de la aplicación de esta disminuir a 1 además el porcentaje promedio de defectos el cual fue de 5.18% de todas las pruebas se logró reducir a 1.09% y el tiempo medio de instalación del sistema que fue 12.48 minutos; se redujo gracias a la metodología a 9.51 minutos por otro lado el tiempo medio de actualización del sistema fue de 3.41 minutos durante las pruebas realizadas en el periodo de estudio para luego reducirse a 2.67 minutos además el tiempo promedio de entrega de cambios solicitados por el cliente fue de 4.09 horas; para luego de aplicar la metodología reducirse a 2.53 minutos y también se pudo observar que el tiempo medio de reparación de errores el cual fue de 23 minutos pasó a 13 minutos como también se pudo denotar la mejora del tiempo entre fallos promedio el cual fue inicialmente 138.2 horas para luego reducirse a 117.05 minutos.

Así también se presentan las teorías relacionadas a las pautas de calidad de programas por ello es comparable con la investigación de Dalla et al (2020) donde también su estudio propone la aplicación métrica para identificar las propiedades de la infraestructura como código tradicionales e independientes además de elaborar un catálogo de medidas de código basadas en estructuras para evaluar los diferentes aspectos la calidad del software.

Además, el presente estudio también es comparable con el trabajo de Atoum (2020) el cual desarrolló una metodología para controlar las evaluaciones frecuentes que se realizaba al software, logrando así el mayor rendimiento en

comparación con los métodos conocidos seleccionados con aportando a esta área raramente investigada y de igual manera en esta tesis la metodología se enfocó en mejorar procesos y lograr alcance y aportes para otras organizaciones también.

Por otro lado, en el estudio realizado por Xin (2019) implemento modelos de calificación de calidad que se centran en aspectos tales como métricas, factores de calidad, métodos de agregación, métodos de evaluación y soporte de herramientas y la construcción de un software de referencia público y diverso que se puede adoptar en diferentes contextos de aplicación lo cual es comparable a nuestro trabajo ya que se desarrolló construcción de esquemas, diseños y modelos del programa software generando un proceso sistematizado que contribuya a la organización donde se aplique.

Además, Hynninen (2018) en su estudio presentó un marco para recopilar métricas de tiempo de ejecución y medir la calidad en uso del software con una interfaz sistemática; para realizar el marco, se siguió el proceso descrito en la norma ISO 15939.

De los resultados del estudio, el marco de medición y el proyecto de prueba de concepto se evaluaron mediante el uso de escenarios descriptivos para el software en la fase de mantenimiento de su ciclo de vida por lo que aportó al marco de medición con la creación de una biblioteca de métricas, lo cual tiene semejanza con nuestro estudio ya que creamos un sistema de métricas de control pero que se aplican de forma sistematizada según el proceso que se vaya a medir.

Se puede observar además que en el trabajo de estudio de Wen-Hong y Xin (2012) busca mejorar la efectividad, visibilidad y especificación de las pruebas del software TT&C a través de un modelo de evaluación de la calidad con alta confiabilidad y demandas en tiempo real obteniendo una idea del proceso de jerarquía analítica (AHP) en nuestro caso se mejoró la efectividad y respuestas del software gracias a la implementación de metodología de mejora y control la calidad del programa.

Otro estudio de suma importancia y comparable al presente estudio fue el realizado por Vargas (2018) el crea una metodología de evaluación de las herramientas para generar sistemas de información de tipo comerciales lo cual logró evaluar y seleccionar de forma eficaz de la igual manera que el presente estudio pudo plantear metodologías de evaluación según los procesos del sistema.

## **VI. CONCLUSIONES**

Las conclusiones de la investigación fueron las siguientes:

Se logró desarrollar una metodología de evaluación de la calidad de software los cuales incluyeron requisitos, guías y formatos de evaluación además métricas de evaluación de calidad y se detalló cada proceso de evaluación en base a los estándares ISO/IEC 25000 e ISO/IEC 14598 y la metodología RUP.

Por otro lado, en base al objetivo específico 1 se concluyó que:

- La metodología con el fin de mejorar los procesos actuales, la eficiencia, calidad y control del desarrollo del programa software desarrolló requisitos de evaluación con base en los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP, los cuales se pueden observar a partir del anexo 14 del presente estudio.

En base al objetivo específico 2 se concluyó:

- La metodología con el fin de evaluar de mejor manera la calidad de software empleo el uso de métricas tales como capacidad de respuesta, disponibilidad, vulnerabilidades, detección de defectos, tiempo de instalación, tiempo de actualización, tiempo de respuesta, entre otros los cuales tienen base en los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.

En base al objetivo específico 3 se concluyó:

- Se desarrolló una metodología completa donde se detalló cada proceso con su respectivo objetivo, alcance, responsable, actividad y documentación necesaria como formatos de evaluación, registros, registro de análisis, entre otros lo cual tienen base en los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.

## **VII. RECOMENDACIONES**



Las recomendaciones para futuras investigaciones son las siguientes:

1. Ampliar la investigación científica con una muestra más amplia de normas internacionales y de documentos a evaluar, con la finalidad de tener mayor respaldo teórico en la investigación.
2. Profundizar el análisis de los resultados históricos que se obtuvieron con la implementación de la metodología propuesta, dado que se evidencia una mejora continua del producto software con la ayuda de la metodología.
3. Desarrollar la investigación científica considerando una muestra o población de mayor tamaño o diversidad, teniendo en cuenta que la implementación de una metodología en una organización necesita de tiempo y recursos tanto humanos como económicos para poder ser implementados con éxito.
4. Se recomienda ampliar la cantidad de indicadores de acuerdo a la realidad que puedan enfrentar las organizaciones en su época actual, debido a la evolución constante de las tecnologías.
5. Se sugiere a las MYPES y PYMES implementar metodologías de evaluación de calidad de software que contemplen los indicadores planteados en la investigación, con ello logran desarrollar software de mayor calidad diferenciándose de los competidores para poder ganar terreno en el mercado empresarial.
6. Incorporar metodologías ágiles a la presente investigación, que ayuden a las organizaciones a poder evaluar la calidad del software de manera más efectiva y rápida considerando la volatilidad en las empresas.
7. Por último, se recomienda hacer mejoras en el proceso de recolección de datos, técnicas y/o instrumentos de medición utilizados, considerando que los conflictos sociales y enfermedades virales pueden ocasionar dificultades al momento de obtener los puntos mencionados anteriormente.

## **REFERENCIAS**

- Atoum, Issa. 2020 A novel framework for measuring software quality-in-use based on semantic similarity and sentiment analysis of software reviews. 1, Amman : Journal of King Saud University - Computer and Information Sciences, 2020, Vol. 32. ISBN.
- Basili V.R., Weiss D.M.: Una metodología para recopilar datos válidos de ingeniería de software, IEEE Transactions on Software Engineering, Nov. (1984)
- Basili, V.R., Caldiera, G., Rombach, H.D.: El enfoque métrico de la pregunta del objetivo. En: Encyclopedic of Software Engineering, Wiley-Interscience, New York (1994)
- Calabrese, J., Muñoz, R., Pasini, A., Esponda, S., Boracchia, M., & Pesado, P. (s.f). Assistant for the Evaluation of Software Product Quality Characteristics Proposed by ISO/IEC 25010 Based on GQM-Defined Metrics. Buenos Aires: Center of the Scientific Research Agency of the Province of Buenos Aires (CICPBA).
- Dalla, Stefano, y otros. 2020 Toward a catalog of software quality metrics for infrastructure code. 110726, Eindhoven: The Journal of Systems & Software, 2020, Vol. 170. ISBN.
- DIAZ-ANTÓN, M., PÉREZ, M., GRIMÁN, A. y MENDOZA MORALES, L., 2003. Propuesta de una Metodología de Desarrollo de Software Educativo bajo un Enfoque de Calidad Sistémica. S.l.: s.n.
- Estévez, Y., & Esteban, I. (2014). Modelo de calidad para evaluar el software desarrollado en el centro de investigación aplicada y desarrollo de ti . Pamplona.
- FERRO SOTO, C., MARTÍNEZ SENRA, A.I. y OTERO NEIRA, M.C., 2009. Ventajas del uso de las TICs en el proceso de enseñanza-aprendizaje desde la óptica de los docentes universitarios españoles. Edutec. Revista Electrónica de Tecnología Educativa, no. 29, pp. a119. ISSN 1135-9250. DOI 10.21556/edutec.2009.29.451.

GAMBOA, Johana Zumba. Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. INNOVA Research Journal, 2018, vol. 3, no 10, p. 20-33.

Grady Booch et al. The Unified Modeling Language User Guide. Addison-Wesley, 1999.

Hernández, Fernández & Baptista (2014). Metodología de la Investigación (6ta ed.). México D.F.: Mc Graw Hill.

Hynninen, Timo, Kasurinen, Jussi y Taipale, Ossi. 2018. Framework for Observing the Maintenance Needs, Runtime Metrics and the Overall Quality-in-Use. 4, Kotka: Journal of Software Engineering and Applications, 2018, Vol. 11. ISBN.

I/E 14598/1. Ciencia de los datos. Ingeniería de Programas. Calificación del programa. Parte 1 - Descripción General.

I/E 14598/2. Ciencia de los datos. Ingeniería de Programas. Calificación del programa. Parte 2 - Planificación y Gestión.

I/E 14598/3. Ciencia de los datos. Ingeniería de Programas. Calificación del programa. Parte 3 - Proceso para Desarrolladores.

I/E 14598/4. Ciencia de los datos. Ingeniería de Programas. Calificación del programa. Parte 4 - Proceso para Compradores.

I/E 14598/5. Ciencia de los datos. Ingeniería de Programas. Calificación del programa. Parte 5 - Proceso para Evaluadores.

I/E 14598/6. Ciencia de los datos. Ingeniería de Programas. Calificación del programa. Parte 6 - Documentación de los Módulos de Calificación.

I/E 14598: Software engineering – Product quality. I/E (2001)

I/E 25000 <http://www.iso25000.com/>.

I/E 9126-1: Software engineering – Product quality. I/E (2001)

ISO 9001: Quality management systems – Requirements. ISO (2001)

James Rumbaugh et al. The Unified Modeling Language Reference Manual. Addison-Wesley, 1999.

Kanellopoulos, Yiannis, y otros. 2010. CODE QUALITY EVALUATION METHODOLOGY USING THE ISO/IEC 9126 STANDARD. Manchester: International Journal of Software Engineering & Applications, 2010. Vol. 1, 3. ISBN.

Krzysztof S. (2005) Evaluation of software quality. Warsaw University of Technology, Nowowiejska 15/19. Warszawa, Poland. [https://www.researchgate.net/publication/228765628\\_Evaluation\\_of\\_software\\_quality](https://www.researchgate.net/publication/228765628_Evaluation_of_software_quality)

LENARDUZZI, V., LOMIO, F., HUTTUNEN, H. y TAIBI, D., 2020. Are SonarQube Rules Inducing Bugs? 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER). S.l.: s.n., pp. 501-511. DOI 10.1109/SANER48275.2020.9054821.

LOPEZ GRANADOS, Evelyn Vanessa. Pago de beneficios sociales. 2018.

Muñoz, Minra, Mejia, J., & Laporte, C. Y. (2020). Implementando I / E 29110 para reforzar cuatro entidades muy pequeñas de México bajo un enfoque ágil. IET DIGITAL LIBRARY. <https://digital-library.theiet.org/content/journals/10.1049/iet-sen.2019.0040>

Muñoz, Mirna, Mejia, J., Peña, A., Lara, G., & Laportec, C. Y. (2019). Transición de los estándares internacionales de ingeniería de software a la academia: análisis de los resultados de la adopción de I / E 29110 en cuatro universidades mexicanas. Computer Standards & Interfaces. <https://www.sciencedirect.com/science/article/abs/pii/S0920548918304173>

Netdata - Monitor everything in real time for free with Netdata. Netdata [en línea], [sin fecha]. [Consulta: 16 junio 2021].

Norma ISO/IEC 25000 | Tecnología Investigación y Academia. [en línea], 2017. [Consulta: 17 julio 2021].

- Owens, D., Khazanchi, D. (2009). Calificación de programas informáticos. In: Kidd, T.T., Editor (2009), Handbook of Research on Technology Project Management, Planning and Operations, Hershey, PA: Information Science Reference (an imprint of IGI Global), Chapter XVI, pp. 242-260.
- Petkov, D., Edgar-Nevill, D., Madachy, R., & O'Connor, R. 2008. Sistemas de información, ingeniería de software y pensamiento de sistemas: desafíos y oportunidades. International Journal of Information Technologies and Systems Approach (IJITSA), 1(1), 62-78
- Philippe Kruchten. The Rational Unified Process - An Introduction. AddisonWesley, 1999.
- Pressman, R. S. (2005) Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill.
- Rational Software Corporation. Rational unified process - best practices for software development teams. A Rational Software White Paper, 1998.
- Sacha, Krzysztof. 2015. Evaluation of software quality. Warsaw: Warsaw University of Technology, 2015. ISSN.
- SUÁREZ TÉLLEZ, L., RAMÍREZ, M., LUNA, V., LAZCANO, A., GUERRERO, J. y RUIZ, B., 2018. *Red Académica de los Seminarios Repensar: Proyecto innovador de profesionalización docente en el Instituto Politécnico Nacional*. S.l.: s.n.
- Timo Hynninen, Jussi Kasurinen, O. T. (2018). Marco para observar las necesidades de mantenimiento, las métricas de tiempo de ejecución y la calidad en uso general. Scientific Research. <https://www.scirp.org/journal/paperinformation.aspx?paperid=83411>
- TORREALBA-PEÑA, M. y MENDOZA MEJIA, J., 2018. Actitud hacia la educación virtual en estudiantes de la licenciatura en administración de la UNESR, núcleo Barquisimeto. S.l.: s.n.,

Vargas, Laura, y otros. 2018. Software Quality Methodology to Train Engineers as Evaluators of Information Systems Development Tools. Tamaulipas: Universal Journal of Educational Research, 2018. Vol. 6, 12. ISBN.

Wen-Hong, Liu y Xin, Wu. 2012. The Software Quality Evaluation Method Based on Software Testing. Nanjing : International Conference on Computer Science and Service System, 2012. Vol. 11, 13227072. ISBN.

Xin Xia, Meng, y otros. 2019. Software quality assessment model: a systematic mapping study. Beijing: Science China Information Sciences volume, 2019. Vol. 62, 191101. ISBN.

## **ANEXOS**



### Anexo 1: Matriz de operacionalización de variables

Tabla 39

Matriz de operacionalización de variables

Variable	Definición conceptual	Definición operacional	Dimensión	Indicadores	Instrumento	Escala de medición
Calidad del programa informático (ISTQB 2017).	Son los requisitos del cliente que el programa informático debe satisfacer y las características que este debe tener según estándar. (ISTQB 2017).	Normas I/E 25000, I/E 14598, metodología RUP	Eficiencia (I/E 25000 e I/E 14598)	Cantidad de código (I/E 25000 e I/E 14598)	Métricas para la calidad de programas (I/E 25000, I/E 14598)	Las indicadas para cada indicador en la norma (I/E 25000, I/E 14598)
				Capacidad de respuesta (I/E 25000 e I/E 14598)		
			Fiabilidad (I/E 25000 e I/E 14598)	Porcentaje de detección de defectos (I/E 25000 e I/E 14598)		
				Tiempo medio entre fallos (I/E 25000 e I/E 14598)		
				Tiempo medio de reparación (I/E 25000 e I/E 14598)		
			Mantenibilidad (I/E 25000 e I/E 14598)	Tiempo de entrega de cambios (I/E 25000 e I/E 14598)		
			Portabilidad (I/E 25000 e I/E 14598)	Tiempo de instalación (I/E 25000 e I/E 14598)		
Tiempo de actualización (I/E 25000 e I/E 14598)						
Seguridad (I/E 25000 e I/E 14598)	Disponibilidad (I/E 25000 e I/E 14598)					
	Número de vulnerabilidades (I/E 25000 e I/E 14598)					

## Anexo 2: Matriz de Consistencia

Tabla 40  
Instrumento de recolección de datos

PROBLEMAS	OBJETIVOS	HIPÓTESIS	VARIABLES	INDICADORES
<p><b>Problema general:</b></p> <p>No se han encontrado modelos de evaluación de calidad de software con requisitos de evaluación, métricas de evaluación de calidad y un proceso de evaluación en base a los estándares ISO/IEC 25000 e ISO/IEC 14598 y la metodología RUP.</p> <p><b>Problemas específicos:</b></p> <ul style="list-style-type: none"> <li>PE1: No se han encontrado metodologías de evaluación de calidad de software con requisitos de evaluación según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</li> <li>PE2: No se han encontrado metodologías de evaluación de calidad de software con métricas de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</li> <li>PE3: No se han encontrado metodologías de evaluación de calidad de software con un proceso de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</li> </ul>	<p><b>Objetivo general:</b></p> <p>Desarrollar una metodología de evaluación de la calidad de software que incluya requisitos de evaluación, métricas de evaluación de calidad y un proceso de evaluación en base a los estándares ISO/IEC 25000 e ISO/IEC 14598 y la metodología RUP.</p> <p><b>Fines específicos:</b></p> <p><b>OE1:</b> Desarrollar una metodología de evaluación de calidad de software con requisitos de evaluación según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</p> <p><b>OE2:</b> Desarrollar una metodología de evaluación de calidad de software con métricas de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</p> <p><b>OE3:</b> Desarrollar una metodología de evaluación de calidad de software con un proceso de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</p>	<p><b>Hipótesis General:</b></p> <p>Se desarrolló una metodología de evaluación de la calidad de software que incluya requisitos de evaluación, métricas de evaluación de calidad y un proceso de evaluación en base a los estándares ISO/IEC 25000 e ISO/IEC 14598 y la metodología RUP.</p> <p><b>Hipótesis específicas:</b></p> <p><b>H1:</b> Se desarrolló una metodología de evaluación de calidad de software con requisitos de evaluación según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</p> <p><b>H2:</b> Se desarrolló una metodología de evaluación de calidad de software con métricas de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</p> <p><b>H3:</b> Se desarrolló una metodología de evaluación de calidad de software con un proceso de evaluación de calidad según los estándares ISO/IEC 25000, ISO/IEC 14598 y metodología RUP.</p>	<p><b>Variable 1:</b> Calidad del programa informático (I/E 14598, 25000)</p>	<ul style="list-style-type: none"> <li>Capacidad de respuesta (I/E 25000 e I/E 14598)</li> <li>Disponibilidad (I/E 25000 e I/E 14598)</li> <li>Número de vulnerabilidades (I/E 25000 e I/E 14598)</li> <li>Porcentaje de detección de defectos (I/E 25000 e I/E 14598)</li> <li>Tiempo de instalación (I/E 25000 e I/E 14598)</li> <li>Tiempo de actualización (I/E 25000 e I/E 14598)</li> <li>Tiempo de entrega de cambios (I/E 25000 e I/E 14598)</li> <li>Cantidad de código (I/E 25000 e I/E 14598)</li> <li>Tiempo medio de reparación (I/E 25000 e I/E 14598)</li> <li>Tiempo medio entre fallos (I/E 25000 e I/E 14598)</li> <li>Portabilidad (I/E 25000 e I/E 14598)</li> </ul>

### Anexo 3: Formularios de registro de indicadores

#### Capacidad de respuesta

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha de Registro
<b>Empresa Investigada</b>			
<b>Nombre del indicador</b>	Capacidad de respuesta		
<b>Fecha de Inicio</b>		<b>Fecha Final</b>	

Ítem	Ruta probada	Número de solicitudes	Tiempo de respuesta (s)
1			
2			
3			

## Disponibilidad

Formato de Registro			
Investigador		Tipo de Prueba	
Empresa Investigada			
Nombre del indicador	Disponibilidad		
Fecha de Inicio		Fecha Final	

Objetivo	Indicador	Fórmula
Procedimiento de identificación de las características de la calidad del programa en función de la eficiencia.	Disponibilidad	$Disponibilidad = \frac{Horas\ Totales - Horas\ parada\ por\ mantenimiento}{Horas\ Totales}$

Ítem	Fecha	Horas Totales	Horas Parados Por Mantenimiento	Diferencia de horas
1				
2				
3				

## Número de vulnerabilidades

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>			
<b>Nombre del indicador</b>	Numero de vulnerabilidades		
<b>Fecha de Inicio</b>		<b>Fecha Final</b>	

Objetivo	Indicador	Fórmula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la seguridad	Numero de vulnerabilidades		

Ítem	Fecha	Ruta del software	Número de Vulnerabilidades
1			
2			

## Porcentaje de detección de defectos

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>			
<b>Nombre del indicador</b>	Porcentaje de detección de defectos		
<b>Fecha de Inicio</b>		<b>Fecha Final</b>	

Objetivo	Indicador	Fórmula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la seguridad	Porcentaje de detección de defectos	$X = \frac{\text{Número de defectos previstos}}{\text{Número de defectos encontrados}} \times 100$	

Ítem	Fecha	Ruta del software	Número defectos previstos	Número de defectos encontrados
1				
2				

## Tiempo de instalación

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>			
<b>Nombre del indicador</b>	Tiempo de instalación		
<b>Fecha de Inicio</b>		<b>Fecha Final</b>	

Objetivo	Indicador	Fórmula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la portabilidad.	Tiempo de instalación		

Ítem	Fecha	Entorno	Hora Inicio	Hora Fin	Tiempo total de instalación (m)

### Tiempo de actualización

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>			
<b>Nombre del indicador</b>	Tiempo de actualización		
<b>Fecha de Inicio</b>		<b>Fecha Final</b>	

Objetivo	Indicador	Fórmula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la portabilidad.	Tiempo de actualización de versión		

Ítem	Fecha	Entorno	Tiempo total de actualización (m)



## Tiempo de entrega al cliente

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>			
<b>Nombre del indicador</b>	Tiempo de entrega de cambios		
<b>Fecha de Inicio</b>		<b>Fecha Final</b>	

Objetivo	Indicador	Fórmula	Detalle
Procedimiento de identificación de la libertad de riesgo del SFTW.	Tiempo de entrega		

Ítem	Fecha	Cambio	Tiempo total de realización (h)

### Cantidad de código

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>			
<b>Nombre del indicador</b>	Cantidad de código		
<b>Fecha de Inicio</b>		<b>Fecha Final</b>	

Ítem	Fecha	Ruta	Número total de líneas de código
1			
2			

### Tiempo medio de reparación

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>			
<b>Nombre del indicador</b>	Tiempo medio de reparación		
<b>Fecha de Inicio</b>		<b>Fecha Final</b>	

Ítem	Fecha	Hora del falló	Hora de la reparación	Total (m)
1				
2				

### Tiempo medio entre fallos

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>			
<b>Nombre del indicador</b>	Tiempo medio entre fallos		
<b>Fecha de Inicio</b>		<b>Fecha Final</b>	

Ítem	Fecha	Hora del falló	Total de horas (h)

### Anexo 4: Comparación de pautas ISO 14598, 25000 y metodología RUP

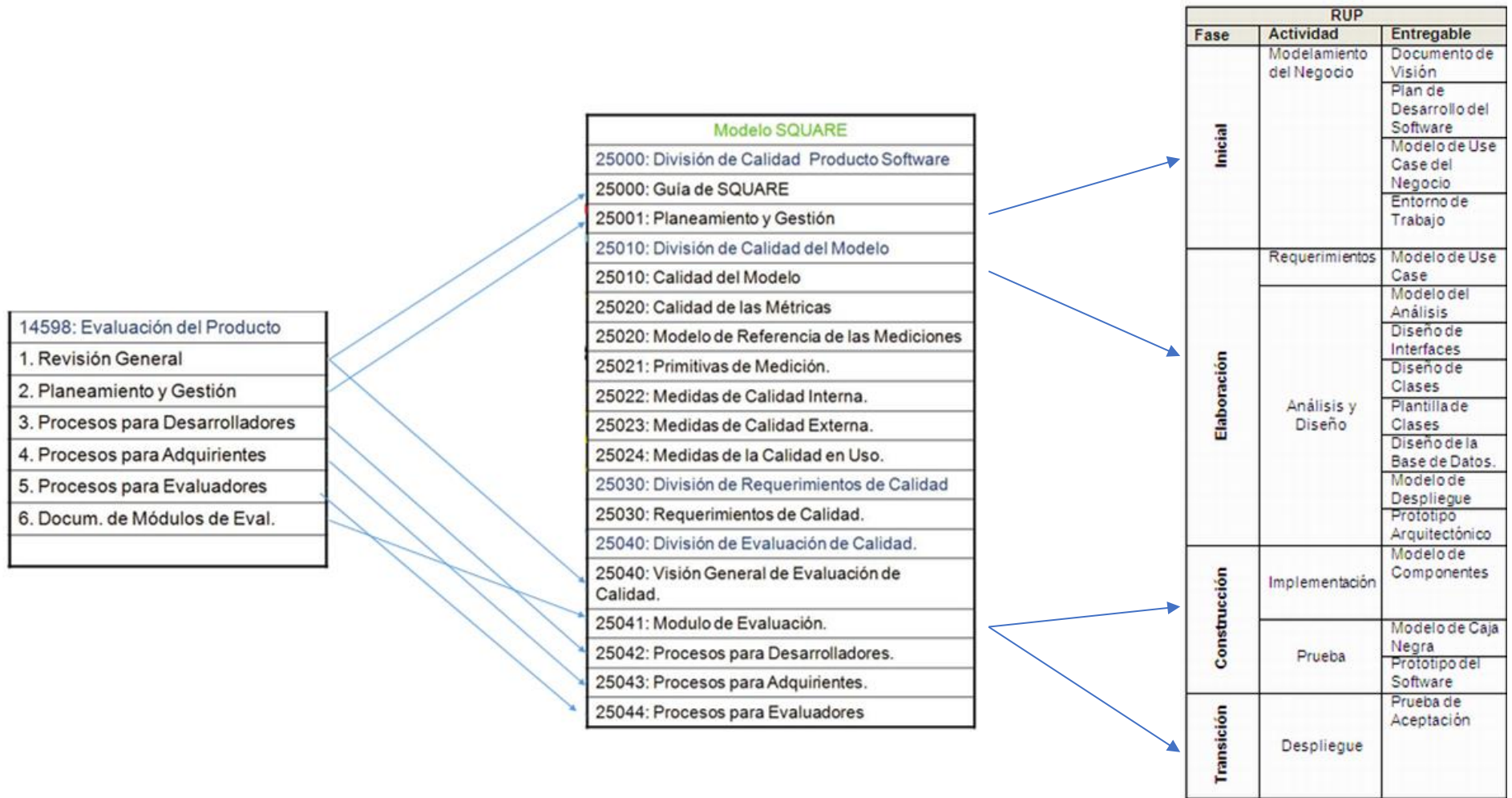


Figura 4. Metodología RUP y comparación de pauta Iso14598, 25000

## Anexo 5: Estructura inicial SPICE I/E 25000(Tecnología, Investigación y Academia,2017)

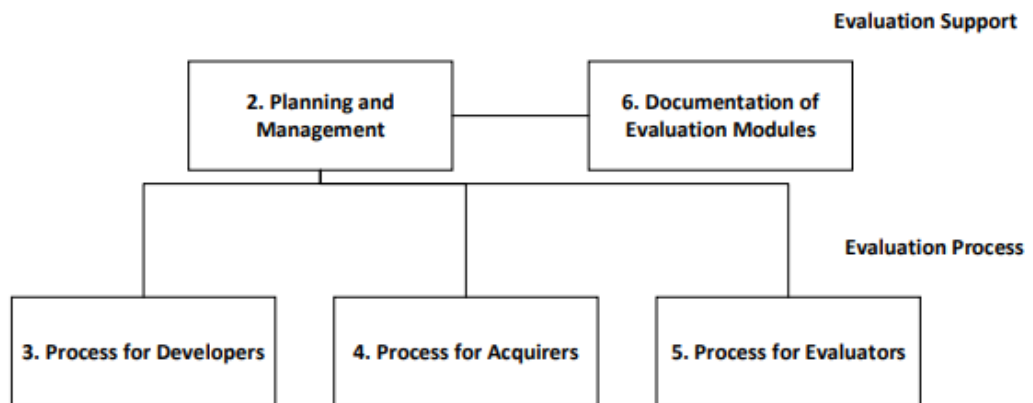
Tabla 41  
*Estructura inicial*

ISO/IEC 25000 (SQuaRE)		
Modelos N	Modelos	Nombres
A: Administración de calidad ISO 2500n	ISO 25000 ISO 25001	Guía de SQUARE Planeación y gestión
B: Modelo de calidad 2501n	ISO 25010	Calidad del modelo
C. Medidas de Calidad 2502n	ISO 25020 ISO 25021 ISO 25022	Calidad de las métricas, Modelo de referencia de las métricas Primitivas de medición Medidas de calidad interna
D: Requerimientos de calidad 2503n	ISO 25023 ISO 25024 ISO 25030	Medidas de calidad externa Medidas de calidad de uso Requerimientos de cali- dad
E: Evaluación de calidad 2504n	ISO 25040 ISO 25041 ISO 25042 ISO 25043 ISO 25044	Visión General de evaluación de calidad Módulo de evaluación Proceso para desarrolladores Proceso para adquirientes Proceso de evaluadores

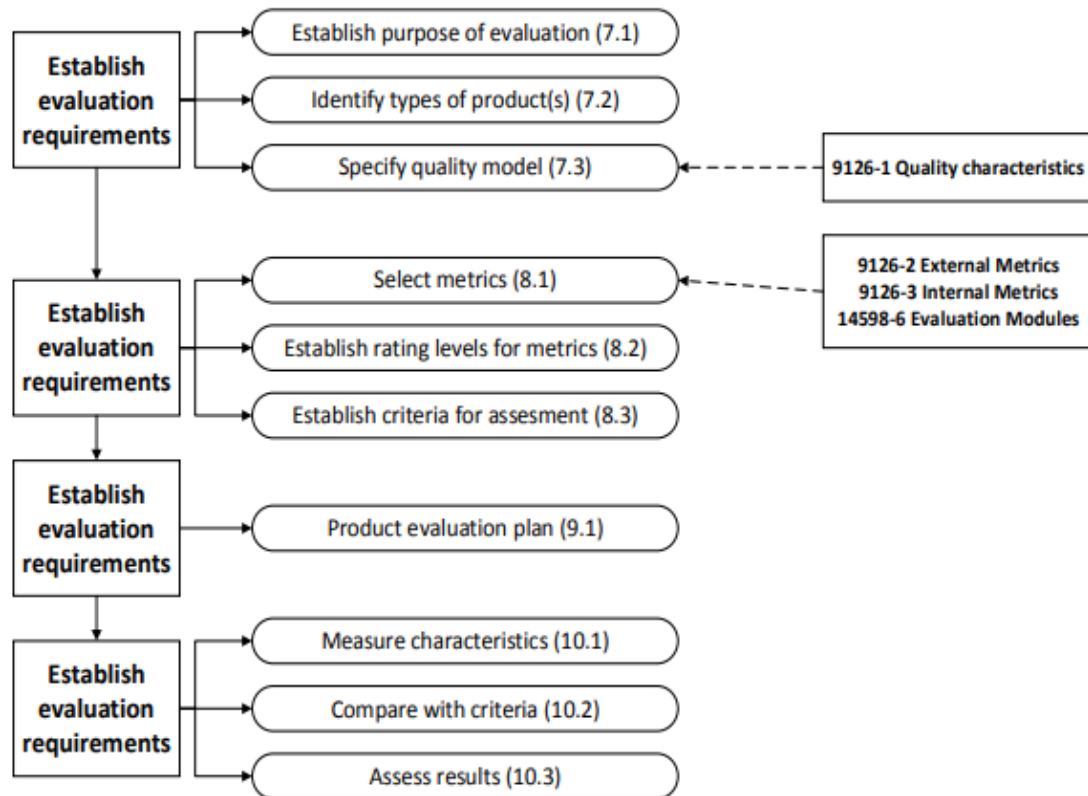
## Anexo 6: Particularidades de la pauta I/E 25000

Características	Subcaracterísticas
Funcionalidad	Adecuación
	Exactitud
Seguridad	Conformidad con la funcionalidad
	Resistencia al acceso
	Resistencia a la copia
	Facilidad de cifrado
	Resistencia a la falsificación
Interoperabilidad	Robustez
	Conformidad con la seguridad
	Compatibilidad de la OSI
	Compatibilidad del software
	Compatibilidad de los datos
	Trazabilidad
Fiabilidad	Conformidad con la interoperabilidad
	Madurez
	Tolerancia a los errores
	Recuperabilidad
	Conformidad con la fiabilidad
	Comprensibilidad
	Facilidad de aprendizaje
Usabilidad	Operabilidad
	Capacidad de ayuda
	Atractibilidad
	Conformidad de uso
	Comportamiento con el tiempo
Eficiencia	Utilización de recursos
	Conformidad con la eficiencia
	Capacidad de ser analizado
Mantenibilidad	Facilidad de cambio
	Estabilidad
	Facilidad de prueba
	Conformidad con la mantenibilidad
	Adaptabilidad
Portabilidad	Facilidad de instalación
	Coexistencia
	Reemplazabilidad
	Conformidad con la portabilidad

## Anexo 7: Relación entre las pautas del Evaluación de calificación y las pautas de apoyo a la calificación (ISO 14598-1, 2012)



**Anexo 8: Relación entre las pautas del Evaluación de calificación y las pautas de apoyo a la calificación (ISO 14598-1, 2012)**





## **Anexo 9: Evaluación de análisis de la calidad del programa mediante la calificación de las propiedades del Software**

### **P01.**

#### **OBJETIVO**

La evaluación tiene como finalidad analizar y verificar la concordancia del software producido con los requerimientos y propiedades explícitamente establecidos con los estándares de desarrollo prefijados y requerimientos implícitos no establecidos formalmente por el usuario. (Rivera y García, 2008)

#### **ALCANCE**

El alcance pretende garantizar la calidad de todo el sistema desde el punto de mantener las propiedades con que cuenta el software elaborado.

#### **PROCESO**

El proceso de análisis de calidad del software mediante su evaluación es abordado como auditoría interna de la siguiente forma (ISO 9001:2000):

- Establecer los elementos de un sistema de gestión de calidad

Se debe definir las políticas del sistema y la concordancia que tenga esta con los requerimientos y propiedades que afecten a la gestión de calidad.

- Documentar el sistema de calidad

Se debe describir el proceso, elaborar manual operativo, controlar los documentos y elaborar diagnóstico del sistema de calidad.

- Soporte del control de calidad

Se debe definir un plan de calidad que aborde sus objetivos, propiedades y mecanismos del sistema.

- Establecer mecanismos de revisión por propiedad.

Identificar propiedades de la calidad para darle respectivo seguimiento a partir del plan de control de calidad del programa.

- Mecanismo de control de calidad

Definir métricas y enfoques de procesos continuos para actividades técnicas tales como análisis de diseño, calidad y gestión del proyecto.

## **Anexo 10: Evaluación de medición del rendimiento del programa por medio del análisis del comportamiento del software en producción.**

### **P02.**

#### **OBJETIVO**

El objeto de esta evaluación es medir el rendimiento del software a través de algunas métricas de productividad que se pueden utilizar en la gestión del software. (Garmus y Herron, 2001).

#### **ALCANCE**

El alcance pretende asegurar métricas primarias basadas en la medida función del comportamiento de producción que rigen al sistema. (Garmus y Herron, 2001).

#### **PROCESO**

Para calificar el Evaluación de medición del rendimiento del programa por análisis de comportamiento del software en producción es el siguiente Garmus y Herron 2001).

1. Establecer y definir el comportamiento del software respecto a la producción de la organización.
2. Realizar análisis y diagnóstico al comportamiento del software en producción.
3. Desarrollar métricas de rendimiento de producción a partir del diagnóstico realizado.
4. Desarrollar un perfil documentado de cada medida de rendimiento.
5. Realizar una lista de check de las métricas que debe cumplir según el perfil establecido con relación al comportamiento del software en producción.

## **Anexo 11: Evaluación de calificación del proceso de elaboración del programa Software.**

### **P03.**

#### **OBJETIVO**

El objetivo de la evaluación es realizar la adecuada calificación de las evaluaciones en el desarrollo de la elaboración del software y así detectar todas las falencias que puedan afectar la calidad con la que se ejecutan los procesos.

#### **ALCANCE**

El alcance pretende asegurar una objetiva calificación de todos sus procesos y evitar fallas para poder generar confianza en sus clientes

#### **PROCESO**

Para calificar la evaluación de elaboración del software se debe constatar que sus actividades se lleven de la manera correcta y con los lineamientos de la ISO 9001-2015.

ETAPA: Exploración

Se necesita recolectar toda la información posible del proceso de elaboración del software y a partir de ello tener un criterio efectivo para realizar el diagnóstico de las actividades, por ello se lleva a cabo estas actividades.

- Proceso de observaciones de cada uno de los procesos que se lleva a cabo.
- Determinar inputs y outputs de información.
- Revisión de la documentación existente
- Identificar las vulnerabilidades y amenazas que está expuesto el sistema.
- Realización de análisis y apreciación de riesgos preliminares.

#### **ETAPA 2: IMPLEMENTACIÓN**

Posteriormente se ejecutará la implementación del proceso de calificación el cual debe ser lógicamente ordenado, clasificado, alcance y la forma en que se han de aplicar debe ser bajo condiciones que garanticen su óptima y utilización, contando con las siguientes actividades:

- Plan del proceso de calificación: Establecer los objetivos, alcances, metodología, recursos y cronograma de calificación.
- Elaborar el programa de calificación: Definir responsabilidades y actividades a desarrollar.
- Identificar y seleccionar los métodos, herramientas, instrumentos y Evaluaciones necesarios para la calificación de los procesos del desarrollo del software.
- Diseñar los papers de trabajo: listas de chequeo, cuestionarios, otros.
- Elaborar el plan de pruebas de análisis y ejecución.
- Presentar el informe final de calificación y documentación

### ETAPA 03: EVALUACIÓN

Se analizan los resultados obtenidos del proceso de calificación del desarrollo del software, presentando informe final con las posibles mejoras o hallazgos presentados en la ejecución del proceso.

## **Anexo 12: Evaluación del análisis del contexto de uso del programa Software.**

### **P04.**

#### **OBJETIVO**

Esta evaluación tiene como finalidad identificar y analizar el caso de uso el contexto de uso del programa y la identificación de factores internos y externos que permita conocer todos los factores que pueden ejercer alguna influencia de algún modo en el sistema. (Giner y Torres, 2008)

#### **ALCANCE**

Este proceso pretende implica la consideración de todos aquellos aspectos que la organización no puede controlar dentro del contexto de uso del programa.

#### **PROCESO**

El proceso de análisis del contexto consiste. (Giner y Torres, 2008) en:

- Identificar el tipo de modelo contexto de uso:  
Para ello se elabora un inventario de agentes involucrados en el sistema.
- Analizar iteraciones del contexto de uso:  
Realizar diagnóstico entre las iteraciones de agentes y necesidades del contexto de uso del programa.
- Diagnosticar caso de uso del contexto:  
A partir de los análisis e identificación del modelo de caso de uso del contexto que cuenta el programa se debe determinar la relación entre agentes y necesidades, así como los problemas con que cuenta estos y afectan al programa.

## **Anexo 13: Evaluación del análisis de medidas de la efectividad del Software.**

### **P05.**

#### **OBJETIVO**

El objetivo de esta evaluación es analizar de forma correcta la efectividad del software de forma integral y considerando características particulares.

#### **ALCANCE**

Esta evaluación pretende alcanzar estudio en primer lugar, lineamientos necesarios para identificar el nivel de efectividad, realizar su diagnóstico y un plan de evaluación para su correspondiente análisis de sus medidas de control de su efectividad.

#### **PROCESO**

Este proceso se fundamenta en normativas como: ISO 9001, ISO 25000

#### **ETAPA 1: EVALUAR LA EFECTIVIDAD**

- Establecer requisitos de evaluación que se deben tener en cuenta para controlar la efectividad.
- Seleccionar las métricas y criterios de evaluación
- Diseñar el plan y cronograma de actividades de medición.
- Ejecutar la evaluación

#### **ETAPA 2: REALIZAR EL DIAGNÓSTICO DE LA EFECTIVIDAD**

- Analizar los resultados de evaluación
- Jerarquizar las métricas por importancia y trascendencia en la efectividad del software
- Diagnosticar puntos débiles de la efectividad del software.
- Realizar reporte de análisis de las medidas de efectividad del software tomando en cuenta los criterios anteriores analizados.

## **Anexo 14: Evaluación de identificación de las características de la calidad del programa en función de la eficiencia.**

### **P06.**

#### **OBJETIVO**

El objetivo de la evaluación es identificar las características de calidad y el impacto del software respecto a su eficiencia por ello la importancia es inconmensurable de los fallos que se puedan encontrar en el sistema y facilitar las posteriores evaluaciones que se quieran realizar (NIST,2002).

#### **ALCANCE**

El alcance pretende mantener la valoración de la efectividad del software implementado o actual dentro de la organización. (NIST,2002)

#### **PROCESO**

Para identificar las características de la calidad en función de la eficiencia se considera la Norma ISO 14598(1999).

Se sigue el siguiente proceso:

Paso 01: Análisis de la calidad de la información

Para ello se toma en cuenta la evaluación del software en los siguientes indicadores.

- Durabilidad del proceso
- Efectividad flexibilidad
- Cantidad de información
- Confiabilidad de utilidad
- Utilización
- Disponibilidad de reusabilidad
- Usabilidad

Paso 02: Definir requisitos de control de calidad

Implica identificar el propósito y requisitos del software respecto a su efectividad.

### Paso 03: Medidas de utilización

Se emplea como mecanismo herramientas que permita evidenciar la efectividad del sistema

### Paso 04: Seleccionar y definir características

Tomando en cuenta los criterios analizados con anterioridad, se identifica las características desde sus herramientas, indicadores y requisitos de control de calidad del software.



## **Anexo 15: Evaluación de identificación de características de calidad del programa en función de la seguridad**

### **P07.**

#### **OBJETIVO**

La presente evaluación busca identificar las características de seguridad que tiene un programa informático (Calabrese et al, 2018)

#### **ALCANCE**

Es aplicable para cualquier software que se desee evaluar a nivel de seguridad. (Calabrese et al, 2018)

#### **PROCESO**

Para evaluar el software a nivel de seguridad se presentan las siguientes preguntas: (Calabrese et al, 2018)

<b>ID</b>	<b>PREGUNTA</b>	<b>SI</b>	<b>NO</b>
S1	¿La contraseña requiere poseer 8 caracteres de identificación?		
S2	¿La contraseña debe contener letras mayúsculas y minúsculas?		
S3	¿La contraseña debe contener número y letras?		
S4	¿La contraseña requiere características especiales?		
S5	¿El sistema utiliza como medida de seguridad la conexión mediante HTTPS?		
S6	¿La base de datos contiene datos encriptados?		
S7	¿Se puede acceder a las diferentes funcionalidades del sistema sin permiso?		

S8	¿Cualquier persona puede acceder a la base de datos?		
S9	¿Cualquier persona puede tener acceso al código del sistema?		
S10	¿Cualquier persona puede acceder al servidor físico?		
S11	¿Cualquier persona puede acceder al servidor remoto?		
S12	¿El sistema cuenta con redireccionamientos a sitios seguros?		
S13	¿El sistema necesita una confirmación de registro a través del uso de correo electrónico?		
S14	¿El sistema permite que se pueda modificar la base de datos por cualquier persona?		
S15	¿El sistema puede modificarse por cualquier persona a través del código del servidor de la aplicación?		
S16	¿El sistema permite inyecciones SQL?		
S17	¿El sistema cuenta con un historial de las acciones y actividades realizadas?		
S18	¿El sistema posee algoritmos descifrados de datos?		
S19	¿El sistema cuenta con un mecanismo de criptografía, como firma digital?		
S20	¿El sistema requiere confirmación en el momento de realizar una acción?		
S21	¿El sistema posee una protección a la hora de realizar una acción?		
S22	¿El sistema de aviso se puede acceder desde una ubicación desconocida?		
S23	¿El sistema informa vía mail las operaciones realizadas?		
S24	¿El sistema tiene un registro de fecha y hora de ingreso?		
S25	¿El sistema registra tanto el tipo de navegador como el sistema de operación que se realizan?		
S26	¿El sistema registra la dirección de IP del navegador que ingresa al sitio?		
S27	¿El sistema realiza una verificación de identidad a través de un certificado digital?		

S28	¿El sistema cuenta con un sistema de inspección y verificación los cuales consta de dos pasos?		
S29	¿Se requiere una clave de segundo nivel para lograr ingresar al sistema?		
S30	¿Se comprueba la identidad del usuario mediante datos biométricos?		
S31	¿El sistema realiza una comprobación de identidad a través de tarjeta de coordenadas?		
S32	¿Se realiza la comprobación de identidad mediante el uso de credenciales?		
S33	¿El sistema realiza una comprobación de identidad a través de la identificación y revisión de una firma electrónica?		

El resultado de dicha evaluación se calcula en porcentaje, considerando el total de aspectos a cumplir dividido entre el total de aspectos. (Calabrese et al, 2018)

**Anexo 16: Evaluación de identificación de características de calidad del programa en función de la usabilidad P08.**

**OBJETIVO**

La presente evaluación busca identificar las características de usabilidad que tiene un programa informático (Calabrese et al, 2018)

**ALCANCE**

Es aplicable para cualquier software que se desee evaluar a nivel de usabilidad. (Calabrese et al, 2018)

**PROCESO**

Para evaluar el software a nivel de usabilidad se presentan las siguientes preguntas: (Calabrese et al, 2018)

ID	PREGUNTA	SI	NO
U1	¿El sistema se cambia de colores y adecua según las necesidades de los usuarios?		
U2	¿Se puede cambiar el tamaño de letra de los textos del sistema?		
U3	¿El sistema puede realizar lectura de pantalla a través de la voz?		
U4	¿El sistema presenta textos difíciles de comprender?		
U5	¿El sistema tiene textos que no tengan trascendencia y sean irrelevantes dentro de sus Evaluaciones?		
U6	¿El sistema posee textos con faltas ortográficas?		
U7	¿El sistema permite eliminar acciones realizadas?		
U8	¿El texto tiene codificación y texto en diferentes idiomas?		

U9	¿El sistema tiene la opción de configurar el lenguaje del sistema de un idioma a otro?		
U10	¿La interfaz del sistema es amigable e interactúo de fácil entendimiento?		
U11	¿El sistema puede indicar las secciones que el usuario está accediendo?		
U12	¿El sistema puede mostrar las áreas que se han accedido hasta el momento?		
U13	¿Se puede realizar una acción con más de un término?		
U14	¿El sistema cuenta con una organización de páginas en un listado?		
U15	¿El sistema presenta consistencia de colores en todas sus secciones?		
U16	¿El sistema posee errores visuales? (Como menús solapados, despegables sin funcionar, textos en lugares no destinados a ello, etc.)		
U17	¿El sistema realiza operaciones y avisa mediante mensajes si se llevó con éxito/ sin éxito?		
U18	¿El sistema permite salir de cada sección utilizando los comandos atrás, cancelar, salir, volver, entre otros?		
U19	¿El sistema cuenta con atajos de teclado para poder utilizar accesos para diferentes funcionalidades?		
U20	¿El sistema cuenta con íconos de acceso a las diferentes funcionalidades?		
U21	¿El sistema tiene medidas de ayuda, donde detalla el error ocurrido y la solución?		
U22	¿El sistema detalla a través de un comentario si se comete un error?		
U23	¿El sistema muestra una alarma de error para prevenir que vuelva a ocurrir el mismo acontecimiento?		
U24	¿El sistema incide en mostrar mensajes de advertencia de error en el interfaz?		
U25	¿El sistema muestra los requerimientos y requisitos que se necesitan en cada uno de los campos?		

U26	¿El sistema indica cual son los campos que se deben llenar de forma obligatoria en el formulario?		
U27	¿El sistema permite ingresar caracteres o datos que difieran con las indicaciones? (Ej.: El sistema permite ingresar letras en un campo de DNI)		
U28	¿En el momento de completar un formulario, existe información pre cargada en algunos de sus campos? (Ej.: El campo país posee una lista despegable con diferentes países)		
U29	¿Las secciones del sistema contiene indicaciones de ayuda sobre las acciones que puedan realizar en el campo?		
U30	¿El sistema posee una sección de ayuda y descripción de las funciones?		
U31	¿El sistema posee una sección de preguntas frecuentes?		
U32	¿El sistema realiza comentario de pregunta, donde se detalla la pregunta si se logró resolver la inquietud de las dudas que encuentra como usuario?		
U33	¿El sistema provee un acceso rápido de ayuda e instrucciones dirigido a los usuarios?		

El resultado de dicha evaluación se calcula en porcentaje, considerando el total de aspectos a cumplir dividido entre el total de aspectos. (Calabrese et al, 2018)

## **Anexo 17: Evaluación de identificación de características de calidad del programa en función de la portabilidad**

### **P09.**

#### **OBJETIVO**

La presente evaluación busca identificar las características de portabilidad que tiene un programa informático (Calabrese et al, 2018)

#### **ALCANCE**

Es aplicable para cualquier software que se desee evaluar a nivel de portabilidad. (Calabrese et al, 2018)

#### **PROCESO**

Para evaluar el software a nivel de portabilidad se presentan las siguientes preguntas: (Calabrese et al, 2018)

<b>ID</b>	<b>PREGUNTA</b>	<b>SI</b>	<b>NO</b>
P1	¿El sistema cuenta con una aplicación móvil?		
P2	¿El sistema logra funcionar correctamente desde cualquier ordenador con diferentes características?		
P3	¿El sistema se puede utilizar adecuadamente desde un navegador como una Tablet?		
P4	¿El sistema puede utilizarse desde cualquier dispositivo con cualquier sistema operativo?		
P5	¿El sistema funciona correctamente desde cualquier navegador que tenga acceso a internet?		
P6	¿El sistema funciona correctamente desde cualquier navegador de un dispositivo móvil?		

El resultado de dicha evaluación se calcula en porcentaje, considerando el total de aspectos a cumplir dividido entre el total de aspectos. (Calabrese et al, 2018)

## **Anexo 18: Evaluación de identificación de las características de la calidad del programa en función del mantenimiento.**

### **P010.**

#### **OBJETIVO**

El objetivo de la evaluación es lograr que el software mantenga una calidad acorde los rápidos avances en la industria, tecnología y otros campos. Por ello el mantenimiento del software es la actividad principal ya que evita costos para desarrollar un nuevo producto y aumente su tiempo de vida útil (P. Bhatt y col, 2004)

#### **ALCANCE**

El alcance pretende mantener la calidad del mantenimiento de todo el software Además se centra en modelar el proceso esencial para el mantenimiento (S. Das y col, 2005)

#### **PROCESO**

Para la calidad del software y sus métricas: ISO 9126, IEEE 1061. • Para el mantenimiento del software: IEEE 1219.

El proceso debe seguir los siguientes pasos:

Procesos principales:

- Verificar documentación: Se analiza las actividades de planificación, diseño, desarrollo, producción de toda la documentación necesaria para distintos actores del proyecto tales como director, ingeniero y usuario.
- Revisar la gestión de la configuración: Se identifica el tipo de configuración, se controla y se pone como meta las fechas de calendario de entrega.
- Seguridad de calidad: Verificar que se cumpla los requisitos especificados y que sigan el plan establecido.
- Validación: Comprobar que el producto sirve para el uso proyectado.



- Revisión conjunta(demo): Tanto el que define el proyecto, como el que desarrolla y usa deben revisar y dar seguimiento al sistema.
- Auditoría: Es un control de externo de la calidad y mantenimiento como actividades complementarias a las anteriores ya mencionadas.

## **Anexo 19: Evaluación de identificación de la libertad de riesgo del software**

### **P011.**

#### **OBJETIVO**

El objetivo de la evaluación es lograr identificar, analizar y evaluar los riesgos que el software pueda tener. Además, Van Scoy exhibe que existen distintas actividades involucradas en la gestión de riesgos, ya que es un proceso continuo.

#### **ALCANCE**

El alcance pretende llegar a todos los procesos dentro del ciclo de vida de vida del software para poder lograr identificar los riesgos dentro del sistema Hall 1998 [02, 05, 06])

#### **PROCESO**

El proceso de identificar consiste en descubrir factores de riesgo antes que estos lleguen a convertirse en problemas y deriven en daños o pérdidas. El cual tiene el siguiente proceso (Bohem, 2015):

- Verificación:  
Se revisa la lista de requerimientos y sus características funcionales que debe tener.
- Análisis de conductores de decisiones:  
Se analiza los factores más determinantes y las consecuencias que pueden tener estos en el software.
- Análisis de supuestos:  
Se simula todas las situaciones que puedan ocurrir a partir de identificación de un posible riesgo.

- **Descomposición:**  
Se fragmenta y analiza tanto las posibilidades de riesgos, así como las decisiones que pueden acarrear a un problema que ponga en peligro la seguridad del sistema.
- **Identificación del riesgo:**  
De los pasos anteriores se selecciona los riesgos encontrados a partir de cada parámetro que se tomó en cada proceso.

## **Anexo 20: Evaluación de contratación de calificador independiente. P012.**

### **OBJETIVO**

El objetivo de esta evaluación es realizar la mejor contratación de un calificador independiente a la organización que pueda evaluar los diferentes procesos u/o desarrollos del software

### **ALCANCE**

Este proceso pretende determinar la mejor evaluación y elección de un calificador externo para la organización, el cual debe ser experto en la elaboración y análisis del software en sus diferentes procesos.

### **PROCESO**

Para este proceso se determinó las siguientes fases:

- **Fase 01:** Elaboración de requerimientos y conocimientos del calificador.
- **Fase 02:** Realizar el perfil competitivo del calificador el cual debe estar acuerdo a las necesidades y los criterios que deba evaluar.
- **Fase 03:**  
Se realiza publicación del puesto, donde debe contar con todos los criterios que deba cumplir el postulante.
- **Fase 04:**  
Se filtra las solicitudes tomando en cuenta todos los criterios del perfil competitivo que ya ha sido definido anteriormente.
- **Fase 05:**

Se realiza entrevista personal y evaluación de diagnóstico del postulante, donde se tomará como criterios a calificar las necesidades, requerimientos y competencias con que debe contar.

- Fase 06:

Se califica a los postulantes y selecciona el que cuente con la más alta puntuación y cumpla con el perfil competitivo del puesto, de caso contrario, se volverá a publicar el requerimiento del calificador externo.

## **Anexo 21: Evaluación de identificación del esquema de calidad del producto Software.**

### **P013.**

#### **OBJETIVO**

El objetivo de esta evaluación tiene como finalidad identificar todo el esquema de calidad que maneja el programa y así fortalecer los procesos que puedan brindar sus productos y servicios de alto nivel a la organización.

#### **ALCANCE**

Este proceso pretende alcanzar a dar seguimiento a todo el sistema del software para poder identificar su esquema de calidad. ISO 9004-2.

#### **PROCESO**

El proceso (ISO 9004-2) consiste en los siguientes pasos:

- Paso 01: Identificar las políticas de calidad
- Paso 02: Analizar los requerimientos generales empleados en la gestión de calidad
- Paso 03: Identificar los criterios de calidad en cada uno de los procesos del sistema.
- Paso 04: Comparar la estructura de calidad según el estándar ISO 9004-2.
- Paso 05: Revisión de control de procesos y documentación.

- Paso 06: Identificación de indicadores de control de calidad del sistema.
- Paso 07: Diagnóstico del esquema de calidad a partir de los sistemas de control de procesos, políticas, documentación e indicadores de calidad del sistema.

## **Anexo 22: Evaluación de identificación del modelo de diseño del producto Software**

### **P014.**

#### **OBJETIVO**

El objetivo de la evaluación es mejorar la calidad de un sistema de software a través del modelado de sus procesos y capturar las principales características del conjunto de actividades realizadas para obtener el software como producto. (AJ Walker et al, 2000).

#### **ALCANCE**

El alcance pretende llegar a todo el diseño del software e identificar el modelo que permite la creación del producto. (People CMM, 1995)

#### **PROCESO**

El modelo sigue los siguientes procesos (Liu et. Al, 2003):

- Paso 01: Identificar las funciones del modelo
- Paso 02: Analizar las funciones sobre el diseño del sistema.
- Paso 03: Revisar los requerimientos, objetos y especificaciones técnicas de los procesos.
- Paso 04: Analizar la estructura de las funciones y los modificadores funcionales.
- Paso 05: Definir los diferentes estados por los que pasa el sistema.
- Paso 06: Definir el diseño conceptual a partir de los resultados de los pasos anteriores.
- Paso 07: Seleccionar y definir el modelo del diseño proceso.

## **Anexo 23: Evaluación de planeación y gestión de desarrollo del producto Software.**

### **P015.**

#### **OBJETIVO**

El objetivo de la Evaluación es lograr la formalización de las actividades relacionadas con el desarrollo del software del sistema informático a partir de la planeación y gestión de los procesos. (J.M. Drake, 2008)

#### **ALCANCE**

El alcance pretende generar procesos sistematizados de planificación y gestión para el desarrollo de productos para la organización o una entidad. (J.M. Drake, 2008)

#### **PROCESO**

Un proceso de planificación y gestión de desarrollo de software busca definir su secuencia de actividades que deben ser seguidas por el equipo de trabajadores para generar un conjunto coherente de productos. (J.M. Drake, 2008)

- Actividad 01:  
A partir de las necesidades del cliente, se da el estudio del análisis de factibilidad, financiamiento, EDT, contrato y acta de inicio del desarrollo del software, para poder seleccionar las herramientas informáticas de gestión de software y configuración de campos personalizados según las necesidades de información y características de la organización.
- Actividad 02:  
Establecer frecuencia para los cortes de control escogido por el equipo, se elabora cronograma según la ejecución de indicadores que se debe planificar por el equipo de trabajo.
- Actividad 03:  
Establecer fechas de inicio y fin de los hitos de ejecución y tareas con los responsables para su cumplimiento. A estos se le debe atribuir los recursos necesarios, el tiempo, costos y calidad.
- Actividad 04:  
Las tareas ejecutadas deben recibir evaluación por parte de la dirección del desarrollo del software además el planificador y/o replanifica tareas a

un nivel más detallando los pasos establecidos para el cumplimiento del proyecto y así se genera de manera continua tanto la gestión como la planificación.

Las actividades del proceso ya antes mencionado deben incluir el cronograma, reportes y administración de acciones correctivas de forma adecuada (Ramos Blanco, y otros, 2011).

## **Anexo 24: Evaluación guía de calificación de producto Software para calificadores.**

### **P016.**

#### **OBJETIVO**

El objetivo de la evaluación es lograr la correcta evaluación de la manera más completa y sistemática para la aplicación en las diversas áreas de ingeniería de software. (Mohammad Zarour,2000)

#### **ALCANCE**

Pretende alcanzar la correcta guía de calificación a todo el sistema integrado del software desarrollado o incorporado por el software.

#### **PROCESO**

La guía de calificación estará basada en la metodología MARES (Metodología para la evaluación de procesos de softwares) esta se elaborará en cinco fases principales (Mohammad Zarour,2000):

1. Definir los criterios de evaluación:  
Se evalúa a través de un cuestionario los métodos, herramientas de apoyo, Evaluaciones, documentación y usuarios.
2. Diagnóstico de evaluación:  
Los desarrolladores del diseño, contentan el cuestionario y producen un informe de evaluación.
3. Análisis de documentos:
  - Se analiza los documentos por método SPA el cual se encarga de Adquirir datos a través de la revisión de documentos.

Aunque ha sido indicado que, para una evaluación simple, las entrevistas o la revisión de documentos es suficiente, todavía creemos que para una comprensión precisa se deben utilizar ambos.

- Estudiar la precisión de los hallazgos.

Los datos recopilados deben consolidarse en hallazgos precisos de acuerdo con criterios definidos. Para el método MARES, no se ha encontrado ningún estudio para definir y probar tal precisión de los hallazgos.

- Realización de la evaluación en 2-8 horas:

La herramienta de evaluación indicó que el método MARES toma 2 días de evaluación, análisis y presentación de resultados: 2 días es justificable para todas estas actividades. La duración mencionada de 2 a 8 horas en realidad especifica solo el tiempo de evaluación, no el tiempo para el análisis y la presentación de resultados. Por lo tanto, el método MARES debe estar sujeto a una duración de evaluación específica que no debe exceder las 8 horas.

- Asegurar la confiabilidad de los resultados de la evaluación.

Aunque la entrevista mencionó que la confiabilidad está garantizada a través de numerosos estudios de casos, no se ha encontrado ningún estudio que mida hasta qué punto los resultados de la evaluación producidos por estos estudios de casos son repetibles. Asegurar la repetibilidad dará confianza a la organización para confiar en los resultados de la evaluación y tomar decisiones adicionales para mejorar. mi. Garantice la integridad. No se encontró ningún estudio para discutir la integridad del método MARES que muestre que el método de evaluación ha tenido en cuenta los elementos esenciales para evaluar cada proceso en el alcance de la evaluación y proporcionar todos los resultados necesarios.

4. Análisis de las fortalezas y debilidades del método evaluado y elaboración del informe de evaluación.
5. Validar evaluación a partir de los hallazgos encontrados.
6. Presentar guía de calificación a partir del método implementado en el proceso de evaluación.

## **Anexo 25: Evaluación guía de codificación para desarrolladores P017.**

### **OBJETIVO**

El propósito de esta evaluación es elaborar un estándar de codificación como guía para que el desarrollador lea y evalúe su propio trabajo. (Pressman, 2015)

### **ALCANCE**

Esta evaluación es aplicable para todos los equipos de desarrolladores que pertenecen a la organización y están implementando la presente metodología de evaluación (Pressman, 2015)

### **PROCESOS**

- a) Diseño: la organización debe establecer un diseño consistente de desarrollo de código de software que evite errores innecesarios (Owens y Khazanchi, 2009)
- b) El código fuente debe utilizar sangría con espacios en lugar de tabulaciones, se recomienda que la sangría debe ser de 4 espacios para evitar problemas de lectura. (Estévez y Esteban, 2014)
- c) Limite la longitud de las líneas de código fuente debe ser de 55 caracteres
- d) Limitar la longitud (número de líneas) de las cláusulas a 24 líneas. (Estévez y Esteban, 2014)
- e) Ser consistente en el uso del espacio entre las comas, cada coma un espacio. (Estévez y Esteban, 2014)
- f) Poner cada subobjetivo en una línea separada
- g) Utilice el espacio vertical de forma coherente para mejorar la legibilidad
- h) Comente los archivos de origen, no solo los predicados dentro de ellos
- i) Utilice el diseño para hacer que los comentarios sean más legibles
- j) Anular los comentarios a la derecha del código, a menos que sean inseparables de las líneas en las que aparecen (Díaz et al, 2003)
- k) Considere usar los comentarios de manera constante como recordatorios
- l) Hacer que las cláusulas sean comprensibles de forma aislada
- m) Sangría de un nivel adicional entre la repetición y el corte correspondiente.
- n) Decida cómo quiere romper las cláusulas largas y los subobjetivos.



- o) Decidir cómo formatear disyunciones y if-then-elses
- p) Considere el uso de herramientas automatizadas para mejorar la legibilidad

## **Anexo 26: Evaluación guía de calificación de producto software para compradores**

### **P018.**

#### **OBJETIVO**

El objetivo de la evaluación es establecer una guía de criterios de evaluación de ofertas, propuestas para seleccionar la oferta, propuesta más útil que mejor cumpla con los requisitos y ofrezca la mejor relación calidad-precio para el comprador de software. (I/E 14598, I/E 25000)

#### **ALCANCE**

La evaluación es aplicable para todo comprador de software que desee evaluar un producto informático de manera rápida utilizando la presente metodología. (I/E 14598, I/E 25000)

#### **PROCESO**

Para que la evaluación se haga de manera correcta, el comprador debe tener en cuenta lo siguiente: (I/E 14598, I/E 25000)

- a) Los criterios de calificación serán proporcionados y adecuados al tipo, naturaleza, condiciones del mercado, complejidad, riesgo, valor y objetivo de la contratación.
- b) En la medida de lo posible, los criterios de calificación deben ser cuantificables
- c) Se incluirá los criterios de calificación completos y la forma específica en que se aplicarán.
- d) Sólo se aplicarán los criterios de calificación y todos los criterios de calificación indicados por la empresa.

e) una vez emitido el documento de adquisición estándar, cualquier modificación de los criterios de calificación se realizará únicamente mediante adendas.

f) los criterios de calificación se aplicarán de manera coherente a todas las Ofertas, propuestas presentadas.

Se tienen los siguientes alcances para la evaluación que se debe tener en cuenta: (I/E 14598, I/E 25000)

a) Costo: calificación del costo utilizando una metodología adecuada a la naturaleza de la contratación, que incluye:

1. precio de oferta / propuesta ajustada.

2. precio de oferta / propuesta ajustada más el costo corriente / recurrente durante la vida útil del activo sobre la base del costo actual neto (costos del ciclo de vida).

b) Calidad: calificación de la calidad mediante una metodología para determinar el grado en que los Bienes, Obras, Servicios distintos de consultoría o Servicios de consultoría cumplen o superan los requisitos.

c) Riesgo: criterios que mitigan el riesgo evaluado relevante.

d) Sostenibilidad: criterios que toman en cuenta los beneficios económicos, ambientales y sociales declarados en apoyo de los objetivos del proyecto, y pueden incluir la flexibilidad de la Propuesta para adaptarse a posibles cambios durante el ciclo de vida.

e) Innovación: criterios que permitan evaluar la innovación en el diseño y / o entrega de los Bienes, Obras, Servicios de No Consultoría o Servicios de Consultoría y que brinden a los Licitantes / Proponentes la oportunidad de incluir, cuando corresponda, en sus Ofertas / Propuestas.

El proceso de calificación debe comenzar inmediatamente después de la apertura de la Solicitud / Oferta / Propuesta con un examen preliminar para verificar la integridad general de la Solicitud / Oferta / Propuesta recibida según lo requieran antes de emprender su examen o evaluación detallados. (Pressman, 2015)

Toda Solicitud / Oferta / Propuesta debe estar sujeta a un examen preliminar. Esta acción permite al comité de evaluación identificar y rechazar Solicitudes / Licitaciones / Propuestas que están incompletas, inválidas o que sustancialmente no responden. (Pressman, 2015)

En el examen preliminar, se debe prestar atención a las deficiencias que, de aceptarse, proporcionarían ventajas injustas al Postulante / Licitante / Proponente. Se debe utilizar un buen juicio: por ejemplo, las simples omisiones o errores que posiblemente se deban a errores humanos no deben ser motivo de rechazo de la Solicitud / Oferta / Propuesta. Rara vez una Solicitud / Oferta / Propuesta es perfecta en todos los aspectos. (Pressman, 2015)

Por lo tanto, la justificación del rechazo debe basarse en la existencia de una o más deficiencias o desviaciones importantes que no se puede permitir rectificar o aceptar en ningún caso, y el rechazo sería justificado y sostenible. Una desviación material es aquella que:

- a) Tiene un efecto sobre la validez de la oferta.
- b) Se haya especificado en los documentos de licitación como motivo de rechazo de la oferta.
- c) Es una desviación de los términos comerciales o de las especificaciones técnicas en los documentos de licitación cuyo efecto sobre el precio de la oferta es sustancial pero no se le puede asignar un valor monetario.

Deben aplicarse las siguientes comprobaciones:

- a) Verificación: La validez de la Solicitud / Oferta / Propuesta requiere que todos los formularios pertinentes estén firmados por la persona o personas autorizadas. Si el Solicitante / Licitante / Proponente es una empresa conjunta, se debe presentar el acuerdo de empresa conjunta; Si el Solicitante / Licitante / Proponente es un agente, se debe proporcionar una autorización del proveedor o fabricante además de cualquier documentación requerida del proveedor o el propio fabricante.
- b) Elegibilidad: Todos los bienes y servicios se originarán en países de origen elegibles. En el caso de plantas y equipos, esta prueba de

elegibilidad se aplica solo al producto terminado ofrecido en la Solicitud / Oferta / Propuesta y a sus componentes principales y claramente identificables.

c) Garantía de Oferta / Propuesta: puede requerir la presentación de una Garantía de Oferta / Propuesta. De ser así, la garantía de la Oferta / Propuesta debe cumplir con los requisitos y debe acompañar a la Oferta / Propuesta.

d) Integridad de la Solicitud / Oferta / Propuesta: A menos que hayan permitido específicamente al Solicitante / Licitante / Proponentes cotizar solo para artículos seleccionados o solo para cantidades parciales de un artículo en particular: aquellos que no ofrecen todos los artículos requeridos normalmente deben considerarse que no responden. Sin embargo, en los contratos de obras, los precios faltantes para artículos de trabajo ocasionales se consideran incluidos en los precios de artículos estrechamente relacionados en otros lugares.

### ***Criterios de calificación técnicos y comerciales (Duque, 2016)***

Son los requisitos mínimos y / o máximos que normalmente se evalúan sobre la base de pasa / no pasa.

Los criterios de calificación (aprobado / reprobado) deben establecerse de tal manera que una evaluación pueda determinar si la Oferta / Propuesta responde sustancialmente a los requisitos técnicos y comerciales.

Los criterios de calificación se pueden aplicar tanto en una solicitud de oferta como en una solicitud de propuesta.

Las desviaciones sustanciales de los requisitos comerciales y las especificaciones técnicas son una base para el rechazo de una Solicitud / Oferta / Propuesta.

Como regla general, las desviaciones materiales son aquellas que, de ser aceptadas, no cumplirían con los propósitos para los cuales se solicita la Solicitud / Oferta / Propuesta, o impedirían una comparación justa con Solicitudes / Licitaciones / Propuestas que cumplen adecuadamente con:

- a) Negarse a asumir responsabilidades y pasivos importantes asignados, como garantías de desempeño y cobertura de seguros;
- b) Incapacidad para cumplir con el cronograma de entrega crítico o el cronograma de trabajo claramente especificado.
- c) Incumplimiento de los criterios de experiencia mínima especificados.
- d) No cumplir con los requisitos técnicos principales (por ejemplo, ofrecer tipos especificados completamente diferentes, capacidad de la planta muy por debajo del mínimo especificado, equipo que no puede realizar las funciones básicas para las que está destinado; y / o
- e) No licitar por el alcance del trabajo requerido (por ejemplo, para las obras completas o un paquete completo o un cronograma completo) como se indica y cuando no hacerlo se haya indicado como inaceptable.

La precalificación y la selección inicial son procesos que se utilizan para preseleccionar a los solicitantes en la adquisición de bienes, obras y servicios distintos de los de consultoría.

La evaluación de las calificaciones de un solicitante solo evalúa la empresa que presentó la solicitud de precalificación / selección inicial. La evaluación no toma en consideración las calificaciones de otras firmas tales como sus subsidiarias, entidades matrices, afiliadas, subcontratistas (que no sean subcontratistas especializados si se permite en el documento de Precalificación / Selección Inicial), o cualquier otra firma diferente de la firma que presentó el Solicitud de Precalificación / Selección Inicial.

### ***Priorización de criterios calificados (Duque, 2016)***

Los criterios de tipo calificado y los subcriterios, según corresponda, se priorizan, se asignan puntos de mérito y se ponderan de acuerdo con su importancia relativa para cumplir con los requisitos. El número de criterios y subcriterios debe mantenerse al mínimo esencial.

Las puntuaciones que se otorgarán a cada criterio y subcriterio se especifican en la hoja de datos. Para ayudar a priorizar los criterios calificados se puede utilizar una matriz de priorización simple.

- a) Empiece por crear una tabla según el ejemplo siguiente con cada criterio identificado como una letra en orden alfabético;
- b) Inserte los criterios en la matriz dos veces: uno en las filas horizontales y una vez en las columnas verticales.
- c) Tome cada pareja por turno. Pídale al equipo de evaluación que determine cuál de los dos están comparados entre sí es más importante en esta adquisición si comparamos el criterio "A" frente a "B". Si el equipo decide que "B" es lo más importante, inserte la letra "B" en el cuadro. Si el equipo de evaluación decide que ambos criterios son iguales, inserte A y B;
- d) Cuente el número total de "A", "B", C, etc.
- e) La letra con el conteo más alto es la más importante y la letra con el conteo más bajo es la menos importante;
- f) Priorizar como 1º, 2º, 3º, etc. sobre la base del recuento más alto para que cada criterio se clasifique frente al otro; y
- g) Discutir y acordar puntos o ponderaciones porcentuales.

### **Resultados** (Duque, 2016)

Los resultados deben presentarse en el formulario de evaluación. Si la Solicitud / Oferta / Propuesta no acepta la aceptación, las razones deben explicarse claramente en notas al pie o en un anexo, según sea necesario.

Dado que el rechazo en esta etapa excluye la Solicitud / Oferta / Propuesta de cualquier consideración adicional, se debe garantizar que la decisión de rechazarla sea justificable.

## **Anexo 27: Evaluación de diseño de arquitectura basada en factores.**

### **P019.**

#### **OBJETIVO**

El objetivo de la evaluación es presentar un paso a paso para diseñar una arquitectura basada en factores. El método recopila inquietudes, extrae factores que influyen de las inquietudes y las analiza por su influencia en los objetivos comerciales y los atributos de calidad del software. (Basili, 1984)

#### **PROCESO** (Dalla et al, 2020)

- a. Extraer las inquietudes de las partes interesadas por medio de entrevistas, la lectura de documentos y las experiencias personales
- b. Las inquietudes pueden estar muy relacionadas con las partes interesadas y puede ser beneficioso filtrar las inquietudes para que solo los hechos de las inquietudes entren en el análisis y no la relación con las partes interesadas.
- c. Para cada cambio en las preocupaciones, el arquitecto de software debe satisfacer nuevos atributos de calidad del software que debe lograr en el sistema con respecto a la arquitectura del software.
- d. Los factores de influencia son parte de las preocupaciones de las partes interesadas e incluyen tendencias, entorno técnico, experiencias previas y demandas del mercado, etc.
- e. Para ser proactivo, el arquitecto debe analizar continuamente las preocupaciones cambiantes y no solo esperar a que las preocupaciones se transformen en requisitos funcionales y no funcionales.
- f. Y finalmente priorizar los factores que influyen en la que se identifican los factores de influencia y sus influencias en los objetivos comerciales y los atributos de calidad del software se documentaron. En lo que se extraen aquellos factores que influyen que tienen un impacto positivo en el objetivo comercial priorizado.
- g. Los factores de influencia deben agruparse en torno a un impacto positivo en un objetivo empresarial específico. En este caso, el arquitecto puede intentar verificar con la dirección que este objetivo empresarial específico es el priorizado en la organización.

## **Anexo 28: Evaluación de gestión del cambio.**

### **P020.**

#### **OBJETIVO**

Proporcionar métodos y evaluaciones estandarizados para cumplir con los requisitos de gestión de cambios que respaldan las operaciones (I/E 25000, Metodología RUP)

#### **ALCANCE**

El alcance previsto de la evaluación de gestión de cambios es cubrir todos los sistemas y plataformas informáticos.

#### **PROCESO**

La gestión de cambios generalmente incluye los siguientes Evaluaciones: (I/E 25000, Metodología RUP)

1. Envío: durante este paso se identifica un cambio y se envía una solicitud de cambio. Se evalúa el cambio, incluida la determinación del nivel de prioridad del servicio y el riesgo del cambio propuesto; determinar el tipo de cambio y el proceso de cambio que se utilizará.
2. Planificación: planifique el cambio, incluido el diseño de implementación, la programación, el plan de comunicación, el plan de prueba y el plan de reversión.
3. Aprobación: obtener la aprobación de la gerencia para el Plan de cambios según sea necesario.
4. Implementación: Implementar el cambio.
5. Revisar: Comunicar y revisar el Plan de Cambio con sus compañeros y / o la Junta Asesora de Cambio con respecto a su éxito o fracaso y si el cambio resultó en una falla en el servicio, definir una estrategia de mitigación para sucesos futuros. Documente todos los aspectos del cambio.
6. Cerrar: etapa en la que la revisión se realiza correctamente y el cambio se cierra.



## **Anexo 29: Evaluación de estudio de los problemas del producto software**

### **P021.**

#### **OBJETIVO**

Objetivo de la evaluación es brindar un método de análisis de problemas y subproblemas de clases reconocidas, para luego identificar las estructuras de dominio y las interfaces en el mundo del problema. (Calabrese et al, 2018)

#### **ALCANCE**

El alcance pretende, sobre todo, ser un enfoque de análisis y estructuración de problemas más que de diseño de soluciones. Se centra en la atención del problema que requiere un esfuerzo consciente y una toma de conciencia. (Calabrese et al, 2018)

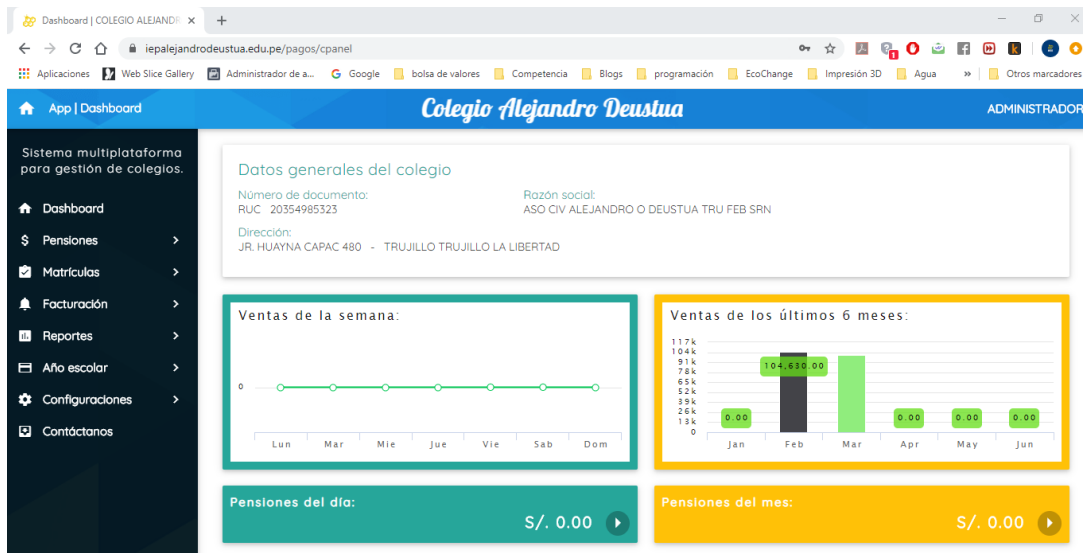
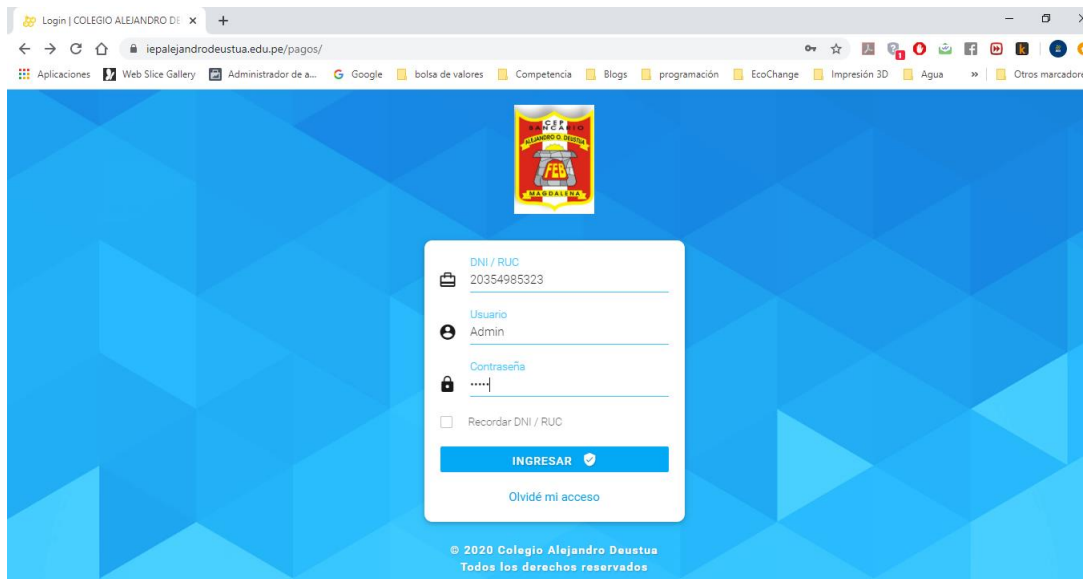
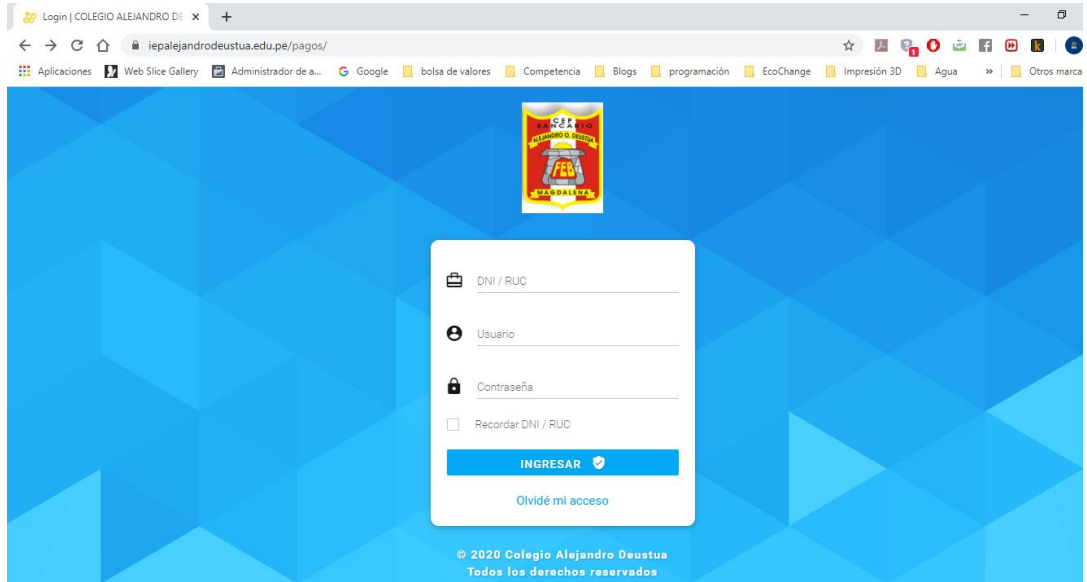
#### **PROCESO**

El análisis del problema debe seguir el siguiente proceso: (Calabrese et al, 2018)

1. Dominios
  - a) Descripciones
    - a. El requerimiento
    - b. Las propiedades del dominio
    - c. La especificación de la máquina.
  - b) Tipos de dominio
    - a. Máquinas
    - b. Dominios causales
    - c. Dominios léxicos
    - d. Dominios pugnables
2. Comportamiento simple
  - a) Respuestas de información simple
  - b) Pantalla de información simple
  - c) Piezas de trabajo simples
3. Descomposición del problema
  - a) Descomposición de afuera hacia adentro
  - b) Descomposición de adentro hacia afuera

- c) Descomposición por tiempo del subproblema
  - d) Reconocimiento de un marco compuesto estándar
  - e) Estandarización por marcos compuestos
4. Un problema más realista
- a) Enfoque de adentro hacia afuera
  - b) Una dificultad de conexión
  - c) Una digresión sobre los dominios modelo
  - d) Dificultades de dos identidades
  - e) Otra dificultad de conexión
  - f) Un problema de visualización de información
  - g) Componiendo la solución
5. Preocupaciones por subproblemas y preocupaciones por la composición
- a) La preocupación por la inicialización
  - b) La preocupación por las roturas
  - c) La preocupación por la confiabilidad
  - d) Interferencia y sincronización
  - e) Coherencia y contradicción

## Anexo 30: Imágenes del software evaluado



Registro de pensiones pagadas | Colegio Alejandro Deustua

Sistema multiplataforma para gestión de colegios.

Dashboard

Pensiones

Registrar pensiones

Pagos de pensiones

Pensiones pagadas

Otros pagos

Matrículas

Facturación

Reportes

Año escolar

Configuraciones

### Registro de pensiones pagadas

Pensiones del día: S/. 0.00

Pensiones del mes: S/. 0.00

Listo de todas las pensiones pagadas.

EDITAR PROCESO EXPORTAR REFRESCAR

Ver	Fecha	Cod. Boleta	Nombres y Apellidos	Concepto
	2020-03-04 14:22:59	EB01-0000522	ARIANA AGUILAR OLANO	matricula_marzo - 101475
	2020-03-04 14:22:33	EB01-0000521	HECTOR ORE QUIROZ	matricula - 101474
	2020-03-04 14:21:37	EB01-0000520	KARLA PRECIADO CRUZ	recuperacion - 101473
	2020-03-04 14:19:53	EB01-0000519	LESLIE PRECIADO CRUZ	recuperacion - 101472
	2020-03-04 14:19:26	EB01-0000518	KIARA PRECIADO CRUZ	recuperacion - 101471
	2020-03-04 14:19:01	EB01-0000517	MARIANA LUCANA PAREDES	matricula - 101470
	2020-03-04 14:18:31	EB01-0000516	MARIA ORTIZ GONZALES	abril a setiembre - 101469
	2020-03-04 14:17:31	EB01-0000515	XIOMARA CASTILLO SANCHEZ	certificado de estudios - 101468
	2020-03-04 14:16:17	EB01-0000514	ADRIANA DOMINGUEZ DE LA CRUZ	... - 101467
	2020-03-04 14:15:12	EB01-0000513	LESLIE PRECIADO CRUZ	julio a diciembre - 101466

Lista de alumnos | COLEGIO ALE | Colegio Alejandro Deustua

Sistema multiplataforma para gestión de colegios.

Dashboard

Pensiones

Matrículas

Nuevo alumno

Nueva matricula

Generar pensiones

Lista de alumnos

Lista de matriculados

Facturación

Reportes

Año escolar

### Lista de alumnos

En este apartado se visualizan todos los alumnos registrados en el sistema

NUEVO REFRESCAR

Acción	DNI	Alumno	Fecha Nacimiento	Dirección
	00000001	AGUILAR OLANO ARIANA	2020-02-15	....
	00000002	LEON REYES ANGHELY	2020-02-15	....
	00000003	ROMERO GUEVARA DARLET	2020-02-15	....
	00000004	SALAZAR MELGAREJO CORAL	2020-02-15	....
	00000006	PASTOR FLORES NAOMI	2020-02-15	....
	00000007	SANCHEZ CRUZ GENESIS	2020-02-15	....

Reporte por Mes | COLEGIO ALE | Colegio Alejandro Deustua

Sistema multiplataforma para gestión de colegios.

Dashboard

Pensiones

Matrículas

Facturación

Reportes

Alumnos inrosos

Reporte de ingresos

Reporte por mes

Año escolar

Configuraciones

Contactanos

### Reporte por Mes

En este apartado se visualizan todos los comprobantes de pago, ordenado por tipo y mes

EXPORTAR REFRESCAR

Tipo de Comprobante: Todos

Id Venta	Código Comprobante	Nombre Alumno	Fecha Comprobante	Tipo de Compi	Mes	Forma de Pago	Pensión	IGV	Total
530	EB01-0000522	ARIANA AGUILAR OLANO	2020-03-04 14:22:59	Boleta	matricula_marzo - ...	Depósito	550	0	550
529	EB01-0000521	HECTOR ORE QUIROZ	2020-03-04 14:22:33	Boleta	matricula - 101474	Depósito	280	0	280
528	EB01-0000520	KARLA PRECIADO CRUZ	2020-03-04 14:21:37	Boleta	recuperacion - 10...	Depósito	80	0	80
527	EB01-0000519	LESLIE PRECIADO CRUZ	2020-03-04 14:19:53	Boleta	recuperacion - 10...	Depósito	40	0	40
526	EB01-0000518	KIARA PRECIADO CRUZ	2020-03-04 14:19:26	Boleta	recuperacion - 10...	Depósito	80	0	80
525	EB01-0000517	MARIANA LUCANA PAREDES	2020-03-04 14:19:01	Boleta	matricula - 101470	Depósito	280	0	280
524	EB01-0000516	MARIA ORTIZ GONZALES	2020-03-04 14:18:31	Boleta	abril a setiembre - ...	Depósito	1680	0	1680
523	EB01-0000515	XIOMARA CASTILLO SANCHEZ	2020-03-04 14:17:31	Boleta	certificado de estu...	Depósito	50	0	50
522	EB01-0000514	ADRIANA DOMINGUEZ DE LA CRUZ	2020-03-04 14:16:17	Boleta	... - 101467	Depósito	1110	0	1110
521	EB01-0000513	LESLIE PRECIADO CRUZ	2020-03-04 14:15:12	Boleta	julio a diciembre - ...	Depósito	1680	0	1680

Lista de grados | COLEGIO ALEJ... x

iepalajandrodeustua.edu.pe/pagos/cpanel

Aplicaciones Web Slice Gallery Administrador de a... Google bolsa de valores Competencia Blogs programación EcoChange Impresión 3D Agua Otros marcadores

AÑO ESCOLAR | Lista de grados **Colegio Alejandro Deustua** ADMINISTRADOR

REFRESCAR

En este apartado se visualizan todos los grados registrados en el sistema

Acción	ID Grado	Grado y Sección	Estado	Año escolar
	1	1° A PRIMARIA	1	2020
	2	2° A PRIMARIA	1	2020
	3	2° B PRIMARIA	1	2020
	4	3° A PRIMARIA	1	2020
	5	3° B PRIMARIA	1	2020
	6	4° A PRIMARIA	1	2020

1-6 de 20

Porcentaje de mora | COLEGIO A... x

iepalajandrodeustua.edu.pe/pagos/cpanel

Aplicaciones Web Slice Gallery Administrador de a... Google bolsa de valores Competencia Blogs programación EcoChange Impresión 3D Agua Otros marcadores

CONFIGURACIONES | Porcentaje de mora **Colegio Alejandro Deustua** ADMINISTRADOR

Sistema multiplataforma para gestión de colegios.

Dashboard Pensiones Matriculas Facturación Reportes Año escolar Configuraciones Ajustes del sistema Contáctanos

Configure aquí el porcentaje de mora se aplicará en los pagos

PORCENTAJE DE MORA (%)  **GUARDAR**

ENVÍO AUTOMÁTICO DE FACTURAS  ¿Enviar automáticamente sus facturas?

Nuevo alumno | COLEGIO ALEJ... x

iepalajandrodeustua.edu.pe/pagos/cpanel

Web Slice Gallery Administrador de a... Google bolsa de valores Competencia Blogs programación EcoChange Impresión 3D Agua viajes Otros marcadores

MATRICULAS | Nuevo alumno **Colegio Alejandro Deustua** ADMINISTRADOR

Sistema multiplataforma para gestión de colegios.

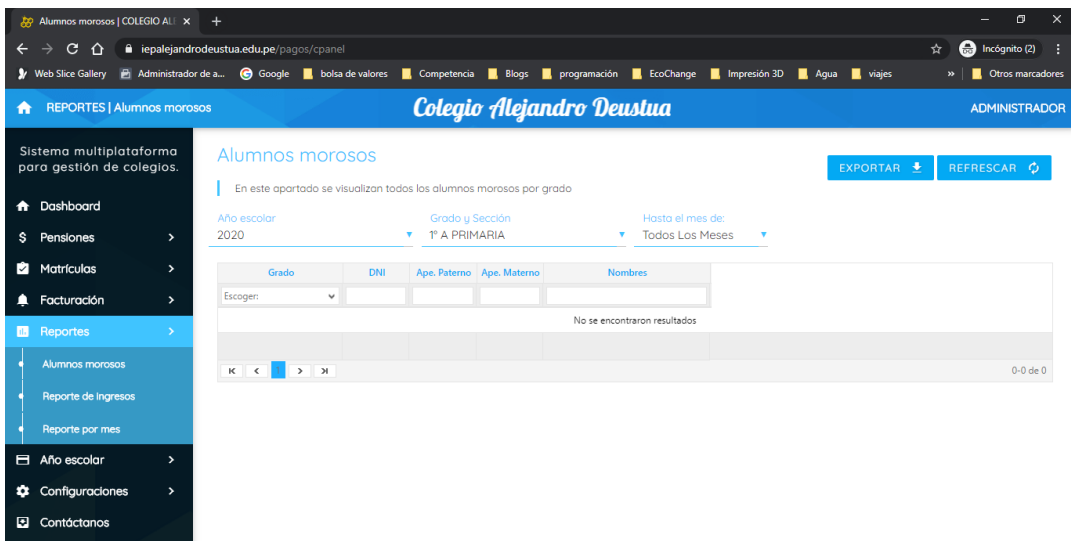
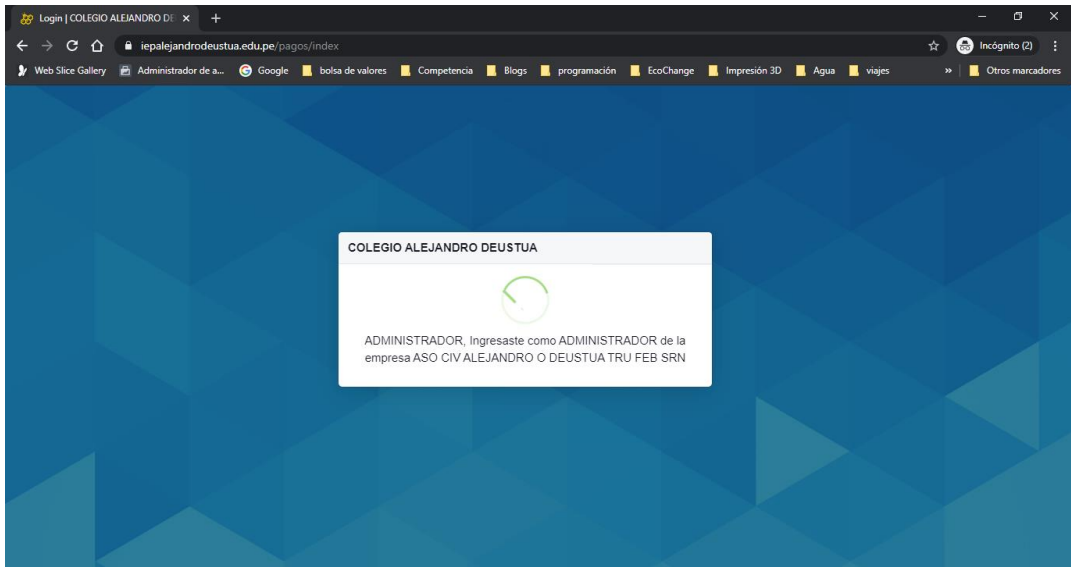
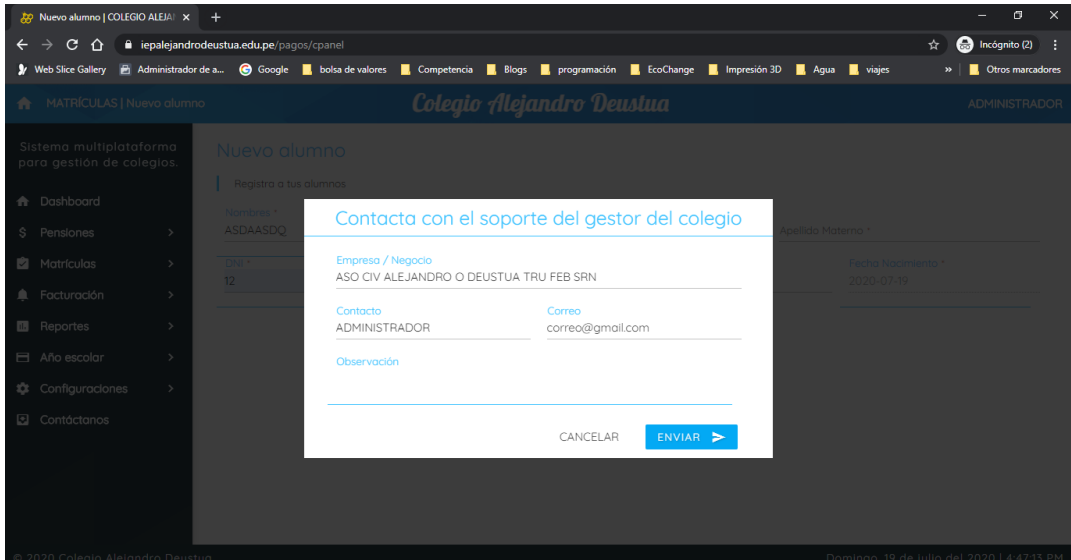
Dashboard Pensiones Matriculas Facturación Estado de comprobantes Numeración legal Reportes Año escolar Configuraciones Contáctanos

Registra a tus alumnos

Nombres \* ASDAASDQ Apellido Paterno \* Apellido Materno \*

DNI \* 12 Dirección Fecha Nacimiento \* 2020-07-19

**REGISTRAR ALUMNO**



**Colegio Alejandro Deustua** ADMINISTRADOR

Reporte por mes

- Año escolar
- Configuraciones
- Contactanos

**Información del certificado**

Este certifi. está destinado a los siguientes propósitos:

- Asegura la identidad de un equipo remoto
- Prueba su identidad ante un equipo remoto
- 1.3.6.1.4.1.1649.1.2.2.52
- 2.23.140.1.2.1

\* Para ver detalles, consulte la declaración de la entidad de ce

**Emitido para:** iepalejandrodeustua.edu.pe

**Emitido por:** cPanel, Inc. Certification Authority

**Válido desde:** 10/05/2020 **hasta:** 9/08/2020

Declaración del emisor

Aceptar

visualizan todos los alumnos morosos por grado

Grado y Sección: 1° A PRIMARIA | Hasta el mes de: Todos Los Meses

DNI	Ape. Paterno	Ape. Materno	Nombres
No se encontraron resultados			

0-0 de 0

Dashboard | COLEGIO ALEJANDRO DEUSTUA

Web Slice Gallery | Administrador de a... | Google | bolsa de valores | Competencia | Blogs | programación | EcoChange | Impresión 3D | Agua | viajes

**Colegio Alejandro Deustua** ADMINISTRADOR

Sistema multiplataforma para gestión de colegios.

- Dashboard
- Pensiones
- Matrículas
- Facturación
- Reportes
- Año escolar
- Configuraciones
- Contactanos

**Datos generales del colegio**

Número de documento: RUC 20354985323 | Razón social: ASO CIV ALEJANDRO O DEUSTUA TRU FEB SRN

Dirección: JR. HUAYNA CAPAC 480 - TRUJILLO TRUJILLO LA LIBERTAD

**Ventas de la semana:**

**Ventas de los últimos 6 meses:**

Mes	Ventas
Feb	104,630.00
Mar	0.00
Apr	0.00
May	0.00
Jun	0.00
Jul	0.00

**Pensiones del día:** S/. 0.00

**Pensiones del mes:** S/. 0.00

5:15 5:14

iepalejandrodeustua.edu.pe

### Colegio Alejandro Deustua

### Porcentaje de mora

Configure aquí el porcentaje de mora se aplicará en los pagos

PORCENTAJE DE MORA (%)  
 0.85

ENVÍO AUTOMÁTICO DE FACTURAS  
 ¿Enviar automáticamente sus facturas?

**GUARDAR**

ASO CIV ALEJANDRO DEUSTUA TRU FEB SRN 480 - TRUJILLO TRUJILLO LA LIBERTAD

JR HUAYNA CAPAC

#### Ventas de la semana:

Día	Ventas
Lun	0
Mar	0
Mie	0
Jue	0
Vie	0
Sab	0
Dom	0

#### Ventas de los últimos 6 meses:

Mes	Ventas
Feb	~100k
Mar	~100k
Apr	~100k
May	~100k
Jun	~100k
Jul	104,630.00

Jul  
Venta total del mes: S/. 0.00

Pensiones del día: S/. 0.00

Pensiones del mes: S/. 0.00

© 2020 Colegio Alejandro Deustua



### Anexo 31: Resultados de la evaluación inicial

Formato de Registro			
Investigador		Tipo de Prueba	Ficha de Registro
Empresa Investigada	Nedley Support SAC		
Nombre del indicador	Capacidad de respuesta		
Fecha de Inicio	01/09/2021	Fecha Final	01/10/2021

Ítem	Ruta probada	Número de solicitudes	Tiempo de respuesta (s)
1	/home	1	1.15436
2	/home	10	11.197292
3	/home	100	26.7068
4	/home/dashboard	1	1.569
5	/home/dashboard	10	14.4348
6	/home/dashboard	100	27.46972
7	/home/registroDeActividades	1	2.3486
8	/home/registroDeActividades	10	22.72086
9	/home/registroDeActividades	100	45.90645
10	/home/cartaDigital	1	2.81366
11	/home/cartaDigital	10	25.4343
12	/home/cartaDigital	100	55.479957
13	/home/DatosEmpresa	1	0.9689
14	/home/DatosEmpresa	10	8.62312
15	/home/DatosEmpresa	100	18.471211
16	/home/Categorías	1	1.5298
17	/home/Categorías	10	10.4735
18	/home/Categorías	100	20.7354
19	/home/Productos	1	2.34268
20	/home/Productos	10	21.552656
21	/home/Productos	100	44.352816

Formato de Registro			
Investigador		Tipo de Prueba	
Empresa Investigada	Nedley Support SAC		
Nombre del indicador	Disponibilidad		
Fecha de Inicio	01/09/2021	Fecha Final	01/10/2021

Objetivo	Indicador	Formula
Procedimiento de identificación de las características de la calidad del programa en función de la eficiencia.	Disponibilidad	$Disponibilidad = \frac{Horas\ Totales - Horas\ parada\ por\ mantenimiento}{Horas\ Totales}$

Ítem	Fecha	Horas Totales	Horas Parados Por Mantenimiento	Diferencia de horas
1	01/09/21	24	0	24
2	02/09/21	24	0	24
3	03/09/21	24	0	24
4	04/09/21	24	0	24
5	05/09/21	24	0	24
6	06/09/21	24	0	24
7	07/09/21	24	1	23
8	08/09/21	24	0	24
9	09/09/21	24	0	24
10	10/09/21	24	0	24
11	11/09/21	24	0	24
12	12/09/21	24	0	24
13	13/09/21	24	0	24
14	14/09/21	24	0	24
15	15/09/21	24	0.5	23.5
16	16/09/21	24	0	24
17	17/09/21	24	0	24
18	18/09/21	24	0	24
19	19/09/21	24	0.5	23.5
20	20/09/21	24	0	24
21	21/09/21	24	0	24
22	22/09/21	24	0	24
23	23/09/21	24	0	24
24	24/09/21	24	0	24

25	25/09/21	24	0	24
26	26/09/21	24	0	24
27	27/09/21	24	0	24
28	28/09/21	24	0	24
29	29/09/21	24	0	24
30	30/09/21	24	0	24
31	01/10/21	24	0	24

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Numero de vulnerabilidades		
<b>Fecha de Inicio</b>	01/09/21	<b>Fecha Final</b>	09/09/21

Objetivo	Indicador	Formula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la seguridad	Numero de vulnerabilidades		

Ítem	Fecha	Ruta del software	Número de Vulnerabilidades
1	01/09/21	/home	0
2	02/09/21	/home/dashboard	2
3	03/09/21	/home/registroDeActividades	5
4	04/09/21	/home/cartaDigital	10
5	05/09/21	/home/DatosEmpresa	0
6	06/09/21	/home/Categorias	5
7	07/09/21	/home/Productos	5
8	08/09/21	/home/Horarios	8
9	09/09/21	/home/MetodoDeEntrega	3

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Porcentaje de detección de defectos		
<b>Fecha de Inicio</b>	01/09/21	<b>Fecha Final</b>	11/09/21

Objetivo	Indicador	Formula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la seguridad	Porcentaje de detección de defectos	$X = \frac{\text{Número de defectos previstos}}{\text{Número de defectos encontrados}} \times 100$	

Ítem	Fecha	Ruta del software	Número defectos previstos	Número de defectos encontrados
1	01/09/21	/home	0	0
2	02/09/21	/home/dashboard	2	2
3	03/09/21	/home/registroDeActividades	4	5
4	04/09/21	/home/cartaDigital	8	10
5	05/09/21	/home/DatosEmpresa	5	5
6	06/09/21	/home/Categorías	5	5
7	07/09/21	/home/Productos	5	5
8	08/09/21	/home/Horarios	8	8
9	09/09/21	/home/MetodoDeEntrega	3	3
10	10/09/21	/home/Paletas	10	9
11	11/09/21	/home/MetodoDePago	5	5

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Tiempo de instalación		
<b>Fecha de Inicio</b>	01/09/21	<b>Fecha Final</b>	01/10/21

Objetivo	Indicador	Formula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la portabilidad.	Tiempo de instalación		

Ítem	Fecha	Entorno	Hora Inicio	Hora Fin	Tiempo total de instalación (m)
1	01/09/21	Servidor	15:20	15:30	10
2	02/09/21	Servidor	16:30	16:45	15
3	03/09/21	Servidor	10:45	11:00	15
4	04/09/21	Servidor	12:10	12:23	13
5	05/09/21	Servidor	10:45	11:00	15
6	06/09/21	Servidor	16:10	16:24	14
7	07/09/21	Servidor	13:30	13:45	15
8	08/09/21	Servidor	11:20	11:30	10
9	09/09/21	Servidor	15:20	15:30	10
10	10/09/21	Servidor	12:10	12:23	13
11	11/09/21	Servidor	11:20	11:34	14
12	12/09/21	Servidor	16:30	16:45	15
13	13/09/21	Servidor	13:20	13:30	10
14	14/09/21	Servidor	13:20	13:30	10
15	15/09/21	Servidor	16:30	16:45	15
16	16/09/21	Servidor	16:10	16:24	14
17	17/09/21	Servidor	16:10	16:24	14
18	18/09/21	Servidor	13:20	13:30	10
19	19/09/21	Servidor	15:20	15:30	10
20	20/09/21	Servidor	12:10	12:23	13
21	21/09/21	Servidor	13:20	13:30	10
22	22/09/21	Servidor	10:45	11:00	15
23	23/09/21	Servidor	16:10	16:24	14
24	24/09/21	Servidor	11:20	11:33	13
25	25/09/21	Servidor	15:20	15:30	10
26	26/09/21	Servidor	13:30	13:30	10

27	27/09/21	Servidor	13:30	13:30	10
28	28/09/21	Servidor	16:30	16:45	15
29	29/09/21	Servidor	13:30	13:40	10
30	30/09/21	Servidor	10:45	11:00	15
31	01/10/21	Servidor	15:20	15:30	10

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Tiempo de instalación		
<b>Fecha de Inicio</b>	01/09/21	<b>Fecha Final</b>	01/10/21

Objetivo	Indicador	Formula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la portabilidad.	Tiempo de actualización de versión		

Ítem	Fecha	Entorno	Tiempo total de actualización (m)
1	01/09/21	Servidor	5
2	02/09/21	Servidor	5
3	03/09/21	Servidor	5
4	04/09/21	Servidor	5
5	05/09/21	Servidor	2
6	06/09/21	Servidor	3
7	07/09/21	Servidor	4
8	08/09/21	Servidor	1
9	09/09/21	Servidor	5
10	10/09/21	Servidor	4
11	11/09/21	Servidor	3
12	12/09/21	Servidor	5
13	13/09/21	Servidor	2
14	14/09/21	Servidor	2
15	15/09/21	Servidor	1
16	16/09/21	Servidor	1
17	17/09/21	Servidor	5
18	18/09/21	Servidor	5
19	19/09/21	Servidor	5

20	20/09/21	Servidor	3
21	21/09/21	Servidor	2
22	22/09/21	Servidor	4
23	23/09/21	Servidor	2
24	24/09/21	Servidor	3
25	25/09/21	Servidor	5
26	26/09/21	Servidor	2
27	27/09/21	Servidor	3
28	28/09/21	Servidor	4
29	29/09/21	Servidor	5
30	30/09/21	Servidor	2
31	01/10/21	Servidor	3

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Tiempo de entrega de cambios		
<b>Fecha de Inicio</b>	01/09/21	<b>Fecha Final</b>	24/09/21



Objetivo	Indicador	Formula	Detalle
Procedimiento de identificación de la libertad de riesgo del SFTW.	Tiempo de entrega		

Ítem	Fecha	Cambio	Tiempo total de realización (h)
1	01/09/21	Vista Responsive Carta	5
2	02/09/21	Creación de cuentas, campo extra en formulario	1
5	05/09/21	Vista Responsive Categorías	4
7	07/09/21	Desactivar imagen categoría	1
9	09/09/21	Vista Responsive método Pago	4
11	11/09/21	Creación de página para contenido no disponible	1
13	13/09/21	Vista Responsive Productos	6
17	17/09/21	Vista Responsive Horarios	6
18	18/09/21	Creación de código QR para compartir	4
22	22/09/21	Vista Responsive Paleta de Colores	10
24	24/09/21	Mensaje de bienvenida por registro	3

Formato de Registro			
Investigador		Tipo de Prueba	Ficha Registro
Empresa Investigada	Nedley Support SAC		
Nombre del indicador	Cantidad de código		
Fecha de Inicio	01/09/21	Fecha Final	11/09/21

Ítem	Fecha	Ruta	Número total de líneas de código
1	01/09/21	/home	1500
2	02/09/21	/home/dashboard	2000
3	03/09/21	/home/registroDeActividades	566
4	04/09/21	/home/cartaDigital	463
5	05/09/21	/home/DatosEmpresa	356
6	06/09/21	/home/Categorias	865
7	07/09/21	/home/Productos	932
8	08/09/21	/home/Horarios	846
9	09/09/21	/home/MetodoDeEntrega	692
10	10/09/21	/home/Paletas	688
11	11/09/21	/home/MetodoDePago	378

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Tiempo medio de reparación		
<b>Fecha de Inicio</b>	01/09/21	<b>Fecha Final</b>	30/09/21

Ítem	Fecha	Hora del falló	Hora de la reparación	Total (m)
1	01/09/21	15:20	16:00	40
2	08/09/21	11:20	11:30	10
3	10/09/21	12:10	12:15	5
4	11/09/21	11:20	12:00	40
5	18/09/21	13:20	13:35	15
6	30/09/21	10:45	11:23	28

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Tiempo medio entre fallos		
<b>Fecha de Inicio</b>		<b>Fecha Final</b>	

Ítem	Fecha	Hora del falló	Total de horas (h)
1	01/09/21	15:20	165
8	08/09/21	11:20	164
10	10/09/21	12:10	48
11	11/09/21	11:20	23
18	18/09/21	13:20	171
30	30/09/21	10:45	285

### Anexo 32: Resultados de la evaluación posterior

Formato de Registro			
Investigador		Tipo de Prueba	Ficha de Registro
Empresa Investigada	Nedley Support SAC		
Nombre del indicador	Capacidad de respuesta		
Fecha de Inicio	01/09/21	Fecha Final	20/10/2021

Ítem	Ruta probada	Número de solicitudes	Tiempo de respuesta (s)
1	/home	1	1.0042932
2	/home	10	8.62191484
3	/home	100	21.36544
4	/home/dashboard	1	1.23951
5	/home/dashboard	10	10.970448
6	/home/dashboard	100	21.7010788
7	/home/registroDeActividades	1	2.066768
8	/home/registroDeActividades	10	18.176688
9	/home/registroDeActividades	100	40.397676
10	/home/cartaDigital	1	2.2790646
11	/home/cartaDigital	10	20.601783
12	/home/cartaDigital	100	46.60316388
13	/home/DatosEmpresa	1	0.726675
14	/home/DatosEmpresa	10	7.5021144
15	/home/DatosEmpresa	100	14.59225669
16	/home/Categorías	1	1.208542
17	/home/Categorías	10	9.111945
18	/home/Categorías	100	18.66186
19	/home/Productos	1	2.0849852
20	/home/Productos	10	16.38001856
21	/home/Productos	100	38.14342176

Formato de Registro			
Investigador		Tipo de Prueba	
Empresa Investigada	Nedley Support SAC		
Nombre del indicador	Disponibilidad		
Fecha de Inicio	20/09/2021	Fecha Final	20/10/2021

Objetivo	Indicador	Formula
----------	-----------	---------

Procedimiento de identificación de las características de la calidad del programa en función de la eficiencia.	Disponibilidad	$Disponibilidad = \frac{Horas\ Totales - Horas\ parada\ por\ mantenimiento}{Horas\ Totales}$
--	----------------	--

Ítem	Fecha	Horas Totales	Horas Parados Por Mantenimiento	Diferencia de horas
1	20/09/2021	24	0	24
2	21/09/2021	24	0	24
3	22/09/2021	24	0.25	23.75
4	23/09/2021	24	0	24
5	24/09/2021	24	0	24
6	25/09/2021	24	0	24
7	26/09/2021	24	0	24
8	27/09/2021	24	0	24
9	28/09/2021	24	0	24
10	29/09/2021	24	0	24
11	30/09/2021	24	0	24
12	1/10/2021	24	0	24
13	2/10/2021	24	0.16	23.84
14	3/10/2021	24	0	24
15	4/10/2021	24	0	24
16	5/10/2021	24	0	24
17	6/10/2021	24	0	24
18	7/10/2021	24	0	24
19	8/10/2021	24	0	24
20	9/10/2021	24	0	24
21	10/10/2021	24	0	24
22	11/10/2021	24	0	24
23	12/10/2021	24	0	24
24	13/10/2021	24	0	24
25	14/10/2021	24	0	24
26	15/10/2021	24	0	24
27	16/10/2021	24	0	24
28	17/10/2021	24	0	24
29	18/10/2021	24	0	24
30	19/10/2021	24	0.17	23.83
31	20/10/2021	24	0	24

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Numero de vulnerabilidades		
<b>Fecha de Inicio</b>	20/09/2021	<b>Fecha Final</b>	20/10/2021

Objetivo	Indicador	Formula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la seguridad	Numero de vulnerabilidades		

Ítem	Fecha	Ruta del software	Número de Vulnerabilidades
1	20/09/2021	/home	0
2	21/09/2021	/home/dashboard	0
3	22/09/2021	/home/registroDeActividades	2
4	23/09/2021	/home/cartaDigital	4
5	24/09/2021	/home/DatosEmpresa	0
6	25/09/2021	/home/Categorias	1
7	18/10/2021	/home/Productos	1
8	19/10/2021	/home/Horarios	2
9	20/10/2021	/home/MetodoDeEntrega	0

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Porcentaje de detección de defectos		
<b>Fecha de Inicio</b>	20/09/2021	<b>Fecha Final</b>	22/10/2021

Objetivo	Indicador	Formula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la seguridad	Porcentaje de detección de defectos	$X = \frac{\text{Número de defectos previstos}}{\text{Número de defectos encontrados}} \times 100$	

Ítem	Fecha	Ruta del software	Número defectos previstos	Número de defectos encontrados
1	20/09/2021	/home	0	0
2	21/09/2021	/home/dashboard	0	0
3	22/09/2021	/home/registroDeActividades	0	1
4	23/09/2021	/home/cartaDigital	1	3
5	24/09/2021	/home/DatosEmpresa	2	1
6	25/09/2021	/home/Categorias	2	1
7	18/10/2021	/home/Productos	2	0
8	19/10/2021	/home/Horarios	1	2
9	20/10/2021	/home/MetodoDeEntrega	0	0
10	22/10/2021	/home/Paletas	4	3
11	22/10/2021	/home/MetodoDePago	1	1

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Tiempo de instalación		
<b>Fecha de Inicio</b>	20/09/2021	<b>Fecha Final</b>	20/10/2021

Objetivo	Indicador	Formula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la portabilidad.	Tiempo de instalación		

Ítem	Fecha	Entorno	Hora Inicio	Hora Fin	Tiempo total de instalación (m)
1	20/09/2021	Servidor	14:20	14:28	8
2	21/09/2021	Servidor	15:30	15:41	11
3	22/09/2021	Servidor	09:45	09:57	12
4	23/09/2021	Servidor	11:10	11:19	9
5	24/09/2021	Servidor	09:45	09:57	12
6	25/09/2021	Servidor	15:10	15:20	10
7	26/09/2021	Servidor	12:30	12:41	11
8	27/09/2021	Servidor	10:20	10:27	7
9	28/09/2021	Servidor	14:20	14:27	7
10	29/09/2021	Servidor	11:10	11:21	11
11	30/09/2021	Servidor	10:20	10:30	10
12	1/10/2021	Servidor	15:30	15:41	11
13	2/10/2021	Servidor	12:20	12:27	7
14	3/10/2021	Servidor	12:20	12:28	8
15	4/10/2021	Servidor	15:30	15:42	12
16	5/10/2021	Servidor	13:10	13:20	10
17	6/10/2021	Servidor	14:10	14:22	12
18	7/10/2021	Servidor	10:20	10:28	8
19	8/10/2021	Servidor	12:20	12:28	8
20	9/10/2021	Servidor	11:10	11:21	11
21	10/10/2021	Servidor	12:20	12:28	8
22	11/10/2021	Servidor	11:45	11:57	12
23	12/10/2021	Servidor	14:10	14:21	11
24	13/10/2021	Servidor	13:20	13:29	9
25	14/10/2021	Servidor	15:20	15:27	7
26	15/10/2021	Servidor	15:30	15:37	7

27	16/10/2021	Servidor	10:30	10:38	8
28	17/10/2021	Servidor	14:30	14:41	11
29	18/10/2021	Servidor	11:30	11:38	8
30	19/10/2021	Servidor	09:45	09:56	11
31	20/10/2021	Servidor	10:20	10:28	8

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Tiempo de instalación		
<b>Fecha de Inicio</b>	20/09/2021	<b>Fecha Final</b>	20/10/2021

Objetivo	Indicador	Formula	Detalle
Procedimiento de identificación de las características de la calidad del programa en función de la portabilidad.	Tiempo de actualización de versión		

Ítem	Fecha	Entorno	Tiempo total de actualización (m)
1	20/09/2021	Servidor	4
2	21/09/2021	Servidor	4
3	22/09/2021	Servidor	4
4	23/09/2021	Servidor	4
5	24/09/2021	Servidor	2
6	25/09/2021	Servidor	2
7	26/09/2021	Servidor	3
8	27/09/2021	Servidor	1
9	28/09/2021	Servidor	4
10	29/09/2021	Servidor	3
11	30/09/2021	Servidor	2
12	1/10/2021	Servidor	4
13	2/10/2021	Servidor	2
14	3/10/2021	Servidor	1
15	4/10/2021	Servidor	1
16	5/10/2021	Servidor	1
17	6/10/2021	Servidor	4



18	7/10/2021	Servidor	4
19	8/10/2021	Servidor	4
20	9/10/2021	Servidor	2
21	10/10/2021	Servidor	1
22	11/10/2021	Servidor	3
23	12/10/2021	Servidor	2
24	13/10/2021	Servidor	2
25	14/10/2021	Servidor	4
26	15/10/2021	Servidor	2
27	16/10/2021	Servidor	2
28	17/10/2021	Servidor	3
29	18/10/2021	Servidor	4
30	19/10/2021	Servidor	2
31	20/10/2021	Servidor	2

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Tiempo de entrega de cambios		
<b>Fecha de Inicio</b>	20/09/2021	<b>Fecha Final</b>	13/10/2021

Objetivo	Indicador	Formula	Detalle
Procedimiento de identificación de la libertad de riesgo del SFTW.	Tiempo de entrega		

Ítem	Fecha	Cambio	Tiempo total de realización (h)
1	20/09/2021	Vista Responsive Carta	3.45
2	21/09/2021	Creación de cuentas, campo extra en formulario	0.69
4	24/09/2021	Vista Responsive Categorías	2.24
5	26/09/2021	Desactivar imagen categoría	0.51
6	28/09/2021	Vista Responsive método Pago	2.8
7	30/09/2021	Creación de página para contenido no disponible	0.6
8	2/10/2021	Vista Responsive Productos	4.08
9	6/10/2021	Vista Responsive Horarios	3.48
10	7/10/2021	Creación de código QR para compartir	2.28
11	11/10/2021	Vista Responsive Paleta de Colores	5.8

12	13/10/2021	Mensaje de bienvenida por registro	1.95
----	------------	------------------------------------	------

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Cantidad de código		
<b>Fecha de Inicio</b>	20/09/2021	<b>Fecha Final</b>	13/10/2021

Ítem	Fecha	Ruta	Número total de líneas de código
1	20/09/2021	/home	1155
2	21/09/2021	/home/dashboard	1780
3	24/09/2021	/home/registroDeActividades	419
4	26/09/2021	/home/cartaDigital	361
5	28/09/2021	/home/DatosEmpresa	270
6	30/09/2021	/home/Categorias	614
7	2/10/2021	/home/Productos	652
8	6/10/2021	/home/Horarios	710
9	7/10/2021	/home/MetodoDeEntrega	533
10	11/10/2021	/home/Paletas	523
11	13/10/2021	/home/MetodoDePago	335

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Tiempo medio de reparación		
<b>Fecha de Inicio</b>	25/09/2021	<b>Fecha Final</b>	20/10/2021

Ítem	Fecha	Hora del falló	Hora de la reparación	Total (m)
1	25/09/2021	14:30	14:55	24.8
2	28/09/2021	10:12	10:16	4
3	30/09/2021	8:48	8:51	3.05
4	9/10/2021	12:58	13:05	7
5	20/10/2021	11:38	11:56	18

Formato de Registro			
<b>Investigador</b>		<b>Tipo de Prueba</b>	Ficha Registro
<b>Empresa Investigada</b>	Nedley Support SAC		
<b>Nombre del indicador</b>	Tiempo medio entre fallos		
<b>Fecha de Inicio</b>	25/09/2021	<b>Fecha Final</b>	20/10/2021

Ítem	Fecha	Hora del falló	Total de horas (h)
6	25/09/2021	14:30	213.2
9	28/09/2021	10:12	59.5
11	30/09/2021	8:48	28.29
20	9/10/2021	12:58	230
31	20/10/2021	11:38	353

## **Anexo 33: Autorización para la realización y difusión de resultados de la investigación**

### **AUTORIZACIÓN PARA LA REALIZACIÓN Y DIFUSIÓN DE RESULTADOS DE LA INVESTIGACIÓN**

Por medio del presente documento, Yo Oscar Manuel Olaya Vásquez, identificado con DNI N° 70672139 y representante legal de la empresa Nedley Support SAC autorizo a Romualdo Molina, Jesús Antonio identificado con DNI N° 70449566 a realizar la investigación titulada: “Metodología de evaluación de la calidad del producto de software en base a las normas ISO/IEC 25000 e ISO/IEC 14598 y la metodología RUP” y a difundir los resultados de la investigación utilizando el nombre de Nedley Support SAC.

Lima, 21 de noviembre de 2021

FIRMA



---

Oscar Manuel Olaya Vásquez

DNI N° 70672139

Gerente General

Nedley Support SAC



**UNIVERSIDAD CÉSAR VALLEJO**

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

**Declaratoria de Originalidad del Autor**

Yo, ROMUALDO MOLINA JESUS ANTONIO estudiante de la FACULTAD DE INGENIERÍA Y ARQUITECTURA de la escuela profesional de INGENIERÍA DE SISTEMAS de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA ESTE, declaro bajo juramento que todos los datos e información que acompañan la Tesis titulada: "METODOLOGÍA DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO DE SOFTWARE EN BASE A LAS NORMAS ISO/IEC 25000 E ISO/IEC 14598 Y LA METODOLOGÍA RUP", es de mi autoría, por lo tanto, declaro que la Tesis:

1. No ha sido plagiada ni total, ni parcialmente.
2. He mencionado todas las fuentes empleadas, identificando correctamente toda cita textual o de paráfrasis proveniente de otras fuentes.
3. No ha sido publicada, ni presentada anteriormente para la obtención de otro grado académico o título profesional.
4. Los datos presentados en los resultados no han sido falseados, ni duplicados, ni copiados.

En tal sentido asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de la información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

<b>Nombres y Apellidos</b>	<b>Firma</b>
ROMUALDO MOLINA JESUS ANTONIO <b>DNI:</b> 70449566 <b>ORCID</b> 0000-0002-1652-9465	Firmado digitalmente por: JROMUALDOM12 el 12-12- 2021 07:46:02

Código documento Trilce: INV - 0520729