

Método de aseguramiento de calidad de la característica de fiabilidad durante el ciclo de desarrollo de software



Trabajo de grado
Modalidad: Trabajo de investigación

Leidy Vanesa Fernández Muñoz
104614020663

Luisa Fernanda Gómez Robles
104614010574

Director: PhD. Francisco José Pino Correa
Codirector: PhD. Sandra Lorena Buitrón Ruiz

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de Investigación y Desarrollo en Ingeniería Del Software - IDIS
Popayán, diciembre de 2022

AGRADECIMIENTOS

En primer lugar, agradezco a Dios y a la virgen por brindarme guía en el camino, en segundo lugar, a mis padres que con su sacrificio dieron la posibilidad a este logro, gracias por su apoyo incondicional y a pesar de que no fue fácil, se logró, a mis hermanos mayores que fueron mi ejemplo a seguir, de no desistir de los sueños y metas que se trazan, a mi hermanita menor que siempre me inspira a ser mejor persona. “Leidy Vanesa Fernández Muñoz”.

Agradezco en primer lugar a Dios por darme la oportunidad de haber podido culminar mis estudios de pregrado, a la Santísima Virgen que ha sido quien me cuidó y alivió en los momentos más difíciles: a mis padres Clariza y José Luis que nunca han dejado de apoyarme hasta verme feliz y realizada, porque como cada uno me lo ha expresado, su felicidad y recompensa siempre será verme feliz y cumpliendo todo lo que me propongo, a mi hija María Paula que ha sido mi compañera fiel desde que inicié mis estudios de pregrado y que ha sido testigo de todo lo que he luchado por llegar hasta aquí; a mis abuelos Lidia, Edylma y Efrén porque siempre han orado por mí y por todos mis logros incluyendo este y finalmente a toda mi familia que han aportado tiempo, oraciones, ánimos y los mejores deseos para mi. “Luisa Fernanda Gómez Robles”.

Agradecimientos a los compañeros Darly, Juan Camilo, Daniel Gómez, Daniel Bravo, Luis Gómez, Carlos Chapid, Francly Narvaez y Victor Rendon quienes nos acompañaron durante nuestro recorrido universitario y se convirtieron en amigos de la vida.

Finalmente agradecer a la Codirectora de este proyecto Sandra Lorena Buitrón, por su tiempo, paciencia, apoyo y guía en este largo proceso para cumplir con el último requisito para obtener nuestro título, además de todo lo aportado en nuestro paso por las materias impartidas por ella que han sido fundamentales en nuestro crecimiento profesional.

Tabla de contenido

Capítulo 1	8
Introducción	8
1.1 Planteamiento del problema	8
1.2 Justificación.....	11
1.3 Objetivos	12
1.3.1 Objetivo General.....	12
1.3.2 Objetivos Específicos.....	12
1.4 Metodología de Investigación	12
Capítulo 2 - Marco teórico y Estado del arte	13
2.1 Marco Teórico.....	13
2.1.1 Calidad de producto software	13
2.1.2 Medición	14
2.1.3 Verificación.....	14
2.1.4 Validación.....	14
2.1.5 Aseguramiento de la calidad.....	14
2.1.6 Técnicas de aseguramiento.....	14
2.1.7 Fiabilidad	15
2.2 Estado del arte	16
2.2.1 Búsqueda de información	16
2.2.2 Discusión.....	20
2.2.3 Aportes.....	21
Capítulo 3 - Método de aseguramiento de calidad de la característica de fiabilidad.....	23
3.1 Análisis conceptual de la fiabilidad	24
3.2 Aspectos para asegurar la calidad de la fiabilidad	24
3.3 Preguntas que permiten asegurar la fiabilidad en el proceso de desarrollo de software.	26
3.4 Técnicas de aseguramiento de calidad encontradas en la literatura consultada.....	35
3.5 Método de aseguramiento de calidad de la característica de fiabilidad	36
3.5.1 Levantamiento de requisitos de fiabilidad.....	43
3.5.2 Aseguramiento de la fiabilidad en etapa de análisis	43
3.5.3 Realizar ajuste del artefacto de análisis	44
3.5.4 Diseño de la solución	45
3.5.5 Aseguramiento de la fiabilidad en etapa de diseño	45
3.5.6 Realizar ajuste del artefacto de diseño.....	46
3.5.7 Codificación del producto	46

3.5.8	Aseguramiento de la fiabilidad en etapa de codificación	47
3.5.9	Realizar ajuste del artefacto de codificación.....	48
3.5.10	Ejecución los casos de prueba.....	48
3.5.11	Aseguramiento de la fiabilidad en etapa de pruebas	48
3.5.12	Realizar ajuste del artefacto de pruebas	49
3.5.13	Desplegar funcionalidades aprobadas	49
3.5.14	Aseguramiento de la fiabilidad en etapa de despliegue.....	50
3.5.15	Ajustar/Mejorar proyecto	50
Capítulo 4 - Evaluación del método		51
4.1	Estructura del Focus Group.....	51
4.2	Realización del focus group.....	52
4.2.1	Fase de planeamiento de la investigación	52
4.2.2	Fase de definición de grupos de discusión	55
4.2.3	Fase de conducción de la sesión de debate	56
4.2.4	Fase de análisis de información y reporte de resultados	58
4.3	Ajustes realizados con respecto a los puntos de mejora	65
4.4	Análisis de validez y limitaciones.....	67
4.5	Conclusiones de la evaluación preliminar a través del Focus group	68
Capítulo 5 - Conclusiones y trabajos futuros.....		70
Referencias Bibliográficas		72

Lista de tablas

Tabla 1. Artículos por editorial.....	16
Tabla 2. Relación de los artículos relevantes para la investigación.	17
Tabla 3. Cubrimiento de las subcaracterística de la fiabilidad en la literatura.	20
Tabla 4. Cubrimiento de las etapas del ciclo de desarrollo de software en la literatura.	20
Tabla 5. Elementos semánticos de Fiabilidad.....	25
Tabla 6. Preguntas propuestas para el aseguramiento de la fiabilidad en etapas de análisis.	27
Tabla 7. Diagramas base para etapa de diseño basados en el diagrama C4.	28
Tabla 8. Preguntas propuestas para el aseguramiento de la fiabilidad en etapas de diseño.	32
Tabla 9. Preguntas propuestas para el aseguramiento de la fiabilidad en etapa de codificación.	33
Tabla 10. Preguntas propuestas para el aseguramiento de la fiabilidad en etapa de pruebas.	34
Tabla 11. Preguntas propuestas para el aseguramiento de la fiabilidad en etapa de despliegue.....	34
Tabla 12. Preguntas para el aseguramiento de la fiabilidad en etapa de mantenimiento.	34
Tabla 13. Técnica de aseguramiento de calidad tradicionales.....	35
Tabla 14. Roles del método.....	37
Tabla 15. Escala de correspondencia para preguntas cuantitativas.	54
Tabla 16. Preguntas cuantitativas del focus group.	54
Tabla 17. Preguntas cualitativas del focus group.	54
Tabla 18. Agenda focus group.	54
Tabla 19. Participantes al focus group.....	56
Tabla 20. Resumen de comentarios por participantes en el focus group.....	58
Tabla 21. Respuestas preguntas cuantitativas.	60
Tabla 22. Respuestas preguntas cualitativas.	62
Tabla 23. Categorización de la información.....	64
Tabla 24. Ajustes realizados respecto a los puntos de mejora.	65

LISTA DE ILUSTRACIONES

Ilustración 1. Modelo conceptual del método para el aseguramiento de la fiabilidad. ...	24
Ilustración 2. Modelo del método propuesto, parte a.	38
Ilustración 3. Modelo del método propuesto, parte b.	39
Ilustración 4. Aseguramiento de la fiabilidad de una etapa.	40
Ilustración 5. Resultados globales del nivel de aseguramiento de la fiabilidad en el proyecto.	41
Ilustración 6. Interpretación del nivel de aseguramiento en la característica.....	42
Ilustración 7. Interpretación del nivel de aseguramiento en los aspectos que cubre la pregunta.....	42
Ilustración 8. Escala del nivel de aseguramiento.	42
Ilustración 9. Aseguramiento de calidad de la característica de fiabilidad en etapa de análisis.	44
Ilustración 10. Aseguramiento de calidad de la característica de fiabilidad en etapa de diseño.	46
Ilustración 11. Aseguramiento de calidad de la fiabilidad en etapa de codificación.....	47
Ilustración 12. Aseguramiento de calidad de la fiabilidad en etapa de pruebas.	49
Ilustración 13. Aseguramiento de calidad de la fiabilidad en etapa de despliegue.	50
Ilustración 14. Aseguramiento de calidad de la fiabilidad en etapa de mantenimiento..	50
Ilustración 15. Pregunta cuantitativa n° 1.	61
Ilustración 16. Pregunta cuantitativa n° 2.	61
Ilustración 17. Pregunta cuantitativa n° 3.	62
Ilustración 18. Promedio de preguntas cuantitativas.....	62

Capítulo 1

Introducción

Al pasar de los años la industria del software ha ido aumentando su impacto y relevancia en el día a día de la sociedad [1]; los productos software creados para propósitos cotidianos deben ser considerados bajo altos niveles de aseguramiento de calidad. Tales niveles refieren especialmente a dos tipos de requisitos, los funcionales y los no funcionales, entre los cuales se destaca la fiabilidad del producto software [2].

La fiabilidad, como una característica de calidad está definida en términos de sub características y elementos semánticos no visibles, esto es, conceptos que están implícitos en dicha definición y que deben descubrirse de manera temprana para que puedan ser tenidos en cuenta dentro del desarrollo de un producto software [3]. En general, las características de calidad como la fiabilidad y otras que se pueden conocer en el modelo de calidad propuesto por la ISO 25010 [4] son poco exploradas dentro del ciclo de desarrollo de software, y en mayor porcentaje en las etapas de diseño, codificación y despliegue [5].

En este capítulo se presenta el planteamiento del problema específico para este contexto, centrado en la característica de fiabilidad. Así mismo, se describen los argumentos considerados al momento de la realización del presente trabajo de investigación, los objetivos planteados, y la estrategia de investigación utilizada para el desarrollo del mismo.

1.1 Planteamiento del problema

Actualmente la industria de software está en aumento debido a la creciente demanda de productos para la vida diaria [1], esta realidad exige por tanto que estos productos se construyan considerando el uso de técnicas variadas que aseguren su calidad frente a ésta demanda del mercado. Así mismo, es relevante el aseguramiento de calidad durante el ciclo de desarrollo de software [6], dado que es un proceso esencial para alcanzar dicha calidad del producto.

Con respecto al aseguramiento de calidad, este permite identificar defectos en el software, para luego poder corregirlos [1],[7] a través de un proceso gradual. Los profesionales de la Industria del software llevan a cabo este proceso, con el fin de garantizar que cada

componente sea creado cumpliendo con todos los requisitos de calidad [8]. De modo que, al momento de la entrega del producto a los clientes, se logre evitar el sobre costo de mantenimiento por defectos no identificados [6]. Del mismo modo, las técnicas tradicionales de aseguramiento de la calidad del software se encuentran limitadas por algunos factores como el tiempo y el costo de desarrollo [1].

Entre los aspectos que cubre el aseguramiento de la calidad se encuentran los requisitos no funcionales que hacen referencia a cómo debe comportarse el sistema, y entre los cuales están: fiabilidad, eficiencia, portabilidad, mantenibilidad, compatibilidad y usabilidad [9]. Estos comportamientos son capacidades que se definen al producto software durante su desarrollo. Sin embargo, al ser éste realizado por personas, es inevitable que contengan defectos o errores [6], los cuales en algún momento pueden conllevar a muchos más errores [10], y por tanto afectar la calidad y confianza en el producto. De acuerdo con [10] y [11], entre los requisitos no funcionales más relevantes, la fiabilidad es una característica de calidad que debe trabajarse con mayor esfuerzo dado que esta permite medir la probabilidad o confianza en el comportamiento del software en su entorno de ejecución. Además, es un factor importante que se ocupa de los errores que están presentes en un sistema. Para esta característica se han definido algunos mecanismos de ingeniería de software como la prevención de fallas, detección de fallas, tolerancia a fallas y eliminación de fallas, que están presentes en el ciclo del desarrollo garantizando efectividad y eficacia [5].

Actualmente los requisitos de fiabilidad de un producto software que se encuentre en proceso de desarrollo no son tenidos en cuenta o son analizados inadecuadamente lo cual, hace que esta característica no sea aprovechada en su totalidad dentro de este proceso [8], por lo que es importante incluirlos desde las primeras etapas del desarrollo, logrando requisitos claros, completos e inequívocos que conlleven a la implementación de la fiabilidad en el sistema [12]. En cuanto al aseguramiento de calidad se pueden encontrar distintas técnicas durante las diferentes etapas del ciclo de desarrollo de software para requisitos funcionales, entre ellas:

- En etapa de análisis, pese a que no hay trabajos que estudien la fiabilidad, se encuentran algunos que se enfocan en la prevención de fallas a través de una recopilación y análisis críticos de los requisitos del sistema, para garantizar el refinamiento adecuado de las funcionalidades definidas [5],[13]. Sin embargo, se dejan por fuera del análisis aspectos como la madurez, la disponibilidad y la capacidad de recuperación [5]. Este último artículo no presenta técnicas de aseguramiento de calidad para esta etapa.
- En la etapa de codificación, se realiza la implementación del código de los requisitos especificados en la etapa de análisis [14]; los aspectos que no hayan sido analizados adecuadamente o de manera completa en la etapa de análisis no serán implementados.
- En la etapa de pruebas [15], se analizan posibles riesgos, y el rendimiento de un producto software [8], lo que permite evaluar un sistema o componente por medios manuales o automáticos para verificar que satisface los requisitos [16]. Frente a esto, en los proyectos de desarrollo de software las pruebas corresponden al 50% del presupuesto [17]. Sin embargo, no abarcan la totalidad de aspectos que se requieren para un aseguramiento de calidad más completo como: seguridad, privacidad y fiabilidad [18].

Por otro lado, entre las técnicas de pruebas han surgido las de aprendizaje automático (MLS) que predicen defectos a través del entrenamiento de datos en software similares y es así como se detectan módulos propensos a errores [1],[19], [20]. Algunas de las técnicas de clasificación o agrupación para la evaluación de fiabilidad son: Bayes Naive, Árboles de decisión, Máquinas de vectores de soporte y Red neuronal recurrente [10]. Pese a esto, las técnicas de clasificación automática requieren un rendimiento muy alto que depende de grandes volúmenes de datos, no obstante, las organizaciones no cuentan con este volumen por lo que los algoritmos pueden no ser capaces de llevar a cabo una generalización [1],[21].

- Finalmente, en la etapa de despliegue, la replicación de recursos software y hardware apoya la característica de fiabilidad, especialmente la tolerancia a fallos y la detección de errores, aun así, la implementación de estos dos aspectos genera altos costos [22],[23].

De acuerdo con este escenario se evidencia la necesidad de ampliar los mecanismos de aseguramiento de la fiabilidad dentro de la calidad del software, para todo el ciclo de

desarrollo de software, a través de alguna solución o propuesta que permita dar respuesta a la pregunta de investigación ¿Cómo asegurar la característica de la fiabilidad del producto durante todo el ciclo de desarrollo de software?

1.2 Justificación

La fiabilidad del software es una base importante para las actividades de toma de decisiones relacionadas con los equipos de desarrollo de software [10], por lo cual se hace necesario hacer un seguimiento temprano y continuo de esta característica y permitir a cada uno de los interesados tomar decisiones más efectivas antes, durante y después del desarrollo del producto software. Además, el encontrar defectos en las primeras etapas del desarrollo de software logra minimizar los recursos asignados para corregir el problema [24] y permiten mejorar la calidad del sistema [25], adicionalmente evita posibles sobrecostos y sobreesfuerzos que se invierten en la solución del error o posibles fallos derivados del mismo [11]. Más aún, los niveles de mejora de procesos reducen la probabilidad de que se produzcan defectos de gran gravedad y, por tanto, disminuyen la incidencia de fallos funcionales catastróficos o importantes [12].

Teniendo en cuenta que la característica de fiabilidad es uno de los principales atributos del aseguramiento de calidad del software [10], es fundamental determinar un método que contemple la característica de fiabilidad desde etapas tempranas del ciclo de desarrollo y que busque mejorar el proceso de aseguramiento de la calidad del desarrollo software. Así mismo, es de gran importancia que éste se encuentre estandarizado y que permita a las empresas del sector tener una guía útil para asegurar la calidad de dicha característica, llevando a cabo un método adaptado no solamente para etapas prontas sino para cada una de las etapas del desarrollo.

1.3 Objetivos

1.3.1 Objetivo General

Definir un método¹ para el aseguramiento de la calidad de la característica de fiabilidad de un producto software, que permita incorporar esta característica desde etapas tempranas del desarrollo de software.

1.3.2 Objetivos Específicos

- Establecer los mecanismos de aseguramiento de la calidad de un producto software a partir de una revisión de la literatura.
- Diseñar un método para el aseguramiento de la calidad de la fiabilidad aplicable desde etapas tempranas del ciclo de desarrollo de software considerando los mecanismos previamente identificados.
- Evaluar de manera preliminar la utilidad del método de aseguramiento de calidad de la fiabilidad propuesto a través de la estrategia de investigación de focus group.

1.4 Metodología de Investigación

La estructura de proyecto multi-ciclo con bifurcación será la utilizada para alcanzar los objetivos planteados del presente trabajo. Esto gracias a que permite adaptarse a la aparición de nuevos problemas y/o sub problemas durante el transcurso del ciclo que se esté llevando a cabo [26]. Los ciclos planteados para la presente investigación se definen a continuación:

Ciclo conceptual:

- a) Realizar una búsqueda en la literatura de técnicas u otras propuestas de investigación que aborden el aseguramiento de calidad para un producto software.
- b) A partir de los trabajos encontrados, identificar, analizar y describir los aspectos que no se han tenido en cuenta en las técnicas que podrían fortalecer a la característica de fiabilidad durante las etapas del ciclo de desarrollo de software.

¹ Modo ordenado y sistemático de proceder para llegar a un resultado o fin determinado [RAE].

Ciclo metodológico:

- a) Identificar los conceptos claves a ser considerados para proponer el método de aseguramiento de calidad.
- b) Definir los componentes que constituyen el método de aseguramiento de calidad para la fiabilidad, considerando las técnicas de aseguramiento de la calidad previamente identificadas.
- c) Establecer el método de aseguramiento integrando los conceptos y componentes.

Ciclo de evaluación:

- a) Diseñar el focus group estableciendo los elementos y procedimientos que serán aplicados al debate.
- b) Definir el grupo de discusión, identificando los perfiles de los participantes al debate.
- c) Desarrollar el focus group, ejecutando los procedimientos definidos y teniendo en cuenta los roles.
- d) Analizar los resultados obtenidos en el focus group y generar conclusiones que determinen la validez o no del método de aseguramiento propuesto.

Ciclo de documentación:

- a) Simultáneamente a los anteriores ciclos, se realizará la elaboración de la monografía del trabajo de grado en la cual se recopilará toda la información durante la ejecución del trabajo.
- b) Elaboración de un artículo científico en borrador para la publicación de los resultados obtenidos.

Capítulo 2 - Marco teórico y Estado del arte

2.1 Marco Teórico

2.1.1 Calidad de producto software

La calidad del producto software es de gran relevancia dentro del desarrollo de software, por tal motivo, actualmente se encuentran una familia de normas ISO/IEC 25000 basadas en ISO/IEC 9126 y en ISO/IEC 14598, cuyo objetivo es la creación de un marco de trabajo común para evaluar la calidad del producto software [2]. Esta familia se encuentra

compuesta por cinco divisiones, entre ellas está el modelo de calidad ISO/IEC 2501n en la cual se encuentra la norma ISO/IEC 25010 en donde define la calidad de producto software frente a las características y subcaracterísticas del producto y su uso [4].

2.1.2 Medición

Es un proceso que recopila, analiza e informa datos objetivos para respaldar una gestión eficaz y demostrar la calidad de los productos, servicios y procesos [27].

2.1.3 Verificación

Es un proceso que proporciona evidencia objetiva de que el sistema o los elementos del sistema satisfacen sus requisitos y características especificadas [27].

2.1.4 Validación

Es un proceso que proporciona evidencia objetiva de que el sistema, cuando está en uso, cumple con sus objetivos comerciales o de misión y los requisitos de las partes interesadas, logrando el uso previsto en su entorno operativo previsto [27].

2.1.5 Aseguramiento de la calidad

Según la ISO 9000 el aseguramiento de la calidad hace parte de la gestión de la calidad, la cual garantiza la confianza en el cumplimiento de los requisitos de calidad [28].

El aseguramiento de calidad de procesos y productos apoya para realizar entregas de productos de alta calidad, al proporcionar al equipo del proyecto software y a los directivos la visibilidad adecuada del trabajo realizado a lo largo de la vida del proyecto [29].

El objetivo del aseguramiento de calidad es proporcionar confianza en que se efectuaran los requisitos de calidad. Se realiza un análisis proactivo de los procesos y resultados del ciclo de vida del proyecto para respaldar que el producto resultante tendrá la calidad deseada y que se seguirán las políticas y procedimientos de la organización y del proyecto. [27].

2.1.6 Técnicas de aseguramiento

Es un conjunto de actividades preestablecidas y sistematizadas, aplicadas al sistema de calidad, que han demostrado ser necesarias para dar confianza adecuada de que un

producto o servicio satisfará los requisitos para la calidad [23]. Dentro de las técnicas según la ISO 1028 [30] se encuentran:

- Auditorías: una examinación independiente del producto, o procesos de software realizada por alguien externo para evaluar cumplimiento de especificaciones, estándares, acuerdos contractuales, y otros criterios.
- Revisión: proceso o reunión durante la cual un producto o proceso de software es presentado a personal involucrado en el proyecto para examinación, retroalimentación y aprobación.
- Inspecciones: examinación visual de un producto de software para detectar e identificar anomalías.
- Revisión técnica: evaluación sistemática de un producto de software realizada por un equipo calificado que examina la conveniencia del producto de software para sus usuarios e identifica discrepancias de especificaciones y estándares.
- Recorridos: técnica de análisis estático, en la que personal interesado comentan acerca de posibles anomalías, violaciones a los estándares de desarrollo y otros problemas.

2.1.7 Fiabilidad

Según la ISO/IEC 25010 la fiabilidad se define como la capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados [4]. Esta característica se divide a su vez en las siguientes características:

- Madurez. Capacidad del sistema o producto que satisface las necesidades de fiabilidad en condiciones normales de operación.
- Disponibilidad. Capacidad del sistema o producto operativo y accesible para su uso cuando se requiere.
- Tolerancia a fallos. Capacidad del sistema o producto para operar según lo previsto en presencia de fallos hardware o software.
- Capacidad de recuperación. Capacidad del sistema o producto software para recuperar los datos afectados y restablecer el estado deseado del sistema en caso de interrupción o fallo.

2.2 Estado del arte

2.2.1 Búsqueda de información

En esta sección se explica cómo se lleva a cabo el proceso de la definición del estado del arte mediante la ejecución de una revisión de la literatura, siguiendo algunos lineamientos presentados en [31], para el aseguramiento de la característica de fiabilidad durante el ciclo de desarrollo de software. La secuencia que se aplicó fue la siguiente:

- i. Definición de las cadenas de búsqueda.
- ii. Ejecución de cadenas de búsqueda.
- iii. Filtrado de resultados por título, palabras clave y resumen.
- iv. Filtrado de resultados por contenido.

Para la definición de la cadena de búsqueda se tuvo en cuenta palabras clave como el proceso, la técnica y la característica. A continuación, se presenta la cadena de búsqueda construida:

("quality assurance" AND "techniques" AND "reliability") AND (LIMIT-TO (SUBJAREA, "COMP"))

Adicionalmente a la cadena de búsqueda resultante se le realizó un último filtro que consistió en incluir solo los artículos publicados desde el año 2014 hasta 22 de Diciembre de 2022 para contar con las publicaciones más recientes, obteniendo un volumen de 79 artículos.

La revisión de la literatura fue apoyada mediante el buscador académico Scopus, el cual recopila los artículos más importantes de diferentes editoriales como ScienceDirect, IEEE, ACM, y Springer, adicionalmente se aprovechó el motor de búsqueda Google Scholar para complementar la recopilación de información. A continuación, se presenta una tabla con la distribución de los artículos encontrados por cada editorial (ver *Tabla 1*):

Tabla 1. Artículos por editorial

Cantidad de artículos	Editorial
3	ACM
6	IEEE

3	Science Direct
4	Springer
4	Google Scholar

Para la primera fase de filtrado de los artículos obtenidos mediante la cadena búsqueda se tuvo en cuenta los siguientes criterios de inclusión:

- El título y las palabras claves, estos deberían contener la característica, el proceso o alguna de las técnicas del proceso.
- El resumen del artículo debería incluir información relevante relacionada con el proceso de aseguramiento de calidad con respecto la característica de fiabilidad.

Después de aplicar los criterios de inclusión se obtuvieron 40 artículos relevantes.

En la segunda fase de filtrado, se tuvo en cuenta el siguiente criterio de exclusión:

- En los ítems de los artículos como la introducción, contexto, propuesta, análisis de resultados o conclusiones que no contenga definiciones, detalles o relaciones de valor sobre el aseguramiento de calidad con respecto a la característica de fiabilidad o alguna de sus características.

Al aplicar el criterio anteriormente descrito se obtuvieron 20 artículos, que formarán parte de los artículos iniciales que se tendrán en cuenta en el estudio. Como resultado del mapeo sistemático en la *Tabla 2*, se presentan las propuestas más relevantes con respecto a la temática abordada en el trabajo.

Es importante explicar que para este protocolo de búsqueda se debe aclarar los términos de mecanismo y técnica. Según la RAE un mecanismo son medios prácticos que se emplean para llegar a un objetivo [32] y una técnica es un conjunto de procedimientos y recursos de que se sirve una ciencia [33]; debido a que en el estado del arte las propuestas no se presentan como mecanismos, se asumirá el concepto de técnica, en el sentido de procedimientos prácticos para lograr un objetivo.

Tabla 2. Relación de los artículos relevantes para la investigación.

Artículo	Etapas de desarrollo	Técnica	Subcaracterística de fiabilidad	Alcance o propósito
[1]	Pruebas automatizadas	Aprendizaje automático (MLS)	Madurez	Prevención de fallas

[5]	Análisis de requisitos Pruebas Despliegue	No especifica	Madurez Disponibilidad	Validación y verificación: Prevención de fallas Detección de fallas Tolerancia de fallas Eliminación de fallas
[6]	Pruebas automatizadas	Máquina de vectores Aprendizaje semisupervisado	Madurez Disponibilidad	Automatizar la clasificación de defectos de software
[8]	Pruebas	Caja Gris Caja negra Caja blanca casos de prueba Base de datos híbrida (técnicas tradicionales y modernas)	Madurez y disponibilidad	Prevención de fallas
[9]	Pruebas	Algoritmo genético basados en filtros de Kalman ² Algoritmo genético estándar (SGA) Algoritmo genético con operadores: mutación, cruzamiento, selección y reemplazo Técnica de emparejamiento de probabilidad (PM)	Madurez y disponibilidad	Optimización de datos de prueba
[10]	Pruebas	Máquina de vectores (RBF, Lineal, Polinomio) Naive Bayes Árbol de Decisiones Red neuronal recurrente (RNN)	Madurez	Prevención de fallas
[11]	Pruebas	Predicción basada en árboles de decisión (Random Forest y J48) Naïve Bayes Bayesian Network Clasificador basado en reglas (PART) Vecinos más cercanos (KNN) máquinas de vectores de soporte (SVM) Redes neuronales artificiales (ANN)	Madurez	Predicción de defectos
[14]	Pruebas	Regresión logística Naïve Bayes J48	Madurez	Predicción de defectos
[15]	Pruebas	Algoritmos codiciosos (GA) 2 óptimos Algoritmos genéticos	Madurez	Medir el rendimiento y satisfacción respecto al producto
[16]	Análisis de requisitos Pruebas	Casos de uso Escenarios de prueba	Madurez	Verificación y validación
[17]	Pruebas	Caja Blanca Caja negra	Madurez	Prevención de fallas

² Algoritmo genético basados en filtros de Kalman: Algoritmo utilizados para la generación de casos de pruebas.

		Pruebas basadas en máquinas de estado finito Pruebas de conformidad Pruebas de mutación		
[18]	Pruebas	Pruebas caja Blanca Pruebas caja negra Pruebas fuzz Revisión de código Auditoría de código Análisis estático Inspección Ejecución simbólica Auditoría Inspección	Tolerancia a Fallos	Seguridad en datos
[19]	Pruebas	Algoritmos de clasificación y regresión Árboles de decisión k-vecinos más cercanos máquinas de vectores de soportes Regresión lineal Perceptrón multicapa.	Disponibilidad	Prevención de fallas
[20]	Pruebas	Aprendizaje Supervisado (SSL) Extended Random Forest (extRF)	Madurez	Evaluación de técnicas para la predicción de defectos
[21]	Pruebas	Redes degenerativas adversarias (GAN) Sobre muestreo SMOTE, ROSS, ADASYN, Borderline-SMOTE (Borderline-SMOTE)	Madurez	Estimar el número de defectos clasificar los defectos
[23]	Producción (despliegue)	Estrategia de recuperación basada algoritmos	Tolerancia a fallos Capacidad de recuperación	Estrategia de recuperación ante un fallo
[34]	Pruebas	Caja blanca Caja negra Caja gris Refactorización de código Análisis de código Técnica Ricca y Tonella ³	Tolerancia a fallos Madurez	Adaptación del proceso de pruebas
[35]	Pruebas	Técnicas SAST ⁴	Madurez Capacidad de recuperación	Hacer pruebas enfocadas en el aspecto de seguridad
[36]	Pruebas	Aprendizaje automático Predicción híbrida	Tolerancia a fallos Madurez	Predicción de fallos y calidad de los equipos
[37]	Pruebas	Redes neuronales Aprendizaje automático Arboles de decisión Máquinas de vectores	Tolerancia a fallos Madurez	Predicción de fallos

³ Técnica Ricca y Tonella: Técnica de testing funcional que soporta testing basado en requerimientos.

⁴ Técnicas SAST: Pruebas de seguridad de aplicaciones estáticas.

2.2.2 Discusión

A partir del análisis de la literatura, se puede evidenciar que; de las subcaracterísticas de fiabilidad, la que ha sido trabajada en un mayor porcentaje es la madurez del software (85%), seguida de las subcaracterísticas disponibilidad, tolerancia a fallos y de capacidad de recuperación (ver *Tabla 3*). Cabe aclarar que un artículo puede soportar a varias subcaracterísticas de la fiabilidad.

Tabla 3. Cubrimiento de las subcaracterística de la fiabilidad en la literatura.

Subcaracterística	Madurez	Disponibilidad	Tolerancia a fallos	Capacidad de recuperación
Porcentaje de artículos	85%	25%	25%	10%

Así mismo, se observa una gran diferencia en cuanto a la consideración de la característica de fiabilidad en las etapas del ciclo de desarrollo de software, como se indica en la *Tabla 4*. En la etapa de pruebas se encuentra la mayor parte de esfuerzos de investigación (95% de las propuestas), en comparación con las etapas de análisis y de despliegue, en las cuales se evidencia un bajo porcentaje de publicaciones, correspondiente al 10% de las publicaciones en cada una de ellas. Las etapas de diseño, codificación y mantenimiento no fueron encontradas en el proceso de revisión de la literatura. En este análisis cabe resaltar que un artículo puede soportar a varias etapas del ciclo de desarrollo de software.

Tabla 4. Cubrimiento de las etapas del ciclo de desarrollo de software en la literatura.

Etapas	Análisis	Diseño	Codificación	Pruebas	Despliegue	Mantenimiento
Porcentaje de artículos	10%	0%	0%	95%	10%	0%

De otro lado, según las publicaciones consultadas, la técnica de aseguramiento en etapas de pruebas más relevante para garantizar la fiabilidad es el uso de algoritmos con un 50%, usados para la predicción y clasificación de defectos, continuando con las técnicas de caja blanca y caja negra con un 20%, entre otras (ver *Tabla 2*). Del mismo modo, se identifica que hay una variedad de 43 técnicas de aseguramiento que han trabajado los autores.

De las 43 técnicas descritas en el estado del arte, 25 técnicas que utilizan el machine learning, el uso de esta disciplina conlleva a las organizaciones a contar, para su

implementación, con grandes volúmenes de datos con el fin de viabilizar el entrenamiento requerido. De las 18 técnicas restantes: (i) 4 de ellas se enfocan en analizar el código fuente en su escritura y forma (refactorización, análisis estático, ejecución simbólica), las cuales consideran aspectos funcionales como los caminos lógicos del código, aspectos no funcionales como seguridad y rendimiento y solo una de ellas permitiría el aseguramiento de la fiabilidad en etapa de desarrollo (análisis estático), (ii) 14 técnicas en etapa de pruebas, tanto automatizadas (fuzz, SAST) como manuales (caja gris, caja negra, caja blanca, escenarios de pruebas, revisiones, inspecciones y auditorías). Finalmente, se puede concluir que, pese a la variedad de técnicas encontradas en la literatura, estas técnicas están siendo abordadas solamente dentro de las etapas de desarrollo y pruebas de software, en otras palabras, no se están utilizando técnicas de aseguramiento de calidad en las demás etapas del ciclo de desarrollo de software.

También es importante indicar que se evidencia que los autores pese a que buscan abordar una sub característica específica, según la aplicación de la cadena de búsqueda, el propósito planteado en estos artículos no contrasta con dicha sub característica, por lo tanto, se evidenció una tendencia a la ambigüedad al momento de la lectura y análisis de la literatura. Estos hallazgos, refleja la posibilidad de trabajar en la definición de un método (mecanismo) que permita el aseguramiento de calidad de la característica de fiabilidad de manera concreta, mejorando el entendimiento y comprensión de esta característica y sus subcaracterísticas, para que puedan ser integradas con las prácticas del ciclo de desarrollo de software.

2.2.3 Aportes

De la información revisada se concluye que puede ser importante ampliar el cubrimiento de la característica de fiabilidad durante las etapas de análisis, pruebas y despliegue. Del mismo modo se debería definir una forma de incluir el aseguramiento de esta característica en las etapas de diseño, codificación y mantenimiento del producto software. Para estas etapas del ciclo de desarrollo de software, puede ser valioso fortalecer el uso de otras técnicas de aseguramiento para la característica de fiabilidad.

Debido a lo mencionado anteriormente, el aporte de la tesis, desde una perspectiva de investigación, es proponer un método para el aseguramiento de calidad de la característica de fiabilidad durante las diferentes etapas del desarrollo del software, contribuyendo de esta

manera a mejorar la calidad en los productos de software. Desde la perspectiva del conocimiento, la contribución es la generación de nuevos mecanismos que permitan integrar los conceptos de fiabilidad y las técnicas de aseguramiento de la calidad durante las etapas del ciclo de desarrollo de software.

Capítulo 3 - Método de aseguramiento de calidad de la característica de fiabilidad

Dado que existen diversas falencias para la definición del aseguramiento de calidad de la característica de fiabilidad, en este capítulo se plantean abordar dos aspectos importantes que se evidencian para esta característica. El primero es el conceptual, en donde se dará mayor claridad a la definición de las subcaracterísticas mediante elementos semánticos de la fiabilidad y un banco de preguntas. El segundo aspecto, es el aseguramiento de calidad de esta característica durante las etapas del ciclo de desarrollo de software a través de una extensión del uso de técnicas existentes. Esto con el propósito de construir un método adaptable para las empresas software y que permita ser adoptado como un proceso organizacional que garantice el aseguramiento de la característica de fiabilidad. Entendiendo como método el modo ordenado y sistemático de proceder para llegar a un resultado o fin determinado (Real Academia Española RAE) [38].

A partir de la información extraída del estado del arte se identifican técnicas propuestas por los autores, considerándolas como mecanismos para el aseguramiento de la calidad de la fiabilidad, lo cual permite cuestionar sobre el qué se debería asegurar de esta característica de las diferentes etapas del ciclo de desarrollo del software. En la *Ilustración 1* se muestra el modelo conceptual que podría resolver este cuestionamiento y que además soporta el método de aseguramiento propuesto. De acuerdo a esto, en éste modelo conceptual se evidencia cómo el método es alimentado en primera instancia por elementos semánticos tomados de [3], los cuales permiten acotar el alcance del concepto de la fiabilidad a trabajar a través del método de aseguramiento; éstos elementos se toman como base para construir un banco de preguntas de acuerdo a las diferentes etapas del ciclo de desarrollo software, así mismo, se apoya de las técnicas encontradas en la literatura y de un artefacto funcional propuesto para lograr aplicar el aseguramiento de calidad de la característica de fiabilidad en escenarios productivos.

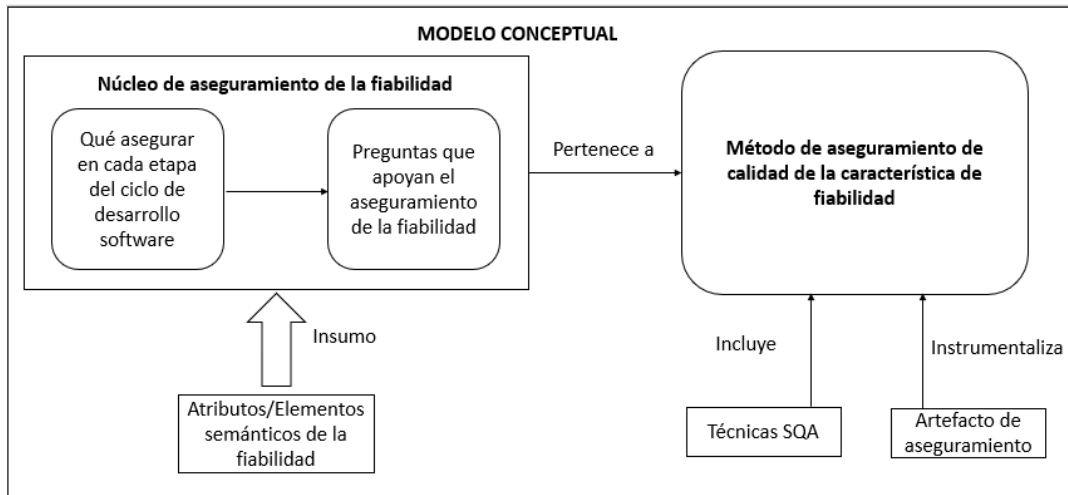


Ilustración 1. Modelo conceptual del método para el aseguramiento de la fiabilidad.

A continuación, se describen las actividades llevadas a cabo para la generación de los componentes del modelo conceptual:

3.1 Análisis conceptual de la fiabilidad

Inicialmente se realizan dos tareas antes de definir el método de aseguramiento de calidad para la característica de fiabilidad durante el ciclo de vida de desarrollo de software, en la primera actividad se define qué aspectos se deben tener en cuenta para asegurar la calidad de la característica de fiabilidad en las etapas de desarrollo software, y en la segunda actividad se plantea una serie de preguntas que tienen la finalidad de guiar al proyecto de desarrollo en el cumplimiento o no cumplimiento de los aspectos que pueden asegurar esta característica.

3.2 Aspectos para asegurar la calidad de la fiabilidad

Para la definición de los aspectos implicados en el aseguramiento de calidad de la característica de fiabilidad en cada etapa del ciclo de desarrollo de software se tomó como punto de partida los elementos semánticos mencionados en el modelo conceptual de la propuesta[3], la cual buscaba concretar aspectos (conceptos) relacionados con la fiabilidad y otras características de calidad para posteriormente proponer una representación (iconográfica) de dichos aspectos. El conjunto de estos aspectos *Tabla 5* permite determinar el alcance del aseguramiento de calidad de esta característica para este trabajo

de investigación y que será considerado durante cada etapa del ciclo de desarrollo de software.

Tabla 5. Elementos semánticos de Fiabilidad.

Característica de calidad	Subcaracterística de calidad	Elemento semántico
Fiabilidad	Madurez	<p>Sobre los datos: (Exactitud = Veracidad + Precisión) -> https://midebien.com/cual-es-ladiferencia-entre-exactitud-y-precision/ La exactitud indica qué tan cerca se encuentra del resultado correcto. Mientras que, la precisión es que tan consistentemente se obtiene el resultado con el mismo método (menos variabilidad). Exactitud semántica y sintáctica Completitud (funcional) Consistencia Credibilidad - veracidad Fidelidad Actualidad Accesibilidad Conformidad Confidencialidad Eficiencia Precisión Trazabilidad Comprensibilidad Disponibilidad Portabilidad Recuperabilidad</p> <p>Sobre la función: Fidelidad lógica Veracidad lógica Manejo de errores de entrada y salida</p>
	Disponibilidad	Horarios de disponibilidad Acuerdos de nivel de servicio
	Tolerancia a fallos	Frecuencia de fallos HW/SW Severidad de fallos HW/SW Integridad/robustez Tiempo entre fallos Viabilidad lógica del fallo (probabilidad de que exista de que ocurra una falla)
	Capacidad de recuperación	Recuperación a fallos Viabilidad de datos Recuperación de datos afectados Reestablecer el sistema

3.3 Preguntas que permiten asegurar la fiabilidad en el proceso de desarrollo de software.

Implementar una estrategia para el aseguramiento de la calidad de la fiabilidad de un producto software, podría incluir el uso de elementos que resultan de gran utilidad para el proceso de desarrollo software como lo es un banco de preguntas [3], por tanto, para el aseguramiento de la calidad de la fiabilidad se establece un banco de preguntas asociadas a cada una de las sub características de fiabilidad al cual se puede acudir para garantizar que se asegure la fiabilidad dentro de sus etapas del ciclo de desarrollo. Este banco propone categorías y preguntas. Cada etapa del ciclo de desarrollo cuenta con un subconjunto de preguntas de aseguramiento y que, a la vez, pertenecen a una sub característica específica. La *Tabla 6, Tabla 8, Tabla 9, Tabla 10, Tabla 11, Tabla 12*, contienen un banco de preguntas, cada una de las tablas hace referencia a una etapa del ciclo de desarrollo. La *Tabla 6, Tabla 8, Tabla 9 y Tabla 10*, se definen preguntas para cada una de las cuatro subcaracterísticas de fiabilidad. En cambio, en la *Tabla 11 y Tabla 12*, se establecen preguntas integradas sin hacer clasificación por cada subcaracterística. Las preguntas fueron elaboradas a partir de los elementos semánticos de la característica de fiabilidad *Tabla 5*, tomados de [3]. Cada elemento semántico fue considerado para elaborar la pregunta que aportaría en el aseguramiento de la característica de fiabilidad en las diferentes etapas del ciclo de vida de desarrollo de software.

Tabla 6. Preguntas propuestas para el aseguramiento de la fiabilidad en etapas de análisis.

Etapa del ciclo de vida de Software: Análisis	
Precondición	Se asume que en la etapa se cuenta con el artefacto de especificación
Subcaracterística de fiabilidad	Preguntas planteadas
Madurez	<ol style="list-style-type: none"> 1. ¿Los mensajes especificados son claros para realizar los diferentes caminos definidos? 2. ¿Los datos han sido especificados conforme a lo que se espera que sea el resultado correcto y con una precisión definida (exactitud)? 3. ¿Los datos que debe manejar el sistema han sido especificados de forma completa funcionalmente (frente al negocio) (Compleitud funcional)? 4. ¿Los datos del sistema han sido especificados con coherencia (estables) y las salidas que se esperan sean las correctas (consistencia)? 5. ¿Los datos del sistema han sido especificados con claridad frente a su verdadero valor (credibilidad-veracidad)? 6. ¿El sistema cuenta con algún mecanismo que protege los datos de individuos o softwares no autorizados a manipularlos (confidencialidad de los datos)? 7. ¿Se cuenta con formas explícitas para observar la evolución de los datos frente a los cambios que puedan tener (Trazabilidad)? 8. ¿Las funciones tienen una abstracción adecuada de la realidad en el sistema (fidelidad-veracidad lógica)?
Disponibilidad	<ol style="list-style-type: none"> 1. ¿Se ha especificado el horario en que debe estar disponible el sistema para los usuarios? 2. ¿Se han especificados los acuerdos de nivel de servicio del sistema?
Tolerancia a fallos	<ol style="list-style-type: none"> 1. ¿Se definen formas de medir la frecuencia de fallos hardware/software? 2. ¿Se cuenta con una forma específica para medir el impacto funcional y sobre los datos de los fallos hardware/software (severidad hardware/software)? 3. ¿El sistema cuenta con mecanismos para auto-gestionar el tiempo entre fallos? 4. ¿El sistema podría determinar la probabilidad de que ocurra un fallo (viabilidad lógica del fallo)?
Capacidad recuperación	<ol style="list-style-type: none"> 1. ¿Algún mecanismo de recuperación a fallos ha sido contemplado en la especificación (incluye elementos como la definición de logs de errores y logs de datos procesado con éxito)? 2. ¿Se establecieron los horarios a realizar copias de seguridad para los datos afectados? 3. ¿Se estableció los tiempos de vigencia de la copia de seguridad para los datos afectados? 4. ¿Se establecieron los tipos de copias de seguridad para los datos afectados? 5. ¿Se definen formas para corroborar el adecuado restablecimiento del sistema?

En la Tabla 7. Diagramas base para etapa de diseño basados en el diagrama C4 se resume los diagramas del modelo C4 [39], en el cual se indica el propósito del diagrama y la naturaleza de los elementos que son abordados.

Tabla 7. Diagramas base para etapa de diseño basados en el diagrama C4.

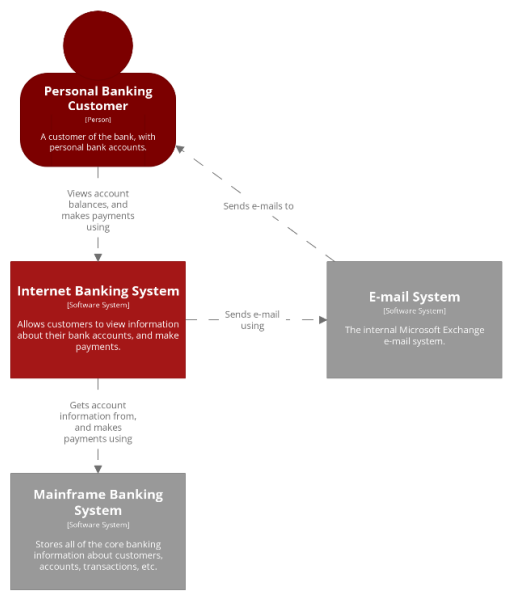
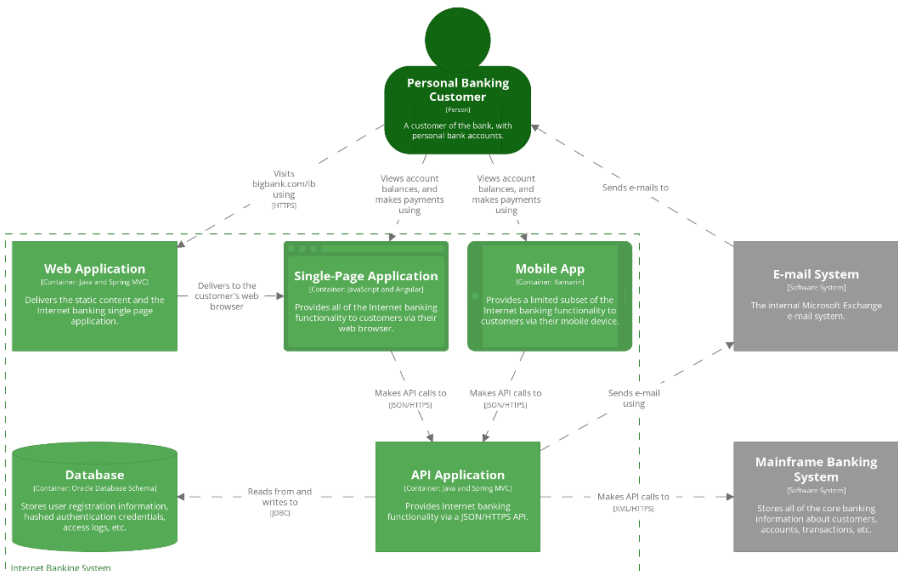
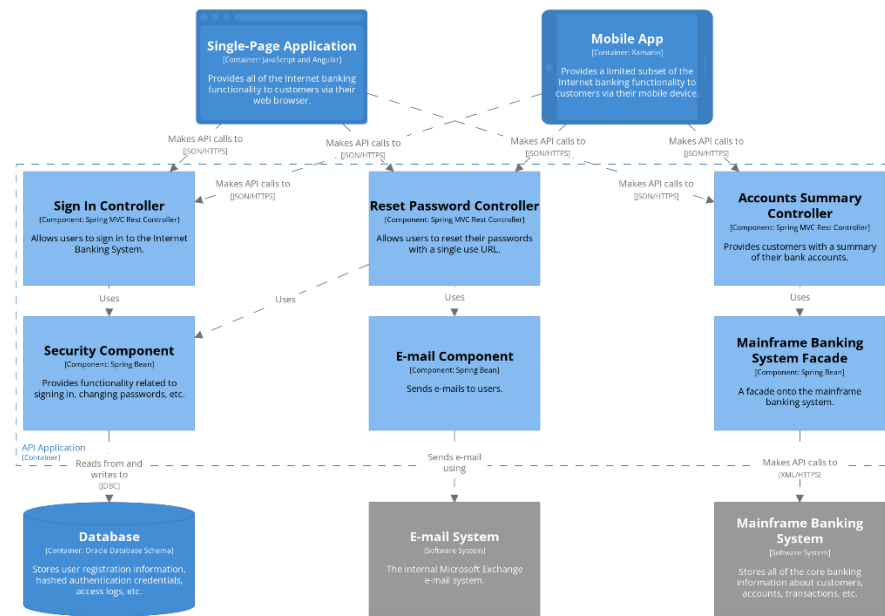
Nombre del diagrama	Propósito
<p>Diagrama de contexto del sistema</p>	<p>Presentar el panorama general del sistema, enfocándose en las personas / actores y los sistemas de software en lugar de las tecnologías, los protocolos y otros detalles de bajo nivel. Se adjunta una imagen como ejemplo de este diagrama tomada de [39].</p>  <pre> graph TD Customer[Personal Banking Customer [Person] A customer of the bank, with personal bank accounts.] IBSystem[Internet Banking System [Software System] Allows customers to view information about their bank accounts, and make payments.] Mainframe[Mainframe Banking System [Software System] Stores all of the core banking information about customers, accounts, transactions, etc.] Email[E-mail System [Software System] The internal Microsoft Exchange e-mail system.] Customer -- "Views account balances, and makes payments using" --> IBSystem IBSystem -- "Sends e-mails to" --> Customer IBSystem -- "Sends e-mail using" --> Email IBSystem -- "Gets account information from, and makes payments using" --> Mainframe </pre> <p>[System Context] Internet Banking System Tuesday, September 27, 2022, 7:30 PM Coordinated Universal Time</p>
<p>Diagrama de contenedor</p>	<p>Presentar la forma de alto nivel de la arquitectura del software y cómo se distribuyen las responsabilidades a través de ella. Se adjunta una imagen como ejemplo de este diagrama tomada de [39].</p>  <pre> graph TD Customer[Personal Banking Customer [Person] A customer of the bank, with personal bank accounts.] subgraph IBSystem [Internet Banking System [Software System]] WA[Web Application [Container: Java and Spring, WC] Delivers the static content and the internet banking single-page application.] SPA[Single-Page Application [Container: JavaScript and Angular] Provides all of the Internet banking functionality to customers via their web browser.] MA[Mobile App [Container: Xamarin] Provides a limited subset of the Internet banking functionality to customers via their mobile device.] AA[API Application [Container: Java and Spring, MVC] Provides Internet banking functionality via a JSON/HTTPS API.] DB[(Database [Container: Oracle database, Schema] Stores user registration information, hashed authentication credentials, access logs, etc.)] WA -- "Delivers to the customer's web browser" --> SPA SPA -- "Makes API calls to [JSON/HTTPS]" --> AA MA -- "Makes API calls to [JSON/HTTPS]" --> AA AA -- "Reads from and writes to [JSON]" --> DB end Mainframe[Mainframe Banking System [Software System] Stores all of the core banking information about customers, accounts, transactions, etc.] Email[E-mail System [Software System] The internal Microsoft Exchange e-mail system.] Customer -- "Visits bigbank.com/ib using [HTTPS]" --> SPA Customer -- "Views account balances, and makes payments using" --> SPA Customer -- "Views account balances, and makes payments using" --> MA Customer -- "Sends e-mails to" --> Email Email -- "Sends e-mail using" --> Mainframe AA -- "Makes API calls to [XML/HTTPS]" --> Mainframe </pre> <p>[Container] Internet Banking System Tuesday, September 27, 2022, 7:30 PM Coordinated Universal Time</p>

Diagrama de componentes

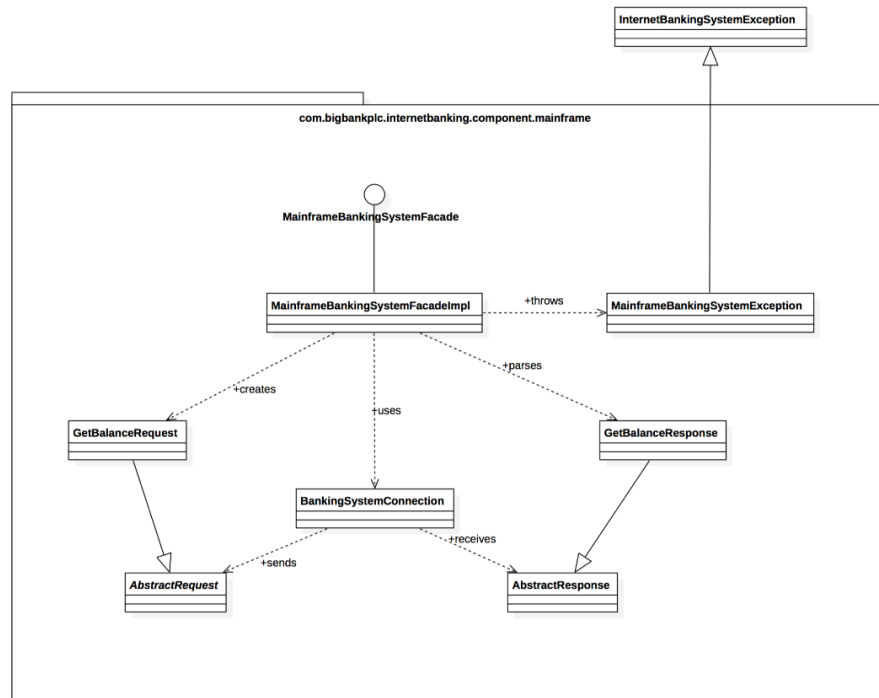
Presentar los principales estructuras e interacciones de componentes que conforman los contenedores. Se adjunta una imagen como ejemplo de este diagrama tomada de [39].



[Component] Internet Banking System - API Application
Tuesday, November 8, 2022, 6:48 PM Coordinated Universal Time

Diagrama de clases

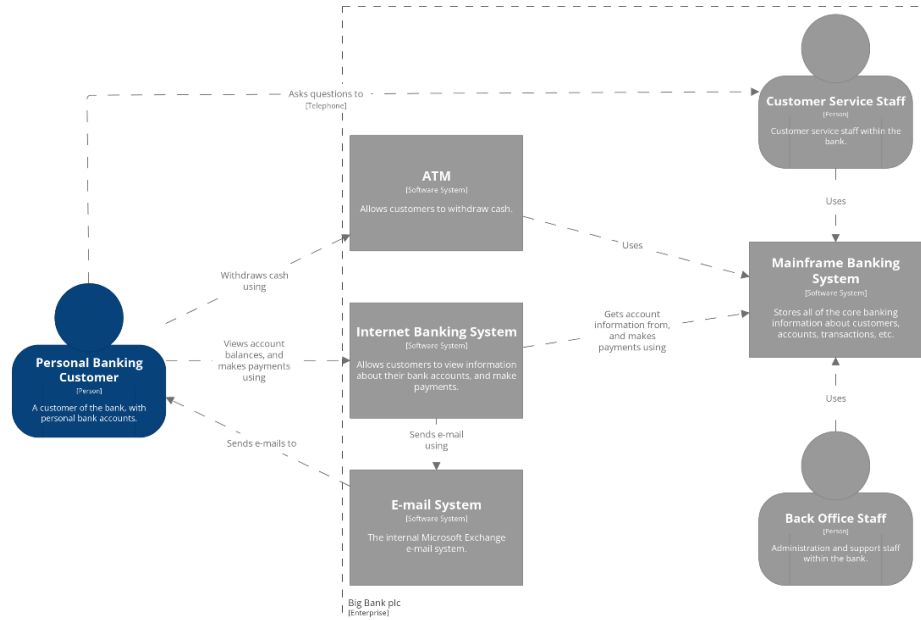
Presentar los atributos y métodos de los componentes más importantes o complejos. Se adjunta una imagen como ejemplo de este diagrama tomada de [39].



Diagramas complementarios

Diagrama de panorama del sistema

Presentar el sistema en alto nivel / nivel empresarial, para mostrar los límites de la organización, los usuarios internos/ externos, y los sistemas internos/externos. Se adjunta una imagen como ejemplo de este diagrama tomada de [39].



[System Landscape] Big Bank plc
Monday, January 31, 2022, 8:56 AM Coordinated Universal Time

Diagrama dinámico

Se basa en un diagrama de comunicación UML (anteriormente conocido como "UML diagrama de colaboración"). Es similar a un diagrama de secuencia UML, aunque permite una disposición de forma libre de los elementos del diagrama con interacciones numeradas para indicar el orden. Se adjunta una imagen como ejemplo de este diagrama tomada de [39].

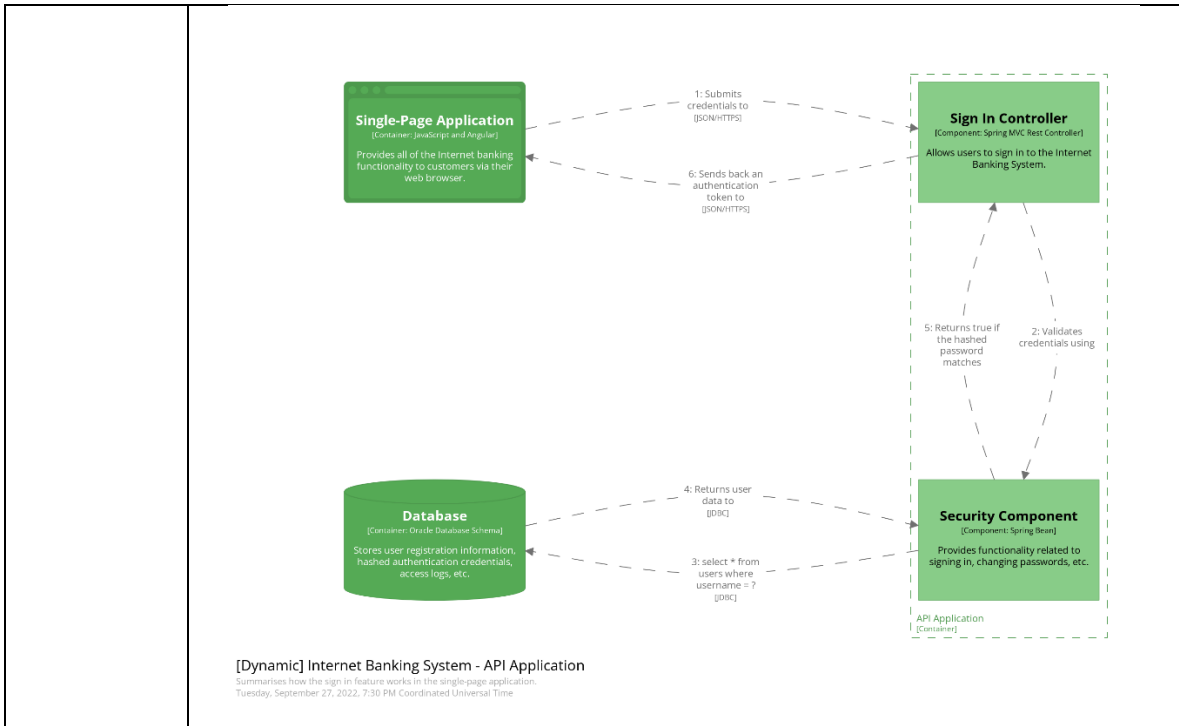


Diagrama de Ilustrar cómo los sistemas de software y/o los contenedores en el modelo estático se asignan a la infraestructura. Se adjunta una imagen como ejemplo de este diagrama tomada de [39].

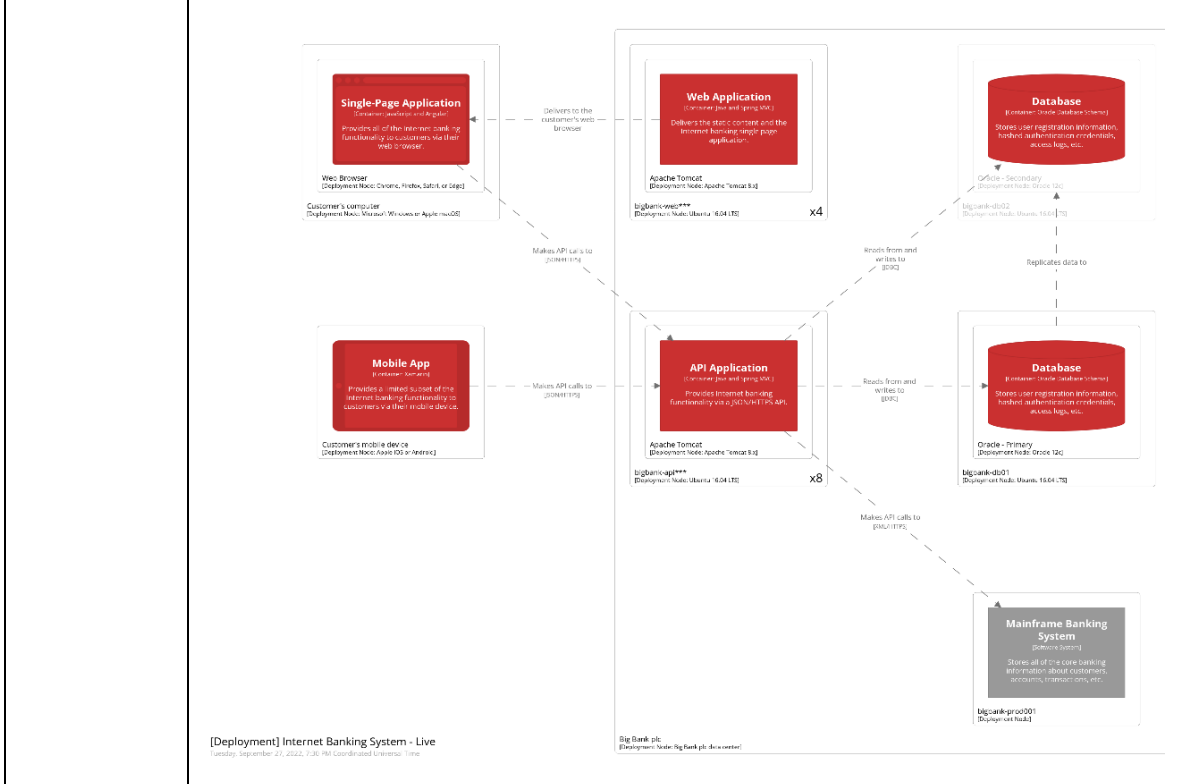


Tabla 8. Preguntas propuestas para el aseguramiento de la fiabilidad en etapas de diseño.

Etapa del ciclo de vida de Software: Diseño	
Subcaracterística de fiabilidad	Preguntas planteadas
Madurez	<ol style="list-style-type: none"> 1. ¿Las secuencias de los diferentes caminos del sistema se ven reflejados correctamente en el diagrama dinámico? 2. ¿En el diagrama dinámico se visualizan los datos conforme a lo que se espera que sea el resultado correcto (exactitud)? 3. ¿En el diagrama dinámico se puede observar la coherencia (estabilidad) de los datos (consistencia)? 4. ¿En el diagrama dinámico se ven reflejados los datos con claridad frente al verdadero valor (credibilidad-veracidad)? 5. ¿En el diagrama dinámico se observa el tratamiento de protección de todos los datos del sistema (confidencialidad)? 6. ¿La evolución frente a los cambios de los datos se ve reflejada en las secuencias del diagrama dinámico? 7. ¿En el diagrama dinámico se puede observar la abstracción adecuada del comportamiento de las funciones (fidelidad-veracidad lógica)?
Disponibilidad	<ol style="list-style-type: none"> 1. ¿En el diagrama de contenedores/componentes se cuenta con algún elemento HW/SW que apoye la disponibilidad del sistema para los usuarios (Acuerdos de Nivel de Servicio)?
Tolerancia a fallos	<ol style="list-style-type: none"> 1. ¿En el diagrama de contenedores/componentes se cuenta con algún elemento HW que apoye la identificación de una falla hardware? 2. ¿En el diagrama de componentes/contenedores se mide de alguna manera la frecuencia de fallos software (severidad)? 3. ¿En el diagrama de componentes/contenedores se cuenta con algún elemento HW/SW que apoye la autogestión del tiempo entre fallos? 4. ¿Para determinar la probabilidad de que ocurra un fallo se encuentra algún elemento HW/SW en el diagrama de componentes/contenedores (viabilidad lógica del fallo)?
Capacidad de recuperación	<ol style="list-style-type: none"> 1. ¿En el diagrama de contenedores/componentes se cuenta con algún elemento HW/SW que permita la recuperación a fallos que se puedan presentar en el sistema? 2. ¿En el diagrama de contexto incluye un documento de apoyo técnico donde se pueda indicar los horarios a restablecer las copias de seguridad de datos afectados? 3. ¿El diagrama de contexto incluye un documento de apoyo que permita definir el tiempo de vigencia de la copia de seguridad de los datos afectados? 4. ¿En el diagrama de contexto se cuenta con un documento de apoyo que detalle la forma en que se debe dar el restablecimiento del sistema en caso de fallo?

Tabla 9. Preguntas propuestas para el aseguramiento de la fiabilidad en etapa de codificación.

Etapa del ciclo de vida de Software: Codificación	
Subcaracterística de fiabilidad	Preguntas planteadas
Madurez	<ol style="list-style-type: none"> 1. ¿En las funciones o métodos implementados en el código fuente se permite la verificación de que los datos están conformes a lo que se espera que sea el resultado correcto y con una precisión definida (exactitud)? 2. ¿En las funciones o métodos codificados se puede corroborar que los datos son coherentes (estables) y sus salidas que se esperan son las correctas (consistencia)? 3. ¿En las funciones o métodos en el código fuente se evidencia que los datos están implementados con claridad frente a su verdadero valor (credibilidad-veracidad)? 4. ¿En las funciones o métodos creados se implementó un mecanismo que protege los datos de individuos o software no autorizados a manipular los datos (confidencialidad)? 5. ¿La evolución de los datos frente a los cambios que puedan tener ha sido implementada en las funciones o métodos del código fuente (trazabilidad)? 6. ¿La abstracción adecuada de la realidad ha sido implementada en las funciones o métodos del código fuente (fidelidad-veracidad)?
Disponibilidad	<ol style="list-style-type: none"> 1. ¿La disponibilidad del sistema para los usuarios ha sido configurada/implementada a través del o los elementos hardware/software definidos en el diseño (Acuerdos de Nivel de Servicio)?
Tolerancia a fallos	<ol style="list-style-type: none"> 1. ¿Existe algún elemento hardware para la identificación de una falla en el sistema? 2. ¿En las funciones o métodos se codificó la medición de la frecuencia de fallos software del sistema? 3. ¿En las funciones o métodos se implementó la medición del impacto funcional y sobre los datos de los fallos hardware/software (severidad)? 4. ¿La autogestión del tiempo entre fallos fue implementada en el código fuente? 5. ¿Fue implementada una función o método en el código fuente que pueda determinar la probabilidad de que ocurra un fallo (viabilidad lógica de los fallos)?
Capacidad de recuperación	<ol style="list-style-type: none"> 1. ¿La recuperación a posibles fallos del sistema ha sido configurada/implementada a través del (los) elementos hardware/software definidos en el diseño (incluye elementos como la creación de logs de errores y logs de datos procesado con éxito)? 2. ¿Los horarios para restablecer las copias de seguridad de datos afectados definidos en el diseño fueron considerados en las configuraciones/implementaciones del sistema? 3. ¿El tiempo de vigencia de la copia de seguridad de los datos afectados definidos en el diseño fueron considerados en las configuraciones/implementaciones del sistema? 4. ¿El restablecimiento del sistema en caso de fallo, definido en el diseño, ha sido configurado/implementado?

Tabla 10. Preguntas propuestas para el aseguramiento de la fiabilidad en etapa de pruebas.

Etapa del ciclo de vida de Software: Pruebas	
Subcaracterística de fiabilidad	Preguntas planteadas
Madurez	<ol style="list-style-type: none"> 1. ¿Se ha diseñado casos de prueba que verifiquen que los datos están conforme a lo que se espera que sea el resultado correcto y con una precisión definida (exactitud)? 2. ¿Existen casos de prueba diseñados que corrobore que los datos arrojados tienen la coherencia (estables) y las salidas que se esperan sean las correctas (consistencia)? 3. ¿Los casos de prueba diseñados verifican que los datos resultantes son claros frente a su verdadero valor (credibilidad-veracidad)? 4. ¿Los casos de prueba creados verifican la protección de los datos antes individuos o softwares no autorizados a manipularlos (confidencialidad de los datos)? 5. ¿Se ha diseñado casos de prueba que verifiquen la evolución de los datos frente a los cambios que puedan tener (trazabilidad)? 6. ¿Se ha diseñado casos de prueba que corrobore la abstracción adecuada de la realidad en las funcionalidades a probar (fidelidad-veracidad lógica)?
Disponibilidad	<ol style="list-style-type: none"> 1. ¿Los casos de prueba diseñados confirman la disponibilidad del sistema para los usuarios (Acuerdos de Nivel de Servicio)?
Tolerancia a fallos	<ol style="list-style-type: none"> 1. ¿Se han diseñado caso de prueba que permitan verificar la identificación del fallo hardware/software? 2. ¿Se han diseñado caso de prueba para verificar la medición de la frecuencia de fallos hardware/software (severidad)? 3. ¿Se diseñaron casos de prueba que verifiquen la autogestión del tiempo entre fallos? 4. ¿Se diseñaron casos de prueba que verifiquen la probabilidad de que ocurra un fallo (viabilidad lógica del fallo)?
Capacidad de recuperación	<ol style="list-style-type: none"> 1. ¿Se diseñaron casos de prueba que permitan corroborar la recuperación de fallos presentados por el sistema (incluye elementos a considerar como logs de errores y logs de datos procesado con éxito)? 2. ¿Los casos de prueba diseñados verificar los horarios de restablecimiento de las copias de seguridad de los datos afectado? 3. ¿El tiempo de vigencia de la copia de seguridad de datos afectados se verifican en los casos de pruebas diseñados? 4. ¿Los casos de prueba diseñados verifican el restablecimiento del sistema en caso de fallos?

Tabla 11. Preguntas propuestas para el aseguramiento de la fiabilidad en etapa de despliegue.

Etapa del ciclo de vida de Software: Despliegue
Preguntas planteadas
¿El sistema que ha sido previamente analizado, diseñado, codificado y probado ha sido llevado al ambiente productivo considerando todos los aspectos de la característica de fiabilidad?

Tabla 12. Preguntas para el aseguramiento de la fiabilidad en etapa de mantenimiento.

Etapa del ciclo de vida de Software: Mantenimiento
Preguntas planteadas
¿Los cambios que han sido identificados, han sido analizados, diseñados, codificados y probados para un nuevo despliegue?

3.4 Técnicas de aseguramiento de calidad encontradas en la literatura consultada

Luego de la construcción de las preguntas, se definió una *Tabla 13* de técnicas de aseguramiento de calidad para la característica de fiabilidad. Esta tabla fue generada dentro del trabajo de investigación a partir de la revisión de literatura realizada, donde se evidencia que las técnicas de aseguramiento se concentran en gran parte en la fase de pruebas y una minoría son técnicas que pueden ser usadas o adaptadas en cualquiera de las fases del ciclo de vida del desarrollo software. A continuación, las técnicas presentes en la literatura:

Tabla 13. Técnica de aseguramiento de calidad tradicionales.

Técnica de aseguramiento	Propósitos	Sugerencia de etapa para aplicar
Revisión Técnica Formal [40]	Evaluar el producto software, identificar discrepancias en las especificaciones y estándares Participa pares técnicos, expertos técnicos en la misma u otras disciplinas Necesita preparación individual antes de la reunión de revisión	Análisis
Auditoría [41]	Auditoría interna o auditoría de primera parte: realizadas por la propia organización Auditoría Externa de segunda parte: realizadas por partes que tienen interés en la organización Auditoría Externa de tercera parte: realizadas por terceros son realizadas por auditorías independientes, organizaciones como los reguladores o los que proporcionan la certificación.	Diseño
Inspección [30]	Detectar e identificar anomalías en el producto de software. Una inspección es una examinación de pares.	Análisis, Diseño Codificación Pruebas Despliegue Mantenimiento
Recorrido [30]	El propósito de un recorrido es educar/dar a conocer a la audiencia un producto de software o un componente específico.	Análisis, Diseño Codificación Prueba, Despliegue Mantenimiento
Caja negra [40]	Encontrar una serie de datos de entrada cuya probabilidad de pertenecer al conjunto de entradas que causan un comportamiento erróneo sea lo más alto posible	Pruebas
Caja Blanca [40]	Comprobar todas las sentencias del programa, y todas las condiciones tanto en sus caminos válidos como no válidos.	Desarrollo
Análisis estático [40]	Respaldar la detección de vulnerabilidades de seguridad, junto con la detección de errores, métricas de calidad y cumplimiento de estándares de codificación.	Análisis, Diseño Codificación, Pruebas Despliegue Mantenimiento
Machine Learning [42]	Reducir la cantidad de tareas que implica las pruebas manuales y evita que los desarrolladores tengan que estar revisando los documentos constantemente para detectar error y fallas, reduciendo el tiempo y el dinero, aumentando la	Pruebas

	cobertura de las pruebas, mejorando su precisión y disminuyendo la cantidad de trabajo del equipo de probadores	
Pruebas de conformidad [43]	En las pruebas de conformidad se verifica que una implementación particular cumple satisfactoriamente con lo estandarizado en un protocolo determinado	Análisis, Diseño Codificación, Pruebas Despliegue Mantenimiento
Pruebas Fuzz [44]	Las pruebas fuzz son semiautomáticas o automáticas donde se proveen datos al azar, inválidos y no esperados en la sección de entrada de datos del software. De esa forma es posible comprobar la seguridad de la entrada en lo que respecta a la validación de datos.	Pruebas
Ejecución simbólica [45]	La técnica de ejecución simbólica simula la ejecución del código transformando variables de tipo en símbolos, permitiendo identificar qué caminos de ejecución se toman de acuerdo con los valores de entrada, pudiendo identificar claramente qué valores ha tomado la variable en cuestión y si lo hace de manera esperada o no	Desarrollo

A continuación, se describe el método de aseguramiento de calidad de la característica de fiabilidad durante el ciclo de desarrollo software.

3.5 Método de aseguramiento de calidad de la característica de fiabilidad

A continuación, se presenta la propuesta del método de calidad de la característica de fiabilidad durante el ciclo de desarrollo de software (ver *Ilustración 2*, *Ilustración 3*), la cual incluye la definición de un rol de aseguramiento por etapa del proyecto de desarrollo quien será el responsable de este aseguramiento, como se muestra en la *Tabla 14*. De acuerdo con esta propuesta, el aseguramiento de calidad de la característica de fiabilidad es realizado por el integrante con mayor dominio de la actividad (etapa del ciclo de vida del desarrollo de software), donde con el apoyo del artefacto de aseguramiento propuesto evidenciará que los elementos semánticos determinados para la fiabilidad (ver parte 3.2) se encuentren presentes en cada artefacto técnico que corresponda a la etapa. Los distintos roles de aseguramiento (analista QA, arquitecto QA, Programador QA, QA e Implementador QA) requerirían contar con al menos 5 años continuos de experiencia en temas de calidad en desarrollo de software, incluyendo conocimientos de estándares y artefactos técnicos de cada etapa del ciclo. La responsabilidad del buen aseguramiento dependerá del grado de experticia de estos roles de acuerdo con los procesos de reclutamiento de las organizaciones.

Tabla 14. Roles del método.

Roles que intervienen en el método	Responsabilidad
Analista QA	El analista QA es el encargado de realizar la actividad de aseguramiento de calidad de la característica de fiabilidad con referencia a los requisitos elicitados (capturados) y descritos en el documento de especificación. Esta tarea es realizada en colaboración con los interesados (Stakeholders del producto), como, por ejemplo: jefes de área, analistas de negocio y entre otros.
Arquitecto QA	El arquitecto QA es un miembro del proyecto o un experto externo que realiza las tareas de la actividad de aseguramiento de calidad de la característica de la fiabilidad con respecto a los diseños del producto software.
Programador QA	El programador QA es el encargado de ejecutar la actividad de aseguramiento de calidad de la característica de fiabilidad sobre los artefactos de salida de la actividad de implementar los requisitos (código fuente).
QA (Quality Assurance)	El QA es el encargado de verificar y validar la característica de fiabilidad en los artefactos de salida del diseño y la ejecución de los casos de prueba planteados en el proyecto.
Implementador QA	El implementador QA es el rol encargado de realizar la actividad de aseguramiento de calidad de la característica de fiabilidad al momento del despliegue del producto software para corroborar que la fiabilidad fue considerada en el análisis, diseño, codificación y pruebas de manera suficiente.

Propósito del método: El método de aseguramiento de calidad propuesto busca que las empresas desarrolladoras de software lo adopten como una estrategia, construida a través de la investigación, que facilite llevar a cabo los procesos organizacionales de aseguramiento de la característica de fiabilidad para el desarrollo de productos software.

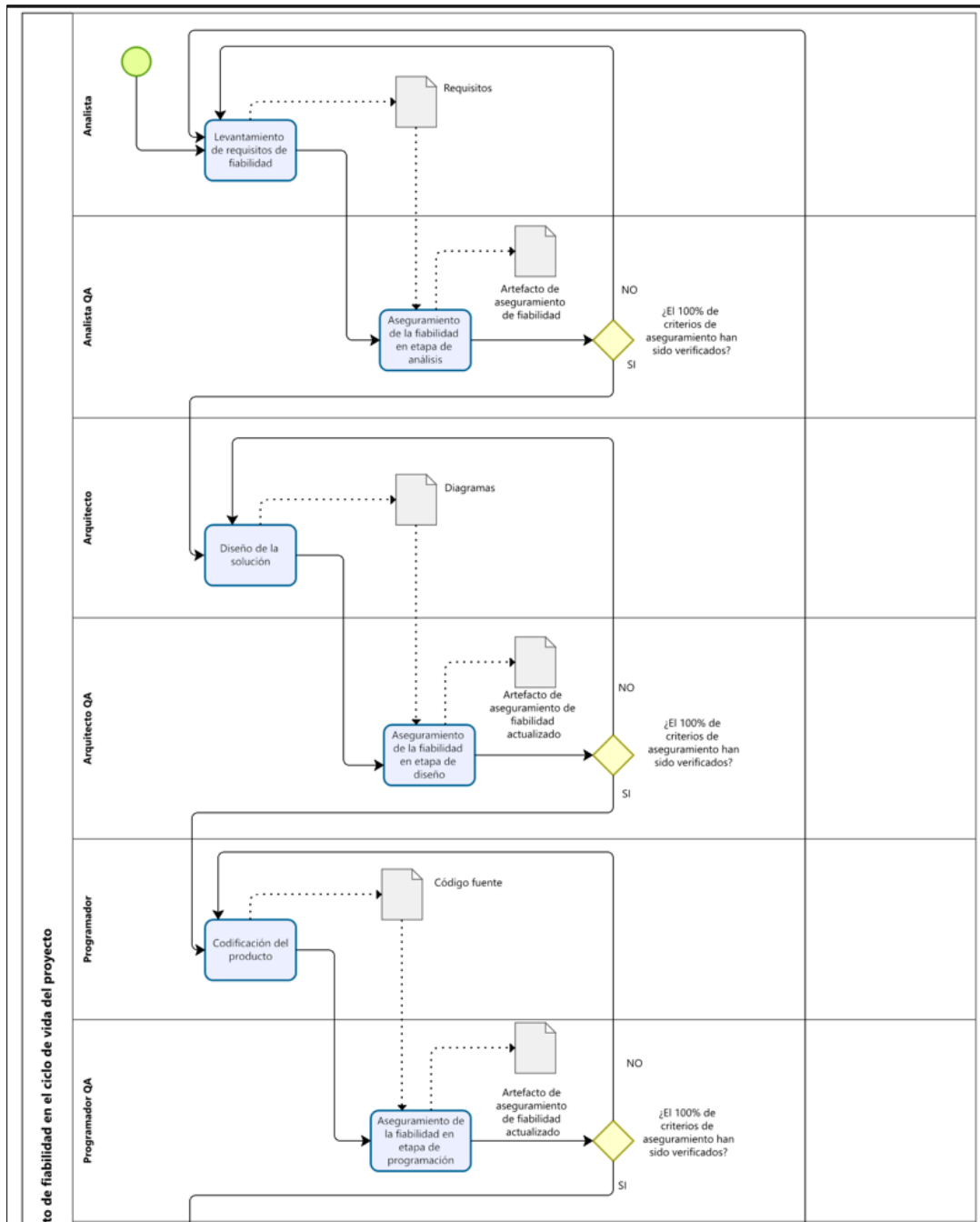


Ilustración 2. Modelo del método propuesto, parte A.

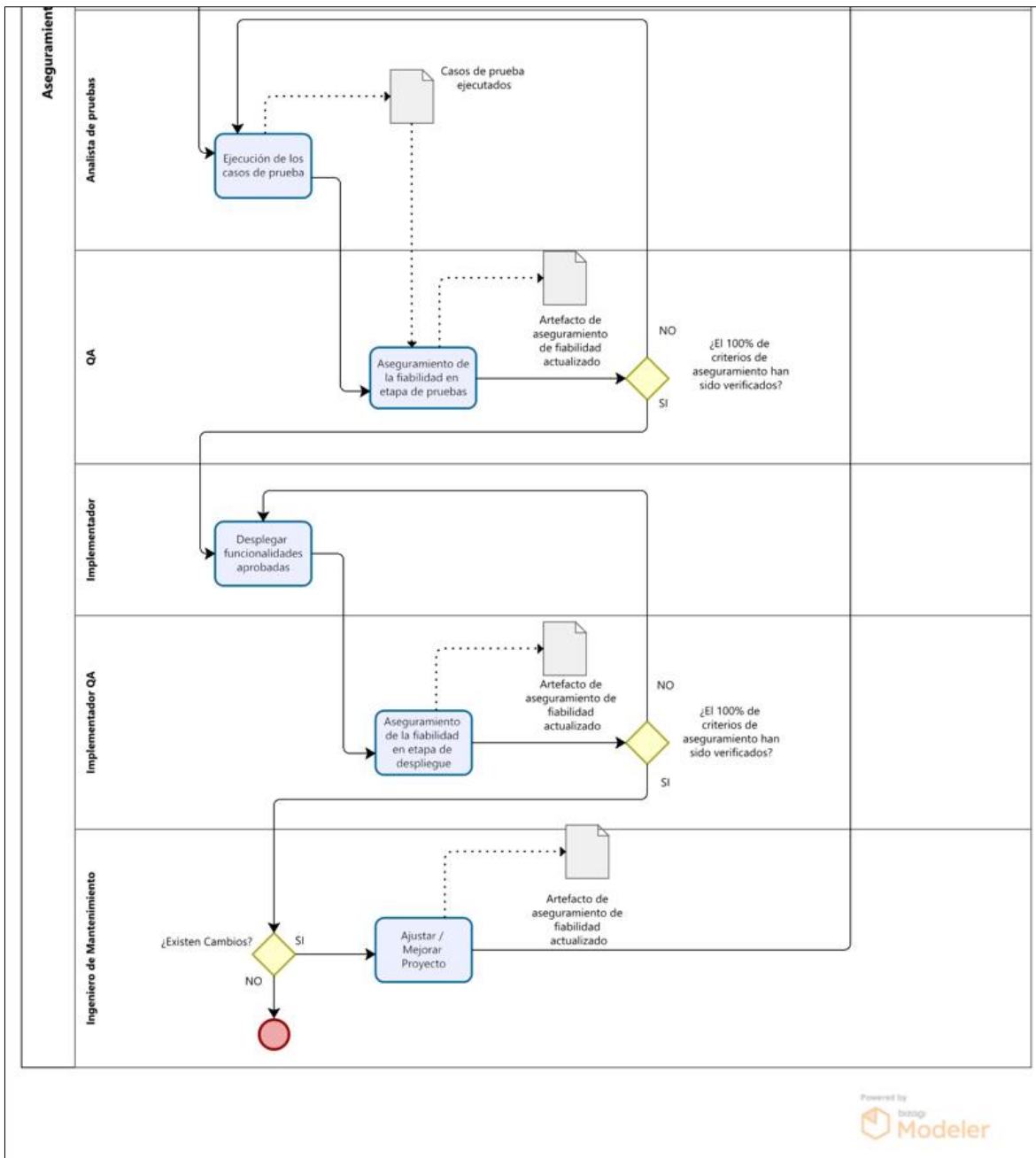


Ilustración 3. Modelo del método propuesto, parte B.

Generalidades del método:

Este método se podría implementar en diversos proyectos, puesto que no está limitado a ninguna metodología específica de desarrollo de software, esto es, el método puede ser utilizado para proyectos bajo metodologías tradicionales o metodologías ágiles de desarrollo.

El método incluye el apoyo de un artefacto de aseguramiento de calidad de la característica de fiabilidad construido en Excel, el cual va siendo actualizado durante el ciclo de desarrollo de software, y que consiste en responder una serie de preguntas en cada una de las etapas del ciclo de desarrollo de software. Vale la pena aclarar que las preguntas sugeridas son diferentes por cada etapa, ya que son formuladas cubriendo los elementos semánticos pero aterrizadas a las actividades propias de cada etapa. A partir de esta información, el artefacto permite:

Visualizar el promedio del avance y el nivel de aseguramiento de calidad de la fiabilidad con respecto a sus sub características (según el modelo de calidad de un producto software establecido en la ISO 25010 [4]) en cada una de las etapas del ciclo de desarrollo de software, los cuales se actualizan a medida que se van respondiendo las preguntas propuestas dentro de cada una de ellas (ver *Ilustración 4*).

Etapas del ciclo de vida de Software: Análisis			
Precondición: Se asume que en la etapa se cuenta con el artefacto de especificación			
Subcaracterística	Preguntas planteadas	Nivel de aseguramiento	
Madurez	¿Los mensajes especificados son claros para realizar los diferentes caminos definidos?	-	
	¿Los datos han sido especificados conforme a lo que se espera que sea el resultado correcto y con una precisión adecuada?	-	
	¿Los datos que debe manejar el sistema han sido especificados de forma completa funcionalmente (frente al usuario)?	-	
	¿Los datos del sistema han sido especificados con coherencia (estables) y las salidas que se esperan sean las correctas?	-	
	¿Los datos del sistema han sido especificados con claridad frente a su verdadero valor (credibilidad-veracidad)?	-	
	¿El sistema cuenta con algún mecanismo que protege los datos de individuos o softwares no autorizados a manipularlos (confidencialidad de los datos)?	-	
	¿Se cuenta con formas explícitas para observar la evolución de los datos frente a los cambios que puedan tener?	-	
	¿Las funciones tienen una abstracción adecuada de la realidad en el sistema (fidelidad-veracidad lógica)?	-	
	Disponibilidad	¿Se ha especificado el horario en que debe estar disponible el sistema para los usuarios?	-
		¿Se han especificados los acuerdos de nivel de servicio del sistema?	-
Tolerancia a fallos	¿Se definen formas de medir la frecuencia de fallos?	-	
	¿Se cuenta con una forma específica para medir el impacto funcional y sobre los datos de los fallos hardware/software (severidad hardware/software)?	-	
	¿El sistema cuenta con mecanismos para autogestionar el tiempo entre fallos?	-	
	¿El sistema podría determinar la probabilidad de que ocurra un fallo (viabilidad lógica del fallo)?	-	
Capacidad de recuperación	¿Un mecanismo de recuperación a fallos ha sido contemplado en la especificación (incluye elementos como la definición de logs de errores y logs de datos procesado con éxito)?	-	
	¿Se establecieron los horarios a realizar copias de seguridad para los datos afectados?	-	
	¿Se estableció los tiempos de vigencia de la copia de seguridad para los datos afectados?	-	
	¿Se establecieron los tipos de copias de seguridad para los datos afectados?	-	
	¿Se definen formas para corroborar el adecuado restablecimiento del sistema?	-	

Subcaracterística	Alto	Básico	No Evidenciado	No considerado	Sin Respuesta	Promedio de Aseguramiento	Nivel de aseguramiento
Madurez	0	0	0	0	8	0,00	-
Disponibilidad	0	0	0	0	2	0,00	-
Tolerancia a fallos	0	0	0	0	4	0,00	-
Capacidad de recuperación	0	0	0	0	5	0,00	-
Total	0	0	0	0	19	0,00	-

Nivel de Aseguramiento	Equivalencia promedio	Color correspondiente	Interpretación
Alto	Igual a 3	verde	El aspecto que cubre la pregunta ha sido considerado con evidencia suficiente
Básico	>= 2 y < 3	amarillo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias son débiles
No Evidenciado	>= 1 y < 2	rojo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias no son claras/suficientes/completas
No Considerado	>= 0 y < 1	gris	El aspecto que cubre la pregunta No ha sido considerado

Ilustración 4. Aseguramiento de la fiabilidad de una etapa.

Visualizar el nivel de aseguramiento de calidad de la característica de fiabilidad para todo el proyecto (ver *Ilustración 5*). Este nivel se muestra en la hoja principal del artefacto de aseguramiento, y se interpreta de la siguiente manera:

Proyecto:								
Lider de Proyecto:								
Subcaracterística	Nivel de Aseguramiento por Etapa						Nivel de Aseguramiento Proyecto	
	Análisis	Diseño	Codificación	Pruebas	Despliegue	Mantenimiento		
Madurez	-	-	-	-			-	
Disponibilidad	-	-	-	-	-	-	-	
Tolerancia a fallos	-	-	-	-			-	
Capacidad de recuperación	-	-	-	-			-	
Avance	-	-	-	-	-	-	-	

Ilustración 5. Resultados globales del nivel de aseguramiento de la fiabilidad en el proyecto.

Los valores cualitativos del recuadro verde hacen referencia al nivel por subcaracterística de aseguramiento de la fiabilidad en cada una de las etapas, estos niveles son tomados del resumen por etapa.

Los valores cualitativos del recuadro amarillo de la última columna de la tabla hacen referencia al nivel de aseguramiento por cada una de las subcaracterísticas de la fiabilidad en todo el proyecto.

Los valores cualitativos del recuadro naranja de la última fila de la tabla hacen referencia al nivel de aseguramiento de la fiabilidad como característica por cada etapa del ciclo de desarrollo del proyecto. El valor que se muestra en el recuadro rojo representa el nivel total de la implementación de la fiabilidad durante el ciclo de desarrollo de software de todo el proyecto.

Estos datos arrojados en el artefacto de aseguramiento de calidad de la característica de fiabilidad permiten una toma de decisiones temprana, ante la ausencia o cumplimiento de los aspectos de calidad abordados en el proyecto a través de sus diferentes etapas. Para ello, el artefacto de aseguramiento propuesto incluye dos tablas de interpretación de los niveles definidos que permiten guiar el estado del aseguramiento para cada etapa como lo muestra la *Ilustración 6* e *Ilustración 7*.

Las escalas del nivel de aseguramiento son dadas haciendo uso de la escala Likert [46] y se propone para cada una su correspondencia e interpretación: 3 es la calificación más alta correspondiente a una consideración suficiente del aspecto y 0 la calificación más baja

correspondiente a aspecto no considerado (ver *Ilustración 8*). Los umbrales serán interpretados acorde a los lineamientos, definiciones y necesidades del proyecto.

Nivel de Aseguramiento	Equivalencia	Color correspondiente	Interpretación
Alto	igual a 3	verde	La característica ha sido considerada en la etapa con evidencias suficientes
Básico	$>= 2$ y < 3	amarillo	La característica ha sido considerada en la etapa pero las evidencias son debiles
No Evidenciado	$>= 1$ y < 2	rojo	La característica ha sido considerada en la etapa pero las evidencias no son claras/suficientes/completas
No Considerado	> 0 y < 1	gris	La característica ha No ha sido consideradoa en la etapa

Ilustración 6. Interpretación del nivel de aseguramiento en la característica.

Nivel de Aseguramiento	Equivalencia	Color correspondiente	Interpretación
Alto	igual a 3	verde	El aspecto que cubre la pregunta ha sido considerado con evidencia suficiente
Básico	$>= 2$ y < 3	amarillo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias son debiles
No Evidenciado	$>= 1$ y < 2	rojo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias no son claras/suficientes/completas
No Considerado	$>= 0$ y < 1	gris	El aspecto que cubre la pregunta No ha sido considerado

Ilustración 7. Interpretación del nivel de aseguramiento en los aspectos que cubre la pregunta.

Precondición: Se asume que en la etapa se cuenta con el artefacto de especificación		
Subcaracterística	Preguntas planteadas	Nivel de aseguramiento
N	¿Los mensajes especificados son claros para realizar los diferentes caminos definidos?	
	¿Los datos han sido especificados conforme a lo que se esp... que sea el resultado correcto y con una precisión definida (exactitud)?	
	¿Los datos que debe manejar el sistema han sido especificados de forma completa funcionalmente (frente al negocio) (Compleitud funcional)?	

Ilustración 8. Escala del nivel de aseguramiento.

El método cuenta con una tabla de apoyo para la actividad de selección de la técnica de aseguramiento, la cual fue presentada en la parte 3.4 del documento (Ver *Tabla 13*). El objetivo de esta tabla es que el proyecto pueda determinar la técnica que mejor se ajuste al tipo de proyecto de manera que ésta apoye al método de aseguramiento propuesto.

A continuación, se describe cada una de las actividades del método de aseguramiento de calidad de la característica de la fiabilidad.

3.5.1 Levantamiento de requisitos de fiabilidad

En esta actividad el rol de analista realiza el proceso de levantamiento de requisitos de fiabilidad. Para esta actividad la empresa está en libertad de elegir el proceso que se ajuste mejor a sus necesidades y metodología de desarrollo que usen. Al finalizar esta actividad se obtendrán los requisitos documentados y especificados.

3.5.2 Aseguramiento de la fiabilidad en etapa de análisis

El aseguramiento de calidad de la característica de fiabilidad en la etapa de análisis la realiza el rol de analista QA en conjunto de interesados del proyecto (Stakeholders) (ver *Tabla 14*). A continuación, se especifican las tareas a desarrollar en esta actividad con más detalle.

Seleccionar Técnica de aseguramiento

Usando la *Tabla 13*, el analista de QA debe analizar las técnicas de aseguramiento descritas y optar por la que mejor se ajusta a las características de la fase, como la metodología implementada y/o los artefactos de salida que ésta misma provee.

Aplicar artefacto de aseguramiento en etapa de análisis

El analista de QA en colaboración de interesados del proyecto (Stakeholders) se encargarán de aplicar, registrar y evidenciar cada una de las respuestas a las preguntas plasmadas en el artefacto de aseguramiento de calidad de la característica de fiabilidad (ver *Ilustración 9*), con el fin de enriquecer y detectar ausencias o incumplimientos de los atributos de la fiabilidad en las especificaciones. Este artefacto permite hacer seguimiento por etapa al nivel de aseguramiento como se indicó en la parte de Generalidades ítem 2.

Etapa del ciclo de vida de Software: Análisis		
Precondición: Se asume que en la etapa se cuenta con el artefacto de especificación		
Subcaracterística de fiabilidad	Preguntas planteadas	Nivel de aseguramiento
Madurez	¿Los mensajes especificados son claros para realizar los diferentes caminos definidos?	-
	¿Los datos han sido especificados conforme a lo que se espera que sea el resultado correcto y con una precisión definida (exactitud)?	-
	¿Los datos que debe manejar el sistema han sido especificados de forma completa funcionalmente (frente al negocio) (Complejidad funcional)?	-
	¿Los datos del sistema han sido especificados con coherencia (estables) y las salidas que se esperan sean las correctas (consistencia)?	-
	¿Los datos del sistema han sido especificados con claridad frente a su verdadero valor (credibilidad-veracidad)?	-
	¿El sistema cuenta con algún mecanismo que protege los datos de individuos o softwares no autorizados a manipularlos (confidencialidad de los datos)?	-
	¿Se cuenta con formas explícitas para observar la evolución de los datos frente a los cambios que puedan tener (Trazabilidad)?	-
Disponibilidad	¿Las funciones tienen una abstracción adecuada de la realidad en el sistema (fidelidad-veracidad lógica)?	-
	¿Se ha especificado el horario en que debe estar disponible el sistema para los usuarios?	-
Tolerancia a fallos	¿Se han especificados los acuerdos de nivel de servicio del sistema?	-
	¿Se definen formas de medir la frecuencia de fallos hardware/software?	-
	¿Se cuenta con una forma específica para medir el impacto funcional y sobre los datos de los fallos hardware/software (severidad hardware/software)?	-
	¿El sistema cuenta con mecanismos para autogestionar el tiempo entre fallos?	-
Capacidad de recuperación	¿El sistema podría determinar la probabilidad de que ocurra un fallo (viabilidad lógica del fallo)?	-
	¿Un mecanismo de recuperación a fallos ha sido contemplado en la especificación (incluye elementos como la definición de logs de errores y logs de datos procesado con éxito)?	-
	¿Se establecieron los horarios a realizar copias de seguridad para los datos afectados?	-
	¿Se estableció los tiempos de vigencia de la copia de seguridad para los datos afectados?	-
	¿Se establecieron los tipos de copias de seguridad para los datos afectados?	-
	¿Se definen formas para corroborar el adecuado restablecimiento del sistema?	-

Subcaracterística	Alto	Básico	No Evidenciado	No considerado	Sin Respuesta	Promedio de Aseguramiento	Nivel de aseguramiento
Madurez	0	0	0	0	8	0,00	-
Disponibilidad	0	0	0	0	2	0,00	-
Tolerancia a fallos	0	0	0	0	4	0,00	-
Capacidad de recuperación	0	0	0	0	5	0,00	-
Total	0	0	0	0	19	0,00	-

Nivel de Aseguramiento	Equivalencia a promedio	Color correspondiente	Interpretación
Alto	igual a 3	verde	El aspecto que cubre la pregunta ha sido considerado con evidencia suficiente
Básico	>= 2 y < 3	amarillo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias son débiles
No Evidenciado	>= 1 y < 2	rojo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias no son claras/suficientes/completas
No Considerado	>= 0 y < 1	gris	El aspecto que cubre la pregunta No ha sido considerado

Ilustración 9. Aseguramiento de calidad de la característica de fiabilidad en etapa de análisis.

3.5.3 Realizar ajuste del artefacto de análisis

De acuerdo con los resultados de la tarea anterior, si hay respuestas de valor “NO EVIDENCIADO” frente al cumplimiento de los criterios de aseguramiento (elementos semánticos cubiertos a través de las preguntas definidas en el método), el método marca en rojo estos aspectos por cubrir antes de pasar a la etapa de diseño de manera que el proyecto debe registrar un plan de acción para solventar el incumplimiento de fiabilidad en las preguntas respectivas. En esta tarea el analista realiza las correcciones pertinentes de acuerdo con el plan de acción definido para continuar con el proceso de desarrollo. Esta tarea deberá ser realizada con el fin de no minimizar el nivel total de aseguramiento de la fiabilidad.

3.5.4 Diseño de la solución

En esta actividad el rol de arquitecto realiza el diseño de la solución a través de la elaboración de los diagramas técnicos del producto software. Para esta tarea se sugiere que se realicen los diagramas descritos en la *Tabla 7*.

3.5.5 Aseguramiento de la fiabilidad en etapa de diseño

El aseguramiento de calidad de la característica de fiabilidad en la etapa de diseño la realiza el rol de arquitecto QA (ver *Tabla 14*).

Seleccionar Técnica de aseguramiento

Usando la *Tabla 13* el arquitecto de QA debe analizar las técnicas de aseguramiento descritas y optar por la que mejor se ajusta a las características de la fase y tipo de proyecto, como la metodología implementada y/o los artefactos de salida que esta misma provee.

Aplicar artefacto de aseguramiento en etapa de diseño

El arquitecto de QA se encarga de aplicar, registrar y evidenciar cada una de las respuestas a las preguntas plasmadas en el artefacto de aseguramiento de calidad de la característica de fiabilidad (ver *Ilustración 10*), con el fin de enriquecer y detectar ausencias o incumplimientos de los atributos de la fiabilidad en las especificaciones. Este artefacto permite hacer seguimiento por etapa al nivel de aseguramiento como se indicó en la parte de Generalidades ítem 2.

Etapas del ciclo de vida de Software: Diseño		
Subcaracterística de fiabilidad	Preguntas planteadas	Nivel de aseguramiento
Madurez	¿Las secuencias de los diferentes caminos del sistema se ven reflejados correctamente en el diagrama dinámico?	-
	¿En el diagrama dinámico se visualizan los datos conforme a lo que se espera que sea el resultado correcto (exactitud)?	-
	¿En el diagrama dinámico se puede observar la coherencia (estabilidad) de los datos (consistencia)?	-
	¿En el diagrama dinámico se ven reflejados los datos con claridad frente al verdadero valor (credibilidad-veracidad)?	-
	¿En el diagrama dinámico se observa el tratamiento de protección de todos los datos del sistema (confidencialidad)?	-
	¿La evolución frente a los cambios de los datos se ve reflejada en las secuencias del diagrama dinámico?	-
Disponibilidad	¿En el diagrama de contenedores/componentes se cuenta con algún elemento HW/SW que apoye la disponibilidad del sistema para los usuarios (Acuerdos de Nivel de Servicio)?	-
	¿En el diagrama de contenedores/componentes se cuenta con algún elemento HW que apoye la identificación de una falla?	-
Tolerancia a fallos	¿En el diagrama de componentes/contenedores se mide de alguna manera la frecuencia de fallos software (severidad)?	-
	¿En el diagrama de componentes/contenedores se cuenta con algún elemento HW/SW que apoye la autogestión del tiempo?	-
	¿Para determinar la probabilidad de que ocurra un fallo se encuentra a algún elemento HW/SW en el diagrama de componentes/contenedores (viabilidad lógica del fallo)?	-
	¿En el diagrama de contenedores/componentes se cuenta con algún elemento HW/SW que permita la recuperación a fallos que se puedan presentar en el sistema?	-
Capacidad de recuperación	¿En el diagrama de contexto incluye un documento de apoyo técnico donde se pueda indicar los horarios a restablecer los copias de seguridad de datos afectados?	-
	¿El diagrama de contexto incluye un documento de apoyo que permita definir el tiempo de vigencia de la copia de seguridad de los datos afectados?	-
	¿En el diagrama de contexto se cuenta con un documento de apoyo que detalle la forma en que se debe dar el restablecimiento de sistema en caso de fallo?	-

Subcaracterística	Alto	Básico	No Evidenciado	No considerado	Sin Respuesta	Promedio de Aseguramiento	Nivel de aseguramiento
Madurez	0	0	0	0	7	0,00	-
Disponibilidad	0	0	0	0	1	0,00	-
Tolerancia a fallos	0	0	0	0	4	0,00	-
Capacidad de recuperación	0	0	0	0	4	0,00	-
Total	0	0	0	0	16	0,00	-

Nivel de Aseguramiento	Equivalencia	Color correspondiente	Interpretación
Alto	igual a 3	verde	El aspecto que cubre la pregunta ha sido considerado con evidencia suficiente
Básico	>= 2, y < 3	amarillo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias son débiles
No Evidenciado	>= 1 y < 2	rojo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias no son claras/suficientes/completas
No Considerado	>= 0 y < 1	gris	El aspecto que cubre la pregunta no ha sido considerado

Ilustración 10. Aseguramiento de calidad de la característica de fiabilidad en etapa de diseño.

3.5.6 Realizar ajuste del artefacto de diseño

De acuerdo con los resultados de la tarea anterior, si hay respuestas de valor “NO EVIDENCIADO” frente al cumplimiento de los criterios de aseguramiento (aspectos semánticos cubiertos a través de las preguntas definidas en el método), el método marcará en rojo estos aspectos por cubrir antes de pasar a la etapa de codificación, de manera que el proyecto debe registrar un plan de acción para solventar el incumplimiento de fiabilidad en las preguntas respectivas.

En esta tarea el arquitecto realiza las correcciones pertinentes de acuerdo con el plan de acción definido para continuar con el proceso de desarrollo. Esta tarea deberá ser realizada con el fin de no minimizar el nivel total de aseguramiento de la fiabilidad.

3.5.7 Codificación del producto

En esta actividad el rol de programador realiza la implementación de los requisitos documentados en la etapa de análisis y diagramados en la de diseño. Al finalizar esta actividad se obtiene el código fuente del producto.

3.5.8 Aseguramiento de la fiabilidad en etapa de codificación

El aseguramiento de calidad de la característica de fiabilidad en la etapa de codificación la realiza el rol de programador QA (ver *Tabla 14*).

A continuación, se especifican las tareas a desarrollar en esta actividad con más detalle.

Seleccionar Técnica de aseguramiento

Usando la *Tabla 13* el programador QA deberá analizar las técnicas de aseguramiento ahí descritas y optar por la que mejor se ajusta a las características de la fase o tipo de proyecto, como la metodología implementada y/o los artefactos de salida que esta misma provee.

Aplicar artefacto de aseguramiento en etapa de codificación

El programador QA se encarga de aplicar, registrar y evidenciar cada una de las respuestas a las preguntas plasmadas en el artefacto de aseguramiento de calidad de la característica de fiabilidad (ver *Ilustración 11*), con el fin de enriquecer y detectar ausencias o incumplimientos de los atributos de la fiabilidad en las especificaciones. Este artefacto permite hacer seguimiento por etapa al nivel de aseguramiento como se indicó en la parte de Generalidades ítem 2.

Etapa del ciclo de vida de Software: Codificación		
Subcaracterística de fiabilidad	Preguntas planteadas	Nivel de aseguramiento
Madurez	¿En las funciones o métodos implementados en el código fuente se permite la verificación de que los datos están conformes a lo que se espera que sea el resultado correcto y con una precisión definida (exactitud)?	-
	¿En las funciones o métodos codificados se puede corroborar que los datos son coherentes (estables) y sus salidas que se esperan son las correctas (consistencia)?	-
	¿En las funciones o métodos en el código fuente se evidencia que los datos están implementados con claridad frente a su verdadero valor (credibilidad-veracidad)?	-
	¿En las funciones o métodos creados se implementó un mecanismo que protege los datos de individuos o software no autorizados a manipular los datos (confidencialidad)?	-
	¿La evolución de los datos frente a los cambios que puedan tener ha sido implementada en las funciones o métodos del código fuente (trazabilidad)?	-
	¿La abstracción adecuada de la realidad ha sido implementada en las funciones o métodos del código fuente (fidelidad-veracidad)?	-
Disponibilidad	¿La disponibilidad del sistema para los usuarios ha sido configurada/ implementada a través de/ o los elementos hardware/software definidos en el diseño (Acuerdos de Nivel de Servicio)?	-
	¿Existe algún elemento hardware para la identificación de una falla en el sistema?	-
Tolerancia a fallos	¿En las funciones o métodos se codificó la medición de la frecuencia de fallos software del sistema?	-
	¿En las funciones o métodos se implementó la medición del impacto funcional y sobre los datos de los fallos hardware/software?	-
	¿La autogestión del tiempo entre fallos fue implementada en el código fuente?	-
Capacidad de recuperación	¿Fue implementada una función o método en el código fuente que pueda determinar la probabilidad de que ocurra un fallo (viabilidad lógica de los fallos)?	-
	¿La recuperación a posibles fallos del sistema ha sido configurada/ implementada a través de/ (los) elementos hardware/software definidos en el diseño (incluye elementos como la creación de logs de errores, logs de datos procesado con éxito)?	-
	¿Los tiempos para establecer las copias de seguridad de datos afectados definidos en el diseño fueron considerados en las configuraciones/implementaciones del sistema?	-
	¿El tiempo de vigencia de la copia de seguridad de los datos afectados definidos en el diseño fueron considerados en las configuraciones/implementaciones del sistema?	-

Subcaracterística	Alto	Básico	No Evidenciado	No considerado	Sin Respuesta	Promedio de Aseguramiento	Nivel de aseguramiento
Madurez	0	0	0	0	6	0,00	-
Disponibilidad	0	0	0	0	1	0,00	-
Tolerancia a fallos	0	0	0	0	5	0,00	-
Capacidad de recuperación	0	0	0	0	4	0,00	-
Total	0	0	0	0	16	0,00	-

Nivel de Aseguramiento	Equivalencia	Color correspondiente	Interpretación
Alto	igual a 3	verde	El aspecto que cubre la pregunta ha sido considerado con evidencia suficiente
Básico	>= 2 < 3	amarillo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias son débiles
No Evidenciado	>= 1 < 2	rojo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias no son claras/suficientes/completas
No Considerado	>= 0 < 1	gris	El aspecto que cubre la pregunta no ha sido considerado

Ilustración 11. Aseguramiento de calidad de la fiabilidad en etapa de codificación.

3.5.9 Realizar ajuste del artefacto de codificación

De acuerdo con los resultados de la tarea anterior, si hay respuestas de valor “NO EVIDENCIADO” frente al cumplimiento de los criterios de aseguramiento (aspectos semánticos cubiertos a través de las preguntas definidas en el método), el método marcará en rojo estos aspectos por cubrir antes de pasar a la etapa de pruebas, de manera que el proyecto debe registrar un plan de acción para solventar el incumplimiento de fiabilidad en las preguntas respectivas. En esta tarea el programador realiza las correcciones pertinentes de acuerdo con el plan de acción definido para continuar con el proceso de desarrollo. Esta tarea deberá ser realizada con el fin de no minimizar el nivel total de aseguramiento de la fiabilidad.

3.5.10 Ejecución los casos de prueba

En esta actividad el rol de analista de pruebas realiza el análisis, diseño y ejecución de los casos de pruebas orientados a los requisitos de fiabilidad documentados para el proyecto. Al finalizar esta actividad se obtiene las evidencias de los casos de prueba ejecutados.

3.5.11 Aseguramiento de la fiabilidad en etapa de pruebas

El aseguramiento de calidad de la característica de fiabilidad en la etapa de codificación realiza el rol de QA (ver *Tabla 14*).

A continuación, se especifican las tareas a desarrollar en esta tarea con más detalle.

Seleccionar Técnica de aseguramiento

Usando la *Tabla 13* el QA debe analizar las técnicas de aseguramiento ahí descritas y optar por la que mejor se ajusta a las características de la fase o tipo de proyecto, como la metodología implementada y/o los artefactos de salida que esta misma provee.

Aplicar artefacto de aseguramiento en etapa de pruebas

El QA se encarga de aplicar, registrar y evidenciar cada una de las respuestas a las preguntas plasmadas en el artefacto de aseguramiento de calidad de la característica de fiabilidad (ver *Ilustración 12*), con el fin de enriquecer y detectar ausencias o incumplimientos de los atributos de la fiabilidad en las especificaciones. Este artefacto

permite hacer seguimiento por etapa al nivel de aseguramiento como se indicó en la parte de Generalidades ítem 2.

Etapas del ciclo de vida de Software: Prueba		
Subcaracterística de fiabilidad	Preguntas planteadas	Nivel de aseguramiento
Madurez	¿Se han diseñado casos de prueba que verifiquen que los datos están conforme a lo que se espera que sea el resultado correcto y con una precisión definida (exactitud)?	-
	¿Existen casos de prueba diseñados que corrobore que los datos arrojadostienen la coherencia (estables) y las salidas que se esperan sean las correctas (consistencia)?	-
	¿Los casos de prueba diseñados verifican que los datos resultantes son claros frente a su verdadero valor (credibilidad-veracidad)?	-
	¿Los casos de prueba creados verifican la protección de los datos antes individuos o softwares no autorizados a manipularlos (confidencialidad de los datos)?	-
	¿Se han diseñado casos de prueba que verifiquen la evolución de los datos frente a los cambios que pueden tener (resiliencia)?	-
Disponibilidad	¿Se han diseñado casos de prueba que corrobore la abstracción adecuada de la realidad en las funcionalidades a probar (fidelidad-veracidad lógica)?	-
	¿Los casos de prueba diseñados confirman la disponibilidad del sistema para los usuarios (Acuerdos de Nivel de Servicio)?	-
Tolerancia a fallos	¿Se han diseñado caso de prueba que permitan verificar la identificación del fallo hardware/software?	-
	¿Se han diseñado caso de prueba para verificar la medición de la frecuencia de fallos hardware/software (severidad)?	-
	¿Se diseñaron casos de prueba que verifiquen la autogestión del tiempo entre fallos?	-
	¿Se diseñaron casos de prueba que verifiquen la probabilidad de que ocurra un fallo (viabilidad lógica del fallo)?	-
Capacidad de recuperación	¿Se diseñaron casos de prueba que permitan corroborar la recuperación de fallos presentados por el sistema (Incluye elementos a considerar como logs de errores y logs de datos procesado con éxito)?	-
	¿Los casos de prueba diseñados verifican los horarios de restablecimiento de las copias de seguridad de los datos afectados?	-
	¿El tiempo de vigencia de la copia de seguridad de datos afectados se verifican en los casos de pruebas diseñados?	-
	¿Los casos de prueba diseñados verifican el restablecimiento del sistema en caso de fallos?	-

Subcaracterística	Alto	Básico	No Evidenciado	No considerado	Sin Respuesta	Promedio de Aseguramiento	Nivel de aseguramiento
Madurez	0	0	0	0	6	0,00	-
Disponibilidad	0	0	0	0	1	0,00	-
Tolerancia a fallos	0	0	0	0	4	0,00	-
Capacidad de recuperación	0	0	0	0	4	0,00	-
Total	0	0	0	0	15	0,00	-

Nivel de Aseguramiento	Equivalencia	Color correspondiente	Interpretación
Alto	igual a 3	verde	El aspecto que cubre la pregunta ha sido considerado con evidencia suficiente
Básico	>= 2 y < 3	amarillo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias son débiles
No Evidenciado	>= 1 y < 2	rojo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias no son claras/suficientes/completas
No Considerado	>= 0 y < 1	gris	El aspecto que cubre la pregunta No ha sido considerado

Ilustración 12. Aseguramiento de calidad de la fiabilidad en etapa de pruebas.

3.5.12 Realizar ajuste del artefacto de pruebas

De acuerdo con los resultados de la tarea anterior, si hay respuestas de valor “NO EVIDENCIADO” frente al cumplimiento de los criterios de aseguramiento (aspectos semánticos cubiertos a través de las preguntas definidas en el método), el método marca en rojo estos aspectos por cubrir antes de pasar a la etapa de despliegue, de manera que el proyecto debe registrar un plan de acción para solventar el incumplimiento de fiabilidad en las preguntas respectivas.

En esta tarea el analista de pruebas realiza las correcciones pertinentes de acuerdo con el plan de acción definido para continuar con el proceso de desarrollo. Esta tarea deberá ser realizada con el fin de no minimizar el nivel total de aseguramiento de la fiabilidad.

3.5.13 Desplegar funcionalidades aprobadas

En esta actividad el rol de implementador realiza el despliegue de las funcionalidades que se encuentran aprobadas sobre el ambiente de producción.

3.5.14 Aseguramiento de la fiabilidad en etapa de despliegue

El aseguramiento de calidad de la característica de fiabilidad en la etapa de despliegue la realiza el rol de Implementador QA (ver *Tabla 14*). Este rol se encarga de verificar si el aseguramiento de calidad de la característica de fiabilidad de las etapas de análisis, diseño, codificación y pruebas han sido completadas y registrar los resultados de esta verificación en el artefacto propuesto (ver *Ilustración 13*). Este artefacto permite hacer seguimiento por etapa al nivel de aseguramiento como se indicó en la parte de Generalidades ítem 2, de manera que solamente cuando la etapa de pruebas tenga un nivel de aseguramiento BÁSICO o ALTO (Se sugiere que el nivel este en ALTO) frente a la fiabilidad, el despliegue podrá ser realizado.

Etapa del ciclo de vida de Software: Despliegue	
Preguntas planteadas	Nivel de aseguramiento
¿El sistema que ha sido previamente analizado, diseñado, codificado y probado ha sido llevado al ambiente productivo considerando todos los aspectos de la característica de fiabilidad?	-

Nivel de Aseguramiento	Equivalencia	Color correspondiente	Interpretación
Alto	igual a 3	verde	El aspecto que cubre la pregunta ha sido considerado con evidencia suficiente
Básico	>= 2 y < 3	amarillo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias son débiles
No Evidenciado	>= 1 y < 2	rojo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias no son claras/suficientes/completas
No Considerado	>= 0 y < 1	gris	El aspecto que cubre la pregunta No ha sido considerado

Ilustración 13. Aseguramiento de calidad de la fiabilidad en etapa de despliegue.

3.5.15 Ajustar/Mejorar proyecto

En esta actividad el rol de Ingeniero de mantenimiento será el encargado de realizar los cambios o ajustes del producto software, solicitados y aprobados por el cliente, que de acuerdo con su criticidad y complejidad podrán requerir ser analizados, diseñados, codificados y probados hasta llegar nuevamente al despliegue. Este ciclo tendrá en cuenta iterativamente los elementos semánticos planteados para la característica de fiabilidad y será registrada su verificación en el artefacto propuesto (ver *Ilustración 14*).

Etapa del ciclo de vida de Software: Mantenimiento	
Preguntas planteadas	Nivel de aseguramiento
¿El cambio que ha sido identificado, ha sido analizado, diseñado, codificado y probado para un nuevo despliegue?	-
¿El cambio que ha sido identificado, ha sido analizado, diseñado, codificado y probado para un nuevo despliegue?	-

¿Cantidad de cambios por realizar?	Alto	Básico	No Evidenciado	No considerado	Sin Respuesta	Promedio de Aseguramiento	Nivel de aseguramiento
2	0	0	0	0	2	0,00	-

Nivel de Aseguramiento	Equivalencia	Color correspondiente	Interpretación
Alto	igual a 3	verde	El aspecto que cubre la pregunta ha sido considerado con evidencia suficiente
Básico	>= 2 y < 3	amarillo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias son débiles
No Evidenciado	>= 1 y < 2	rojo	El aspecto que cubre la pregunta ha sido considerado pero las evidencias no son claras/suficientes/completas
No Considerado	>= 0 y < 1	gris	El aspecto que cubre la pregunta No ha sido considerado

Ilustración 14. Aseguramiento de calidad de la fiabilidad en etapa de mantenimiento.

Capítulo 4 - Evaluación del método

Para la evaluación del método de aseguramiento de calidad de la característica de fiabilidad descrito en el anterior capítulo, se realiza un focus group el cual permite realizar debates planificados y diseñados estratégicamente para obtener las percepciones personales de los miembros del grupo en un área definida de interés para la investigación [47]. La elección del método de focus group se centra además en que éste permite: obtener realimentación de los participantes sobre preguntas de investigación o nuevos conceptos, reconocer pasadas experiencias que puedan estudiarse con mayor detalle empleando otros métodos, realizar la evaluación inicial de potenciales soluciones, recopilar recomendaciones de lecciones aprendidas o generar ideas, obtener realimentación con respecto a la manera en que los modelos o conceptos son presentados o documentados y descubrir importantes motivaciones [48].

En esta sección se describe inicialmente la estructura teórica del método de investigación, y posteriormente se presenta la aplicación del método. Finalmente, se detallan los productos de trabajo generados a través de éste método.

4.1 Estructura del Focus Group

Este método se basa en la estructura procedimental propuesta por M. Mendoza, et. al en [48], la cual describe un proceso satisfactorio para implementar un focus group en ingeniería de software. La estructura define cuatro etapas, descritas a continuación:

- a) Planteamiento de la investigación, donde se establecen los alcances de la aplicación del proceso, la preparación del material, agenda, instrumento del debate.
- b) Definición de grupos de discusión, Basado en la caracterización, definición y selección de los participantes.
- c) Conducción de las sesiones del Focus Group, llevar a cabo la agenda establecida en la primera etapa con el grupo de discusión seleccionado.
- d) Análisis de la información y reporte de resultados, el objetivo de esta etapa es realizar el análisis de la información cuantitativa y cualitativa obtenida en los debates generados en el Focus group

4.2 Realización del focus group

El método de aseguramiento de calidad de la característica de fiabilidad fue sometido a evaluación mediante la estrategia de investigación del focus group y con la información recopilada en éste, la versión inicial fue ajustada con el fin de mejorar la propuesta, esto es, siguiendo la metodología de investigación definida para el trabajo de investigación. A continuación, se presenta en detalle cada una de las fases llevadas a cabo para realizar el focus group.

4.2.1 Fase de planeamiento de la investigación

Definición del problema de investigación

El objetivo con el que se aplica el focus group es: Evaluar la utilidad en el método de aseguramiento de calidad de la característica de fiabilidad propuesto. Como base para el planeamiento se envió previamente a los participantes un resumen del método propuesto y un documento extendido con todas las actividades del proceso y sus tareas descritas a detalle.

Preparación de materiales y métodos, a cumplir por parte del grupo investigador.

Posteriormente se definen los procedimientos y técnicas que abrirán la sesión de debate, así mismo se plantean las estrategias que permitirán obtener la información objetivo. También se generan los documentos de insumo, los cuales serán diligenciados por los participantes durante el transcurso de la sesión; se compartirá la ficha de asistencia al focus group, la evaluación del método construido, la definición de los protocolos de la sesión y la definición con mayor precisión de la agenda. Todos estos materiales y métodos descritos serán útiles para evaluar los “Recursos objeto de debate” en siguientes etapas de acuerdo con las tareas que se describen a continuación:

Definición de estructura, en la que se definen los aspectos protocolarios para el debate.

Fecha: 06 octubre 2022.

Hora de inicio: 06:00 p.m.

Hora de finalización: 07:10 p.m.

Lugar: Sesión virtual.

Actividad: Sesión de debate del focus group.

Tema para tratar: Evaluación del método para el aseguramiento de la característica de la fiabilidad durante todo el ciclo de desarrollo software.

Objeto de investigación: Método para el aseguramiento de la característica de la fiabilidad durante todo el ciclo de desarrollo software.

Moderador: PhD. Sandra Lorena Buitrón Ruiz

Relator: Luisa Fernanda Gómez Robles - Leidy Vanesa Fernández Muñoz

Participantes: Ing. Erika Dayana Rosero Giraldo, Ing. Juan Sebastián Páez, Ing. Omar Ricardo Bonilla Hernández, Esp. Andrés Felipe Navia Z, Ing. Melissa Muñoz G, Ing. Karen Zúñiga, Ing. Daniel Eduardo Paz.

Objetivo del Focus Group

Evaluar la utilidad del método propuesto acorde a la experticia de los participantes.

Medidas utilizadas para la evaluación

Para el método de aseguramiento de la característica de la fiabilidad durante todo el ciclo de desarrollo software se considera necesario evaluar de manera cuantitativa (ver *Tabla 16*) la utilidad de éste, además se establecen preguntas cualitativas (ver *Tabla 17*) para que los participantes pudieran dar su opinión respecto al método.

La estrategia para evaluar la utilidad del método se basa en la definición de la palabra Utilidad según la RAE [49]. La cual divide esta cualidad en tres aspectos fundamentales; conveniencia, interés y provecho. De acuerdo con esto, se plantean tres preguntas, cada una con el objetivo de evaluar cada uno de los aspectos mencionados, así: una pregunta cuantitativa para conveniencia, otra para interés y finalmente una pregunta cualitativa para provecho. Para evaluar las preguntas cuantitativas (ver *Tabla 16*) del focus group, se hace uso de la escala Likert [46] y se propone para cada una su correspondencia; donde 5 es la calificación más alta correspondiente a Totalmente de acuerdo con el aspecto medido y 1 la calificación más baja correspondiente a Totalmente en desacuerdo con el aspecto medido (ver *Tabla 15*).

Tabla 15. Escala de correspondencia para preguntas cuantitativas.

Escala Likert	Correspondencia
5	Totalmente de acuerdo con el aspecto medido
4	De acuerdo con el aspecto medido
3	Ni de acuerdo ni en desacuerdo con el aspecto medido
2	En desacuerdo con el aspecto medido
1	Totalmente en desacuerdo con el aspecto medido

Las preguntas cualitativas son necesarias para que los participantes brinden retroalimentación al estado actual del proyecto, su beneficio en la práctica y futuras propuestas de mejora. Así, impulsar la posteridad del método construido mediante próximas optimizaciones.

Tabla 16. Preguntas cuantitativas del focus group.

No	Preguntas
1	¿Considera conveniente este método para el aseguramiento de la calidad de la característica de la fiabilidad en los proyectos software?
2	¿Considera que estaría interesado en hacer uso del método propuesto en los proyectos software que conoce o lidera?
3	¿Considera que el método propuesto podría ser usado de manera cómoda para en los proyectos software en los que usted participa?

Tabla 17. Preguntas cualitativas del focus group.

No	Preguntas
1	Del estado actual del método propuesto ¿Qué cree usted que se pueda mejorar?
2	Del estado actual del método propuesto ¿Qué nuevos aspectos/elementos pertenecientes al método cree usted que se deberían incluir?
3	¿Cuál(es) podría(n) ser el (los) beneficio(s) que usted considera que este método podría generar si se implementa en los proyectos software que actualmente trabaja?

Para la sesión del focus group se define la siguiente agenda Tabla 18.

Tabla 18. Agenda focus group.

Actividad	Responsable	Duración
Introducción al focus group	Sandra Lorena Buitrón	5 min
Descripción del método de aseguramiento de la característica de la fiabilidad durante todo el ciclo de desarrollo software	Luisa Gómez y Vanesa Fernández	20 min
Debate y recomendaciones de los participantes	Participantes	35 min
Evaluación del método	Participantes	10 min

Definición de instrumentos, materiales y métodos a ser empleados

Materiales utilizados:

- Documento sintetizado del método.
- Evaluación del método de aseguramiento de calidad de la característica de fiabilidad (Formulario de Google).
- Presentación del método en Power Point.

Definición de métodos de captura y registro de información derivada del debate

Puesto que el focus group es de manera virtual y con el ánimo de recopilar de manera más clara y específica las recomendaciones de los asistentes, se tomó la decisión de grabar un video de la totalidad del debate, se debe tomar apuntes de todas las apreciaciones de cada uno de los participantes y de igual manera cada participante debe diligenciar el formulario de Google de preguntas para la evaluación del método de aseguramiento de calidad de la característica de fiabilidad.

Por lo tanto, los métodos de captura y registro de información del debate definidos son:

- Video de la sesión del focus group
- Apuntes de las apreciaciones de los participantes del focus group
- Evaluación (Google Forms) del método propuesto diligenciada por los participantes

Definición de métodos de análisis de información para generar el procesamiento de lo generado en el debate.

Después de la sesión se debe realizar una revisión por parte de los moderadores para el análisis de la información obtenida, extrayendo la información más relevante para refinar el método de aseguramiento de calidad de la característica de fiabilidad.

4.2.2 Fase de definición de grupos de discusión

La caracterización y selección de los participantes fue definido de la siguiente manera:

Selección de participantes

Definición del perfil del participante: (i) Profesionales con mínimo 5 años de experiencia en la industria del software, (ii) profesionales con mínimo 2 años de experiencia en los roles relacionados con el aseguramiento de calidad.

Identificación de participantes potenciales: En esta sesión se tuvo en cuenta que los participantes al focus group tuvieran conocimientos y/o experiencia de aseguramiento de calidad y ciclos de desarrollo de software. En la *Tabla 19* se presenta las personas convocadas para la sesión de debate.

Tabla 19. Participantes al focus group.

Nombre	Formación académica	Nivel académico	Rol que desempeña en la empresa	Años de experiencia
Erika Dayana Rosero Giraldo	Ingeniera de sistemas	Profesional	QA Senior	5
Juan Sebastián Páez	Ingeniero de sistemas	Profesional	Analista Admón. Software / Aseguramiento de calidad	3
Omar Ricardo Bonilla Hernández	Ingeniero de sistemas	Profesional	Analista de pruebas	2
Andrés Felipe Navia Z	Ingeniero Sistemas Especialista en TIC para la educación.	Especialista	Líder de QA	14
Melissa Muñoz G	Ingeniera de sistemas	Profesional	QA Ingeniero automatizador	5
Karen Zúñiga	Ingeniera de sistemas	Profesional	Ingeniera de pruebas	3
Daniel Eduardo Paz Perafán	Ingeniero de sistemas	Profesional	Docente del departamento de sistemas	7

4.2.3 Fase de conducción de la sesión de debate

Secuencia de conducción

El desarrollo del debate fue llevado a cabo con el grupo de expertos, siendo coordinado por la moderadora y los relatores, e integrado por los participantes, haciendo uso del planeamiento, materiales, y el artefacto resultante de la primera fase. Siguiendo la agenda definida previamente, se inicia la sesión del focus group a las 6:05 p.m. del 6 de octubre de 2022, en la sala de virtual adjunta al correo de invitación. Asisten la mayoría de los invitados al focus group y se da inicio bajo la contextualización la PhD. Sandra Lorena Buitrón. Seguido se realizó una exposición del método propuesto, haciendo uso de una presentación

que involucró: el propósito del método, la descripción de actividades en general y las tareas que están al interior de estas. La sesión se conduce acorde al protocolo definido bajo una dinámica de discusión y realimentación de comentarios, ideas, ventajas, desventajas y opiniones acerca de la propuesta método de aseguramiento de calidad de la característica de fiabilidad.

Al finalizar la discusión se continúa solicitándoles el diligenciamiento del formulario de evaluación de la propuesta con el fin de capturar algunas medidas iniciales en términos cuantitativos acerca de la utilidad la propuesta de investigación.

Captura de información

La toma de información fue realizada por los relatores: Luisa Fernanda Gómez Robles y Leidy Vanesa Fernández Muñoz, registrando los conceptos, detalles, características y aportes más relevantes ofrecidos por los participantes sobre cada una de las actividades y en general del método propuesto debatido en esta sesión. Además, fueron empleados como técnicas de captura de información: la grabación de video y el formulario de evaluación del método.

Rol del moderador

Este rol fue desarrollado por parte de la PhD Sandra Lorena Buitrón. Es importante recalcar que el vínculo del moderador con el método logró un control en cuanto a: las intervenciones y la gestión de la actividad, con el fin de seguir la agenda propuesta.

Productos de trabajo

Durante el planeamiento y la ejecución del focus group se obtuvieron una serie de documentos (productos de trabajo), que relacionan información acerca de conceptos, técnicas e instrumentos empleados en este método. Algunos de ellos son descritos a continuación:

- Documento sintetizado del método describiendo las actividades en detalle, un archivo Excel donde se encontraba el artefacto de aseguramiento de calidad de la característica de fiabilidad durante el ciclo de desarrollo software y una imagen con el modelo del método.
- Protocolo del debate y la agenda respectiva.

-Reporte de participantes, en este documento describe las características de los expertos que asistieron al debate.

-Reporte de análisis de resultados, en este documento se registran las conclusiones y características del debate.

4.2.4 Fase de análisis de información y reporte de resultados

Análisis de la información

El análisis de la información se realiza en primer lugar de las anotaciones obtenidas al momento del debate y se complementan con la grabación en video de la sesión del focus grupo teniendo en cuenta el siguiente orden para su análisis:

- Revisión de la información recolectada en el medio respectivo
- Categorización de los comentarios como punto de mejora / punto positivo

En segundo lugar, se realiza el análisis de la información obtenida a través del diligenciamiento del formulario de evaluación del método propuesta (evaluación cuantitativa):

- Revisión y organización de los puntajes asignados por cada participante.
- Organización de las calificaciones obtenidas teniendo en cuenta la escala de valores y sus interpretaciones.

Reporte de resultados

En la Tabla 20. Resumen de comentarios por participantes en el focus group se encuentran registrados los comentarios aportados al momento del focus group por parte de los participantes donde aportaron sus apreciaciones respecto al método propuesto.

Tabla 20. Resumen de comentarios por participantes en el focus group.

Nombre	Comentarios
Erica Dayana Rosero	"Excelente propuesta, ojalá se pudiera implementar en todos los modelos de negocio que manejamos como QA's"
Juan Sebastián Páez	En la industria hace falta un método que asegure la fiabilidad En la empresa donde actualmente trabajo estos artefactos no son tenidos en cuenta a pesar de su importancia. La condición de asegurar un 100% el aseguramiento en cada etapa podría formar un pico de botella, ya que al no completarse este porcentaje el desarrollo se tendría que parar, y haciendo que se consuman los recursos innecesariamente.

	<p>Se sugiere tener una variable del porcentaje de aseguramiento de fiabilidad acordada con el stakeholders o la empresa.</p> <p>En la etapa de mantenimiento debería tenerse en cuenta otros roles como el stakeholders, ya que la opinión de ellos en esta etapa es de vital importancia para asegurar la fiabilidad.</p>
Omar Ricardo Bonilla	<p>El método propuesto es de gran utilidad para los QA</p> <p>Al acotar las respuestas de las preguntas del artefacto de aseguramiento con un SI o un NO podría dejar un poco cerrado el hecho de medir de una forma más precisa la fiabilidad. Se recomienda que los valores más allá del SI o un NO, se cree una escala de medición o tabla definida, ya que se pueden existir preguntas que se pueden cumplir parcial o completamente.</p> <p>En etapa de despliegue agregar más preguntas apuntando a las condiciones que se pueden presentar en un despliegue, dado que la fiabilidad se puede ver afectada por condiciones internas o externas del sistema debido a que los ambientes productivos son complejos.</p> <p>La etapa de mantenimiento está bien definida en el método dado que para realizar este tipo de cambios o mejoras al sistema se debe contemplar nuevamente todas las etapas.</p> <p>Por último, se sugiere agregar más roles que puedan aportar al aseguramiento de calidad de la fiabilidad como analistas de seguridad, entre otros.</p>
Andrés Felipe Navia Z	<p>"El artefacto es muy útil, muy útil, la verdad no se tiene uno igual y tan descrito".</p> <p>Es aconsejable que el artefacto tenga más flexibilidad de acuerdo con el porcentaje de aseguramiento.</p> <p>Para el artefacto se podría crear una guía para que el dueño del producto pueda tomar decisiones respecto a lo que debe considerar necesario de asegurar para el proyecto.</p> <p>En marcos de trabajo ágiles y con ciclo de desarrollo cortos no es muy claro ya que podría no alcanzar a responderse la totalidad de preguntas propuestos en el método.</p>
Melissa Muñoz G	<p>Contemplar cambiar el porcentaje total por uno parcial de cumplimiento de la fiabilidad o que poder escoger los aspectos a priorizar según el tipo de empresa o proyecto abordado para así no generar estancamientos.</p> <p>Tener en cuenta roles Cloud ya que este atributo de calidad es abarcado por el rol de arquitecto y en la actualidad arquitectos iCloud. Involucrar el rol de arquitecto iCloud y sus funcionalidades en el método.</p>
Karen Zúñiga	<p>En el ambiente de pruebas se puede sugerir que haya una data similar a la de producción para poder mitigar mucho más los riesgos generados en producción.</p> <p>El rol QA y analista QA son redundantes en el método propuesto.</p> <p>Empresa pequeñas o medianas no van a poder tener el rol Implementador QA, es mejor que al rol de implementador pueda monitorear con aspectos de la fiabilidad en el sistema y si falla algo, reporta y corrige. Herramientas de monitoreo orientado a calidad</p> <p>En una metodología ágil se entrega el producto por partes y puede que al usar el artefacto no va a pasar a otra etapa ya que no cumpliría el 100% de la etapa de análisis.</p> <p>Definir una estructura del método para metodologías ágiles y otra para tradicionales</p> <p>Definir mejor el rol de Ingeniero de mantenimiento, ya que los cambios lo aprueban en el área de producto (etapa de análisis), el ingeniero de mantenimiento solo se encarga de hacer los ajustes en la etapa de despliegue.</p> <p>Se sugiere definir actividades opcionales u obligatorias, esto aporta a distinguir que actividades las puede realizar pequeñas empresas, MiPymes (mediana empresa) o grandes empresas.</p>

Daniel Eduardo Paz	<p>“Las preguntas planteadas son bastantes completas, ayudan a asegurar la fiabilidad”.</p> <p>La escala respuesta para el artefacto (si/no) deben considerarse una escala más flexible, ya que una respuesta depende del contexto, los requisitos, el proyecto a desarrollar.</p> <p>Se aconseja que el artefacto de aseguramiento este acompañado de un glosario para evitar problema de ambigüedades</p> <p>En la etapa de análisis exactamente donde se habla de la capacidad de recuperación se debe hablar también de los logs, para posteriormente analizar y aumentar la capacidad de recuperación del sistema.</p>
--------------------	---

Resultados de calificaciones obtenidas

Tabla 22 se exponen las calificaciones y opiniones realizadas por los participantes al método explicado en el focus group. Así mismo, en las *Ilustración 15*, *Ilustración 16*, e *Ilustración 17* presentan las gráficas correspondientes al promedio obtenido en cada pregunta cuantitativa y en la *Ilustración 18* se presenta el promedio total de la utilidad del método.

Tabla 21. Respuestas preguntas cuantitativas.

N°	Preguntas Cuantitativas	Número de personas que marcaron la opción					Promedio
		5	4	3	2	1	
1	¿Considera conveniente este método para el aseguramiento de la calidad de la característica de la fiabilidad en los proyectos software? <i>Ilustración 15.</i>	3	4	–	–	–	4.4
2	¿Considera que estaría interesado en hacer uso del método propuesto en los proyectos software que conoce o lidera? <i>Ilustración 16.</i>	2	5	–	–	–	4.3
3	¿Considera que el método propuesto podría ser usado de manera cómoda para en los proyectos software en los que usted participa? <i>Ilustración 17.</i>	1	3	2	1	–	3.6
Promedio del grado de utilidad del método de aseguramiento de la fiabilidad							4.1

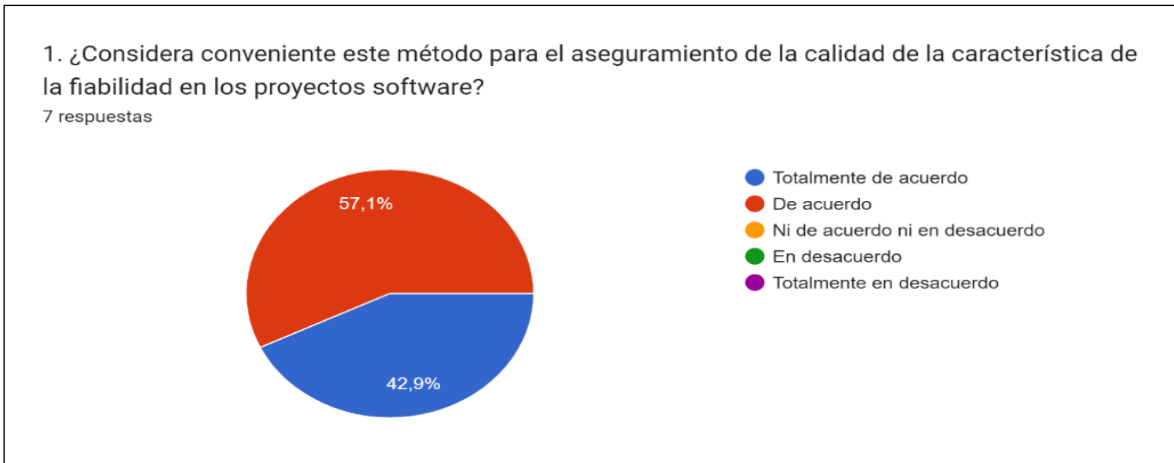


Ilustración 15. Pregunta cuantitativa N° 1.



Ilustración 16. Pregunta cuantitativa N° 2.



Ilustración 17. Pregunta cuantitativa N° 3.

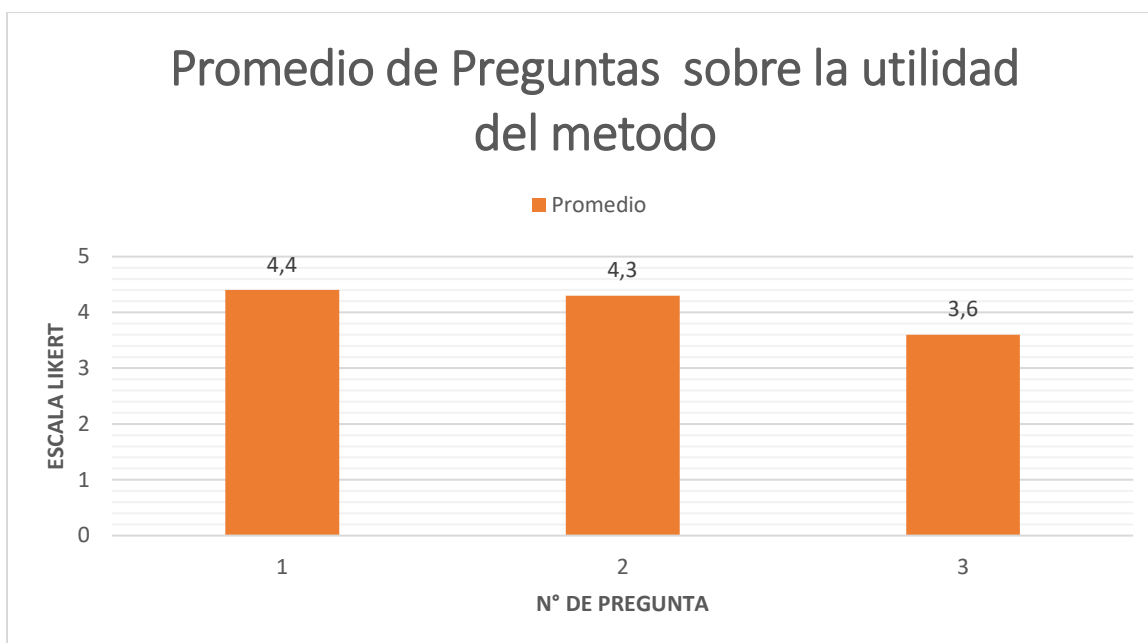


Ilustración 18. Promedio de preguntas cuantitativas.

Tabla 22. Respuestas preguntas cualitativas.

No.	Preguntas Cuantitativas	No. Participante	Respuesta
1	¿Considera conveniente este método para el aseguramiento de la calidad de la característica de la fiabilidad en los proyectos software?	1	Si
		2	Sería interesante tener una variable (Definida por el negocio o naturaleza del proyecto) la cual permita evitar cuellos de botella en el ciclo de desarrollo sw.
		3	Incluir una escala de medición
		4	Definir una escala de evaluación de ítems de cada etapa más flexible, de tal manera de que se tenga un margen de error permitido para pasar a la siguiente etapa. Tratar de involucrar una etapa de validación en preproducción.
		5	Se puede generar una propuesta más flexible en métricas para procesos ágiles
		6	El método se ve secuencial, a mi modo de ver aplica solo para metodologías tradicionales, no para ágiles.
		7	Hay varios párrafos sin justificar. Hay preguntas que tiene errores gramaticales. No queda claro que en cada actividad hay que usar un conjunto de preguntas aparte.
2	¿Considera que estaría interesado en hacer uso del método propuesto en	1	Metodologías ágiles
		2	Es muy importante dentro del ciclo de vida contar con la participación de stakeholders o usuarios que no estén directamente relacionados con el ciclo de desarrollo y

	los proyectos software que conoce o lidera?		puedan brindar una opinión de la fiabilidad desde el negocio.
		3	Incluir una escala de medición y más preguntas sobre despliegue
		4	Podrían definir en el instrumento un ítem que haga referencia al "no aplica" para preguntas que no sean del contexto del proyecto o producto a evaluar.
		5	Considerar una priorización más marcada de los sub- atributos, puesto que en entornos ágiles no se cumplirá siempre con el 100% de los requisitos
		6	Hacer pruebas en la etapa de codificación. Proponer monitoreo continuo en producción a través de herramientas.
		7	Es necesario complementar las preguntas con un glosario de términos. El glosario permitiría usar la propuesta cómodamente. Debes existir una correlación entre los requisitos de fiabilidad y las preguntas propuestas. Falta un párrafo que establezca que antes de usar la propuesta debe existir una capacitación que considere x, y, z aspectos.
		3	¿Considera que el método propuesto podría ser usado de manera cómoda para en los proyectos software en los que usted participa?
2	Actualmente, en la industria aún existen problemas de ambigüedad con los requisitos y especificaciones, por lo tanto, es importante plantear este tipo de preguntas durante el ciclo de desarrollo lo que permitiría llevar al usuario final el producto deseado.		
3	Podría reducir los costos de carencia de fiabilidad de productos software en ambientes productivos		
4	Se puede prevenir el riesgo de inyectar errores desde etapas tempranas del ciclo de vida del desarrollo.		
5	Mayor trazabilidad al cumplimiento del RNF		
6	Un software con mayor estabilidad.		
7	<p>Considero que se aumentaría la capacidad de fiabilidad en el producto software final, desde la etapa de requisitos, hasta la implementación y mantenimiento. Las preguntas son un excelente instrumento que permite determinar si se ha considerado la fiabilidad en diferentes productos de trabajo de cada actividad asociada al ciclo de desarrollo del software.</p> <p>Hay un análisis completo y holístico alrededor de los diferentes elementos a considerar en las etapas del ciclo de desarrollo de software y posteriormente existe una correlación con las preguntas para establecer si los elementos de los productos software son fiables.</p> <p>Ahora considero que debe existir una etapa previa de capacitación en la empresa, que permita utilizar su propuesta.</p>		

Categorización de la información

En la *Tabla 23* se presenta los puntos de mejora y puntos positivos aportados por los participantes en el momento del debate del focus group. Los puntos de mejora son definidos como posibles cambios o refinamientos al método propuesto y los puntos positivos se catalogan como características que se resaltan en el método por sus posibles ventajas y beneficios.

Tabla 23. Categorización de la información.

Puntos de mejora (PM)	Puntos positivos (PP)
<p>PM1 Considerar la posibilidad de tener una variable que le permita al artefacto ser más flexible, de acuerdo con el porcentaje de fiabilidad que se asigne al proyecto que se está trabajando.</p> <p>PM2 Es muy importante dentro del ciclo de vida contar con la participación de stakeholders o usuarios que no estén directamente relacionados con el ciclo de desarrollo y puedan brindar una opinión de la fiabilidad desde el negocio.</p> <p>PM3 En la etapa de despliegue se aconseja la creación de más preguntas enfocadas a condiciones propias del despliegue.</p> <p>PM4 Considerar crear una guía para que el dueño del producto para que pueda tomar decisiones respecto a lo que podría ser necesario de asegurar en el proyecto.</p> <p>PM5 Se propone agregar un ítem al artefacto que permita hacer referencia a la opción “No aplica”, para preguntas que no requiera de aseguramiento en el proyecto.</p> <p>PM6 Se aconseja crear un glosario que acompañe al artefacto de aseguramiento para evitar problemas de ambigüedad.</p> <p>PM7 Para la subcaracterística de fiabilidad relacionada con capacidad de recuperación se debe considerar agregar nuevas preguntas relacionadas con la creación de logs.</p> <p>PM8 En el ambiente de pruebas se sugiere que haya una data similar a la de producción para poder mitigar mucho más los riesgos generados en producción.</p> <p>PM9 Se sugiere que las funciones del rol de implementador QA las asuma el implementador monitoreando el sistema con aspectos de la fiabilidad apoyándose en herramientas de monitoreo orientado a calidad.</p> <p>PM10 Se recomienda definir una estructura del método para metodologías ágiles y otra para tradicionales.</p>	<p>PP1. Propuesta de gran utilidad para el aseguramiento de calidad de la característica de fiabilidad.</p> <p>PP2. El artefacto contiene preguntas completas para el aseguramiento, además este permite tener trazabilidad al cumplimiento de los requisitos no funcionales.</p> <p>PP3. En etapa de mantenimiento, tener en cuenta que el ajuste o mejora deba pasar por todas las etapas nuevamente reafirma el compromiso con la calidad.</p> <p>PP4. El artefacto de aseguramiento es útil y bien descrito.</p> <p>PP5. El método ayuda a prevenir la inyección de errores desde etapas tempranas del ciclo de vida de desarrollo software.</p> <p>PP6. El método podría ayudar a reducir los costos de carencia de fiabilidad en ambientes productivos.</p> <p>PP7. El método permite tener un producto con mayor estabilidadPP8. Hay un análisis completo y holístico.</p>

<p>PM11 Se sugiere que el rol de Ingeniero de mantenimiento sea definido de manera más clara y real posible, ya que los cambios lo aprueban en el área de producto (etapa de análisis) y no este rol.</p> <p>PM12 Se recomienda definir actividades opcionales u obligatorias, esto aporta a distinguir que actividades las puede realizar pequeñas empresas, MiPymes (mediana empresa) o grandes empresas. (no porque el método está redactado de forma general).</p> <p>PM13 Tratar de involucrar una etapa de validación en preproducción.</p> <p>PM14 Se recomienda aclarar que para cada etapa del método el conjunto de preguntas es diferentes.</p> <p>PM15 Se propone redactar un párrafo o incluir una capacitación previa en donde se indican los aspectos para tener en cuenta para la utilización del método propuesto.</p>	
--	--

4.3 Ajustes realizados con respecto a los puntos de mejora

A partir de los puntos de mejora presentados en la *Tabla 24* se define refinar el método en los siguientes aspectos:

Tabla 24. Ajustes realizados respecto a los puntos de mejora.

Punto de mejora sugerido	Respuesta frente a punto de mejora
<p>PM1 Considerar la posibilidad de tener una variable que le permita al artefacto ser más flexible, de acuerdo con el porcentaje de fiabilidad que se asigne al proyecto que se está trabajando.</p>	<p>Respuesta a PM1. Para volver flexible el artefacto, se elimina el porcentaje de cumplimiento en cada etapa y se incluye una escala de valores cualitativos (“NO CONSIDERADO”, “NO EVIDENCIADO”, “BASICO” y “ALTO”) para definir en qué grado concreto de cumplimiento se encuentra el elemento semántico evaluado en cada pregunta, si en algún caso éste elemento semántico no va a ser considerado (momentáneamente o durante todo el proyecto) se deberá usar el valor “NO CONSIDERADO”, y en caso de que el elemento semántico no haya sido asegurado (siendo considerado en el proyecto), se deberá usar el valor “NO EVIDENCIADO” de la escala que dispone el instrumento.</p>
<p>PM2 Es muy importante dentro del ciclo de vida contar con la participación de stakeholders o usuarios que no estén directamente relacionados con el ciclo de desarrollo y puedan brindar una opinión de la fiabilidad desde el negocio.</p>	<p>Respuesta a PM2. Para agregar el rol del Stakeholders al método, se lo involucra dentro de las responsabilidades del rol de analista QA, de manera que se propone que trabajen en colaboración al momento de realizar la tarea de aseguramiento de la fiabilidad en etapa de análisis.</p>
<p>PM3 En la etapa de despliegue se aconseja la creación de más preguntas enfocadas a condiciones propias del despliegue.</p>	<p>Respuesta a PM3. Este punto de mejora no se incluirá en el método, porque cada proyecto y cada organización tienen diferentes aspectos a considerar en los momentos de sus despliegues, incluyendo tecnologías. Sin embargo, el método de aseguramiento de calidad de la característica de fiabilidad es una propuesta general que puede ser adaptada.</p>

<p>PM4 Considerar crear una guía para que el dueño del producto para que pueda tomar decisiones respecto a lo que podría ser necesario de asegurar en el proyecto.</p>	<p>Respuesta a PM4. Este punto de mejora no se incluirá, ya que el método ya tiene una dinámica de uso que permite a través de la escala de valores lograr una flexibilidad al momento de considerar los aspectos referentes a la fiabilidad, incluyendo un valor “NO SERÁ CONSIDERADO”</p>
<p>PM5 Se propone agregar un ítem al artefacto que permita hacer referencia a la opción “No aplica”, para preguntas que no requiera de aseguramiento en el proyecto.</p>	<p>Respuesta a PM5. Este punto si es tenido en cuenta para el refinamiento del método; en la escala de valores se incluye un valor “NO CONSIDERADO” que significa que el elemento semántico respectivo quedará por fuera del proceso de aseguramiento según determinación del proyecto. El “NO APLICA” podría entenderse como una rotunda exclusión del elemento semántico por parte del método propuesto, lo que anularía además su relevancia dentro de la definición de la característica de fiabilidad de un producto software.</p>
<p>PM6 Se aconseja crear un glosario que acompañe al artefacto de aseguramiento para evitar problemas de ambigüedad .</p>	<p>Respuesta a PM6. Este punto es tenido en cuenta ampliando el significado de algunas palabras en las preguntas del artefacto de aseguramiento para tener más claridad al momento de responder por parte de los roles del método.</p>
<p>PM7 Para la subcaracterística de fiabilidad relacionada con capacidad de recuperación se debe considerar agregar nuevas preguntas relacionadas con la creación de logs.</p>	<p>Respuesta a PM7. En la subcaracterística de recuperación a fallos se amplía la pregunta especificando el uso de los logs en los mecanismos.</p>
<p>PM8 En el ambiente de pruebas se sugiere que haya una data similar a la de producción para poder mitigar mucho más los riesgos generados en producción.</p>	<p>Respuesta a PM8. Este punto no es tenido en cuenta ya que son decisiones que son tomadas por las propias empresas y que hacen parte de la configuración del ambiente de pruebas específicamente.</p>
<p>PM9 Se sugiere que las funciones del rol de implementador QA las asuma el implementador monitoreando el sistema con aspectos de la fiabilidad apoyándose en herramientas de monitoreo orientado a calidad.</p>	<p>Respuesta a PM9. Este punto de mejora no es tenido en cuenta ya que el rol de implementador QA está definido en el método para que se pueda realizar un efectivo aseguramiento de la calidad bajo la perspectiva del aseguramiento cruzado.</p>
<p>PM10 Se recomienda definir una estructura del método para metodologías ágiles y otra para tradicionales</p>	<p>Respuesta a PM10. Se mantendrá una sola estructura del método con ajustes referentes a la flexibilidad (escala de valores de las repuestas, eliminación de condición del 100%) para ser usado en metodologías ágiles o tradicionales.</p>
<p>PM11 Se sugiere que el rol de Ingeniero de mantenimiento sea definido de manera más clara y real posible, ya que los cambios lo aprueban en el área de producto (etapa de análisis) y no este rol.</p>	<p>Respuesta a PM11. De acuerdo con los procedimientos y los roles existentes en cada empresa u organización, cuando se haya aprobado un cambio o mejora, el rol que realiza dicho cambio es el Ingeniero de Mantenimiento, por tal motivo se sigue manteniendo este rol así definido en el método.</p>
<p>PM12 Se recomienda definir actividades opcionales u obligatorias, esto aporta a distinguir que actividades las puede realizar pequeñas empresas, MiPymes (mediana empresa) o grandes empresas. (no porque el método está redactado de forma general).</p>	<p>Respuesta a PM12. No se tiene en cuenta este punto de mejora, ya que el método tiene actividades sugeridas a nivel estándar y que según el criterio de cada empresa se optará por cumplirlas o no, sin tener en cuenta el tamaño de la organización o empresa.</p>

<p>PM13 Tratar de involucrar una etapa de validación en preproducción.</p>	<p>Respuesta a PM13. No se tiene en cuenta este punto de mejora ya que la etapa de preproducción no está creada en todas las empresas, esta etapa es incluida en algunos negocios que ven la necesidad de incluirlo, así que no haría parte del ciclo de vida de desarrollo general al que se está apuntando.</p>
<p>PM14 Se recomienda aclarar que para cada etapa del método el conjunto de preguntas es diferentes.</p>	<p>Respuesta a PM14. Este punto de mejora será tenido en cuenta en el momento de explicar el artefacto de aseguramiento para más claridad del método propuesto.</p>
<p>PM15 Se propone redactar un párrafo o incluir una capacitación previa en donde se indican los aspectos para tener en cuenta para la utilización del método propuesto.</p>	<p>Respuesta a PM15. Los aspectos para tener en cuenta están definidos por los elementos semánticos y si se decide realizar una capacitación al respecto o realizar un análisis previo a la utilización del método, será decisión de la empresa u organización</p>

4.4 Análisis de validez y limitaciones

Dentro del plan de validez, se diseñaron los siguientes mecanismos para este fin:

- **Validez de constructo:** Previo al focus group se envió un documento resumen que contenía el método propuesto con el fin de contextualizar a los participantes antes de la sesión. Durante la ejecución del focus group se usó un instrumento que contenía criterios de utilidad del método en procesos de desarrollo de software lo que permitió recolectar información muy concreta en términos cuantitativos y como complemento, unos temas cualitativos (preguntas abiertas). Posterior a la sesión del focus se realizó un refinamiento de los puntos de mejora apoyado en la información obtenida en el debate; este refinamiento incluyó asegurar la tarea de verificar que la información recolectada fuera idónea, por tanto, se hizo una extracción de las ideas que aportarían un mejoramiento del método propuesto, cumpliendo con su propósito. A través de la verificación de la información aportada finalmente permite tener una traza desde la pregunta de investigación, datos recolectados y el análisis de la información para poder lograr una adecuada evaluación del método.
- **Validez interna:** A través de las preguntas (ver parte 3.3) que fueron definidas a partir de unos elementos semánticos que permitieron caracterizar de forma explícita el concepto o la definición de la fiabilidad, se podría conseguir el cubrimiento del aseguramiento de calidad de esta característica en las diferentes etapas del ciclo de desarrollo de software.

- **Validez externa:** Frente a la validación ante expertos se utilizó el único focus group realizado y a partir de éste, se obtuvieron los puntos de mejora que fueron refinados, teniendo en cuenta que dentro de la metodología de investigación multi-ciclo con bifurcación existe un punto de refinamiento.
- **Fiabilidad:** El grado de fiabilidad se determina a través de la experiencia de los participantes en la sesión del focus group en procesos relacionados al aseguramiento de la calidad. Se presume que los resultados obtenidos a través del focus group podrían ser similares al ser aplicados para otros participantes con perfiles similares.

La limitación identificada en el focus group fue:

- Frente a la experiencia de los participantes, debido a que no fue fácil conseguir un perfil de asegurador de calidad en las organizaciones debido a que este perfil no es conocido de manera explícita en la industria, ya que el perfil es muy estricto tanto en términos de años de experiencia y dominio del área del aseguramiento en las organizaciones para hacerlo participe en el focus group. Está limitante se mitigó buscando participantes que tuvieran perfiles que estuvieran cerca de actividades relacionadas al aseguramiento como pruebas funcionales o automatizadas y tareas del ciclo de desarrollo de software.

4.5 Conclusiones de la evaluación preliminar a través del Focus group

De acuerdo con los resultados obtenidos a través de las mediciones cualitativas y cuantitativas se concluye que el método de aseguramiento de calidad de la característica de fiabilidad durante el ciclo de desarrollo software es útil para el propósito para lo que fue definido específicamente porque el promedio de las características de conveniencia, interés y comodidad al momento de usarlo en los proyectos de desarrollo de software se indicó un valor de 4.1 lo que significa que los expertos están de acuerdo con la utilidad del método propuesto.

Adicionalmente las preguntas abiertas permitieron corroborar que la propuesta es interesante y posee diversos beneficios si se implementa en la industria; Mejora la

estabilidad y la trazabilidad del software, beneficia la prevención de riesgos y disminuye la ambigüedad existente en los requisitos y especificaciones con los que se trabaja durante el ciclo de desarrollo. Sin embargo, se requiere aplicar ajustes en aspectos de flexibilidad como lo son las márgenes de tolerancia y ajustes de entendimiento al aplicarse en metodologías ágiles. También se visualizan nuevos aspectos a incluir, como la priorización de subcaracterísticas y la consideración de nuevos roles.

Finalmente, luego del proceso de evaluación preliminar y partiendo de la retroalimentación obtenida de cada uno de los participantes se logra identificar que es muy valiosos el uso de técnicas adecuadas (como el focus group) puesto que permiten un enriquecimiento de nuevas ideas, las cuales pueden ser analizadas y categorizadas, posibilitando diversos trabajos futuros y mejoras al estado actual del método.

Del mismo modo, se pudo obtener como resultado del focus group que:

- Ninguna de las empresas conocidas o donde se ha realizado una labor por parte de los encuestados utilizan o conocen artefactos que posibiliten el aseguramiento de la característica de fiabilidad en sus proyectos.
- Los encuestados coinciden en que el método propuesto sería de gran utilidad si es incorporado a la industria.
- El método permite asegurar la fiabilidad y ser trazable durante todo el ciclo de vida de desarrollo.
- Los roles definidos cumplen con el objetivo del método propuesto.
- Los aspectos pueden ser desconocidos para algunos trabajadores (roles del ciclo) pero pueden ser aclarados antes de aplicar el método.
- La participación de los stakeholders dentro del método es de gran relevancia ya que ellos brindan una opinión de fiabilidad desde el punto de vista del negocio
- El método se ajusta a cualquier tipo de metodología ya que sus actividades y tareas son claras y no se basan en ninguna en especial que haga que se excluyan las demás.

Capítulo 5 - Conclusiones y trabajos futuros

De acuerdo a la metodología de investigación definida para este trabajo de grado, en el estado del arte se pudo evidenciar que las etapas del ciclo de desarrollo de software como diseño, codificación y mantenimiento en términos de aseguramiento de calidad son las menos cubiertas, ya que según la literatura abordada las técnicas de aseguramiento son enfocadas en su mayoría a la etapa de pruebas, y el cubrimiento disminuye aún más cuando se aborda este aseguramiento de calidad de requisitos no funcionales como lo es específicamente la característica de la fiabilidad. Del mismo modo, el estado del arte arrojó que la gran mayoría de técnicas de aseguramiento de calidad son enfocadas en la etapa de pruebas.

Posteriormente, y considerando los elementos relevantes de la literatura frente al núcleo de aseguramiento de la fiabilidad que abarca sus atributos/elementos semánticos, las preguntas que apoyan este aseguramiento y la técnicas, se planteó un método de aseguramiento de calidad que permite tener un cubrimiento de la característica de fiabilidad en cada una de la etapa del ciclo de desarrollo software, al mismo tiempo este método proporciona un artefacto de aseguramiento, el cual se convierte en un instrumento fundamental para la aplicación del método en escenarios productivos. Por último, se logró, luego del análisis de expertos, a través de un focus group, que el método fuera analizado frente a su utilidad en proyectos de desarrollo de software, obteniendo una aprobación importante desde el punto de vista de sus expertos. Algunas de las mejoras propuestas por los expertos que evaluaron la versión presentada fueron incorporadas en la versión final del presente trabajo de grado y otras se plantean como trabajo futuro, a saber: la flexibilización de las escalas de respuesta en el artefacto logra ser implementada para que se adapte a diversos porcentajes de fiabilidad planificados por las empresas hacia sus proyectos, se refinaron las preguntas planteadas en el artefacto buscando mejorar la facilidad de entendimiento en cada una de ellas y se adicionaron roles que permitirán abarcar con mejor precisión los involucrados que influyen en cada uno de los proyectos software.

Además, se puede concluir que el método podría ser de interés para las empresas del sector software, puesto que ayuda a mejorar la estabilidad y la trazabilidad en los proyectos, beneficia la prevención de riesgos y disminuye la ambigüedad existente en los requisitos y especificaciones con los que se trabaja durante el ciclo de desarrollo. Este método se convierte en una base para futuras implementaciones que permitan el aseguramiento de la

característica de fiabilidad de productos software en metodologías o proyectos futuros. De este modo, el método apunta a convertirse en un producto software con potencial en el mercado de aseguramiento de la calidad de software, puesto que queda abierto para futuras mejoras e implementaciones que perfeccionaran su adopción en la industria, abriendo paso a la diversificación de estrategias de uso, disponibilidad y gestión de los datos, como lo pueden ser: la trazabilidad, la disponibilidad, la recolección y el análisis de los datos relacionados con los aspectos de fiabilidad de los productos software para diferentes proyectos de desarrollo. Lograr esta construcción como herramienta tecnológica hace parte del trabajo futuro de ésta investigación.

Referencias Bibliográficas

- [1] A. Kalsoom, M. Maqsood, M. A. Ghazanfar, F. Aadil, y S. Rho, *A dimensionality reduction-based efficient software fault prediction using Fisher linear discriminant analysis (FLDA)*, vol. 74, n.º 9. Springer US, 2018.
- [2] ISO, «Portal ISO 25000», *ISO 25000 CALIDAD DE SOFTWARE Y DATOS*, 2005. <https://iso25000.com/> (accedido sep. 11, 2021).
- [3] S. L. Buitrón, «FRAMEWORK PARA LA ELICITACIÓN DE REQUISITOS NO FUNCIONALES USANDO SU REPRESENTACIÓN», 2021.
- [4] International Organization for Standardization, «ISO 25010», *iso25000.com*. 2011, Accedido: sep. 11, 2021. [En línea]. Disponible en: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>.
- [5] M. Ahmadi, B. B. Rad, y M. O. Thomas, «Tailoring software development methodologies for reliability», *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, n.º 3, pp. 117-121, 2018.
- [6] T. Choeikiwong y P. Vateekul, «Two stage model to detect and rank software defects on imbalanced and scarcity data sets», *IAENG International Journal of Computer Science*, vol. 43, n.º 3, pp. 344-355, 2016.
- [7] A. K. A. Takanen, JD Demott, C Miller, *Fuzzing for software security testing and quality assurance*, vol. 4, n.º 3. 2018.
- [8] H. N. Aleem, M. M. Baig, y M. M. Khan, «Efficient software testing technique based on hybrid database approach», *International Journal of Advanced Computer Science and Applications*, vol. 10, n.º 7, pp. 349-356, 2019, doi: 10.14569/ijacsa.2019.0100748.
- [9] A. Aleti y L. Grunske, «Test data generation with a Kalman filter-based adaptive genetic algorithm», *Journal of Systems and Software*, vol. 103, n.º 2015, pp. 343-352, 2015, doi: 10.1016/j.jss.2014.11.035.
- [10] L. E. García Reyes, «Software reliability assessment using machine learning

technique», *Journal of Chemical Information and Modeling*, vol. 53, n.º 9, pp. 1689-1699, 2013.

- [11] S. Reddivari y J. Raman, «Software quality prediction: An investigation based on machine learning», *Proceedings - 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science, IRI 2019*, pp. 115-122, 2019, doi: 10.1109/IRI.2019.00030.
- [12] D. E. Harter, C. F. Kemerer, y S. A. Slaughter, «Does software process improvement reduce the severity of defects? A longitudinal field study», *IEEE Transactions on Software Engineering*, vol. 38, n.º 4, pp. 810-827, 2012, doi: 10.1109/TSE.2011.63.
- [13] B. Lewis, I. Smith, M. Fowler, y J. Licato, «Detecting problems in the database access code of large scale systems - an industrial experience report», *28th Modern Artificial Intelligence and Cognitive Science Conference, MAICS 2017*, n.º May 2016, pp. 189-190, 2017, doi: 10.1145/1235.
- [14] B. Mpofo y E. Mnkandla, «Software defect prediction using process metrics elasticsearch engine case study», *Proceedings - 2016 3rd International Conference on Advances in Computing, Communication and Engineering, ICACCE 2016*, pp. 254-259, 2017, doi: 10.1109/ICACCE.2016.8073757.
- [15] N. Srinivasan, «A study on humanizing software test effort and quality», *Journal of Theoretical and Applied Information Technology*, vol. 72, n.º 1, pp. 133-141, 2015.
- [16] A. Hussain, A. Nadeem, y M. T. Ikram, «Review on formalizing use cases and scenarios: Scenario based testing», *Proceedings of 2015 International Conference on Emerging Technologies, ICET 2015*, 2016, doi: 10.1109/ICET.2015.7389203.
- [17] A. Ibias, M. Núñez, y R. M. Hierons, «Using mutual information to test from Finite State Machines: Test suite selection», *Information and Software Technology*, vol. 132, n.º December, p. 106498, 2021, doi: 10.1016/j.infsof.2020.106498.
- [18] C. C. Yeh, H. L. Lu, C. Y. Chen, K. K. Khor, y S. K. Huang, «CRAXDroid: Automatic android system testing by selective symbolic execution», *Proceedings - 8th International Conference on Software Security and Reliability - Companion, SERE-C 2014*, pp. 140-148, 2014, doi: 10.1109/SERE-C.2014.32.
- [19] V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss, y P. Tonella, «Testing

- machine learning based systems: a systematic mapping», *Empirical Software Engineering*, vol. 25, n.º 6, pp. 5193-5254, 2020, doi: 10.1007/s10664-020-09881-0.
- [20] F. H. Alshammari, «Software Defect Prediction and Analysis Using Enhanced Random Forest (extRF) Technique: A Business Process Management and Improvement Concept in IOT-Based Application Processing Environment», *Mobile Information Systems*, vol. 2022, 2022, doi: 10.1155/2022/2522202.
- [21] Z. Eivazpour y M. R. Keyvanpour, *Adversarial Samples for Improving Performance of Software Defect Prediction Models*, vol. 45. 2020.
- [22] J. Xu, J. Tang, K. Kwiat, W. Zhang, y G. Xue, «Survivable virtual infrastructure mapping in virtualized data centers», *Proceedings - 2012 IEEE 5th International Conference on Cloud Computing, CLOUD 2012*, pp. 196-203, 2012, doi: 10.1109/CLOUD.2012.100.
- [23] A. Zhou *et al.*, «Cloud service reliability enhancement via virtual machine placement optimization», *IEEE Transactions on Services Computing*, vol. 10, n.º 6, pp. 902-913, 2017, doi: 10.1109/TSC.2016.2519898.
- [24] P. Dhiman, Manish, y R. Chawla, «A clustered approach to analyze the software quality using software defects», *Proceedings - 2012 2nd International Conference on Advanced Computing and Communication Technologies, ACCT 2012*, pp. 36-40, 2012, doi: 10.1109/ACCT.2012.1.
- [25] S. Henry, «Object-oriented metrics that predict maintainability», *Journal of Systems and Software*, vol. 23, n.º 2, pp. 111-122.
- [26] F. J. Pino, M. Piattini, y G. H. Travassos, «Managing and developing distributed research projects in software engineering by means of action-research», *Revista Facultad de Ingeniería*, n.º 68, pp. 61-74, 2013.
- [27] R. Of *et al.*, «Draft International Standard Iso / Fdis», vol. 2009, 2009.
- [28] O. I. de N. ISO, «ISO 9001:2015(es), Sistemas de gestión de la calidad — Requisitos», 2015. <https://www.iso.org/obp/ui/es/#iso:std:iso:9001:ed-5:v1:es> (accedido sep. 11, 2021).
- [29] Software Engineering Institute, «CMMI for Development, Version 1.3», *Software*

Engineering Process Management Program, n.º November, pp. 1-520, 2010.

- [30] «IEEE SA - IEEE 1028-2008». <https://standards.ieee.org/ieee/1028/4402/> (accedido oct. 24, 2022).
- [31] O. Kaiwartya *et al.*, «Guidelines for performing Systematic Literature reviews in Software Engineering», *IEEE Access*, vol. 4, pp. 5356-5373, 2016, doi: 10.1109/ACCESS.2016.2603219.
- [32] «mecanismo | Definición | Diccionario de la lengua española | RAE - ASALE». <https://dle.rae.es/mecanismo> (accedido ene. 04, 2023).
- [33] «técnico, técnica | Definición | Diccionario de la lengua española | RAE - ASALE». <https://dle.rae.es/técnico> (accedido ene. 04, 2023).
- [34] M. Ahmed, R. Ibrahim, y N. Ibrahim, «An adaptation model for android application testing with refactoring», *International Journal of Software Engineering and its Applications*, vol. 9, n.º 10, pp. 65-74, 2015, doi: 10.14257/ijseia.2015.9.10.07.
- [35] J. Yang, L. Tan, J. Peyton, y K. A Duer, «Towards Better Utilizing Static Application Security Testing», *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, pp. 51-60, 2019, doi: 10.1109/ICSE-SEIP.2019.00014.
- [36] Z. Shahbazi y Y. C. Byun, «Integration of blockchain, iot and machine learning for multistage quality control and enhancing security in smart manufacturing», *Sensors*, vol. 21, n.º 4, pp. 1-21, 2021, doi: 10.3390/s21041467.
- [37] P. S. Sabnis y S. D. Joshi, «A Literature Survey on Optimization and Validation of Software Reliability Using Machine Learning», *International Journal of Computing and Digital Systems*, vol. 11, n.º 1, pp. 841-849, 2021, doi: 10.12785/IJCDS/110169.
- [38] «método | Diccionario esencial de la lengua española | RAE - ASALE». <https://www.rae.es/desen/método> (accedido dic. 02, 2022).
- [39] «The C4 model for visualising software architecture». <https://c4model.com/> (accedido oct. 26, 2022).
- [40] P. De y E.-N. Básico, «Probador Certificado del ISTQB ®», 2018.

- [41] C. Y. Laporte y A. April, «Software Audits», *Software Quality Assurance*, pp. 210-248, 2017, doi: 10.1002/9781119312451.ch6.
- [42] A. Jiménez y J. Díaz, «Revisión sistemática de literatura: Técnicas de aprendizaje automático (Machine Learning)», *Cuaderno activa*, vol. 13, n.º 1, pp. 113-121, 2021.
- [43] P. Daniel y M. Echenique, «Plataforma de Pruebas de Conformidad LoRaWAN Tesis», 2020.
- [44] B. P. Miller *et al.*, «Fuzz Revisited - A re-examination of the reliability of UNIX utilities and services», *October*, vol. 1525, n.º October 1995, pp. 1-23, 1995, [En línea]. Disponible en: <http://www.eecs.northwestern.edu/~robby/courses/395-495-2009-fall/fuzz-revisited.pdf>.
- [45] C. L. Vidal, R. F. Schmal, S. Rivero, y R. H. Villarroel, «Una revisión sobre la ejecución simbólica de programas computacionales», *Informacion Tecnologica*, vol. 25, n.º 3, pp. 115-124, 2014, doi: 10.4067/S0718-07642014000300014.
- [46] I. E. Allen y C. A. Seaman, «Likert Scales and Data Analyses», Accedido: oct. 23, 2022. [En línea]. Disponible en: www.sloan-c.org.
- [47] J. Kontio, L. Lehtola, y J. Bragge, «Using the focus group method in software engineering: Obtaining practitioner and user experiences», *Proceedings - 2004 International Symposium on Empirical Software Engineering, ISESE 2004*. pp. 271-280, 2004, doi: 10.1109/ISESE.2004.1334914.
- [48] M. Mendoza-Moreno, C. González-Serrano, y F. J. Pino, «Focus group como proceso en Ingeniería de Software: Una experiencia desde la práctica», *DYNA (Colombia)*, vol. 80, n.º 181, pp. 51-60, 2013.
- [49] «utilidad | Definición | Diccionario de la lengua española | RAE - ASALE». <https://dle.rae.es/utilidad> (accedido oct. 23, 2022).