

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Software

Diseño de una guía metodológica de calidad de software para las fases de ejecución de pruebas y puesta en producción de software de la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte.

Trabajo de grado previo a la obtención del título de Ingeniero de Software presentado ante la ilustre Universidad Técnica del Norte.

Autor:

Carvajal Tito Yamilex Elizabeth

Director:

MSc. Caraguay Procel Jorge Adrián. Ing.

Ibarra – Ecuador

2024



UNIVERSIDAD TÉCNICA DEL NORTE

DIRECCIÓN DE BIBLIOTECA

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004121818		
APELLIDOS Y NOMBRES:	Carvajal Tito Yamilex Elizabeth		
DIRECCIÓN:	Pugacho bajo, carrera 27 de febrero y pasaje 09 de octubre		
EMAIL:	yecarvajalt@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0979477620

DATOS DE LA OBRA	
TÍTULO:	Diseño de una guía metodológica de calidad de software para las fases de ejecución de pruebas y puesta en producción de software de la dirección de desarrollo tecnológico e informático de la universidad técnica del norte
AUTOR (ES):	Carvajal Tito Yamilex Elizabeth
FECHA: DD/MM/AAAA	05/12/2024
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> GRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	Ingeniero en software
ASESOR /DIRECTOR:	Director: Ing. Jorge Caraguay, MSc. Asesora: PhD. Cathy Guevara

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 05 días del mes de diciembre de 2024

EL AUTOR:


Srta. Yamilex Carvajal
1004121818

CERTIFICACIÓN DIRECTOR

Ibarra 05 de diciembre del 2024

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE TITULACIÓN

Por medio del presente yo MSc. Jorge Caraguay, certifico que la Srta. Yamilex Elizabeth Carvajal Tito portador de la cédula de ciudadanía número 1004121818, ha trabajado en el desarrollo del proyecto de grado **“Diseño de una guía metodológica de calidad de software para las fases de ejecución de pruebas y puesta en producción de software de la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte”**, previo a la obtención del Título de Ingeniero en Software realizado con interés profesional y responsabilidad que certifico con honor de verdad.

Es todo en cuanto puedo certificar a la verdad

Atentamente

Ing. Jorge Caraguay Procel, MSc.
DIRECTOR DE TRABAJO DE GRADO

DEDICATORIA

El presente trabajo se lo dedico a mi hermana Myrian Carvajal, quien desde mi infancia ha sido mi guía y apoyo incondicional y ha sembrado en mi responsabilidad, perseverancia, constancia y deseos de superación, siempre me ha enseñado que los problemas son solo oportunidades para crecer y que, ante cualquier obstáculo, la solución está en no rendirse. Sus consejos han sido fundamentales en cada paso de mi vida, este logro no habría sido posible sin ellos.

Gracias Dios por concederme la mejor de las hermanas.

Yamilex Elizabeth Carvajal Tito

AGRADECIMIENTO

Agradezco a mi madre por brindarme la oportunidad de acceder al estudio. Por su esfuerzo y sacrificio constante, por sus consejos y darme el impulso para seguir adelante.

A mi novio, Abg. Martín Paucar, quien ha estado a mi lado en los momentos y situaciones más complicadas, siempre apoyándome y motivándome con las palabras adecuadas.

A mis amigos Leslie, Francisco, Stalin, Stiphen, por su amistad, por cada uno de los momentos que hemos compartido, por haber hecho de este camino por la vida universitaria más fácil y por brindarme su apoyo para seguir estos años.

Al docente MSc. Mauricio Rea, por todo el conocimiento adquirido en sus asignaturas, y por su valiosa amistad.

Al docente MSc. Jorge Caraguay por brindarme la accesibilidad al desarrollo de este trabajo de investigación, por su paciencia y conocimientos durante todo el proceso de titulación.

Muchas gracias a todos,

Yamilex

TABLA DE CONTENIDOS

DEDICATORIA	V
AGRADECIMIENTO	VI
TABLA DE CONTENIDOS	VII
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XI
RESUMEN	XIII
ABSTRACT	XIV
INTRODUCCIÓN	2
Antecedentes	2
Planteamiento del problema	3
OBJETIVOS	5
Objetivo General	5
Objetivos Específicos	5
ALCANCE	5
METODOLOGÍA	7
JUSTIFICACIÓN	8
CAPÍTULO 1	10
Marco Teórico	10
1.1. Fundamentos para elaborar una guía metodológica	10
1.1.1. Elementos de una guía o manual	11
1.2. Calidad de software.....	11
1.2.1. Gestión de calidad.....	12
1.2.2. Plan de calidad	13
1.2.3. Aseguramiento de calidad de software SQA.....	14
1.2.4. Control de calidad de software QC.....	15
1.2.5. Costo de la calidad de software.....	16
1.2.6. Métricas de calidad de software	16
1.3. Pruebas de software	18
1.3.1. Principios de las pruebas.....	18
1.4. ISO/IEC/IEEE 29119 Software Testing	19
1.4.1. ISO/IEC/IEEE 29119 – 1: Conceptos y definiciones.....	20
1.4.2. ISO/IEC/IEEE 29119 – 2: Procesos de prueba	21
1.4.3. ISO/IEC/IEEE 29119 – 3: Documentación de pruebas.....	36
1.4.4. ISO/IEC/IEEE 29119 – 4: Técnicas de pruebas.....	37

1.4.5.	ISO/IEC/IEEE 29119 – 5: Pruebas dirigidas por palabras clave	44
1.5.	Prácticas recomendadas por la ISO/IEC/IEEE 29119.....	45
1.6.	Puesta en producción.....	46
1.7.	ISO/IEC/IEEE 20000-1:2018 Tecnología de la información - Gestión de servicios	46
1.7.1.	Gestión de la configuración.....	46
1.7.2.	Gestión de control de cambios	48
1.7.3.	Gestión de entrega y despliegue.....	49
CAPÍTULO 2	52
Diseño de la guía	52
2.1.	Descripción del área de estudio	52
2.2.	Diseño y tipo de investigación.....	55
2.2.1.	Metodología de investigación	56
2.3.	Desarrollo de la investigación	57
2.3.1.	Área de desarrollo del DDTI	57
2.3.2.	Instrumentos	57
2.3.3.	Definición de variables	60
2.3.4.	Resultados de la encuesta realizada al área de desarrollo.....	62
2.3.5.	Resultados de la Entrevista realizada al subdirector del DDTI	70
2.3.6.	Análisis y revisión documental	80
2.4.	Diseño de la guía metodológica	82
Introducción	83
2.4.1.	Propósito de la guía.....	83
2.4.2.	Alcance general	84
2.4.3.	Definiciones y términos clave.....	84
2.4.4.	Roles generales.....	86
Fase de pruebas	86
2.4.5.	Planificación de pruebas	86
2.4.6.	Diseño de pruebas.....	90
2.4.7.	Ejecución de pruebas.....	93
2.4.8.	Seguimiento y control de pruebas	93
2.4.9.	Documentación.....	96
2.4.10.	Evaluación de resultados de las pruebas	96
Fase de puesta en producción	99
2.4.11.	Planificación de despliegue	99
2.4.12.	Control de cambios	101
2.4.13.	Procedimientos de despliegue	102

2.4.14. Cierre de despliegue	103
CAPÍTULO 3	104
Propuesta operativa	104
3.1. Planificación de pruebas	104
3.2. Diseño de pruebas.....	110
3.3. Ejecución de pruebas.....	111
3.4. Seguimiento y control de pruebas	113
3.5. Planificación de despliegue	116
3.6. Control de cambios	118
3.7. Procedimientos de despliegue.....	119
3.8. Cierre de despliegue.....	120
CAPÍTULO 4	123
Validación	123
4.1. Prueba de concepto	123
4.2. Revisión por pares.....	123
4.3. Resultados de la prueba de concepto.....	125
4.3.1. Análisis de resultados	125
CONCLUSIONES	136
RECOMENDACIONES	139
BIBLIOGRAFÍA	141
ANEXOS	144

ÍNDICE DE FIGURAS

Figura 1 Diagrama de Ishikawa.....	4
Figura 2 Diagrama Estructural	6
Figura 3 Diagrama Metodológico	7
Figura 4 Estructura de ISO/IEC/IEEE 29119	20
Figura 5 ISO/IEC/IEEE 29119 Parte 1: conceptos y vocabulario	21
Figura 6 ISO/IEC/IEEE 29119 Parte 2: procesos de prueba	22
Figura 7 Modelo de proceso de pruebas de tres capas ISO/IEC/IEEE 29119	23
Figura 8 Proceso de prueba organizacional	24
Figura 9 Estrategia de prueba y proceso de planificación.....	25
Figura 10 Proceso de seguimiento y control de pruebas.....	28
Figura 11 Proceso de finalización de pruebas	30
Figura 12 Proceso de diseño e implementación de pruebas	31
Figura 13 Entorno de prueba y proceso de gestión de datos	33
Figura 14 Proceso de ejecución de pruebas	35
Figura 15 Proceso de notificación de incidentes de prueba.....	35
Figura 16 ISO/IEC/IEEE 29119 Parte 3: Documentación	37
Figura 17 ISO/IEC/IEEE 29119 Parte 4: Técnicas de prueba	43
Figura 18 ISO/IEC/IEEE 29119 Parte 5: Pruebas dirigidas por palabras clave.....	44
Figura 19 Dirección de Desarrollo Tecnológico e Informático.....	55
Figura 20 Gráfico de barras apiladas – Estructura Organizativa.....	62
Figura 21 Gráfico de barras apiladas – Calidad de Software.....	63
Figura 22 Gráfico de barras apiladas – Metodologías y estándares de desarrollo.....	65
Figura 23 Gráfico de barras apiladas – Pruebas de Software.	66
Figura 24 Gráfico de barras apiladas – Puesta en producción.	69
Figura 25 Gráfico de barras apiladas - Aspectos generales.....	126
Figura 26 Gráfico de barras apiladas - Planificación y roles de pruebas.....	127
Figura 27 Gráfico de barras apiladas - Estrategia y ejecución de pruebas.....	129
Figura 28 Gráfico de barras apiladas - Evaluación de resultados de las pruebas.....	130
Figura 29 Gráfico de Barras Apiladas - Puesta en producción	131
Figura 30 Gráfico de Barras Apiladas - Satisfacción general	133

ÍNDICE DE TABLAS

Tabla 1 Búsqueda bibliográfica	10
Tabla 2 Elementos de una guía o manual	11
Tabla 3 Principios de gestión de calidad	13
Tabla 4 QA Constructivo	14
Tabla 5 QA Analítico.....	14
Tabla 6 Control de calidad mejorado	15
Tabla 7 Tipos de costos de calidad de software	16
Tabla 8 Principios de las pruebas	18
Tabla 9 Tareas del proceso de prueba organizacional.....	24
Tabla 10 Tareas del proceso de estrategia de prueba y proceso de planificación	26
Tabla 11 Tareas del proceso de seguimiento y control.....	28
Tabla 12 Tareas del proceso de finalización de pruebas.....	30
Tabla 13 Tareas del proceso de proceso de diseño e implementación de pruebas	31
Tabla 14 Tareas del proceso de entorno de prueba y proceso de gestión de datos	33
Tabla 15 Tareas del proceso de ejecución de prueba	35
Tabla 16 Tareas del proceso de notificación de incidentes de prueba	36
Tabla 17 Documentación de pruebas	36
Tabla 18 Técnicas de Pruebas Basadas en Especificaciones.....	38
Tabla 19 Técnicas de Pruebas Basadas en Estructuras	40
Tabla 20 Técnicas de Pruebas basadas en Experiencia	43
Tabla 21 Prácticas recomendadas para cumplir con la ISO 29119	45
Tabla 22 FODA DDTI.....	53
Tabla 23 Personal del DDTI.....	57
Tabla 24 Variable independiente: Guía Metodológica de Calidad de Software	60
Tabla 25 Variable dependiente: Calidad del Software Desarrollado por el DDTI.....	61
Tabla 26 Roles y responsabilidades para la fase de pruebas.	86
Tabla 27 Roles y responsabilidades para la fase de puesta en producción.	86
Tabla 28 Prioridad de pruebas	92
Tabla 29 Identificación de documentación - plan de pruebas	104
Tabla 30 Formato para listar los requisitos funcionales	105
Tabla 31 Partes interesadas	106
Tabla 32 Comunicación de las pruebas	106
Tabla 33 Registro de riesgos	106
Tabla 34 Criterios de finalización	107
Tabla 35 Métricas	107
Tabla 36 Ambiente de pruebas.....	108
Tabla 37 Herramientas	108
Tabla 38 Proceso de re-testing y regresión	108
Tabla 39 Criterios de suspensión y reanudación.....	108
Tabla 40 Actividades de prueba	109
Tabla 41 Cronograma.....	109
Tabla 42 Identificación de documento - Diseño de casos de prueba	110
Tabla 43 Criterios de suposiciones y limitaciones - casos de prueba	110
Tabla 44 Formato para listar los casos de prueba	111
Tabla 45 Identificación de documento - ejecución de pruebas.....	111
Tabla 46 Criterios de suposiciones y limitaciones - ejecución de pruebas.....	112
Tabla 47 Registro de ejecución de pruebas.....	113

Tabla 48 Casos de prueba ejecutados.....	113
Tabla 49 Identificador de documento - seguimiento y control de pruebas.....	113
Tabla 50 Criterios de éxito	114
Tabla 51 Estado de las pruebas	114
Tabla 52 Progreso del plan de pruebas.....	115
Tabla 53 Lista de factores	115
Tabla 54 Métricas para la ejecución de pruebas	115
Tabla 55 Lista de nuevos riesgos	115
Tabla 56 Lista de requisitos previos para el despliegue	116
Tabla 57 Lista de los miembros del equipo.....	116
Tabla 58 Lista de riesgos - planificación de despliegue.....	117
Tabla 59 Cronograma - planificación de despliegue	117
Tabla 60 Identificación del documento - control de cambios	118
Tabla 61 Lista del historial de cambios	118
Tabla 62 Lista del estado de los cambios.....	119
Tabla 63 Identificación del documento - procedimiento de despliegue.....	119
Tabla 64 Pasos para la ejecución de despliegue.....	119
Tabla 65 Pasos para seguir el plan de retroceso.....	120
Tabla 66 Identificador de documento - cierre de despliegue	120
Tabla 67 Resultados del Despliegue	121
Tabla 68 Incidencias en el despliegue.....	121
Tabla 69 Identificación de obstáculos	121
Tabla 70 Practicas exitosas	121
Tabla 71 Revisión por pares	124
Tabla 72 Promedio por pregunta	134
Tabla 73 Comentarios de expertos	134

RESUMEN

En el presente trabajo, se diseñó una guía metodológica para las fases de ejecución de pruebas y puesta en producción de software, en la Dirección de Desarrollo Tecnológico e Informático (DDTI) de la Universidad Técnica del Norte (UTN). Se determinó que existe falta de una metodología formal en las fases de pruebas y puesta en producción, generando retrasos en entregas, errores en producción, fallas en retroalimentación, situaciones que afectan la calidad y confiabilidad de los sistemas propuestos, en desarrollo y en producción.

El desarrollo del marco teórico permitió abordar temas como: fundamentos para elaborar una guía o manual, calidad de software, así como los conceptos, lineamientos y prácticas recomendadas por cada una de las normas ISO/IEC/IEEE 29119 e ISO/IEC/IEEE 20000-1.

Se contextualiza la situación actual y línea base del DDTI, para definir las características y variables determinantes en el objeto de estudio; además, se aplicó investigación de campo para concluir con el análisis e interpretación de resultados, análisis y revisión documental y plantear formalmente la guía metodológica.

Con el fin de demostrar su aplicabilidad, se procedió a definir con aspectos prácticos el procedimiento para aplicar la guía metodológica en el DDTI. En el Anexo A, se describe con más detalle los pasos a seguir, específicamente para encontrar hallazgos, tomando como insumo el portafolio docente y estudiante.

Se utilizó, la prueba de concepto, como método de validación teórica, con tres expertos en ingeniería de software, quienes evaluaron la efectividad de la guía metodológica de calidad de software, emitiendo criterios muy favorables a la guía, producto de esta investigación.

Palabras clave: guía metodológica, ejecución de pruebas, puesta en producción, calidad de software, prueba de concepto, validación teórica.

ABSTRACT

In this study, a methodological guide was designed for the phases of software testing execution and production deployment at the Dirección de Desarrollo Tecnológico e Informático (DDTI) of the Universidad Técnica del Norte (UTN). It was determined that there is a lack of a formal methodology for the testing and deployment phases, leading to delivery delays, production errors, feedback failures, and issues that compromise the quality and reliability of the proposed, developing, and deployed systems.

The development of the theoretical framework addressed topics such as the fundamentals for creating a guide or manual, software quality, and the concepts, guidelines, and recommended practices outlined in the ISO/IEC/IEEE 29119 and ISO/IEC/IEEE 20000-1 standards.

The current situation and baseline of the DDTI were contextualized to define the key characteristics and variables of the study object. Additionally, field research was conducted to conclude with the analysis and interpretation of results, documentary review, and formal proposal of the methodological guide.

To demonstrate its applicability, the procedure for implementing the methodological guide at the DDTI was defined with practical aspects. Annex A provides a more detailed description of the steps to follow, specifically for identifying findings using the teacher and student portfolios as inputs.

The proof of concept was used as a theoretical validation method, involving three software engineering experts who evaluated the effectiveness of the methodological guide for software quality. Their feedback was highly favorable, highlighting the guide as a valuable outcome of this research.

Keywords: methodological guide, testing execution, deployment, software quality, proof of concept, theoretical validation.

INTRODUCCIÓN

Antecedentes

La calidad en el desarrollo y producción de software es esencial en el éxito de cualquier sistema, ya que impacta directamente en su funcionalidad, confiabilidad y capacidad para satisfacer las necesidades del usuario. Garantizar la calidad implica implementar procesos estratégicos de prueba y validación que permitan identificar y corregir defectos, asegurando que el producto cumpla con los estándares establecidos y con ello lograr que los usuarios confíen en sistemas de mejor rendimiento, fortaleciendo la relación entre desarrolladores y clientes y promoviendo percepciones positivas del producto. (Mejía Trejo, 2024)

La Dirección de Desarrollo Tecnológico e Informático (DDTI) de la Universidad Técnica del Norte es una unidad responsable del desarrollo, mantenimiento y soporte de herramientas y sistemas de gestión administrativa y académica. Sin embargo, el proceso de ejecución de las pruebas y puesta en producción de software, carecen de una estructura formalizada que garantice resultados consistentes y controlados.

Además, no cuentan con una guía específica y formal para la ejecución de pruebas y puesta en producción, lo que limita la efectividad de cada uno de los procesos y reduce la capacidad del equipo para abordar de manera integral los desafíos asociados con la calidad del software.

La escasa documentación generada en el proceso de ejecución de pruebas y puesta en producción es insuficiente ya que no proporciona un registro claro y completo, esto dificulta el seguimiento de las actividades realizadas, la verificación de los resultados, la trazabilidad de las pruebas realizadas y la implementación de mejoras continuas.

Planteamiento del problema

Según un artículo publicado en la 24ª Conferencia Internacional sobre Informática de Objetos Distribuida, señala que: “la ejecución de pruebas de software no se logra realizar en su totalidad, porque la documentación sobre los procesos de desarrollo es muy escasa y a su vez no existe el personal adecuado que posea los conocimientos necesarios en calidad de software, para llevar a cabo una planificación de pruebas”. (Napoleño, 2020)

En Ecuador, muchas de las empresas pequeñas y medianas (PYMES) que se enfocan en el desarrollo de software no cuentan con una certificación de calidad, porque estas no logran cumplir con los múltiples requerimientos que exige un estándar, además existe una baja utilización de herramientas de pruebas, afectando la calidad de estas y que surjan errores no detectados y sistemas inestables. (Terán-Ávila, 2021)

Según (Pantaleo, 2016) manifiesta que: “La calidad de software es el cumplimiento de las funciones y características especificadas de forma explícita en los requerimientos, con la finalidad de entregar al usuario final un producto útil, confiable y libre de errores”. Sin embargo, en el área de desarrollo del DDTI, es escasa la valoración de los aspectos mencionados.

La Dirección de Desarrollo Tecnológico e Informático (DDTI) de la Universidad Técnica del Norte, es una unidad agregadora de valor, su capital humano lo conforman once desarrolladores, cumple con la función de asesoría y apoyo a las autoridades universitarias, facilitando la producción de información para toma de decisiones.

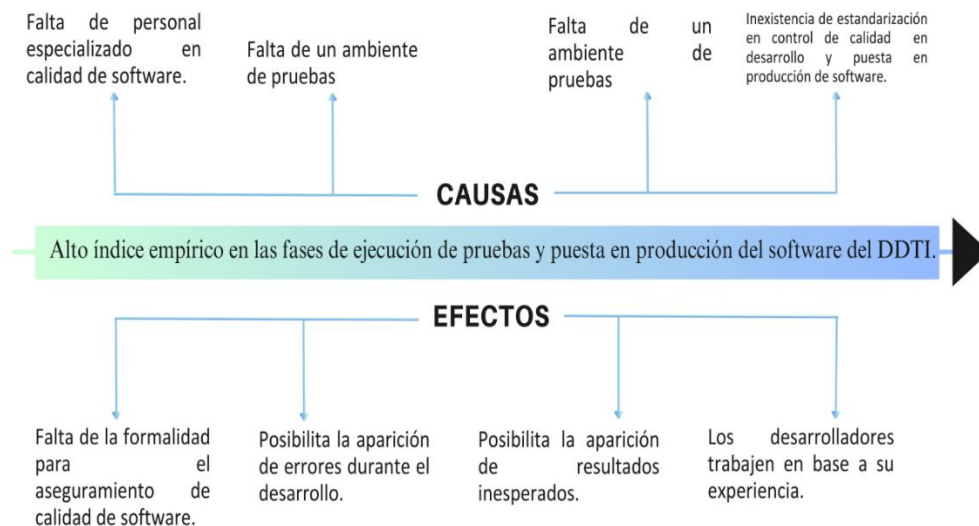
El DDTI no cuenta con una guía formal para la ejecución de pruebas de software, estas se realizan eventualmente por los propios desarrolladores de las aplicaciones desde la etapa inicial de un proyecto, ocasionando errores involuntarios en la calidad del producto, esto dificulta su detección y no facilita toma de decisiones oportunas, afectando en

indicadores de eficiencia y satisfacción de usuarios. Al no contar con un especialista QA en el DDTI, no se ha establecido porcentajes de error en los productos desarrollados.

La falta de adopción de una adecuada guía formal provoca la ausencia de artefactos relacionada a casos de prueba, resultados, seguimiento y control; a esto se suma la posible pérdida de tiempo y recursos, que causan demoras en el proceso de desarrollo y retrasos en la entrega de un proyecto, lo que provoca que no se garantice la calidad del software.

Como producto de entrevistas realizadas a los programadores del DDTI, se construyó el diagrama de Ishikawa con la participación de los principales actores, ver figura 1, concluyendo que no se cuenta con una guía formal a medida para aplicar conceptos de calidad de software en el proceso de desarrollo y manifiestan su interés por contar con una guía metodológica de calidad de software que se adapte a las tecnologías adoptadas y estandarizadas para el desarrollo de software.

Figura 1 Diagrama de Ishikawa.



Nota: Fuente: Elaboración propia.

OBJETIVOS

Objetivo General

Establecer una guía metodológica para las fases de ejecución de pruebas y puesta en producción del software de autoría del DDTI.

Objetivos Específicos

- Describir los principios y lineamientos de la norma ISO/IEC/IEEE 29119 en referencia a las fases de ejecución de pruebas y puesta en producción que sean adaptables al software de autoría del DDTI.
- Elaborar una guía metodológica de ejecución de pruebas, basado en el estándar ISO/IEC/IEEE 29119, adaptada a las necesidades del DDTI.
- Aplicar la guía metodológica en el submódulo de portafolio docente y estudiante.
- Validar la guía metodológica propuesta mediante prueba de concepto.

ALCANCE

La norma ISO/IEC/IEEE 29119 es un conjunto de estándares que tratan todas las etapas del ciclo de vida de las pruebas de software, con el propósito de concientizar la importancia de la calidad y confiabilidad del software, mediante la adopción de buenas prácticas en el área de las pruebas. (Veenendaal, 2016)

La norma consta de varias secciones, las cuales se enfocan en aspectos importantes sobre planificación, diseño, gestión, ejecución, documentación y evaluación de resultados de las pruebas realizadas, de igual forma cada sección se encamina a que las pruebas se realicen de manera efectiva utilizando las técnicas apropiadas. (Ardila, n.d.)

El presente proyecto tiene como propósito proponer una guía metodológica para la ejecución de pruebas de software en la Dirección de Desarrollo Tecnológico e Informático, utilizando como base los principios y estándares establecidos por la norma ISO/IEC/IEEE 29119, además realizar un análisis detallado de las prácticas aplicadas e

identificar las fortalezas y debilidades de los miembros del área de desarrollo, en cuanto a la aplicación del estándar ISO/IEC/IEEE 29119.

Para la aplicación de la norma, se estudiaron conceptos, procesos y técnicas recomendadas por la norma; para implementarlos en la guía metodológica propuesta, se definieron los pasos específicos que deben seguirse, incluyendo la planificación, diseño, ejecución y evaluación de las pruebas de software, ver figura 2.

A manera de caso de uso de la guía metodológica propuesta, se utilizaron dos submódulos del módulo del portafolio electrónico universitario del SIIU, en donde se recopilaban datos sobre su funcionalidad y usabilidad, posteriormente se evaluaron los resultados obtenidos.

Figura 2 Diagrama Estructural



Nota: Fuente: Elaboración propia.

METODOLOGÍA

La figura 3 describe el proceso metodológico seguido para cumplir con los objetivos propuestos.

En el cumplimiento del primer objetivo se realizó una revisión de fuentes y repositorios bibliográficos, con información confiable, precisa y actualizada para obtener una sólida fundamentación teórica, con la finalidad de trabajar dentro del contexto adecuado y fortalecer el conocimiento existente.

El segundo objetivo se cumplió mediante la recolección de información, utilizando encuestas y entrevistas dirigidas al área de desarrollo, que permitieron conocer el proceso de ejecución de pruebas de software existente, así como la problemática del proceso y los efectos que ha causado y con ello proceder a la elaboración de la guía metodológica propuesta.

Para el tercer objetivo se aplicó la guía metodológica propuesta a dos de los submódulos académicos del SIIU. Por último, se validó la guía metodológica propuesta con docentes expertos mediante prueba de concepto.

Figura 3 Diagrama Metodológico



Nota: Fuente: Elaboración propia.

JUSTIFICACIÓN

La Dirección de Desarrollo Tecnológico e Informático (DDTI) de la Universidad Técnica del Norte, es el área encargada de garantizar la calidad del software desarrollado tanto para el sistema académico, como administrativo y financiero de la institución, por este motivo en el presente trabajo de titulación se utilizó como base la norma ISO/IEEE 29119, ya que esta define estándares y prácticas para la ejecución de pruebas, lo que permitió crear y adoptar una guía metodológica basada en las necesidades del DDTI, así como fomentar la cultura de calidad y mejora continua.

Además, el desarrollo de este trabajo de grado busca fortalecer uno de los Objetivos de Desarrollo Sostenible, relacionado con el objetivo N°9 que es: “Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación”, ya que al momento de adoptar una guía metodológica estandarizada promueve la eficiencia y calidad del Sistema Informático Integrado Universitario (SIIU). (Naciones Unidas, n.d.)

Es importante mencionar que también cumple con el objetivo N°16 que es: “Promover sociedades justas, pacíficas e inclusivas”, ya que con esta guía metodológica propuesta se busca garantizar la calidad del software enfocándose en la transparencia, la integridad y la confianza en las instituciones educativas y con ello promover entornos justos en el ámbito educativo. (Naciones Unidas, n.d.)

Justificación Tecnológica. – La adopción de una guía metodológica estandarizada según los principios de la norma ISO/IEEE 29119, garantiza que el Sistema Informático Integrado Universitario (SIIU) cumpla con los requisitos y expectativas de los usuarios, mejorando su calidad y confiabilidad.

Justificación Institucional. – La guía metodológica propuesta mejora la calidad de software utilizado en la universidad, además de optimizar los recursos y tiempos

dedicados a las pruebas y de esta manera ofrecer a la comunidad educativa un sistema de calidad.

Justificación Metodológica. – Es importante establecer una guía metodológica de ejecución de pruebas, porque con esto se promueve el uso de estándares para mejorar las actividades y procesos repetitivos de un área, inclusive facilita la comunicación, la colaboración y la consistencia para generar la documentación necesaria de los registros de las pruebas realizadas.

CAPÍTULO 1

Marco Teórico

1.1. Fundamentos para elaborar una guía metodológica

La presente investigación se fundamentó en la revisión sistemática de literatura (SLR), que permitió evaluar y seleccionar trabajos relevantes de la ingeniería de software, en aspectos de calidad de software, ejecución de pruebas y despliegue, se extrajo y sintetizó los hallazgos de estos estudios.

Según (García, 2022) describe el proceso adecuado para una correcta revisión sistemática de literatura, detallado a continuación:

- Planificar: pregunta de investigación, definir objetivos, términos de búsqueda, criterios de inclusión y exclusión.
- Hacer: organizar y almacenar estudios, relevancia a cada estudio y extraer los datos importantes de los estudios seleccionados.
- Reportar: sintetizar los datos extraídos y presentar los hallazgos de manera clara y ordenada.

Los motores de búsqueda, repositorios bibliográficos y cadenas de búsqueda utilizados en esta investigación se detallan en la tabla 1.

Tabla 1 Búsqueda bibliográfica

Fuente bibliográfica	Cadena de búsqueda
IEEE Xplore	("software quality" OR "software defect") AND ("testing phase" OR "production phase") AND ("methodology" OR "guideline")
ScienceDirect	"software quality assurance" AND ("testing" OR "production")
Google Scholar	("testing phase" AND "production phase")
Scopus	("testing phase" AND "production phase")

Nota: Fuente: Elaboración propia.

1.1.1. Elementos de una guía o manual

En esta sección se describe los hallazgos de la revisión de la literatura. Con base en los resultados de la investigación, detallamos la estructura y los componentes de una guía metodológica en la tabla 2, según las perspectivas de los autores:

Tabla 2 Elementos de una guía o manual

Autor	Tipo de documento	Estructura
De acuerdo a (Robles, 2017) en su publicación de “Guía Metodológica. ¿Qué es? ¿Cómo se realiza?”, propone la siguiente estructura.	Guía	<ul style="list-style-type: none">• Prólogo• Resumen Ejecutivo• Introducción• Desarrollo• Conclusiones
Según (Cárdenas, 2020) establece la siguiente estructura para una guía metodológica.	Guía	<ul style="list-style-type: none">• Alcance• Roles• Planificación• Diseño• Construcción• Ejecución• Monitoreo• Finalización
Según (Páez, 2013) un manual debe tener la siguiente estructura.	Manual	<ul style="list-style-type: none">• Introducción• Alcance• Requisitos Generales• Desarrollo• Monitoreo y control• Seguimiento y evaluación• Documentación

Nota: Fuente: Elaboración propia

Por la naturaleza de esta investigación, para la elaboración de la guía metodológica de calidad de software, se resolvió utilizar la estructura de la guía del autor Cárdenas, porque se ajusta a los parámetros utilizados en el DDTI.

1.2. Calidad de software

La calidad de software es un aspecto esencial en el área de desarrollo ya que esto garantiza que los sistemas o aplicaciones cumplan no solo con el funcionamiento, si no que brinde

al usuario la confianza de que los atributos que solicitó se encuentren plasmados en dichos sistemas.

En el marco de los estándares internacionales de calidad, la ISO 9000:2015 proporciona la siguiente definición: “la calidad de los productos y servicios se determina por la capacidad para satisfacer a los clientes, que incluye no solo su función y desempeño, sino también el impacto previsto y el no previsto sobre las partes interesadas.” (ISO 9000, 2015)

La definición de (Pressman, 2010) sobre la calidad de software es: “la relación que existe entre las características y requisitos proporcionados formalmente por el usuario, en la entrega final del producto de software.”

La calidad de software es el nivel que posee un producto, en sus requisitos y componentes especificados por el cliente. Además de satisfacer las necesidades o expectativas del usuario y cumplir con un software funcional, seguro y fácil de mantener. (López Echeverry et al., 2008)

1.2.1. Gestión de calidad

La gestión de calidad comprende una colección de técnicas y métodos útiles dentro de una organización para el control de calidad de procesos, productos y servicios con la finalidad de que sean consistentes y satisfagan las necesidades de los clientes. (Camisón et al., 2007)

En el glosario de términos que proporciona la norma ISO 9000:2015 menciona que la gestión de calidad es: “la coordinación de actividades para dirigir y controlar una organización en lo que respecta a la calidad.” (ISO 9000, 2015)

Los principios de la gestión de calidad se detallan a continuación en la tabla 3:

Tabla 3 Principios de gestión de calidad

Principios de Gestión de Calidad	
Enfoque al cliente	Las organizaciones se orientan a cumplir las necesidades actuales y futuras de los clientes, entregando productos y servicios de calidad.
Liderazgo	En una organización, un líder debe establecer una visión clara y generar un ambiente de trabajo en donde exista la motivación y compromiso.
Participación del personal	Los miembros de un equipo de trabajo deben ser participativos y empoderarse de las actividades para contribuir a la gestión de calidad.
Enfoque basado en procesos	Las actividades deben cumplir con un proceso eficiente, para obtener un buen resultado.
Enfoque de sistema para la gestión	Comprensión y gestión de los procesos interrelacionados como un sistema, favorece al logro de los objetivos de una organización.
Mejora continua	En cada uno de los procesos, productos y servicios, la mejora continua es un objetivo principal y permanente, para el desempeño global.
Enfoque basado en hechos para la toma de decisión	Los datos, objetivos e información son necesarios ya que estos proporcionan una comprensión sólida de los problemas y permite tomar decisiones eficaces.
Relaciones mutuamente beneficiosas con el proveedor	Una relación de colaboración y confianza entre la organización y un proveedor es esencial para garantizar la calidad de los productos y servicios.

Nota: Fuente (Camisón et al., 2007)

1.2.2. Plan de calidad

La norma ISO 10005:2018 menciona que un plan de calidad son las especificaciones de los procedimientos, responsabilidades y recursos asociados que se aplicarán a un producto, servicio o proyecto para satisfacer necesidades, expectativas y requisitos relacionadas con un caso específico. (ISO 10005, 2018)

Un plan de calidad es un documento que se enfoca en la aplicación de estándares de calidad, en el cual se detallan políticas y procedimientos necesarios para garantizar la calidad en todas las etapas de desarrollo y ejecución de un producto.

1.2.3. Aseguramiento de calidad de software SQA

Son procesos que se ejecutan a lo largo del ciclo de vida del desarrollo de software; define estándares, auditorías, revisiones, pruebas, capacitaciones, riesgos, procesos de requisitos, procesos de diseño, entre otros.

El SQA es un proceso que da soporte a la entrega de productos de alta calidad, proporcionando al personal del proyecto y a los gerentes, en todos los niveles, la visibilidad apropiada y la realimentación sobre los procesos y los productos de trabajo asociados, durante toda la vida del proyecto. (Valencia et al., 2019)

El Comité Internacional de Certificaciones de Pruebas de Software (ISTQB, n.d.) considera dos tipos de aseguramiento de calidad de software, detallados a continuación en las tablas 4 y 5:

Tabla 4 QA Constructivo

QA Constructivo	
Técnico	Organizacional
Métodos	Guías
Herramientas	Estándares
Lenguajes	Check List
	Regulaciones
	Requisitos Legales

Nota: Fuente: (SSTQB, 2023)

Tabla 5 QA Analítico

QA Analítico		
Estático	Dinámico	
	Caja Blanca	Caja Negra
Revisión/tutoriales	Cobertura de declaración	Partición de equivalencia
Análisis de flujo de control	Cobertura de sucursales	Análisis de valor límite
Análisis de flujo de datos	Cobertura de condición	Pruebas de transacción de estado
Métrica/analizador del compilador	Cobertura de ruta	Tablas de decisiones
		Pruebas basadas en casos de uso

Nota: Fuente: (SSTQB, 2023)

1.2.4. Control de calidad de software QC

En el libro “Calidad de Sistemas de Información” se señala: El control de la calidad es un proceso que se lleva a cabo durante todo el proyecto, mediante éste se monitorean y se registran los resultados de la ejecución de las actividades de gestión de calidad con el fin de evaluar su rendimiento y asegurar que las salidas del proyecto son correctas, completas y satisfacen las expectativas de los interesados clave para conseguir su aceptación final. (Piattini et al., 2018)

En una conferencia sobre “Mantenimiento y Evolución del Software” realizada por la IEEE se propone un control de calidad mejorado en donde las métricas constituyen la base, pero los ingenieros de calidad deben analizar e interpretar los resultados de las métricas dentro de su contexto y transformarlos en actividades específicas y manejables que los desarrolladores puedan abordar de manera efectiva. (Steidl et al., 2014).

El proceso de control de calidad mejorado se muestra en la siguiente tabla 6:

Tabla 6 Control de calidad mejorado

Control de calidad mejorado	
Actividad	Ideas conceptuales
Transparente	El ingeniero de calidad coordina y define con la dirección y desarrolladores, un objetivo que se adapte a cada sistema y criterios específicos que cubra el contexto de la empresa.
Procesable	El ingeniero de calidad tiene la responsabilidad de interpretar manualmente los resultados y especificar tareas de refactorización procesables y comprensibles.
Vinculante	Por lo general, el ingeniero de calidad realiza una documentación de los criterios de calidad que se cumplen o se violan, para que este sea revisado por gerencia, con el fin de proporcionar más recursos o generar actividades específicas.

Nota: Fuente: (Steidl et al., 2014)

El proceso de control de calidad está diseñado para ser transparente (todas las partes implicadas están de acuerdo en el objetivo y las consecuencias de los fallos), procesable

(la medición debe provocar acciones) y vinculante (los ingenieros de calidad, desarrolladores y gestores siguen las normas acordadas).

1.2.5. Costo de la calidad de software

Según (Ventura, n.d.) considera que: “el costo de la calidad de software en un proyecto es primordial porque incluye tanto los costos tangibles como intangibles, es decir refleja los gastos asociados en todo el ciclo de vida del software, desde la concepción hasta el mantenimiento.”

A continuación, en la tabla 7 se detalla los diferentes tipos de costos referentes a la calidad de software.

Tabla 7 Tipos de costos de calidad de software

Tipos de costos de calidad de software	
Prevención	Evitar defectos en cuanto planificación, definición de procesos, políticas y estándares, especificación de requerimientos, recopilación de métricas y herramientas.
Evaluación	Descubrir la condición de la calidad del software mediante la revisión de especificaciones de requerimientos, diseño y componentes, verificaciones y validaciones en general.
Fallas Internas	Corrección de defectos previos.
Fallas Externas	Corrección de problemas detectados por el usuario mediante un mantenimiento o actualizaciones al software.

Nota: Fuente: (IEEE Computer Society, 2014)

1.2.6. Métricas de calidad de software

Según Shyamal, Asanka y Wickramaaranchchi, en su artículo “Un enfoque integral para evaluar la calidad del código de software a través de un modelo de calidad flexible” menciona que: la dificultad para especificar métricas de calidad de software es un desafío que abarca una amplia gama de factores como usabilidad, seguridad, rendimiento y confiabilidad, además depende de las expectativas y necesidades de los usuarios. No existen métricas únicas, pero la mayoría puede basarse en el contexto en el que se encuentra el software y con ello no afectar la calidad, por otro lado, los

requisitos evolucionan con el tiempo y las métricas deben ser flexibles y actualizarse constantemente. (Shyamal et al., 2023).

El uso de métricas establece cómo se debe ajustar el software a los requisitos implícitos y explícitos del cliente. Es decir, la medición para que el sistema se adapte a los requisitos establecidos. (Redrován et al., 2017)

Por otra parte, las métricas de calidad deben permitir lo siguiente:

- Indicar la calidad del producto
- Evaluar la productividad de los desarrolladores
- Evaluar los beneficios en términos de productividad y calidad
- Establecer una línea base para la estimación
- Ayudar a justificar el uso de nuevas herramientas o formación adicional

En el trabajo de titulación elaborado por (Laos, 2020) detalla que, en el análisis de resultados de las pruebas realizadas para verificar el alta calidad basado en el mejoramiento continuo, las estadísticas e indicadores permitieron no solo la vista global del software, si no también, diseñar planes de mejora generales y específicos. En el proceso de evaluación, se hicieron uso de las métricas que se detallan a continuación:

- Cobertura: indica el porcentaje de pruebas ejecutadas sobre el total de pruebas diseñadas.
- Defectos: indica el porcentaje de la densidad de defectos encontrados sobre el total de pruebas ejecutadas.
- Efectividad: indica el porcentaje de defectos detectados por QA sobre el total de pruebas detectadas en total.
- Calidad: indica el porcentaje de pruebas ejecutadas en forma satisfactoria sobre el total de pruebas diseñadas.

1.3.Pruebas de software

Bern Bruegge en su libro “Ingeniería de software orientado a objetos” indica que: “Las pruebas de software es un proceso sistemático cuyo objetivo es evaluar y verificar la calidad del funcionamiento de software, además de poder detectar errores y posibles defectos antes de entregar al usuario o se ponga en producción.” (Bruegge & Dutoit, 2002)

Según SWEBOK, “La ejecución de pruebas de software consiste en evaluar el comportamiento del funcionamiento de un programa, mediante la aplicación de pruebas unitarias, integración y depuración, con la finalidad de establecer la calidad de un producto.” (IEEE Computer Society, 2014)

Las pruebas de software se las puede definir como un elemento crítico para la garantía de calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación. (Alonso et al., 2005)

1.3.1. Principios de las pruebas

El ISTQB establece los fundamentos de siete principios para el “testing”, detallados en la tabla 8:

Tabla 8 Principios de las pruebas

Principios de las pruebas	
El proceso de prueba demuestra la presencia de defectos, no su ausencia	La tarea esencial de las pruebas es hallar defectos con el propósito de realizar correcciones y mejoras. El conocimiento y la experiencia en el área son factores muy importantes para encontrar la mayor cantidad de defectos.
La prueba exhaustiva es imposible	Este tipo de pruebas se deben aplicar siempre y cuando existan los recursos disponibles como el tiempo y dinero suficiente, en lugar de ello se debe aplicar una estrategia en donde se haga uso de técnicas de prueba efectivas y centrarse en características, prioridades y áreas críticas del sistema
La prueba temprana ahorra tiempo y dinero	En las etapas iniciales de desarrollo es vital realizar pruebas para corregir defectos, mismas que pueden partir desde las

	historias de usuario, y así evitar mayores complicaciones durante las pruebas finales y la puesta en producción
Los defectos se agrupan	La identificación de patrones de defectos proporciona el hallazgo de fallas, que por lo general suelen agruparse en áreas o módulos del software, lo cual ayuda al equipo de pruebas a enfocarse en el área problemática y a colaborar con los desarrolladores para solucionar los problemas subyacentes.
Las pruebas se desgastan	El aplicar las mismas pruebas en nuevas versiones de un software, se vuelven cada vez más ineficaces. Para contrarrestar este efecto se debe diversificar los casos de prueba y actualizar las pruebas agregando nuevos escenarios para cubrir las áreas con problemáticas
La prueba depende del contexto	Es esencial tener en cuenta que no existe una única solución o enfoque de prueba que se ajuste a todos los proyectos, ya que las pruebas deben ser versátiles y funcionar en una variedad de situaciones, sobre todo al contexto específico de la empresa
Falacia de la ausencia de defectos	Las pruebas no garantizan que un software este completamente libre de errores y la ausencia de estos no debe asumirse como un indicador de calidad, ya que después de la puesta en producción puede aparecer nuevos problemas, es decir que las pruebas deben continuar mientras el software se utilice y se mantenga.

Nota: Fuente: (Torrera, 2023)

1.4.ISO/IEC/IEEE 29119 Software Testing

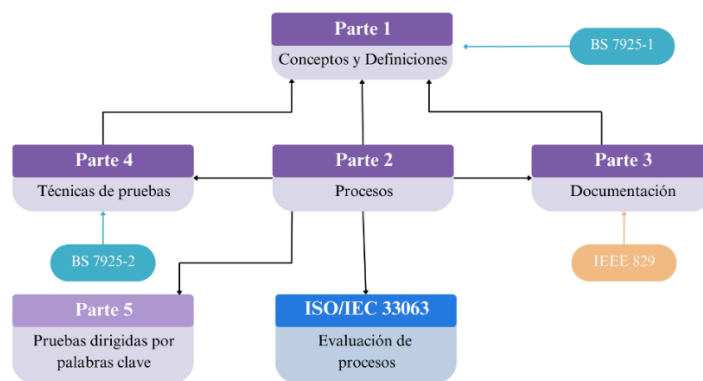
La ISO/IEC/IEEE 29119 contiene documentos reconocidos internacionalmente, orientados al soporte de pruebas en una variedad de aplicaciones y en cualquier estado de ciclo de vida, permite trabajar áreas del “testing” que no han sido correctamente tratadas por otros estándares o normativas. (Reid, 2023)

La ISO/IEC/IEEE 29119 realiza pruebas a través de un proceso específico, incluso define técnicas de diseño y medidas para lograr los niveles de calidad adecuados para un producto de software. (Veenendaal, 2016)

En mayo de 2007, la ISO aprobó la creación de un nuevo conjunto de normas para pruebas de software, fundamentado en estándares previos de IEEE y BSI, como IEEE 829, IEEE 1008, BS 7925-1 y BS 7925-2. Dado que no había un grupo especializado en pruebas de software dentro del SC7, se formó un equipo de trabajo denominado "Pruebas de

software" (WG26), que para 2013 ya contaba con la representación de más de 20 países. Originalmente, el conjunto de normas estaba compuesto por cuatro partes, pero se añadió una quinta enfocada en la evaluación de procesos, desarrollada en colaboración con los grupos ISO WG10 (Evaluación de procesos) y WG26. Además, se ha iniciado la creación de un estándar independiente sobre pruebas basadas en palabras clave. (Reid, 2023)

Figura 4 Estructura de ISO/IEC/IEEE 29119



Nota: Fuente: (Reid, 2023)

La figura 4 ilustra cómo los estándares existentes se integran en las partes 1 a 4 de los nuevos estándares ISO/IEC/IEEE 29119, los cuales se basan en cuatro entidades fundamentales con los procesos de prueba como eje central. La documentación de pruebas se genera a partir de la ejecución de estos procesos, mientras que las técnicas para llevar a cabo las pruebas se establecen dentro de los procesos y se detallan de manera independiente. Además, la terminología utilizada en las demás partes de este modelo se define en el vocabulario correspondiente. (Reid, 2023)

1.4.1. ISO/IEC/IEEE 29119 – 1: Conceptos y definiciones

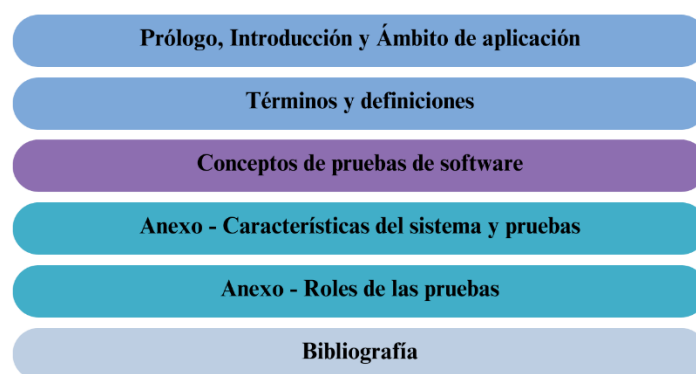
Describe conceptos comunes de cada una de las partes de la ISO/IEC/IEEE 29119, es un estándar informativo y no impone requisitos específicos que los usuarios deban cumplir. En sí proporciona conceptos, terminología y ejemplos prácticos para su

aplicación en la implementación de los estándares. (“ISO/IEC/IEEE 29119 - Parte 1: Conceptos Generales,” 2022)

A continuación, en la figura 5 se muestra la estructura de la parte 1 que incluye:

- Conceptos básicos de pruebas de software: explica los principios fundamentales, como la relación entre las pruebas, gestión de riesgos y gestión de proyectos. Además, aborda los diferentes tipos de procesos de prueba, cómo se diseñan y ejecutan.
- Anexos: el primer anexo detalla las características específicas de diferentes sistemas (como sistemas autónomos, móviles, tiempo real y web) y los tipos de pruebas que estos sistemas suelen requerir, además, el anexo A.15 corresponde a este trabajo de investigación, mientras que el segundo anexo describe las diversas funciones que puede desempeñar un evaluador en el contexto de las pruebas de software.

Figura 5 ISO/IEC/IEEE 29119 Parte 1: conceptos y vocabulario



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 1: Conceptos Generales,” 2022)

1.4.2. ISO/IEC/IEEE 29119 – 2: Procesos de prueba

Define los procesos de prueba de software y comprende varias cláusulas especificadas en la figura 6. A diferencia de la Parte 1, esta sección no es informativa, sino que incluye

requisitos específicos que los usuarios deben cumplir, imponiendo acciones obligatorias para quienes siguen la norma. (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

Figura 6 ISO/IEC/IEEE 29119 Parte 2: procesos de prueba



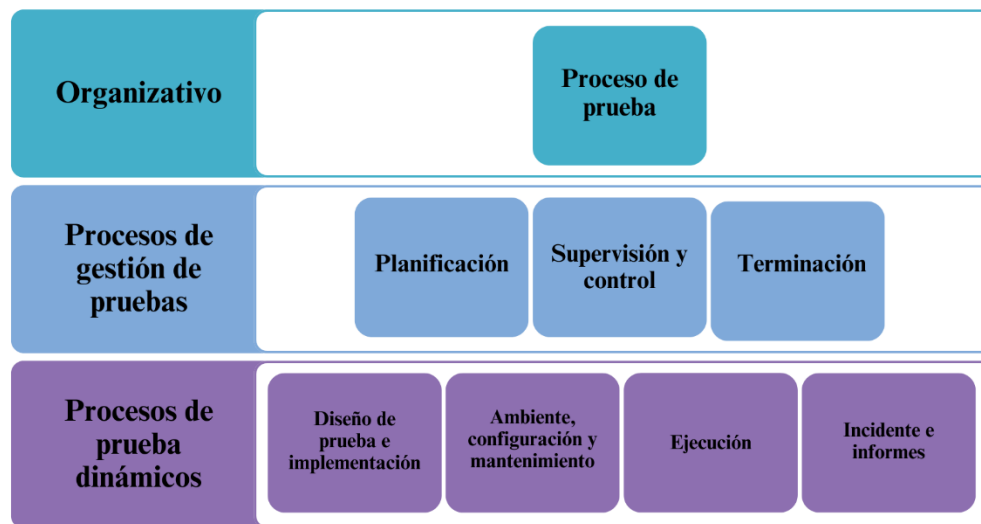
Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

Los requisitos de esta sección corresponden a la necesidad del usuario, convirtiéndose en un estándar normativo, además se compone de un modelo de proceso de pruebas de tres capas, detallado a continuación:

1. Capa Superior (Proceso de Pruebas Organizativo): se encarga de generar y mantener la política y las prácticas de prueba de la organización, asegurando que las pruebas se alineen con los objetivos y estándares organizativos. (Reid, 2023)
2. Capa Intermedia (Procesos de Gestión de Pruebas): se gestiona la estrategia y planificación de las pruebas, el seguimiento y control de las mismas, y la finalización de las actividades de prueba. Incluye procesos para asegurar que las pruebas se planifiquen, monitoreen y concluyan de manera efectiva. (Reid, 2023)

3. Capa Inferior (Procesos Dinámicos de Pruebas): se dedica al diseño y ejecución de las pruebas dinámicas, lo que incluye la creación y ejecución de casos de prueba. Los procesos en esta capa son específicos para las pruebas que implican la ejecución del software. (Reid, 2023)

Figura 7 Modelo de proceso de pruebas de tres capas ISO/IEC/IEEE 29119

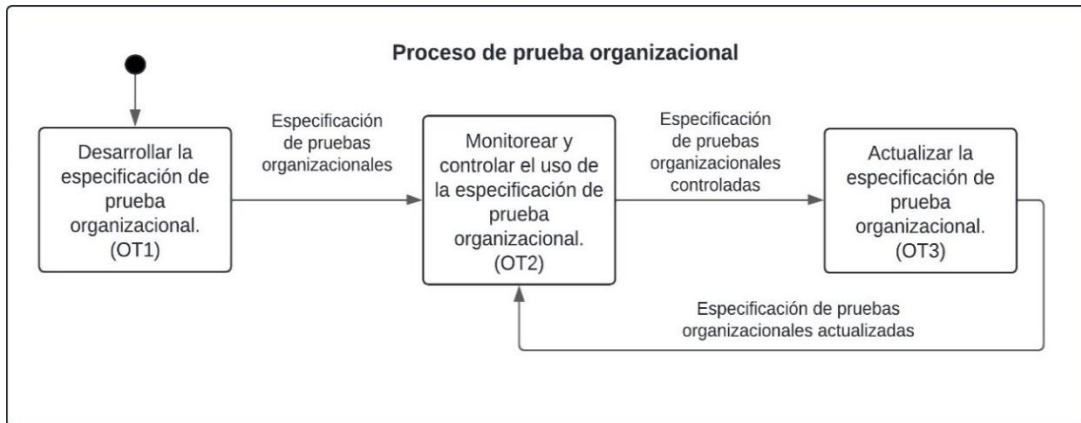


Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

La figura 7 describe la estructura de los procesos de prueba definidos en la sección de procesos de prueba, que incluye un conjunto completo de ocho procesos. Cada uno de estos procesos se desglosa en actividades, y cada actividad, a su vez, se compone de varias tareas. Las tareas representan el nivel más detallado de la estructura y especifican acciones que pueden ser requeridas, recomendadas u opcionales.

El proceso de pruebas organizacionales comprende actividades para la creación, revisión y mantenimiento de especificaciones de prueba, detallado en la figura 8.

Figura 8 Proceso de prueba organizacional



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

A continuación, en la tabla 9 se detalla cada una de las tareas correspondientes a las actividades del proceso de prueba organizacional

Tabla 9 Tareas del proceso de prueba organizacional

Proceso de pruebas organizacionales	Actividades	Tareas
Proceso de prueba	Desarrollar la especificación de prueba organizacional	<p>Los requisitos para las especificaciones de prueba organizacionales se identificarán a partir de las prácticas de prueba actuales dentro de la organización, de las partes interesadas o se desarrollarán por otros medios.</p> <p>Los requisitos de especificación de la prueba organizativa se utilizarán para crear la especificación de la prueba organizativa</p> <p>La aprobación del contenido de la especificación de la prueba organizativa se obtendrá de las partes interesadas.</p> <p>La disponibilidad de la especificación de pruebas de la organización se comunicará a las partes interesadas en la organización.</p>
	Monitorear y controlar el uso de la especificación de prueba organizacional	<p>Se debe monitorear el uso de la especificación de prueba organizacional para determinar si se está utilizando efectivamente dentro de la organización.</p> <p>Se tomarán las medidas adecuadas para fomentar la alineación de las partes interesadas con la especificación de pruebas de la organización.</p>
	Actualizar la especificación de prueba organizacional	<p>Debe revisarse la retroalimentación sobre el uso de la especificación de prueba organizativa.</p> <p>Se debe considerar la eficacia del uso y gestión de la especificación de prueba organizativa y se debe determinar y aprobar cualquier retroalimentación y cambios para mejorar su eficacia.</p>

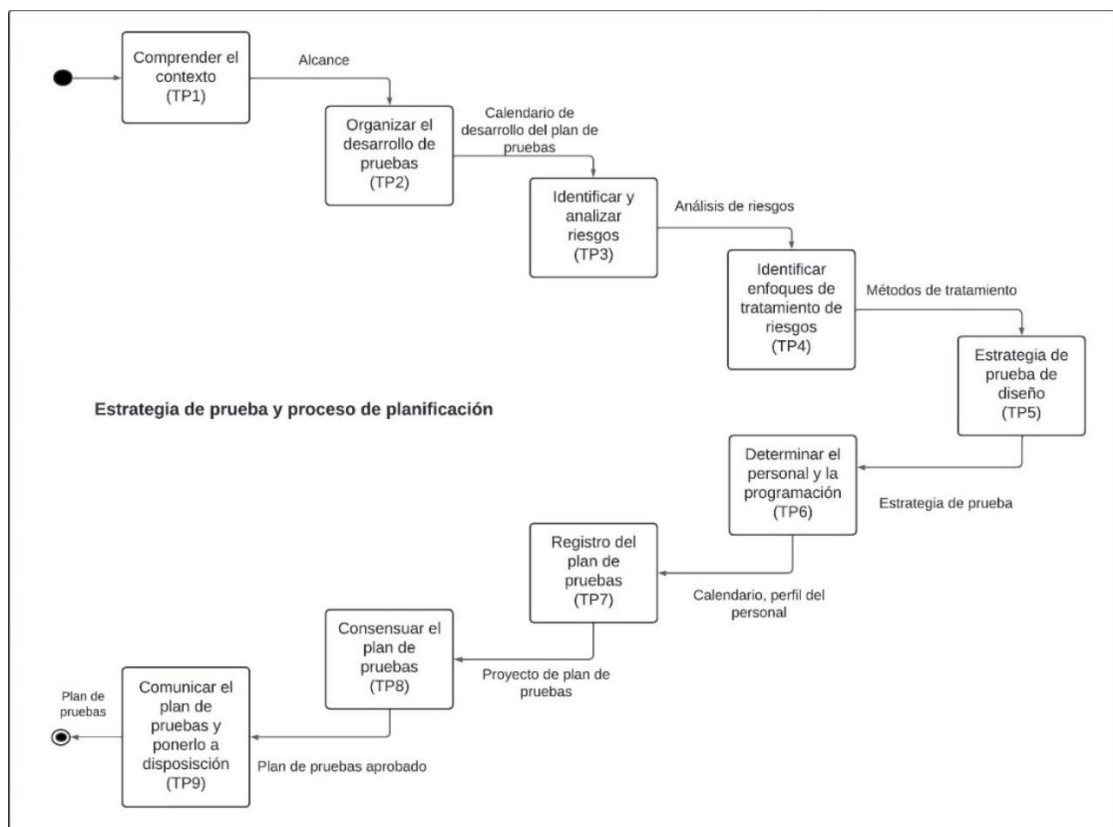
Cuando se hayan determinado y aprobado cambios en la especificación de prueba organizativa, dichos cambios deberán aplicarse.

Todos los cambios en la especificación de pruebas de la organización se comunicarán a toda la organización, incluidas todas las partes interesadas.

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

El proceso de estrategia de prueba y proceso de planificación comprende las actividades detalladas en la figura 9. El proceso se muestra de forma secuencial, pero en la práctica se puede llevar a cabo de forma iterativa, dependiendo de la escala y naturaleza de los proyectos a implementar.

Figura 9 Estrategia de prueba y proceso de planificación



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

En la siguiente tabla 10, se detalla las tareas asignadas a cada una de las actividades de la estrategia de prueba y proceso de planificación

Tabla 10 Tareas del proceso de estrategia de prueba y proceso de planificación

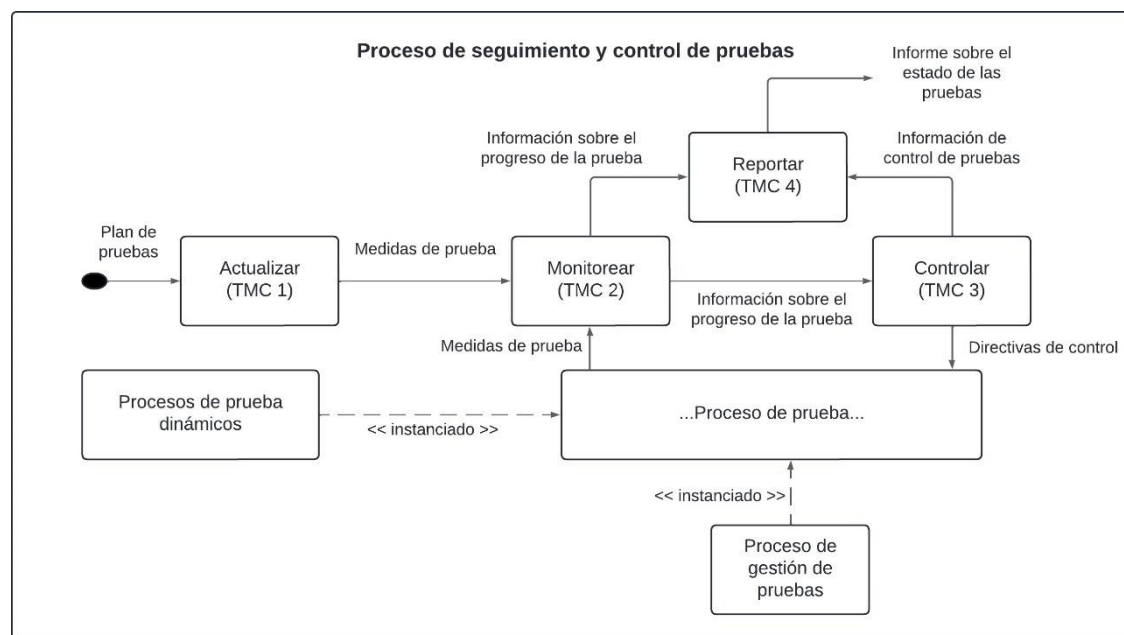
Proceso de gestión de pruebas	Actividades	Tareas
Estrategia de prueba y proceso de planificación	Comprender el contexto	<p>Se obtendrá una comprensión del contexto y del alcance de las pruebas del software para apoyar la preparación del plan de pruebas.</p> <p>La comprensión del contexto y de los requisitos de las pruebas de software debe obtenerse identificando a las partes interesadas pertinentes e interactuando con ellas.</p> <p>Se identificarán y registrarán las bases de las pruebas.</p> <p>Deberá iniciarse la planificación de la comunicación y registrarse las líneas de comunicación.</p>
	Organizar el desarrollo del plan de pruebas	<p>Basándose en los requisitos de las pruebas identificados en la actividad de comprensión del contexto (TP1), se identificarán y programarán aquellas actividades que deban realizarse para completar el diseño de las pruebas y la planificación de las mismas.</p> <p>Deberán identificarse las partes interesadas necesarias para participar en estas actividades.</p> <p>La aprobación de las actividades, el calendario y los participantes se obtendrá de las partes interesadas.</p> <p>Se debe organizar la participación de las partes interesadas.</p>
	Identificar y analizar los riesgos	<p>Se revisarán todos los riesgos que se hayan identificado previamente para identificar aquellos que estén relacionados con las pruebas de software y/o puedan tratarse con ellas.</p> <p>Se identificarán los riesgos adicionales que estén relacionados con las pruebas de software y/o puedan ser tratados por éstas. Cualquier riesgo identificado que no esté relacionado con las pruebas de software deberá comunicarse a las partes interesadas pertinentes.</p> <p>Los riesgos se clasificarán utilizando un sistema de clasificación adecuado que, como mínimo, distinga entre riesgos del proyecto y del producto.</p> <p>A cada riesgo se le asignará un nivel de exposición (por ejemplo, considerando su impacto y probabilidad).</p> <p>Los resultados de esta evaluación de riesgos deberán ser aprobados por las partes interesadas.</p> <p>Se registrarán los resultados de esta evaluación de riesgos.</p>
	Identificar enfoques de tratamiento de riesgos	<p>Se determinarán los medios adecuados para tratar los riesgos, en función del tipo de riesgo, la clasificación y el nivel de exposición al riesgo.</p> <p>Se registrarán los medios identificados para tratar los riesgos.</p>
	Diseñar la estrategia de pruebas	<p>Se diseñará una estrategia de pruebas que tenga en cuenta la base de las pruebas, los riesgos y las</p>

	<p>limitaciones organizativas, del proyecto y del producto.</p> <p>Se identificarán las actividades necesarias para aplicar la estrategia de pruebas.</p> <p>Se determinarán los parámetros que se utilizarán para el seguimiento y control de las pruebas. (TMC1 a TMC4)</p> <p>Se determinarán los requisitos de los datos de las pruebas.</p> <p>Se determinarán los requisitos del entorno y las herramientas de ensayo.</p> <p>Se identificarán los entregables de las pruebas y se registrará su grado de formalidad y frecuencia de comunicación.</p> <p>Se realizará una estimación inicial de los recursos necesarios para llevar a cabo el conjunto completo de actividades necesarias para aplicar la estrategia de pruebas.</p> <p>Se registrará la estrategia de pruebas.</p> <p>Se obtendrá la aprobación de la estrategia de pruebas por parte de los interesados.</p>
Determinar el personal y la programación	<p>Deberán identificarse las funciones y competencias del personal necesario para llevar a cabo las pruebas descritas en la estrategia de pruebas.</p> <p>Cada actividad de prueba requerida en la estrategia de pruebas se programará en función de las estimaciones dependencias y disponibilidad de personal.</p> <p>La aprobación de la dotación de personal y la programación se obtendrá de las partes interesadas pertinentes.</p>
Registro del plan de pruebas	<p>Las estimaciones finales de las pruebas se calcularán basándose en la estrategia de pruebas diseñada en la actividad de diseño de la estrategia de pruebas (TP5) y en la dotación de personal y la programación acordadas en la actividad de determinación de la dotación de personal y la programación (TP6).</p> <p>La estrategia de pruebas identificada en la actividad de diseño de la estrategia de pruebas (TP5), el perfil del personal y el calendario acordados en la actividad de determinación del personal y el calendario (TP6), y las estimaciones finales calculadas en la tarea anterior se incorporarán al plan de pruebas.</p>
Consensuar el plan de pruebas	<p>Se recabarán las opiniones de las partes interesadas sobre el plan de pruebas.</p> <p>Se resolverán los conflictos entre el plan de pruebas y las opiniones de las partes interesadas.</p> <p>Se actualizará el plan de pruebas para tener en cuenta las opiniones de las partes interesadas.</p> <p>Obtener la aprobación del plan de pruebas de las partes interesadas.</p>
Comunicar el plan de pruebas y ponerlo a disposición	<p>El plan de pruebas estará disponible.</p> <p>La disponibilidad del plan de pruebas se comunicará a las partes interesadas.</p>

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

El proceso de monitoreo y control de pruebas, como se lo muestra en la figura 10, examina si las pruebas progresan de acuerdo con el plan de pruebas y las especificaciones de pruebas de la organización, como la política de pruebas de la organización y las prácticas de pruebas de la organización.

Figura 10 Proceso de seguimiento y control de pruebas



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

Las actividades del proceso de seguimiento y control de pruebas constan de las siguientes tareas, detalladas en la tabla 11.

Tabla 11 Tareas del proceso de seguimiento y control

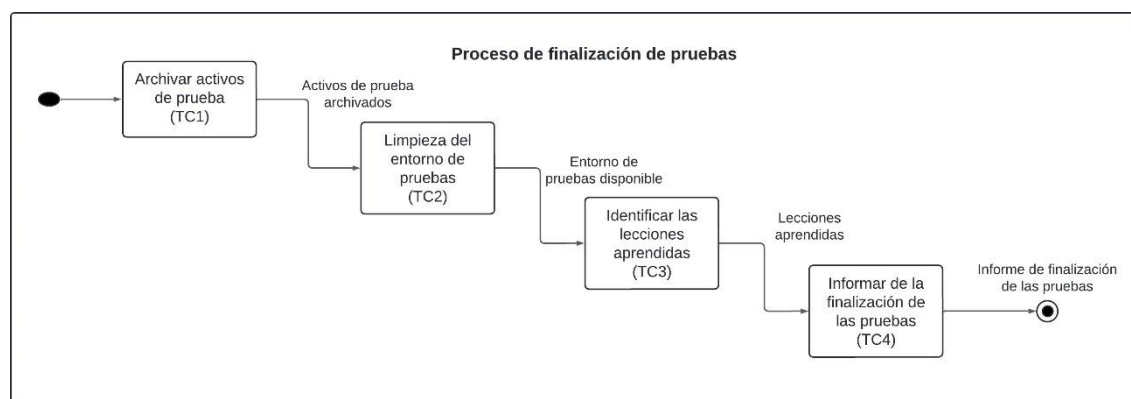
Proceso de gestión de pruebas	Actividades	Tareas
Proceso de seguimiento y control de pruebas	Configuración	Deben identificarse medidas adecuadas para supervisar el progreso con respecto al plan de pruebas, si estas medidas no están ya definidas en el plan de pruebas o en el documento de prácticas de pruebas de la organización.
		Deben identificarse medios adecuados para identificar riesgos nuevos y cambiantes, si no están ya definidos en el plan de pruebas o en el documento de prácticas de pruebas de la organización.
		Se establecerán actividades de supervisión, como informes sobre el estado de las pruebas y recopilación de métricas de pruebas, para recopilar las medidas

	<p>identificadas en las primeras tareas, y en el plan de pruebas y el documento de prácticas de pruebas de la organización.</p>
	<p>Se recopilarán y registrarán las mediciones de las pruebas.</p>
	<p>El progreso con respecto al plan de pruebas se supervisará utilizando las medidas de las pruebas recopiladas.</p>
Monitorear	<p>Se identificarán las desviaciones de las actividades de ensayo previstas y se registrarán los factores que bloquean el progreso.</p>
	<p>Se identificarán y analizarán los nuevos riesgos para determinar los que requieren tratamiento mediante pruebas y los que deben comunicarse a otras partes interesadas.</p>
	<p>Se supervisarán los cambios en los riesgos conocidos para identificar los que requieren tratamiento mediante pruebas y los que deben comunicarse a otras partes interesadas.</p>
	<p>Se realizarán las acciones necesarias para implementar el plan de pruebas.</p>
	<p>Se llevarán a cabo las acciones necesarias para aplicar las directrices de control recibidas de los procesos de gestión de nivel superior.</p>
	<p>Se identificarán las acciones necesarias para gestionar las divergencias entre las pruebas reales y las planificadas.</p>
Controlar	<p>Se identificarán los medios para tratar los nuevos riesgos identificados y modificados.</p>
	<p>Según proceda:</p> <ol style="list-style-type: none"> 1) se emitirán directivas de control para introducir cambios en la forma de realizar las pruebas. 2) los cambios en el plan de pruebas se realizarán en forma de actualizaciones del plan de pruebas 3) los cambios recomendados se comunicarán a las partes interesadas.
	<p>El progreso de las pruebas con respecto al plan de pruebas se comunicará a las partes interesadas en un informe de situación de las pruebas correspondiente al período de notificación especificado.</p>
Informar	<p>Los nuevos riesgos y los cambios en los riesgos existentes se actualizarán en el registro de riesgos y se comunicarán a las partes interesadas pertinentes.</p>

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

El proceso de finalización de pruebas, como se muestra en la figura 11, se realiza cuando se ha obtenido el acuerdo de que las actividades de prueba estén completas.

Figura 11 Proceso de finalización de pruebas



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

A continuación, en la tabla 12, se muestra las tareas correspondientes a las actividades del proceso de finalización de pruebas

Tabla 12 Tareas del proceso de finalización de pruebas

Proceso de gestión de pruebas	Actividades	Tareas
Proceso de finalización de pruebas	Archivar activos de prueba	<p>Aquellos activos de prueba que puedan ser útiles en el futuro o que se prevea reutilizar en una fecha posterior deberán identificarse, ponerse a disposición utilizando los medios adecuados y archivarse.</p> <p>La disponibilidad de los activos de prueba reutilizables se registrará en el informe de finalización de la prueba y se comunicará a las partes interesadas pertinentes.</p>
	Limpiar el entorno de prueba	<p>el entorno de ensayo se restablecerá a un estado predefinido al término de todas las actividades de ensayo.</p>
	Identificar lecciones aprendidas	<p>Se registrarán las lecciones aprendidas durante el proyecto.</p> <p>Los resultados se registrarán para su inclusión en el informe de finalización de la prueba y se comunicarán a las partes interesadas pertinentes.</p>
	Informe de finalización de prueba	<p>La información pertinente se recopilará a partir de los siguientes documentos, entre otros</p> <ol style="list-style-type: none"> 1) planes de pruebas (por ejemplo, plan de pruebas del proyecto, plan de pruebas del sistema o plan de pruebas de rendimiento) 2) resultados de las pruebas 3) informes de estado de las pruebas 4) informes de finalización de pruebas de nivel o tipo de prueba 5) informes de incidencias.

La información recopilada se evaluará y resumirá en el informe de finalización de la prueba.

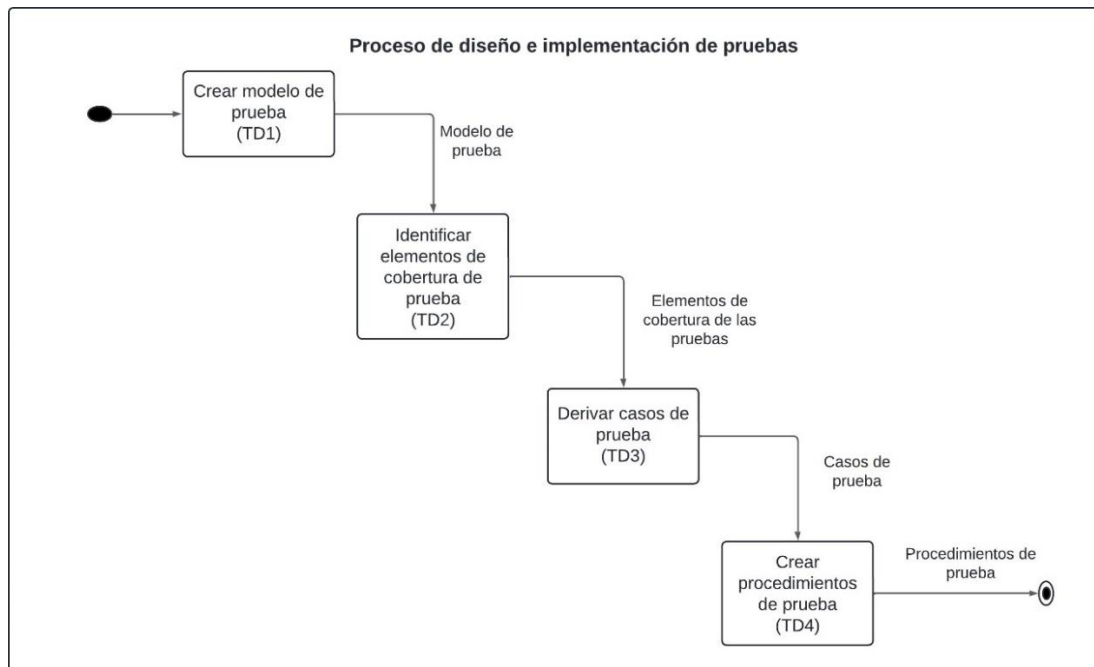
Se obtendrá la aprobación del informe de finalización de la prueba de las partes interesadas responsables.

El informe de finalización de la prueba aprobado se distribuirá a las partes interesadas pertinentes.

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

El proceso de diseño e implementación de pruebas, que se muestra en la figura 12, se utiliza para derivar casos de prueba y procedimientos de prueba.

Figura 12 Proceso de diseño e implementación de pruebas



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

Las actividades del proceso de diseño e implementación de pruebas constan de las siguientes tareas, detalladas en la tabla 13.

Tabla 13 Tareas del proceso de proceso de diseño e implementación de pruebas

Proceso de prueba dinámicos	Actividades	Tareas
Proceso de diseño e implementación de pruebas	Crear modelo de prueba	Se analizarán las bases de la prueba para comprender los requisitos del elemento de prueba.

	<p>La estrategia de ensayo se utilizará para identificar las partes y/o características del elemento de ensayo que deben someterse a ensayo.</p> <p>La notación del modelo de ensayo se decidirá en función de la cobertura de ensayo requerida.</p> <p>El modelo de prueba se creará para el elemento de prueba.</p>
Identificar los elementos de cobertura de las pruebas	<p>Los elementos de cobertura de la prueba que se ejercitarán mediante la prueba se obtendrán aplicando técnicas de diseño de la prueba al modelo de prueba para alcanzar los criterios de cobertura de finalización de la prueba especificados en el plan de prueba.</p> <p>Los elementos de cobertura de la prueba se priorizarán utilizando los niveles de exposición al riesgo documentados en la actividad de identificación y análisis de riesgos (TP3).</p> <p>Los elementos de cobertura de la prueba se registrarán en la especificación del caso de prueba.</p> <p>Se registrará la trazabilidad entre la base de la prueba, el modelo de prueba y los elementos de cobertura de la prueba.</p>
Obtener casos de prueba	<p>Se derivarán uno o más casos de prueba determinando las condiciones previas, seleccionando los valores de entrada y, en caso necesario, las acciones para ejercitar los elementos de cobertura de la prueba seleccionados, y determinando los resultados esperados correspondientes.</p> <p>Los casos de prueba se priorizarán utilizando los niveles de exposición al riesgo documentados en la actividad de identificación y análisis de riesgos (TP3).</p> <p>Los casos de prueba se registrarán en la especificación de casos de prueba.</p> <p>Se registrará la trazabilidad entre la base de la prueba, el modelo de prueba, los elementos de cobertura de la prueba y los casos de prueba.</p> <p>Las partes interesadas aprobarán la especificación de los casos de prueba.</p>
Crear procedimientos de prueba	<p>Los procedimientos de ensayo se derivarán ordenando los casos de ensayo según las dependencias descritas por las condiciones previas y posteriores y otros requisitos de ensayo.</p> <p>Se identificarán los datos de prueba y los requisitos del entorno de prueba que no estén ya incluidos en el plan de pruebas.</p> <p>Los procedimientos de prueba se priorizarán utilizando los niveles de exposición al riesgo documentados en la actividad de identificación y análisis de riesgos (TP3).</p> <p>Los procedimientos de prueba se registrarán en la especificación del procedimiento de prueba.</p> <p>Se registrará la trazabilidad entre la base de la prueba, el modelo de prueba, los elementos de cobertura de la prueba, los casos de prueba y los</p>

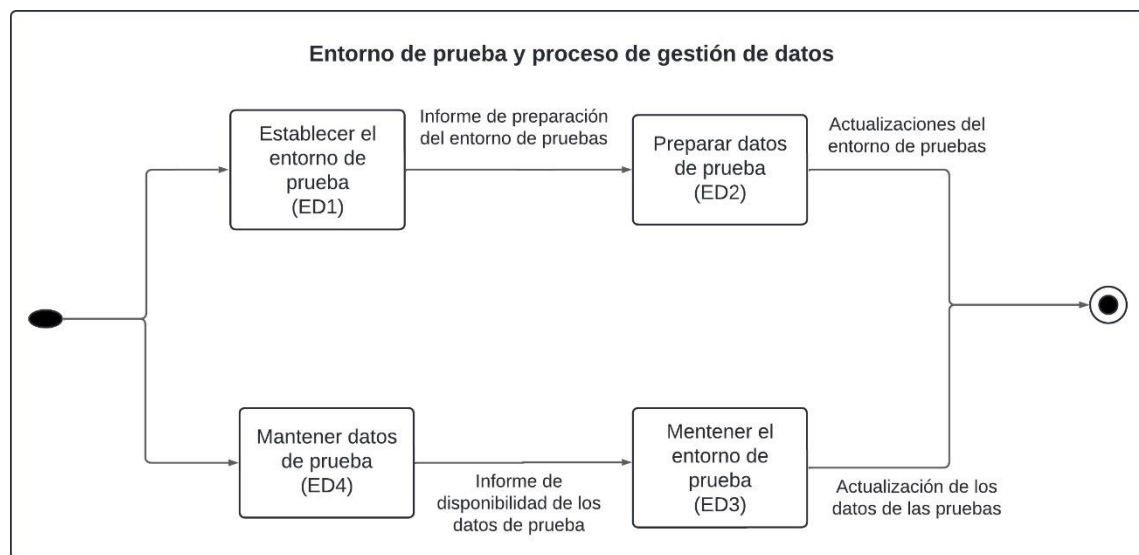
procedimientos de prueba (y/o los scripts de prueba automatizados).

La especificación del procedimiento de prueba deberá ser aprobada por las partes interesadas.

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

El entorno de prueba y el proceso de gestión de datos, que se muestra en la figura 13, se utiliza para establecer y mantener el entorno en el que se ejecutan las pruebas y gestionar los datos de prueba correspondientes.

Figura 13 Entorno de prueba y proceso de gestión de datos



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

A continuación, en la tabla 14, se muestra las tareas correspondientes a las actividades del proceso de entorno de prueba y proceso de gestión de datos.

Tabla 14 Tareas del proceso de entorno de prueba y proceso de gestión de datos

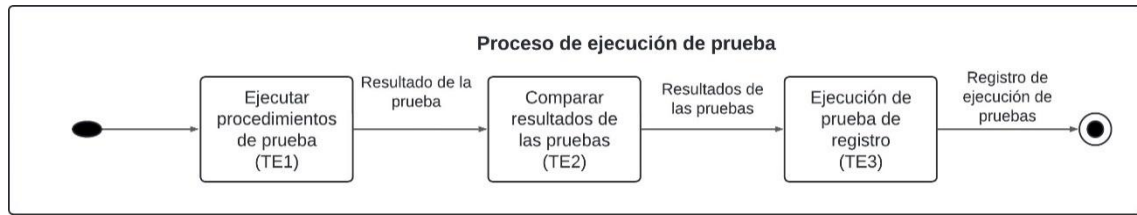
Proceso de prueba dinámicos	Actividades	Tareas
Entorno de prueba y proceso de gestión de datos	Establecer el entorno de pruebas	Basándose en el plan de pruebas, los requisitos detallados generados como resultado del proceso de diseño e implementación de las pruebas, los requisitos de las herramientas de prueba y la escala/formalidad de las pruebas, se realizará lo siguiente: 1) planificar la configuración del entorno de pruebas 2) diseñar el entorno de pruebas

	<p>3) determinar el grado de gestión de la configuración que debe aplicarse (si procede)</p> <p>4) construir el entorno de pruebas</p> <p>5) configurar las herramientas de prueba para apoyar las pruebas (si procede)</p> <p>6) instalar y configurar el elemento de prueba en el entorno de prueba</p> <p>7) verificar que el entorno de pruebas cumple los requisitos del entorno de pruebas</p> <p>8) en su caso, garantizar que el entorno de pruebas cumple los requisitos definidos.</p> <p>El estado del entorno de pruebas se registrará y comunicará a través del informe de preparación del entorno de pruebas a las partes interesadas pertinentes.</p> <p>El informe de preparación del entorno de pruebas incluirá una descripción de las diferencias conocidas entre</p>
Preparar datos de prueba	<p>Sobre la base del plan de pruebas, los requisitos detallados generados como resultado del proceso de diseño y ejecución de las pruebas, y la escala/formalidad de las pruebas, se llevará a cabo lo siguiente</p> <p>1) planificar la preparación de los datos de prueba</p> <p>2) preparar los datos de prueba</p> <p>3) configurar los datos de prueba para respaldar las pruebas</p> <p>4) verificar que los datos de prueba cumplen los requisitos de los datos de prueba</p> <p>5) en caso necesario, garantizar que los datos de prueba cumplen los requisitos definidos.</p> <p>El estado de los datos de prueba se registrará y comunicará a través del informe de preparación de datos de prueba a las partes interesadas pertinentes.</p>
Mantener el entorno de prueba	<p>El entorno de pruebas se mantendrá tal y como se define en los requisitos del entorno de pruebas.</p> <p>Los cambios en el estado del entorno de pruebas se comunicarán a las partes interesadas pertinentes.</p>
Mantener los datos de prueba	<p>Los datos de la prueba se mantendrán según lo definido por los requisitos de los datos de la prueba.</p> <p>Los cambios en el estado de los datos de ensayo se comunicarán a las partes interesadas pertinentes.</p>

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

El proceso de ejecución de pruebas, que se muestra en la figura 14, se utiliza para ejecutar los procedimientos de prueba generados como resultado del proceso de diseño e implementación de pruebas en el entorno de pruebas establecido por el entorno de pruebas y el proceso de gestión de datos.

Figura 14 Proceso de ejecución de pruebas



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

La tabla 15 muestra a detalle las tareas asignadas para el proceso de ejecución de prueba

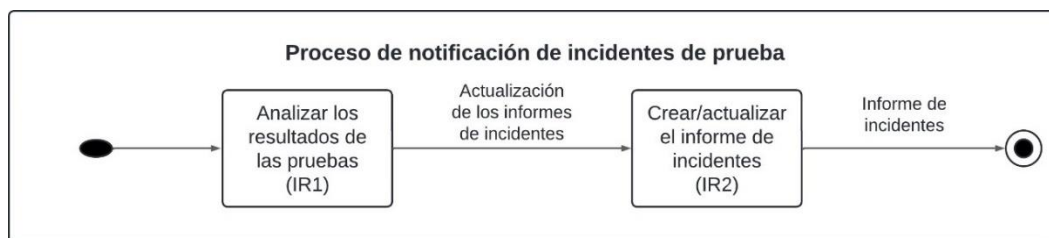
Tabla 15 Tareas del proceso de ejecución de prueba

Proceso de prueba dinámicos	Actividades	Tareas
Proceso de ejecución de prueba	<i>Ejecutar procedimientos de prueba</i>	Se ejecutarán uno o varios procedimientos de ensayo en el entorno de ensayo preparado. Se observarán los resultados reales de cada caso de prueba del procedimiento de prueba. Se registrarán los resultados reales.
	<i>Comparar los resultados de las pruebas</i>	Se compararán los resultados reales y esperados de cada caso de prueba del procedimiento de prueba. Se determinará el resultado de la ejecución de los casos de prueba del procedimiento de prueba. Si se supera una nueva prueba, será necesario actualizar un informe de incidencias mediante el proceso de notificación de incidencias de pruebas.
	<i>Ejecución de prueba de registro</i>	Se registrarán los detalles de la ejecución de las pruebas, tal como se especifica en el plan de pruebas.

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

El proceso de notificación de incidentes de prueba, que se muestra en la figura 15, se utiliza para informar a las partes interesadas incidentes de prueba que requieren acciones adicionales identificados como resultado de la ejecución de prueba.

Figura 15 Proceso de notificación de incidentes de prueba



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

La tabla 16 detalla las tareas correspondientes a cada una de las actividades del proceso de notificación de incidentes de prueba.

Tabla 16 Tareas del proceso de notificación de incidentes de prueba

Proceso de prueba dinámicos	Actividades	Tareas
Proceso de notificación de incidentes de prueba	Analizar los resultados de la prueba	Cuando un resultado de la prueba se refiera a un incidente planteado anteriormente, se analizará el resultado de la prueba y se actualizarán los detalles del incidente.
		Cuando el resultado de una prueba indique que se ha detectado un nuevo problema, se analizará el resultado de la prueba y se determinará si se trata de un incidente que requiere notificación, un elemento de acción que se resolverá sin necesidad de notificar el incidente, o si no requiere la adopción de ninguna otra medida.
	Crear/actualizar informe de incidentes	Los elementos de acción se asignarán a una persona adecuada para su resolución. Se identificará y comunicará/actualizará la información que debe registrarse sobre el incidente. El estado de los incidentes nuevos y/o actualizados se comunicará a las partes interesadas pertinentes.

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 2: Procesos de Prueba,” 2021)

1.4.3. ISO/IEC/IEEE 29119 – 3: Documentación de pruebas

La parte 3 proporciona plantillas para los diferentes tipos de documentación de pruebas de software los cuales se organiza en tres categorías principales detallados en la tabla 17:

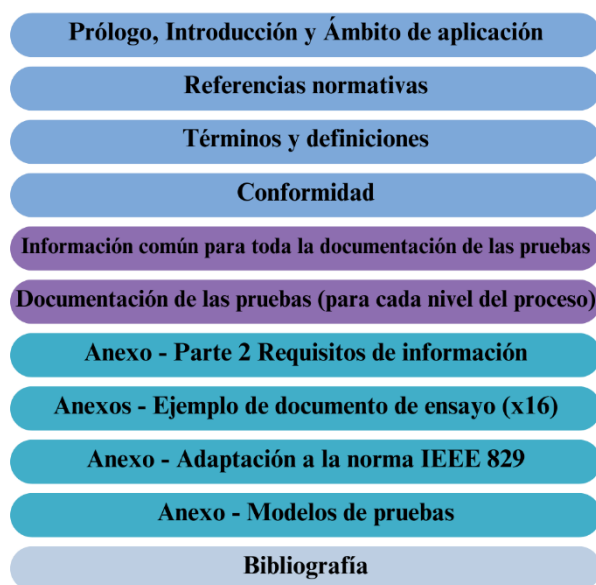
Tabla 17 Documentación de pruebas

Documentación de pruebas	
Documentación de pruebas organizativa	Incluye la política de pruebas y las prácticas organizativas de pruebas.
Documentación de gestión de pruebas	Comprende el plan de pruebas, informes de estado de las pruebas e informes de finalización de las pruebas
Documentación de pruebas dinámicas	Incluye especificaciones del modelo de prueba, especificaciones de los casos de prueba, especificaciones de los procedimientos de prueba, requisitos de los datos de prueba e informe de disponibilidad, requisitos del entorno de prueba e informe de disponibilidad, resultados de las pruebas, registro de ejecución de las pruebas e informe de incidentes.

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 3: Documentación de Pruebas,” 2021)

La figura 16 muestra la estructura de la parte 3, la cual contiene anexos que incluyen 32 documentos de ejemplo, abarcando 16 tipos de documentos de prueba diferentes, adecuados tanto para proyectos ágiles como tradicionales. Esta sección está estrechamente vinculada con la Parte 2 (Procesos de pruebas), ya que los resultados de los procesos definidos en la parte 2 corresponden a la documentación definida en la parte 3. El primer anexo refleja esta relación al enumerar los requisitos de información derivados de la parte 2, indicando qué información debe documentarse para cumplir con las normas.

Figura 16 ISO/IEC/IEEE 29119 Parte 3: Documentación



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 3: Documentación de Pruebas,” 2021)

1.4.4. ISO/IEC/IEEE 29119 – 4: Técnicas de pruebas

La parte 4 se centra en las técnicas de diseño de prueba y las medidas de cobertura correspondientes. La parte 2 de la norma requiere que los usuarios desarrollen estrategias de prueba que especifiquen las técnicas de diseño de casos de prueba y los criterios de realización de pruebas a alcanzar y en la parte 4 define una amplia gama de

técnicas de prueba proporcionando ejemplos de su aplicación en los anexos. (“ISO/IEC/IEEE 29119 - Parte 4: Técnicas de Prueba,” 2021)

A continuación, en las tablas 18,19 y 20, se detallan una lista de técnicas de diseño de prueba que se incluyen en la parte 4 del estándar.

Tabla 18 Técnicas de Pruebas Basadas en Especificaciones

Técnicas de Pruebas Basadas en Especificaciones	
Partición de equivalencias	Consiste en dividir los datos de entrada de un sistema en grupos o clases que se espera que el sistema trate de manera similar. En lugar de probar cada dato individualmente, se selecciona un valor representativo de cada grupo, porque se asume que, si uno de los datos en ese grupo pasa la prueba, los demás también lo harán.
Método del árbol de clasificación	Se utiliza para visualizar y organizar las diferentes combinaciones de entradas y decisiones que puede tomar un sistema. La idea básica es representar las condiciones de prueba en forma de un árbol, donde cada "rama" representa una decisión o un resultado posible. Al final de cada rama, se puede definir qué caso de prueba corresponde a esa combinación específica de entradas.
Análisis de valores límite	Se centra en verificar cómo se comporta un sistema en los extremos de los rangos de entrada. La idea es que los errores a menudo ocurren en los límites, por lo que es importante probar no solo los valores dentro del rango permitido, sino también los que están en los bordes y justo fuera de él.
Pruebas de sintaxis	Se centra en verificar que los datos ingresados en un sistema cumplen con las reglas y formatos establecidos. Esta técnica se utiliza para asegurarse de que la entrada de datos es correcta desde un punto de vista estructural
Técnicas de diseño de pruebas combinatorias	<p>Todas las combinaciones probadas Esta técnica implica crear casos de prueba para cubrir todas las posibles combinaciones de entradas en un sistema. Aunque proporciona una cobertura exhaustiva, puede generar un número muy elevado de pruebas, lo que puede ser poco práctico en sistemas complejos.</p> <p>Obtener casos de prueba Consiste en derivar casos de prueba a partir de un conjunto de condiciones o parámetros. Se seleccionan combinaciones de entradas relevantes para garantizar que se prueben los diferentes aspectos del sistema, buscando un equilibrio entre la cobertura y el esfuerzo requerido.</p> <p>Pruebas por pares En esta técnica, se prueban todas las combinaciones posibles de dos parámetros a la vez. Por ejemplo, si tienes dos variables (A y B), pruebas cada combinación de A1 con B1, A1 con B2, y así sucesivamente. Esto es eficiente,</p>

	<p>ya que se ha demostrado que la mayoría de los defectos surgen de la interacción de solo dos factores.</p> <p>Prueba de cada opción Esta técnica implica asegurarse de que cada opción o valor posible de un parámetro se pruebe al menos una vez. Es útil para verificar que cada entrada en un sistema funciona correctamente por sí sola, garantizando que todas las características individuales se han validado.</p> <p>Prueba de elección de base En esta técnica, se seleccionan las entradas más representativas de un grupo de valores posibles para crear casos de prueba. Se enfoca en elegir valores típicos, extremos y algunos casos especiales para asegurar una buena cobertura sin necesidad de probar todas las combinaciones.</p>
Prueba de tabla de decisiones	<p>Se utiliza para organizar y representar las condiciones y acciones de un sistema de manera clara y estructurada. Consiste en crear una tabla que muestra todas las combinaciones posibles de condiciones (o entradas) y las acciones correspondientes que deben realizarse en función de esas condiciones.</p>
Gráficos de causa – efecto	<p>Son una técnica de prueba utilizada para identificar y visualizar la relación entre diferentes factores (causas) y los resultados que producen (efectos) en un sistema. Esta técnica ayuda a entender cómo ciertos cambios o entradas en el sistema pueden influir en su comportamiento.</p> <p>En un gráfico de causa-efecto, las causas se representan como entradas o condiciones que pueden afectar el resultado, mientras que los efectos son las salidas o resultados esperados del sistema. Al crear este gráfico, se puede ver de manera clara cómo interactúan las diferentes causas para generar ciertos efectos, lo que facilita la identificación de qué combinaciones deben ser probadas.</p>
Pruebas de transición de estado	<p>Se centra en verificar el comportamiento de un sistema al cambiar de un estado a otro. Esta técnica es especialmente útil en sistemas donde el comportamiento varía según el estado actual del sistema, como aplicaciones que tienen diferentes modos de operación.</p> <p>En este tipo de pruebas, se identifican los distintos estados en los que puede estar el sistema y las transiciones que pueden ocurrir entre ellos. Luego, se crean casos de prueba para verificar que el sistema responde correctamente a las acciones que provocan estas transiciones.</p>
Prueba de escenario	<p>Se utiliza para evaluar el comportamiento de un sistema en situaciones específicas que imitan el uso real por parte de los usuarios. En lugar de centrarse en pruebas aisladas de funciones individuales, las pruebas de escenario se enfocan en cómo interactúan diferentes componentes del sistema en un contexto más amplio.</p> <p>En una prueba de escenario, se crean casos que representan flujos de trabajo o situaciones típicas que un</p>

	<p>usuario podría encontrar al utilizar la aplicación. Por ejemplo, en un sistema de compras en línea, un escenario podría incluir un usuario que busca un producto, lo agrega al carrito, completa la compra y recibe una confirmación.</p>
Pruebas aleatorias	<p>Implica la selección de datos o casos de prueba de manera aleatoria, en lugar de seguir un patrón específico o predeterminado. Esta técnica se utiliza para simular un uso real del sistema y descubrir errores que podrían no ser evidentes en pruebas planificadas.</p> <p>En las pruebas aleatorias, los testers ingresan datos o realizan acciones sin un enfoque estructurado, lo que permite explorar el sistema de manera más amplia y espontánea. Esto puede ayudar a identificar problemas inesperados, fallos de rendimiento o comportamientos inusuales que podrían surgir en situaciones no anticipadas.</p>
Pruebas metamórficas	<p>Diseñada para verificar el comportamiento de un sistema cuando no se conocen los resultados esperados. En lugar de comparar la salida del sistema con un resultado específico, esta técnica se basa en propiedades conocidas que deben mantenerse a través de diferentes entradas.</p> <p>La idea es identificar relaciones o transformaciones entre los datos de entrada y las salidas, donde se espera que, al modificar los datos de una manera particular, la salida también cambie de acuerdo con esas reglas. Por ejemplo, si se está probando una función de suma, si se suma un número a un conjunto de datos y luego se elimina el mismo número, se espera que el resultado sea el mismo que el original.</p>
Pruebas basadas en requisitos	<p>Se enfoca en verificar que el software cumple con los requisitos especificados. En este enfoque, los casos de prueba se derivan directamente de los requisitos funcionales y no funcionales del sistema.</p> <p>La idea es asegurarse de que cada requisito tenga al menos un caso de prueba asociado que valide su implementación. Esto implica analizar los documentos de requisitos y crear pruebas que demuestren que el software se comporta de la manera esperada según esas especificaciones.</p>

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 4: Técnicas de Prueba,” 2021)

Tabla 19 Técnicas de Pruebas Basadas en Estructuras

Técnicas de Pruebas Basadas en Estructuras	
Prueba de declaración	Se centra en verificar que cada línea de código en un programa se ejecute al menos una vez durante las pruebas. Este enfoque se utiliza para asegurarse de que todas las partes del código estén operativas y que no

	<p>haya segmentos no utilizados que puedan contener errores.</p> <p>En esta técnica, los testers diseñan casos de prueba de manera que cada declaración o línea de código sea ejecutada. Esto ayuda a identificar problemas potenciales que pueden no ser evidentes si ciertas partes del código nunca se ejecutan.</p>
Prueba de rama	<p>Se centra en verificar que todas las decisiones lógicas en un programa se evalúan en todas sus posibles direcciones. Esto significa que cada "rama" de las estructuras de control del código, como los condicionales "if" o "switch", debe ser ejecutada al menos una vez durante las pruebas.</p> <p>El objetivo de la prueba de rama es asegurarse de que el software se comporta correctamente en todas las situaciones posibles que pueden surgir de estas decisiones.</p>
Prueba de decisión	<p>Se centra en evaluar la lógica de las decisiones dentro de un programa. Esta técnica se utiliza para garantizar que todas las decisiones, como las condiciones en estructuras "if" o "switch", se evalúen correctamente y que el software tome las acciones adecuadas según esas decisiones.</p> <p>En la prueba de decisión, los testers diseñan casos de prueba que cubren todas las posibles condiciones que podrían influir en el flujo del programa. Esto incluye tanto los casos en los que la condición es verdadera como aquellos en los que es falsa.</p>
Prueba de condición de rama	<p>Se enfoca en evaluar las condiciones individuales dentro de las decisiones lógicas en un programa. A diferencia de la prueba de rama, que se centra en las decisiones completas, esta técnica analiza cada condición que puede influir en una decisión para asegurarse de que se evalúe correctamente, tanto cuando son verdaderas como cuando son falsas.</p>
Prueba de combinación de condiciones de rama	<p>Se enfoca en verificar cómo interactúan múltiples condiciones dentro de las decisiones lógicas en un programa. A diferencia de otras técnicas que analizan condiciones individuales o decisiones completas, esta técnica examina todas las combinaciones posibles de condiciones para asegurarse de que el sistema responda correctamente a cada una de ellas.</p>
Pruebas de cobertura de decisión/condición modificada MSDC	<p>Son una técnica de prueba que combina dos enfoques: la cobertura de decisiones y la cobertura de condiciones. Esta técnica se utiliza para garantizar que tanto las decisiones lógicas como las condiciones</p>

individuales en un programa se evalúen correctamente en todas sus variantes.

El objetivo de las pruebas MSDC es asegurar que cada decisión en el código se evalúe en sus dos posibles resultados (verdadero y falso) y que cada condición dentro de esas decisiones también se evalúe tanto en su estado verdadero como en su estado falso. Sin embargo, a diferencia de la cobertura completa, esta técnica permite que algunas condiciones no se evalúen en cada caso, siempre que se verifique su efectividad en al menos una evaluación.

Pruebas de flujos de datos

Pruebas de todas las definiciones

Esta técnica se asegura de que todas las definiciones de variables se evalúen en el programa. Esto implica identificar y probar cada punto en el que una variable se define o inicializa para garantizar que se haya establecido correctamente.

Pruebas de todos los usos - c (control)

En esta técnica, se verifican todos los usos de una variable donde se controla su flujo, como cuando se toma una decisión basada en su valor. Esto garantiza que las condiciones en las que se utiliza la variable se evalúen adecuadamente.

Pruebas de todos los usos - p (predicción)

Esta técnica se centra en todos los usos de una variable que afectan la predicción del resultado, es decir, donde la variable se utiliza para calcular o influir en una salida. Se busca asegurar que cada uso que impacta en el resultado final se pruebe adecuadamente.

Pruebas para todos los usos

Esta técnica implica probar cada uso de las variables, tanto en contextos de control como de predicción, asegurando que todas las interacciones y dependencias se verifiquen para garantizar que el sistema maneje correctamente los datos.

Comprobación de todas las rutas

En esta técnica, se examinan todas las rutas posibles que los datos pueden seguir a través del sistema. Esto significa identificar y probar cada camino que los datos pueden tomar desde su definición hasta su uso, asegurando que cada ruta se evalúe y funcione como se espera.

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 4: Técnicas de Prueba,” 2021)

Tabla 20 Técnicas de Pruebas basadas en Experiencia

Técnicas de Pruebas basadas en Experiencia	
Detección de errores	<p>Se centra en identificar defectos o fallos en el software a través de la ejecución de casos de prueba. Esta técnica implica realizar pruebas en el sistema para observar su comportamiento y verificar que funcione como se espera.</p> <p>El objetivo principal de la detección de errores es encontrar y documentar problemas en el software antes de que se implemente en un entorno real. Durante este proceso, los testers diseñan y ejecutan una variedad de casos de prueba, que pueden incluir pruebas funcionales, pruebas de rendimiento y pruebas de seguridad, entre otras.</p>

Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 4: Técnicas de Prueba,” 2021)

La figura 17 muestra la estructura básica de la Parte 4, al igual que en la Parte 3, los anexos con ejemplos de aplicación de las técnicas de cada una de las técnicas definidas en la norma.

Figura 17 ISO/IEC/IEEE 29119 Parte 4: Técnicas de prueba



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 4: Técnicas de Prueba,” 2021)

1.4.5. ISO/IEC/IEEE 29119 – 5: Pruebas dirigidas por palabras clave

La figura 18 muestra la estructura de la parte 5 la cual proporciona una guía detallada para el desarrollo de especificaciones, estructuras o automatizaciones de pruebas basadas en palabras clave. Los criterios establecidos en esta sección permiten a los profesionales intercambiar componentes de pruebas, incluyendo casos de prueba, datos de prueba y palabras clave, garantizando así la interoperabilidad entre diferentes sistemas y herramientas. Este enfoque está diseñado para ser accesible a cualquier persona interesada en desarrollar o utilizar pruebas basadas en palabras clave, proporcionando una base común para la creación y el intercambio de especificaciones de pruebas. (“ISO/IEC/IEEE 29119 - Parte 5: Pruebas Basadas En Palabras Clave,” 2016)

Figura 18 ISO/IEC/IEEE 29119 Parte 5: Pruebas dirigidas por palabras clave



Nota: Fuente: (“ISO/IEC/IEEE 29119 - Parte 5: Pruebas Basadas En Palabras Clave,” 2016)

1.5.Prácticas recomendadas por la ISO/IEC/IEEE 29119

Según (Reid, 2023) en su guía del profesional sobre las pruebas de software, se detallan prácticas clave que deben seguirse para cumplir con los estándares establecidos por la norma ISO 29119, detallados en la tabla 21:

Tabla 21 Prácticas recomendadas para cumplir con la ISO 29119

Prácticas recomendadas por la ISO 29119	
Planificación	Se debe definir claramente los objetivos y alcance de cada una de las pruebas y de la misma manera establecer roles y responsabilidades del equipo de pruebas. Además, es importante estimar el esfuerzo necesario en función de la complejidad del sistema y los recursos disponibles, así como planificar la disponibilidad de los entornos de prueba y las herramientas necesarias para su correcta ejecución.
Diseño de pruebas	En esta práctica se debe seguir un enfoque basado en riesgos, lo que significa que las pruebas deben priorizarse de acuerdo con la criticidad de las funcionalidades y los riesgos identificados. Asimismo, se deben establecer criterios claros de entrada y salida para cada fase del proceso de pruebas. Los casos de prueba diseñados deben cubrir una amplia gama de escenarios, incluyendo tanto pruebas funcionales como no funcionales, de seguridad y de rendimiento, el presente estudio toma en cuenta únicamente las pruebas funcionales. Es fundamental que exista trazabilidad entre los casos de prueba y los requisitos del sistema, lo que asegura que todos los aspectos críticos del software han sido verificados.
Ejecución	Se recomienda que las pruebas se ejecuten en un entorno lo más cercano posible al de producción para obtener resultados más realistas. Durante esta fase, es importante registrar y gestionar todos los incidentes que se presenten. Además, la documentación detallada garantiza que las pruebas puedan repetirse si es necesario. En cuanto a la eficiencia, siempre que sea posible, se sugiere la automatización de pruebas para reducir el tiempo de ejecución y evitar errores humanos.
Seguimiento y control	Es una práctica que requiere un enfoque sistemático que abarque desde la planificación hasta el monitoreo y control de todas las actividades relacionadas, así como establecer las métricas de pruebas, como el número de defectos encontrados durante las pruebas, cobertura de pruebas, tiempo de ejecución, deben ser monitoreadas de cerca para asegurar el progreso y la calidad del proyecto. De igual manera, es fundamental revisar si la cobertura de los requisitos y los riesgos identificados ha sido suficiente antes de avanzar a la siguiente fase de desarrollo.
Documentación	Es una práctica crucial para asegurar la trazabilidad y para permitir una revisión clara de los resultados. Esto incluye desde la creación de casos de prueba hasta la documentación de incidentes y la elaboración de informes finales que incluyan defectos encontrados y recomendaciones para futuras fases. La calidad de esta documentación permite un análisis efectivo y facilita el mantenimiento del software.
Evaluación	Se debe realizar revisiones periódicas con las partes interesadas y aplicar lecciones aprendidas de proyectos anteriores para mejorar la eficiencia de futuras pruebas. Además, la implementación de herramientas de automatización y soporte, como las utilizadas en la integración y

despliegue continuo, es altamente recomendada para asegurar la calidad del software de manera continua y eficiente.

Nota: Fuente: (Reid, 2023)

1.6. Puesta en producción

La Puesta en Producción de Software es el proceso final en el ciclo de vida del desarrollo de software, donde una aplicación o sistema que ha completado todas las fases de desarrollo y pruebas es implementado en un entorno de producción. Este proceso asegura que el software sea instalado, accesible y disponible para su uso por los usuarios finales o clientes, garantizando su operatividad en un contexto real. (Pressman, 2010)

1.7. ISO/IEC/IEEE 20000-1:2018 Tecnología de la información - Gestión de servicios

Es una norma internacional que posee un marco de trabajo en donde las organizaciones pueden planificar, establecer, implementar, operar, monitorear, revisar, mantener y mejorar un sistema de gestión de servicios (SGS) de TI, cuyos requisitos están diseñados para ser aplicados por prestadores de servicios de TI de todos los tipos, tamaños y naturaleza de los servicios que ofrecen, con la finalidad de asegurar procesos estandarizados y eficientes y con ello garantizar la calidad y la continuidad del servicio. (“ISO/IEC 20000-1: - Gestión Del Servicio: Requisitos,” 2018)

1.7.1. Gestión de la configuración

La gestión de la configuración implica la creación de una definición documentada para cada tipo de elemento de configuración (CI, por sus siglas en inglés). Esta documentación debe incluir detalles como la descripción del CI, sus relaciones con otros CI y componentes del servicio, su ubicación, la versión, y cualquier petición de cambio, problemas o errores conocidos asociados. Este nivel de detalle asegura que cada CI esté claramente definido y comprendido dentro del sistema. Para garantizar una adecuada

gestión de los CI, es fundamental utilizar una Base de Datos de Gestión de la Configuración (CMDB). Todos los CI deben identificarse de manera única y registrarse en la CMDB, la cual debe ser gestionada de manera efectiva para asegurar la precisión de los datos y el control de acceso. La correcta gestión de la CMDB es esencial para mantener la integridad de los datos, asegurando que la información sea fiable y que los accesos estén debidamente controlados y actualizados. (*Norma ISO/IEC 20000-1, n.d.*)

Además, es necesario contar con procedimientos documentados que cubran el registro, control y seguimiento de las versiones de los CI. Estos procedimientos ayudan a mantener la integridad de los servicios y a gestionar los riesgos asociados con los CI. La existencia de estos procedimientos permite una supervisión continua y garantiza que los cambios en los CI sean controlados y gestionados adecuadamente. (*Norma ISO/IEC 20000-1, n.d.*)

La auditoría y el monitoreo son también componentes clave de la gestión de la configuración. Se requiere que los proveedores de servicios auditen regularmente los registros almacenados en la CMDB y tomen las acciones necesarias cuando se identifican deficiencias. Este proceso de auditoría asegura que se mantenga la precisión y confiabilidad de la información contenida en la CMDB. (*Norma ISO/IEC 20000-1, n.d.*)

La información almacenada en la CMDB juega un papel crucial en el proceso de gestión de cambios. Se utiliza para apoyar la evaluación de las peticiones de cambio, asegurando que cualquier modificación en los CI sea cuidadosamente considerada y evaluada antes de su implementación. Esto garantiza que los cambios en el entorno de TI se gestionen de manera ordenada y controlada. Por último, se destaca la importancia de mantener la trazabilidad y la capacidad de auditoría de los cambios realizados en los CI. Esto

garantiza la integridad de los servicios y la precisión de los datos en la CMDB. También se enfatiza la necesidad de almacenar copias maestras de los registros en un lugar seguro, garantizando la accesibilidad y seguridad de la documentación relevante. El texto aclara que el alcance del proceso de gestión de la configuración no incluye la gestión de activos financieros, delimitando claramente el enfoque del proceso. (*Norma ISO/IEC 20000-1*, n.d.)

1.7.2. Gestión de control de cambios

La gestión de cambios se enfoca en la importancia de establecer una política clara y documentada que defina los elementos bajo control y los criterios para determinar los cambios que pueden tener un impacto significativo en los servicios o en los clientes. Este proceso abarca la clasificación, evaluación y aprobación de las solicitudes de cambio, asegurando que todas las modificaciones se manejen de manera estructurada y controlada. (“ISO/IEC 20000-1: - Gestión Del Servicio: Requisitos,” 2018)

En la gestión de cambios, se considera crucial que cualquier eliminación o transferencia de servicios sea clasificada adecuadamente para evaluar su impacto. Se requiere un procedimiento documentado que guíe el registro, la clasificación y el control de los cambios, lo que incluye tanto cambios normales como de emergencia. La colaboración entre el proveedor de servicios y el cliente es esencial para acordar y definir qué constituye un cambio de emergencia y para asegurar que todas las solicitudes de cambio estén bien documentadas y gestionadas. (*Norma ISO/IEC 20000-1*, n.d.)

El proceso de gestión de cambios implica que todas las peticiones deben ser trazables y auditables, con un alcance definido y una clasificación adecuada. Los cambios que tienen un impacto significativo deben gestionarse a través de procesos más rigurosos de diseño y transición de servicios, mientras que otros cambios pueden gestionarse a través

de procesos más estándar. Es fundamental que los cambios se evalúen utilizando información obtenida de la gestión de la configuración y otros procesos relevantes, para asegurar que los riesgos asociados con los cambios se manejen adecuadamente. (*Norma ISO/IEC 20000-1, n.d.*)

La toma de decisiones sobre la aprobación de cambios debe considerar no solo los riesgos, sino también los impactos potenciales en los servicios, las necesidades del cliente y la viabilidad técnica y financiera. Después de aprobar los cambios, es importante que estos sean probados y que se planifique y comunique el despliegue de manera efectiva a todas las partes interesadas. Además, es esencial prever y planificar actividades para revertir o corregir cambios si estos no resultan ser satisfactorios. (*Norma ISO/IEC 20000-1, n.d.*)

Finalmente, una vez que los cambios se despliegan, los registros en la CMDB deben actualizarse para reflejar el nuevo estado de los elementos de configuración. El proveedor de servicios tiene la responsabilidad de revisar la efectividad de los cambios y tomar las medidas necesarias para corregir cualquier problema que surja. Para mejorar continuamente el proceso, se recomienda analizar las tendencias en las peticiones de cambio a intervalos regulares, registrando los resultados y utilizando esta información para identificar oportunidades de mejora. (*Norma ISO/IEC 20000-1, n.d.*)

1.7.3. Gestión de entrega y despliegue

El proceso de gestión de la entrega y despliegue es crucial en el ciclo de vida del servicio, ya que asegura que los servicios o componentes de software nuevos o modificados se implementen en el entorno de producción de manera controlada y efectiva. Este proceso implica la colaboración estrecha entre el proveedor del servicio y el cliente para establecer una política de entrega que defina la frecuencia, los tipos de

entrega y los métodos de implementación adecuados. La planificación de las entregas se coordina con el proceso de gestión de cambios, lo que permite identificar y mitigar cualquier problema o error que pueda surgir durante la implementación. (“ISO/IEC 20000-1: - Gestión Del Servicio: Requisitos,” 2018)

Uno de los aspectos clave es la necesidad de documentar y acordar con el cliente los procedimientos para la gestión de entregas, incluidas las entregas de emergencia que deben manejarse a través de un procedimiento específico. Para garantizar que las entregas cumplan con los estándares acordados, se utilizan entornos controlados de aceptación y pruebas antes del despliegue final. Estos entornos permiten verificar que todos los criterios de aceptación se cumplen, caso contrario, se toman las medidas necesarias para corregir cualquier deficiencia. (*Norma ISO/IEC 20000-1, n.d.*)

Además, el despliegue en el entorno de producción debe garantizar que la integridad del hardware, software y otros componentes del servicio se mantenga intacta durante todo el proceso. Es esencial planificar y probar posibles actividades de reversión en caso de que una entrega falle o necesite repararse. El éxito o fracaso de cada entrega se monitorea rigurosamente, incluyendo el análisis de incidentes relacionados y el impacto en el cliente. Los resultados y conclusiones de estos análisis se documentan para identificar áreas de mejora y optimizar futuros despliegues. (*Norma ISO/IEC 20000-1, n.d.*)

Finalmente, la gestión de la entrega y despliegue no solo se enfoca en la implementación técnica, sino que también proporciona información valiosa al proceso de gestión de cambios, apoyando la evaluación de los impactos de las peticiones de cambio en los planes de despliegue y en la entrega misma. Esto asegura que el servicio entregado

cumpla con las expectativas y requisitos tanto del cliente como del proveedor. (*Norma ISO/IEC 20000-1, n.d.*)

CAPÍTULO 2

Diseño de la guía

2.1.Descripción del área de estudio

Misión

La Dirección de Desarrollo Tecnológico e Informático, gestiona el desarrollo de las TICs como herramientas de apoyo para fortalecer la formación de profesionales de excelencia, éticos, críticos, humanistas, líderes y emprendedores con responsabilidad social, procurando que la automatización mejore continuamente la calidad en los procesos de investigación, transferencia de saberes, de conocimientos científicos, tecnológicos y de innovación, procurando que sirva de soporte para alcanzar una efectiva vinculación con la comunidad, que sea plataforma para aplicar criterios de sustentabilidad y así contribuir al desarrollo social, económico, cultural, ecológico de la región y del país.

Visión

La Dirección de Desarrollo Tecnológico e Informático en el año 2025, tendrá automatizado los procesos definidos de la actividad universitario, utilizando criterios de industria 4.0, aplicando técnicas y herramientas de inteligencia de negocios, incrementando el nivel de utilización del software con que cuenta la UTN; además, se implementarán nuevos servicios tecnológicos y el inicio para la certificación de la ISO 29110 correspondiente al desarrollo de software, se ajustará permanentemente a las necesidades institucionales las bondades del SIIU, se proyecta subir todas las aplicaciones de misión crítica a CLOUD y saltar a nuevas implementaciones de infraestructura de red que permitan ampliar la cobertura de servicio en los campus de la Universidad.

Análisis FODA de la Dirección de Desarrollo Tecnológico e Informático se detalla en la siguiente tabla 22.

Tabla 22 FODA DDTI

Fortalezas	Oportunidades
<ul style="list-style-type: none"> • Se cuenta con el Sistema Integrado de Información Universitario, con sus componentes principales, debidamente probado y en funcionamiento, con altos niveles de seguridad, calidad, eficiencia y efectividad. • Predisposición e iniciativa del personal informático en actualizarse en las nuevas tecnologías emergentes informáticas, en el fomento de nuevos campos y componentes tecnológicos en el área de su competencia. • Implementación de tecnologías emergentes: AuraQuantic, Tableau, Quipux, Uniportal 3.0, portafolios, Aula Virtual, Repositorio Digital, office 365, etc. • Infraestructura de red de datos: Anillo de Fibra Óptica a 10 Gbps que enlaza todos los edificios del campus del Olivo, Telefonía IP y sistema E1, Red Inalámbrica Wifi5 y Wifi 6 en todo el campus del Olivo, Internet y Red Avanzada 3Gb con CEDIA, Red con cableado categoría 6 y 6A, enlaces de Fibra Óptica en: Granja La Pradera, Colegio UTN, HSVP, Granja Yuyucocha, Planta Textil, Centro Infantil. • Setenta y siete módulos que conforman el Sistema de Información Integrado de Gestión Universitaria, proyectos con alto nivel de investigación e innovación (I+i) que le han posicionado a la universidad en el contexto nacional como una de las mejores y debidamente acreditadas • Personal con alto grado de pertinencia institucional. • Imagen positiva y eficiente en el área de influencia de la UTN. • Personal comprometido y efectivo en todas las áreas de trabajo, con capacidad técnica y experiencia profesional. • Capacidad de innovación e integración de los sistemas de información. • Apoyo de autoridades para el desarrollo de la gestión de TICs en la UTN 	<ul style="list-style-type: none"> • Aplicación permanente de las TICs en los requerimientos que la educación superior y la comunidad relacionada lo necesiten. • Iniciativa para mejorar las relaciones de trabajo y prospección con entidades externas líderes en el desarrollo de TICS, proveedores de software y hardware, con el fin de estar a la vanguardia en el desarrollo tecnológico. • Se logre satisfacer la creciente demanda por servicios informáticos de todos los actores universitarios. • Disponibilidad de encontrar en el mercado tecnologías de punta. • El avance Tecnológico proporciona un abanico de posibilidades que pueden ser aplicadas en los procesos sistemáticos. • Dada la inminente necesidad, por crecimiento en cantidad y calidad, es necesario la implementación de un nuevo espacio físico para el DDTI. • Interés creciente por parte de los funcionarios del DDTI en capacitaciones sobre nuevas tecnologías. • Interés por parte de la Universidad Técnica del Norte, en consolidarse como una Institución de Educación Superior digital – Smart University. • Mantener mecanismos de participación democrática a través de nuestro portal Web. • Implementación de IPV6 en todo el Campus Universitario. • Implementación de Red Inalámbrica en todo el Campus Universitario. • Tecnología actualizada que permite la interconexión entre los diferentes sistemas institucionales. • Asignación de recursos financieros que permiten soportar el crecimiento institucional • Desarrollos de nuevas tecnologías y buenas prácticas de gestión enfocadas a los sistemas educativos.

Debilidades	Amenazas
<ul style="list-style-type: none"> • Ambientes reducidos y mal ubicados para la ejecución del trabajo del DDTI. • Ubicación del área física del DDTI, es inadecuada. • Pocos convenios y programas de capacitación continua al personal de esta unidad. • Escasos recursos económicos para disponer de una infraestructura informática acorde a las necesidades. • Insuficiente personal en nuevas áreas del conocimiento tecnológico. • Falta de incorporación de buenas prácticas y estándares de las TIC. • Falta de difusión del Catálogo de servicios de TI. 	<ul style="list-style-type: none"> • Que los procesos a automatizar no sean definidos por las unidades responsables de los mismos, de manera oportuna y precisa • Constante amenazas a la seguridad informática, falta personal especializado. • Elevados costos de hardware y software. • Falta de personal para atender el excesivo número de peticiones de servicio de los usuarios y actores universitarios. • Se recorte el presupuesto para la adquisición de equipos de cómputo y Licencias de software. • Retraso en la entrega de insumos y repuestos. • Resistencia de los funcionarios a utilizar las innovaciones que se realizan en los sistemas de información. • Procedimiento lento en las adquisiciones. • Riesgo de ocurrencia de desastres naturales. • Limitaciones gubernamentales en la asignación de recursos económicos.

Nota: Fuente: DDTI

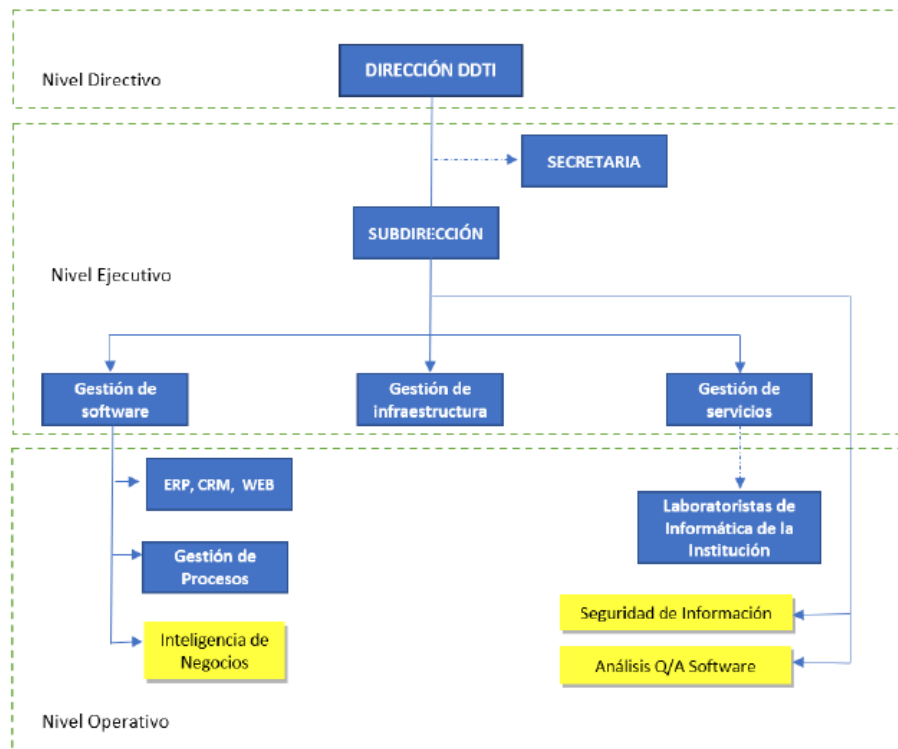
Organigrama interno del DDTI

Nivel Directivo: El nivel directivo establecido en la estructura organizacional es ejercido por el director, y es el que define políticas y lineamientos para la ejecución de las actividades que realiza el personal de la Dirección.

Nivel Ejecutivo: El nivel ejecutivo establecido en la estructura organizacional es ejercido por el subdirector y los responsables de las unidades, mismos que implementarán las políticas y lineamientos establecidos por la dirección, coordinarán y supervisarán el trabajo de su equipo.

Nivel Operativo: El nivel operativo establecido en la estructura organizacional, es ejercido por los grupos de trabajo de cada área los mismos que ejecutarán el trabajo conforme a los lineamientos establecidos desde la dirección.

Figura 19 Dirección de Desarrollo Tecnológico e Informático



Nota: Fuente: DDTI

2.2. Diseño y tipo de investigación

Para el desarrollo de este trabajo se realizó una investigación aplicada con enfoque exploratorio y descriptivo. A continuación, se explica en detalle por qué este enfoque es el más adecuado:

- La investigación aplicada se enfoca en la resolución de problemas prácticos específicos. En este caso, el problema identificado es la falta de una guía formal para la ejecución de pruebas y puesta en producción de software en el DDTI y como resultado de esta investigación se tendrá una aplicación directa en la mejora de los procesos de desarrollo y calidad del software en el DDTI.
- El enfoque exploratorio es adecuado porque inicialmente se necesita investigar y entender a fondo la situación actual del DDTI, incluyendo sus prácticas actuales de pruebas de software, las tecnologías utilizadas y las necesidades específicas de

los desarrolladores. Además, este enfoque permitirá entender las causas del problema identificado.

- El enfoque descriptivo permitirá documentar y caracterizar detalladamente las prácticas y procesos actuales del DDTI, así como describir y medir el impacto de la guía metodológica en términos de eficiencia, calidad y satisfacción del usuario, proporcionando una visión clara del antes y después de su implementación.

2.2.1. Metodología de investigación

Para la elaboración de este trabajo se siguió la siguiente estructura:

1. Recolección de datos

- Entrevistas y Encuestas: se realizó entrevistas y encuestas con los desarrolladores y usuarios finales para recolectar datos cualitativos y cuantitativos sobre las prácticas actuales y las expectativas respecto a la guía metodológica.
- Revisión Documental: se analizó la documentación existente sobre las prácticas de desarrollo y pruebas de software en el DDTI.

2. Análisis de datos

- Análisis Cualitativo: se analizó los datos cualitativos obtenidos de las entrevistas y observaciones para identificar patrones y temas recurrentes.
- Análisis Cuantitativo: se utilizó métodos estadísticos para analizar los datos cuantitativos recolectados mediante encuestas y registros de pruebas, para evaluar la efectividad de la Guía Metodológica de Calidad de Software, identificando áreas de mejora y facilitando la toma de decisiones basada en datos.

3. Diseño y aplicación de la guía

- Desarrollo de la Guía: Basándose en el análisis de datos, se diseñó la guía metodológica adaptada a las necesidades del DDTI.
- Se aplicó la guía describiendo los resultados de los hallazgos de cómo funcionan los submódulos, portafolio estudiantil y portafolio docente, para validar su efectividad antes de una implementación a gran escala.

4. Evaluación y mejora

- Evaluación de Resultados: se comparó los resultados obtenidos antes y después de la aplicación de hallazgos, utilizando los indicadores definidos.
- Ajustes y Mejoras: se realizó ajustes a la guía metodológica basada en los resultados de la evaluación y el feedback recibido, recomendando que se continúe con esta actividad periódicamente.

2.3. Desarrollo de la investigación

2.3.1. Área de desarrollo del DDTI

La investigación se realiza con la participación de: director, subdirector, personal del área de desarrollo de la Dirección de Desarrollo Tecnológico e Informático (DDTI), definiendo la población de la siguiente manera:

Tabla 23 Personal del DDTI

Población	Frecuencia	Porcentaje
Director DDTI	1	7.69%
Subdirector DDTI	1	7.69%
Desarrolladores	11	84.62%
Total	13	100%

Nota: Fuente: Elaboración propia.

2.3.2. Instrumentos

Para la recolección de la información se utilizarán los siguientes instrumentos:

- Cuestionario de encuesta para el Área de Desarrollo
- Cuestionario de entrevista para subdirector del DDTI.
- Cuestionario de encuesta para los usuarios finales (docentes y estudiantes)

2.3.3. Definición de variables

Tabla 24 Variable independiente: *Guía Metodológica de Calidad de Software*

Variable	Conceptualización	Dimensiones	Indicadores	Técnicas e instrumentos
Guía Metodológica de Calidad de Software	La Guía Metodológica de Calidad de Software se refiere a un conjunto de procedimientos, estándares y mejores prácticas documentadas que orientan la ejecución de pruebas de software y la puesta en producción. Basada en la norma ISO/IEC/IEEE 29119, esta guía busca mejorar la calidad del software desarrollado mediante procesos sistemáticos y estructurados.	Aplicación de la norma ISO/IEC/IEEE 29119	Número de procesos de prueba documentados según la norma	Revisión documental
		Diseño de casos de prueba	Número de casos de prueba definidos, cobertura de los casos de prueba	Entrevista
		Ejecución de pruebas	Frecuencia de ejecución de pruebas, tiempo promedio de ejecución de pruebas.	Revisión de reportes de pruebas
		Documentación y seguimiento	Precisión de la documentación de pruebas basado en el número de defectos registrados y su seguimiento, Calidad de los reportes de pruebas	Revisión de documentación Encuesta

Nota: Fuente: Elaboración propia

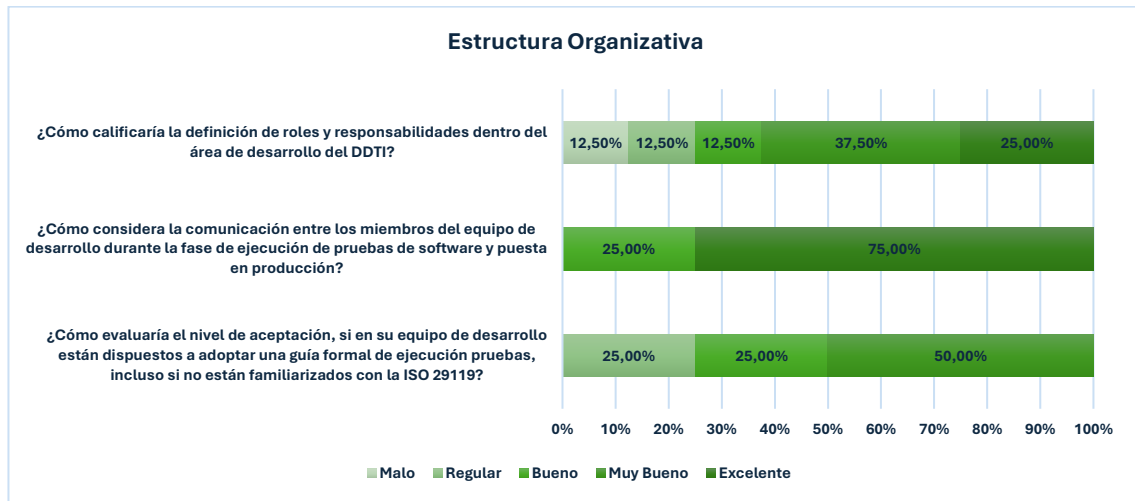
Tabla 25 Variable dependiente: Calidad del Software Desarrollado por el DDTI

Variable	Conceptualización	Dimensiones	Indicadores	Técnicas e instrumentos
Calidad del Software Desarrollado por el DDTI	La Calidad del Software Desarrollado por el DDTI se refiere al grado en que el software cumple con los requisitos funcionales y no funcionales, es libre de defectos, es eficiente, y satisface las expectativas de los usuarios finales (docentes y estudiantes). Esta calidad se mide en términos de eficiencia del proceso de pruebas, calidad del producto final, satisfacción del usuario y productividad del equipo de desarrollo.	Eficiencia del proceso de pruebas	Tiempo de ejecución de pruebas, tasa de detección de defectos.	Revisión de reportes de pruebas
		Calidad el producto final	Severidad de defectos. Clasificación de los defectos según su severidad (altos, medios, bajos).	Revisión de informes de defectos
		Satisfacción de usuario	Índice de satisfacción de usuario, número de reclamaciones.	Encuesta con usuarios finales
		Productividad del equipo de desarrollo	Tasa de adopción de la guía, número de proyectos completados a tiempo.	Revisión de cronogramas de proyectos

Nota: Fuente: Elaboración propia

2.3.4. Resultados de la encuesta realizada al área de desarrollo

Figura 20 Gráfico de barras apiladas – Estructura Organizativa

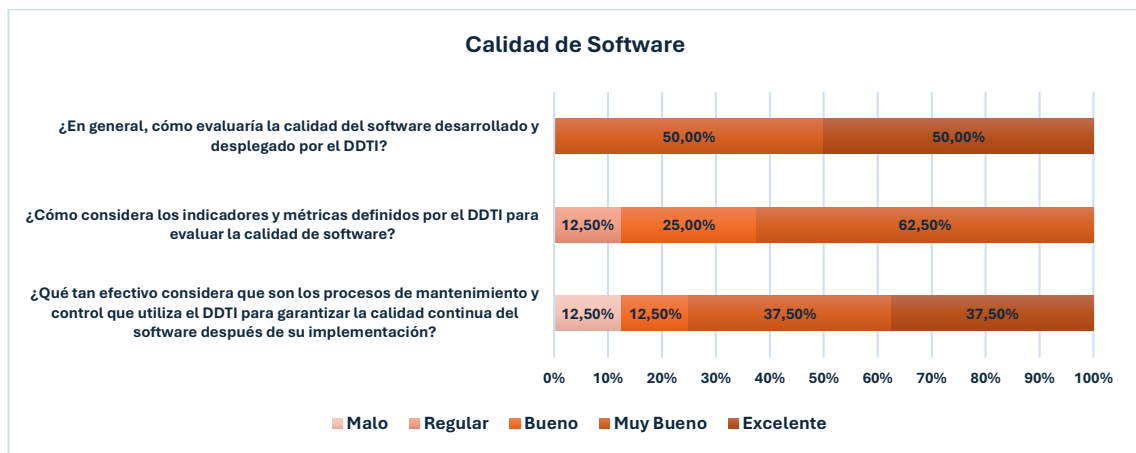


Nota: Fuente: Elaboración Propia

Una adecuada asignación de roles y responsabilidades dentro del equipo de desarrollo tiene un impacto directo en la calidad de la comunicación entre sus miembros. En este sentido, los resultados obtenidos indican que una mayor claridad en las funciones facilita una mejor interacción entre los miembros del equipo. Según la figura 20, el 75% de los encuestados califican la comunicación como "Muy Buena" o "Excelente", lo que refleja una percepción positiva sobre la fluidez y eficiencia en los intercambios dentro del equipo. Esta percepción positiva se refleja también en la valoración sobre la definición de roles, donde el 50% de los participantes consideran que esta definición es "Muy Buena" o "Excelente". Estos resultados sugieren que la claridad en las responsabilidades de cada miembro del equipo contribuye significativamente a la mejora de la comunicación, permitiendo una ejecución más eficaz de las pruebas y una transición más fluida en el proceso de puesta en producción. La importancia de una estructura organizativa bien definida se resalta como un factor clave para optimizar la interacción y el trabajo en equipo dentro del área de desarrollo.

La disposición del equipo de desarrollo para adoptar una guía formal de ejecución de pruebas, incluso sin estar familiarizados con la ISO 29119, muestra una estrecha relación con la percepción de la comunicación dentro del equipo durante la fase de pruebas y puesta en producción. Según la figura 20, un 75% de los encuestados califican la comunicación dentro del equipo como "Muy Buena" o "Excelente", lo que indica que, en general, los miembros del equipo se sienten cómodos y son capaces de comunicarse de manera efectiva durante el proceso de pruebas. Este resultado está relacionado con el nivel de aceptación de la adopción de una guía formal de ejecución de pruebas, donde el 50% de los participantes evalúan positivamente la disposición del equipo para implementar dicho proceso. Esto sugiere que, a pesar de la falta de familiaridad con la norma ISO 29119, la buena comunicación facilita la aceptación de nuevas metodologías y procedimientos dentro del equipo, lo que refuerza la importancia de una comunicación abierta y eficiente para la adopción de prácticas formales en el desarrollo de software.

Figura 21 Gráfico de barras apiladas – Calidad de Software

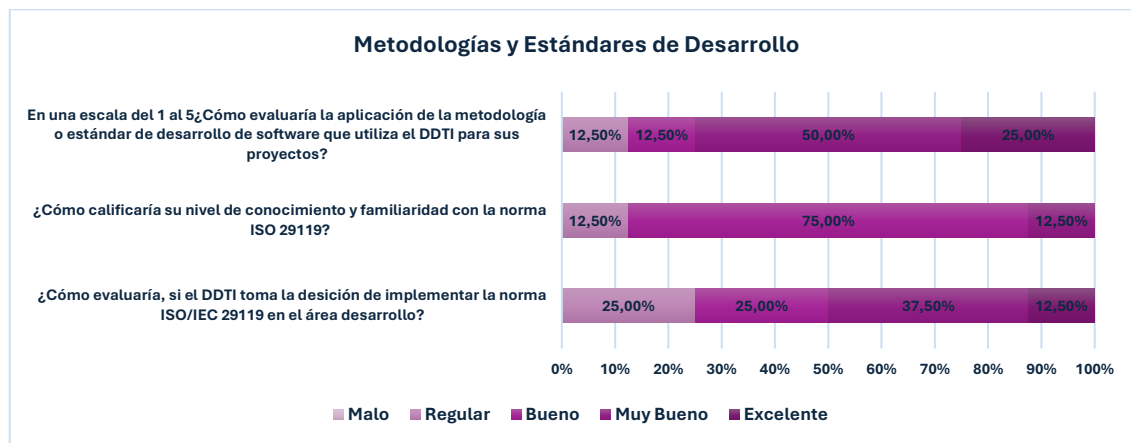


Nota: Fuente: Elaboración Propia.

Los resultados obtenidos muestran una relación directa entre la percepción de la calidad del software y la valoración de los indicadores y métricas utilizadas por el DDTI para evaluar dicha calidad. La mitad de los encuestados califica la calidad del software como "Muy Buena" o "Excelente", lo que refleja una evaluación positiva de los productos entregados. Esta percepción favorable se refleja también en la evaluación de los indicadores y métricas definidas por el DDTI, con un 62,5% de los participantes considerando que estas herramientas son adecuadas para medir la calidad del software. La coincidencia en estas valoraciones sugiere que los miembros del equipo perciben que una buena calidad del software está respaldada por la efectividad de las métricas utilizadas, lo que refuerza la importancia de contar con indicadores claros y alineados con los objetivos de calidad del desarrollo. Sin embargo, los porcentajes más bajos en la evaluación de ambos aspectos también señalan que existen áreas de mejora en la implementación y la comunicación de estas métricas, lo que podría influir en la percepción general de la calidad del software y su evaluación.

Los procesos de mantenimiento y control son valorados positivamente por una parte considerable de los encuestados, también se observa que un porcentaje significativo no los percibe como totalmente efectivos. Esto podría indicar que, aunque existen mecanismos establecidos para garantizar la calidad continua del software, aún persisten áreas de oportunidad en su implementación y seguimiento. La percepción de la efectividad de estos procesos depende en gran medida de la capacidad del equipo para adaptarse a cambios y mantener la calidad en el tiempo, lo cual es un desafío constante que requiere de un enfoque integral y bien coordinado.

Figura 22 Gráfico de barras apiladas – Metodologías y estándares de desarrollo.

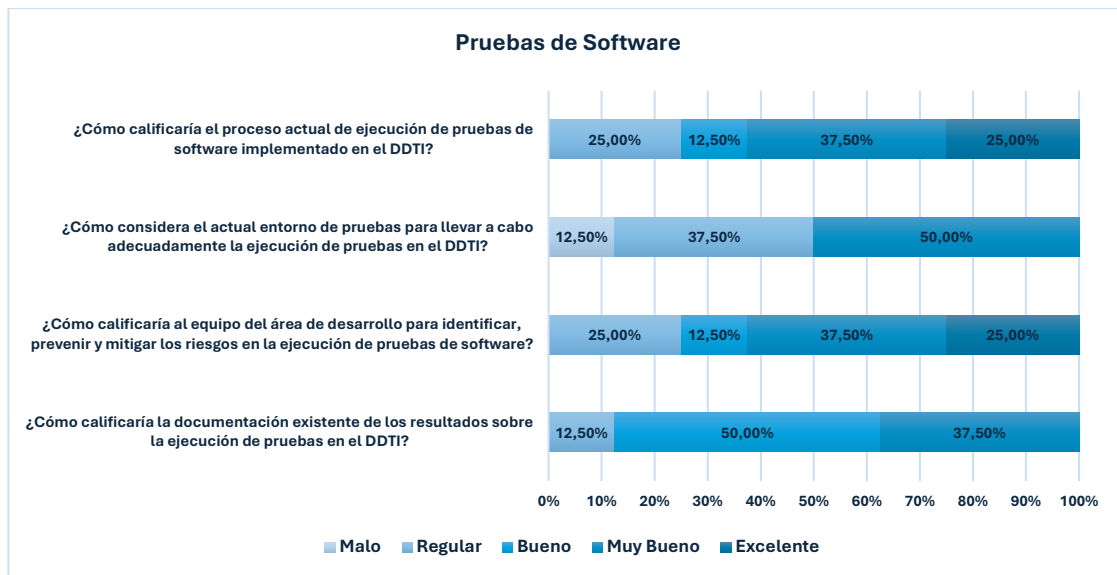


Nota: Fuente: Elaboración propia.

La evaluación de la aplicación de la metodología o estándar de desarrollo de software utilizado por el DDTI muestra una tendencia positiva, aunque también pone de manifiesto algunas áreas que requieren atención. El 50% de los encuestados califica la aplicación de la metodología como "Muy Buena", lo que indica una apreciación favorable general. Sin embargo, un 25% de los participantes lo evaluaron como "Excelente", lo que sugiere que existe un reconocimiento aún más sólido de la eficacia del enfoque metodológico. Por otro lado, un 25% de los encuestados lo calificaron como "Regular" o "Malo", lo que apunta a posibles áreas de mejora en la implementación o en la percepción de la metodología utilizada. Esto podría estar relacionado con la necesidad de ajustar ciertos aspectos de la metodología para que se adapte mejor a las características específicas de los proyectos y al contexto del DDTI. En general, los resultados indican que, aunque la mayoría valora positivamente el uso de una metodología estructurada, aún existen oportunidades para fortalecer su aplicación y garantizar una mayor consistencia en la percepción del equipo sobre su efectividad.

El análisis de los resultados refleja que, a pesar de que la mayoría de los encuestados tiene un nivel de conocimiento "Muy Bueno" o "Excelente" respecto a la norma ISO 29119, aún existen ciertos desafíos en la disposición del equipo para adoptar formalmente esta norma en el proceso de desarrollo. Un alto porcentaje de los participantes que tienen un buen conocimiento de la norma también consideran favorable su implementación en el área de desarrollo, lo que indica que un mayor nivel de familiaridad con la norma contribuye positivamente a la aceptación de su aplicación. Sin embargo, la diferencia en las respuestas, con un 25% de los encuestados evaluando la adopción como "Regular" o inferior, sugiere que el conocimiento por sí solo no es suficiente para garantizar la implementación exitosa de la norma. Esto podría reflejar la necesidad de abordar cuestiones prácticas o contextuales relacionadas con la adaptación de la norma ISO 29119 al entorno específico del DDTI. La implementación de estándares como la ISO 29119 requiere no solo conocimiento, sino también un compromiso más amplio en términos de recursos, capacitación y adaptaciones a las necesidades del equipo, para asegurar que los beneficios de la norma sean completamente comprendidos y aprovechados.

Figura 23 Gráfico de barras apiladas – Pruebas de Software.



Nota: Fuente: Elaboración propia.

Una clara relación entre el proceso de ejecución de pruebas de software y el entorno de pruebas disponible para llevar a cabo estas pruebas se refleja en las respuestas obtenidas. A pesar de que una proporción significativa de los encuestados considera que el proceso actual de ejecución de pruebas es adecuado, los resultados muestran también que el entorno de pruebas presenta desafíos importantes. Aunque un 50% de los participantes evalúa el entorno de pruebas como adecuado para ejecutar las pruebas de manera eficiente, solo el 37,5% califica positivamente el proceso de ejecución, lo que sugiere que, aunque el entorno de pruebas puede ser adecuado en términos de infraestructura, hay aspectos del proceso de ejecución que aún requieren mejoras. Esta discrepancia resalta la importancia de asegurar que no solo el entorno esté bien preparado, sino que también los procesos asociados sean igualmente sólidos y estén alineados con las expectativas del equipo, para garantizar la efectividad en la ejecución de las pruebas de software.

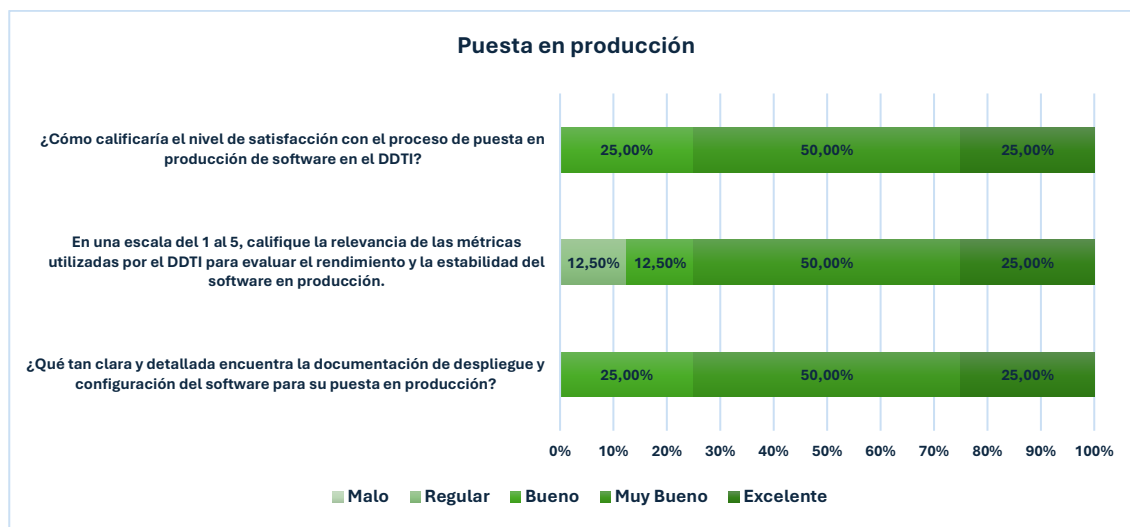
El proceso de ejecución de pruebas y la capacidad del equipo de desarrollo para identificar, prevenir y mitigar riesgos están estrechamente relacionados en la percepción general del

equipo. Aunque el 37,5% de los encuestados considera que el proceso de ejecución de pruebas implementado es adecuado, la misma proporción valora positivamente la capacidad del equipo para gestionar los riesgos en este proceso. Sin embargo, un 25% de los participantes califica tanto el proceso de ejecución de pruebas como la capacidad del equipo para mitigar riesgos como "Malo" o "Regular", lo que sugiere que existen áreas críticas en las que se perciben fallas o deficiencias. Este patrón refleja que, si bien el equipo tiene un enfoque en la mitigación de riesgos, los resultados indican que este esfuerzo podría no ser suficiente o efectivo en algunos casos, lo que afecta directamente la calidad y la eficacia de la ejecución de pruebas. La relación entre estos dos factores resalta la necesidad de fortalecer tanto el proceso de pruebas como las capacidades del equipo para manejar de manera proactiva los riesgos asociados, garantizando así una mayor fiabilidad en las pruebas de software.

La percepción sobre la ejecución de pruebas de software y la documentación asociada a los resultados muestra una relación directa en cuanto a la calidad de ambos aspectos. Aunque un 37,5% de los encuestados evalúa el proceso de ejecución de pruebas de manera positiva, solo un 50% considera que la documentación relacionada con los resultados es adecuada, con una significativa proporción (12,5%) calificando esta documentación de forma menos favorable. Esto sugiere que, aunque el proceso de ejecución de pruebas podría estar en marcha de manera satisfactoria, la falta de una documentación adecuada podría estar limitando la efectividad y trazabilidad de las pruebas realizadas. La calidad de la documentación es fundamental no solo para respaldar los resultados obtenidos, sino también para facilitar el seguimiento, análisis y mejora continua del proceso de pruebas. La falta de documentación clara y completa podría generar incertidumbre sobre los resultados

y la eficiencia del proceso, lo que a su vez podría impactar negativamente en la percepción general de las pruebas de software implementadas.

Figura 24 Gráfico de barras apiladas – Puesta en producción.



Nota: Fuente: Elaboración propia.

La percepción de satisfacción con el proceso de puesta en producción y la claridad de la documentación relacionada con el despliegue y la configuración del software están estrechamente vinculadas. Un 50% de los encuestados califica positivamente ambos aspectos, indicando que una documentación detallada y clara influye directamente en la satisfacción general con el proceso de producción. Sin embargo, el 25% de los participantes califica tanto la satisfacción con el proceso como la claridad de la documentación como "Regular" o "Malo", lo que sugiere que, aunque la mayoría valora positivamente estos aspectos, aún existen áreas en las que se perciben deficiencias. Esta variabilidad refleja la importancia de una documentación exhaustiva para garantizar la transparencia y la eficiencia en la puesta en producción del software. Si bien algunos miembros del equipo parecen estar satisfechos con los procedimientos y la documentación disponible, otros

podrían necesitar más claridad o detalles en los mismos, lo que podría estar afectando la percepción global del proceso de producción.

La percepción sobre la relevancia de las métricas utilizadas para evaluar el rendimiento y la estabilidad del software en producción refleja una tendencia mayormente positiva, aunque con áreas que requieren atención. La mitad de los encuestados considera que las métricas son "Muy Buenas" o "Excelentes", lo que sugiere que, en general, se perciben como adecuadas para evaluar los aspectos clave del software en producción. Sin embargo, un 25% de los participantes aún las considera "Buenas" o "Regulares", lo que indica que hay margen para mejorar la efectividad o la aplicabilidad de estas métricas en el entorno de producción. Esta variabilidad sugiere que, aunque las métricas son valoradas positivamente por la mayoría, aún existen dudas o áreas de mejora que podrían optimizarse para asegurar que realmente estén alineadas con los objetivos de monitoreo y control del rendimiento del software a largo plazo. La mejora en la claridad, precisión o alcance de las métricas podría contribuir a una mejor evaluación de la estabilidad y rendimiento del software una vez en producción.

2.3.5. Resultados de la Entrevista realizada al subdirector del DDTI

La entrevista se realizó al subdirector del DDTI, responsable del proceso de Gestión de Desarrollo de Software en la Dirección de Desarrollo Tecnológico e Informático.

El objetivo de la entrevista es conocer el proceso de desarrollo establecido en el DDTI, específicamente a las pruebas de software.

A continuación, se realiza un extracto de lo más importante manifestado por el subdirector, en dieciocho preguntas abiertas:

En lo referente a estructura y funciones actuales del DDTI, indicó que en su nivel directivo está estructurado por dirección y subdirección con apoyo administrativo de secretaría, así mismo en el área operativa tiene tres áreas fundamentales: área de desarrollo de software, que está orientando a la gestión del desarrollo de proyectos de software, desde la construcción hasta el mantenimiento de estos. Además del desarrollo y mantenimiento de los portafolios electrónicos, portal web de la institución y aplicaciones móviles; la segunda área es de infraestructura, redes y telecomunicaciones que se encarga de brindar mantenimiento y actualización de data center, red física e inalámbrica de la universidad y la tercera es el área de atención al usuario, que consiste básicamente en dar soporte en cuanto a adquisición, mantenimiento preventivo y correctivo de los diferentes activos tecnológicos de la institución.

La definición de roles y responsabilidades dentro del equipo del área de desarrollo se basa en dos puntos clave: las necesidades de la institución y la selección del personal técnico adecuado. En primer lugar, las necesidades se determinan a partir de los requerimientos de las diversas áreas de usuarios, que pueden surgir debido a problemáticas específicas o a la necesidad de mejoras en los procesos. Para cubrir estas necesidades, se busca personal idóneo que tenga conocimientos en las reglas de negocio y en las herramientas tecnológicas utilizadas en la institución. Por ejemplo, si se necesita desarrollar un software de planificación estratégica, se buscará a alguien con conocimientos en planificación institucional y las herramientas tecnológicas correspondientes. Además, se menciona que el personal es preparado y capacitado para los roles asignados, y que la selección se realiza mediante un concurso de méritos y oposición, garantizando que se elija al candidato más adecuado para el puesto específico. Esto asegura que el equipo de desarrollo esté compuesto

por personas que no solo entienden las necesidades del negocio, sino que también tienen las competencias técnicas para cumplir con sus responsabilidades.

El DDTI enfrenta varios desafíos relacionados con la calidad del software, los cuales se agrupan en áreas clave como la seguridad, la eficiencia en los tiempos de respuesta y la capacidad de almacenamiento. Estos desafíos no son exclusivos del DDTI, sino que son comunes en todas las instituciones y empresas, tanto públicas como privadas.

- Seguridad: Un esquema sólido de seguridad es primordial. El software debe ser capaz de proteger los datos y mantener la integridad de la información contra posibles amenazas.
- Eficiencia en Tiempos de Respuesta: El sistema debe ser eficiente, proporcionando tiempos de respuesta rápidos para asegurar una buena experiencia de usuario. La eficiencia en los tiempos de respuesta es crucial para mantener la operatividad y la satisfacción del usuario final.
- Capacidad de Almacenamiento: La capacidad de manejar grandes volúmenes de datos es otro desafío significativo. Con el creciente volumen de información que se maneja en los tiempos actuales, es vital que el sistema tenga una capacidad adecuada para almacenar y gestionar estos datos sin comprometer el rendimiento.

Para abordar estos desafíos, el DDTI utiliza el Sistema Informático Integrado Universitario (SIIU), que se asegura de incorporar estas características esenciales. Se enfatiza que la calidad del software se mejora cuando se garantiza la seguridad, se optimizan los tiempos de respuesta y se asegura una alta disponibilidad, prácticamente cercana al 100%. El SIIU también debe ser capaz de manejar grandes volúmenes de información eficientemente, lo cual contribuye a la percepción general de calidad del software. En resumen, los desafíos

de seguridad, eficiencia y capacidad de almacenamiento están estrechamente relacionados con la calidad del software y son abordados mediante la implementación de soluciones robustas como el SIIU.

El DDTI utiliza encuestas de satisfacción del usuario como principales indicadores para evaluar la calidad del software. Estas encuestas son generadas y administradas por un sector externo a la dirección, en particular la Facultad de Ingeniería en Ciencias Aplicadas (FICA).

Las encuestas están debidamente sustentadas teóricamente y aprobadas metodológicamente en el mercado, asegurando que los datos recolectados sean fiables y relevantes. Las encuestas se distribuyen periódicamente a tres grupos clave de usuarios: estudiantes, docentes y personal administrativo.

El objetivo de estas encuestas es recopilar información sobre cómo los usuarios perciben la agilidad y el cumplimiento de los requerimientos del software en la práctica. Esto incluye evaluar si el sistema es ágil, cumple con los requerimientos propuestos y si estos son entregados a tiempo.

En resumen, el DDTI se basa en las encuestas de satisfacción del usuario administradas por la FICA para obtener retroalimentación valiosa y detallada sobre la calidad del software. Estas encuestas permiten al DDTI medir aspectos clave como la agilidad y el cumplimiento de los requerimientos, proporcionando una visión práctica y basada en la experiencia real de los usuarios sobre el rendimiento del software.

El mantenimiento y control del software después de su implementación se realiza de manera continua y está estrechamente vinculado con la evaluación de la satisfacción del usuario. Dado que las reglas de negocio y los procesos sustantivos de la institución, como la

docencia, la investigación y la vinculación con la sociedad, son dinámicos y cambian con el tiempo, es crucial adaptar el software a estas variaciones. Para garantizar la calidad continua del software, se llevan a cabo encuestas de satisfacción del usuario de manera periódica, aproximadamente cada seis meses.

En resumen, la calidad continua del software se garantiza mediante un proceso de mantenimiento permanente y adaptativo, apoyado por encuestas de satisfacción del usuario que se realizan cada seis meses. Esta retroalimentación constante permite al equipo de desarrollo realizar los ajustes necesarios para mantener y mejorar la calidad del software en respuesta a las cambiantes necesidades de la institución.

El DDTI utiliza metodologías ágiles para el desarrollo de software. Esta decisión se debe a la necesidad de realizar desarrollos rápidos, en contraste con el enfoque anterior, donde se utilizaba la metodología RUP (Rational Unified Process) para la construcción del Sistema Informático Integrado Universitario (SIIU). La metodología RUP fue adecuada para la construcción inicial del ERP y el portafolio, que tomó más de 10 años. Sin embargo, para satisfacer las necesidades actuales, se han adoptado metodologías ágiles.

En particular, el DDTI ha incorporado elementos de SCRUM y ha seguido la metodología sugerida por la herramienta de desarrollo APEX, una solución proporcionada por la corporación ORACLE. APEX tiene sus propias fases de desarrollo definidas, y el DDTI ha ajustado su enfoque para alinearse con estas fases. Además, se ha creado una documentación técnica específica para esta metodología, denominada APEX UTN, que se ha convertido en la metodología propia de la institución.

Las fases de esta metodología combinada incluyen:

- Análisis de Requisitos: Identificación y documentación de las necesidades y requerimientos del proyecto.
- Diseño: Creación de la arquitectura y diseño detallado del sistema.
- Construcción: Desarrollo del software conforme a los diseños establecidos.
- Pruebas: Verificación y validación del software para asegurar su calidad y funcionalidad.
- Despliegue: Implementación del software en el entorno de producción.
- Mantenimiento: Realización de mejoras y correcciones continuas para asegurar el funcionamiento óptimo del software.

En el área de desarrollo del DDTI, la comunicación y participación entre los miembros del equipo durante la ejecución de pruebas y la puesta en producción de software es limitada. Cada técnico tiene asignado un módulo específico y es responsable del desarrollo y las pruebas de ese módulo en particular. Esta estructura implica que los técnicos no tienen una necesidad frecuente de interactuar con otros técnicos de diferentes módulos, ya que cada uno trabaja de manera autónoma en sus propias responsabilidades.

La comunicación lógicamente se da en el contexto de despliegues y compilaciones de soluciones. Es en estos momentos cuando los técnicos deben coordinarse para asegurar que las diferentes partes del software se integren correctamente y funcionen como un todo. Sin embargo, fuera de estas actividades específicas de integración y despliegue, la interacción entre técnicos es mínima.

El DDTI, la ejecución de pruebas de software sigue una metodología que se centra en la capacitación del usuario final y la validación de los requisitos solicitados por estos usuarios. El proceso descrito implica que, una vez que los técnicos completan las pruebas internas de

sus módulos, se procede a capacitar al usuario final. Esta capacitación tiene como objetivo asegurarse de que los usuarios finales entiendan el funcionamiento del software y puedan validar que los requisitos iniciales han sido cumplidos. La validación por parte del usuario final es un paso crucial para confirmar que el desarrollo satisface sus expectativas y necesidades.

Al final del proceso, se formaliza la satisfacción del usuario mediante la elaboración de actas de entrega y recepción. Estas actas sirven como documentación oficial que confirma que los usuarios finales están satisfechos con el desarrollo e implementación de los requerimientos solicitados.

El DDTI cuenta con un entorno de pruebas adecuado para la ejecución de pruebas, alojado en la nube de ORACLE. Esta infraestructura en la nube incluye tanto las herramientas de desarrollo como las bases de datos, las cuales están centralizadas para todos los procesos institucionales y normalizadas según el eje de desarrollo.

Para garantizar que las aplicaciones funcionen correctamente antes de su puesta en producción, el DDTI ha creado un ambiente espejo en la nube, que replica el entorno de producción en el entorno de desarrollo y pruebas. Este ambiente espejo permite a los desarrolladores realizar pruebas en condiciones que emulan las del entorno de producción, asegurando así que las aplicaciones se comporten como se espera una vez implementadas.

Sin embargo, debido a consideraciones de costos, las capacidades de procesamiento y almacenamiento del entorno de pruebas son menores que las del entorno de producción. A pesar de estas limitaciones, el DDTI dispone de un ambiente de pruebas funcional que

permite realizar las pruebas necesarias para asegurar la calidad del software antes de su despliegue.

El área de desarrollo del DDTI implementa varias estrategias y prácticas para identificar, prevenir y mitigar los riesgos en la ejecución de pruebas de software. Una de las estrategias clave es la separación de los entornos de desarrollo y pruebas del entorno de producción, no solo de manera lógica, sino también geográficamente. Esta separación física se realiza según las políticas de seguridad de ORACLE, la corporación que proporciona la infraestructura en la nube utilizada por el DDTI.

Esta separación geográfica y lógica ayuda a mitigar los riesgos al asegurar que cualquier problema o fallo en el entorno de pruebas no afecte el entorno de producción. Además, esta estrategia proporciona una capa adicional de seguridad, ya que los entornos están protegidos en diferentes ubicaciones físicas, lo cual es una práctica recomendada para prevenir posibles vulnerabilidades y asegurar la integridad de los datos.

Además de la separación física y lógica, el DDTI se beneficia de las robustas políticas de seguridad implementadas por ORACLE. Estas incluyen herramientas avanzadas para la protección contra ataques de negación de servicios (DDoS), antimalware, y otras tecnologías de seguridad de primera línea. Estas medidas ayudan a prevenir intrusiones y asegurar que el entorno de pruebas y desarrollo esté protegido contra diversas amenazas de seguridad.

En el DDTI, la documentación del proceso de ejecución de pruebas no está formalizada de manera extensa. Actualmente, la documentación existente se centra en la documentación técnica del desarrollo del software y en las actas de entrega y recepción.

La documentación técnica describe cómo se desarrolló el software, pero no incluye un registro detallado y estructurado del proceso de pruebas. Las pruebas de usuario son parte del proceso de aceptación, y se documentan en las actas de entrega y recepción. Estas actas son muy específicas y detalladas, delimitando todos los componentes y describiendo de manera granular las pruebas realizadas y los resultados obtenidos. Este enfoque asegura que haya una referencia formal sobre lo que se entregó y cómo se verificó su funcionalidad.

Sin embargo, esta forma de documentación tiene limitaciones, ya que se enfoca en los resultados de las pruebas y la aceptación por parte del usuario, pero carece de una estructura formal que incluya todos los aspectos del proceso de pruebas, como los casos de prueba, los resultados detallados de cada prueba, las incidencias encontradas y las acciones correctivas tomadas.

El proceso de puesta en producción de software en el DDTI sigue una serie de pasos estructurados que involucran la colaboración entre el desarrollador y el administrador de la base de datos. Una vez que el desarrollador ha completado el desarrollo y realizado las pruebas en el entorno de desarrollo y pruebas, el siguiente paso es transferir el trabajo al administrador de la base de datos.

El desarrollador entrega al administrador todos los elementos necesarios para la puesta en producción, incluyendo:

- **Objetos de Base de Datos:** Esto incluye la creación de bases de datos, vistas y cualquier otro objeto de base de datos necesario para el desarrollo y pruebas.
- **Archivos de Aplicación:** Esto abarca los archivos que componen el frontend, tales como fórmulas, reportes y otros componentes esenciales del software.

El administrador de la base de datos revisa estos elementos para asegurar que todo esté en orden y cumple con los estándares necesarios. Una vez revisados y validados, el administrador procede a desplegar estos elementos en el entorno de producción, asegurando que el software esté disponible y funcional para los usuarios finales.

Actualmente, el DDTI no utiliza métricas definidas para evaluar el rendimiento y la estabilidad del software en producción. En lugar de ello, la evaluación se basa principalmente en la observación y la experiencia de los usuarios finales. Esto significa que la principal forma de detectar problemas de rendimiento o estabilidad es a través del feedback directo de los usuarios, quienes comunican cualquier inconveniente tan pronto como lo experimentan.

La documentación de despliegue y configuración del software para la puesta en producción en el DDTI no está definida formalmente.

El área de desarrollo no dispone mayor conocimiento y familiaridad del área de desarrollo con la norma ISO 29119

En el DDTI no han existido intentos previos de implementar la norma ISO 29119.

La adopción de la norma ISO/IEC 29119 podría ofrecer múltiples beneficios al DDTI, principalmente al proporcionar una estructura más formal y exhaustiva para la ejecución de pruebas de software. Actualmente, aunque las pruebas se realizan con cierto grado de formalidad, la implementación de esta norma permitiría elevar ese nivel de formalidad y rigurosidad, cubriendo más detalles y requisitos específicos para el desarrollo de pruebas.

El área de desarrollo del DDTI no está capacitada ni cuenta con las herramientas necesarias para implementar la norma ISO 29119. Esto significa que el equipo de desarrollo no tiene

la formación específica ni las herramientas técnicas requeridas para seguir los estándares establecidos por esta norma internacional de pruebas de software.

2.3.6. Análisis y revisión documental

Se solicitó al DDTI cuatro documentos clave: proceso de construcción de software, metodología de desarrollo en Oracle APEX, documentación del ambiente de pruebas y producción y manuales completos del portafolio docente y estudiante. Tras realizar un análisis y revisión de estos documentos, se pudo identificar lo siguiente:

1. Proceso de Construcción de Software y Metodología de Desarrollo de Aplicaciones en Oracle APEX - UTN

Proceso Formal de Pruebas: No se menciona explícitamente un marco formal para la ejecución de pruebas (como ISO 29119, ISTQB u otros estándares internacionales). Esto es preocupante ya que la formalización de un proceso de pruebas garantiza consistencia, trazabilidad y una mejor gestión de la calidad a lo largo del ciclo de vida del software.

Actividades Detalladas de Pruebas: Se hace referencia a pruebas unitarias y funcionales, pero no se detallan las actividades exactas, pasos o criterios que deben seguirse para asegurar que las pruebas se realicen de manera adecuada. También se mencionan pruebas de integración y automatización, pero no queda claro si hay un plan específico de pruebas ni si hay criterios de aceptación definidos.

Documentación y Métricas: No se menciona cómo se documentan los resultados de las pruebas ni si existe un sistema de seguimiento de errores o incidencias. Tampoco se hace referencia a las métricas de rendimiento o calidad que se utilizan para validar los resultados y garantizar la mejora continua del software.

Herramientas de Pruebas: Aunque se mencionan algunas herramientas básicas para la depuración y pruebas (modo de depuración de páginas, SQL Workshop, entre otras), no se hace referencia a herramientas de gestión de pruebas o plataformas para pruebas automatizadas. Esto podría limitar la capacidad de realizar pruebas exhaustivas y garantizar una cobertura completa.

Pruebas en Diferentes Escenarios: Se menciona que se realizan pruebas en el entorno de desarrollo, pero no está claro si el DDTI tiene un entorno específico para pruebas. Además, sería importante conocer si se llevan a cabo pruebas en distintos escenarios como pruebas de estrés, rendimiento y seguridad para evaluar cómo se comporta la aplicación bajo diversas condiciones.

Normativas de Calidad (ISO): No se menciona si el proceso de pruebas en el DDTI sigue algún estándar de calidad como ISO 29119 (pruebas de software) o ISO 25000 (calidad del producto), lo cual es preocupante dado que la ausencia de normativas estandarizadas puede implicar falta de rigurosidad en el proceso de pruebas.

2. Documentación del Ambiente de Pruebas y Entorno de Producción

No disponen de documentación formal sobre el uso del ambiente de pruebas y producción, ya que estos dependen del proveedor del servicio de la nube, Oracle. Si bien hacen uso de estos ambientes, no se cuenta con información documentada que describa detalladamente su funcionamiento o administración.

3. Manual del Portafolio Docente y Portafolio Estudiante

Aunque los manuales de usuario detallan todas las funcionalidades de los submódulos, como el portafolio estudiante y docente, actualmente estos documentos no están al alcance

de los usuarios finales. No se ha publicado ningún enlace en la página oficial de la universidad que permita a los usuarios acceder a los mismos, lo que podría mejorar el entendimiento y uso de las funcionalidades del sistema.

El DDTI enfrenta una serie de desafíos en cuanto a la calidad del software, especialmente en lo que respecta a la formalización y documentación de los procesos de pruebas, la falta de un marco estandarizado como la norma ISO 29119 y la falta de herramientas y capacitación específicas para su implementación. Estos desafíos, junto con la ausencia de documentación clara sobre los entornos de pruebas y producción, limitan la capacidad del DDTI para garantizar una calidad consistente y controlada en las fases de pruebas y puesta en producción. La falta de un proceso formal y estandarizado, junto con la carencia de métricas de rendimiento y seguimiento adecuado de errores, resalta la necesidad urgente de diseñar una guía metodológica de calidad de software para las fases de pruebas y puesta en producción. Esta guía, basada en la norma ISO/IEC/IEEE 29119, permitiría establecer procedimientos claros, asegurar la trazabilidad y mejorar la calidad del software a través de un enfoque estructurado y coherente, garantizando que todos los aspectos del ciclo de vida del software se gestionen con rigurosidad y eficiencia.

2.4. Diseño de la guía metodológica

De acuerdo con el estudio realizado en el capítulo uno se definió que la estructura a seguir para el diseño de la guía metodológica será la propuesta por Cárdenas (2020).

Introducción

La presente guía metodológica tiene como objetivo formalizar y estandarizar los procesos de pruebas y puesta en producción del software desarrollado en la Dirección de Desarrollo Tecnológico e Informático (DDTI) de la Universidad Técnica del Norte. Su propósito es establecer una estructura clara y consistente que facilite la implementación de prácticas de calidad, proporcionando criterios de aceptación, roles definidos y actividades específicas, así como una adecuada documentación para asegurar que cada etapa del proceso cumpla con los estándares requeridos. Esta guía se presenta como un recurso adaptable, aplicable tanto a proyectos actuales como futuros, y orientada a mejorar la calidad y eficiencia del software desarrollado en el DDTI.

2.4.1. Propósito de la guía

La guía metodológica de calidad de software tiene como propósito, formalizar y estandarizar los procesos de pruebas y puesta en producción, asegurando la calidad del software desarrollado en la Dirección de Desarrollo Tecnológico e Informático (DDTI) de la Universidad Técnica del Norte (UTN).

A través de esta metodología, se podrá:

- Establecer criterios de aceptación.
- Definir roles, responsabilidades y actividades.
- Establecer la documentación y entregables del proceso de pruebas y puesta en producción.

2.4.2. Alcance general

La fase inicial de aplicación de la guía metodológica abarca únicamente la ejecución de pruebas funcionales, pruebas de usabilidad y despliegue de software, específicamente en los submódulos Portafolio Docente y Portafolio Estudiante del Sistema Informático Integrado Universitario (SIIU).

Es importante mencionar que la guía puede ser adaptada y aplicada a otros proyectos de software existentes y por desarrollarse en el DDTI, considerando las características y necesidades específicas de cada proyecto, como base para propiciar evolución de la guía y se pueda realizar otro tipo de pruebas

2.4.3. Definiciones y términos clave

Para una mejor comprensión de la guía, se definen los siguientes términos clave, los cuales son extraídos de las normas ISO 29119 e ISO 20000-1:

Criterios de finalización: se definen cuándo una actividad de prueba o conjunto de pruebas se considera completado.

Resultado esperado: es la respuesta o comportamiento que se anticipa del sistema bajo prueba cuando se ejecuta un caso de prueba.

Pruebas basadas en requisitos: es una técnica de prueba que se basa en la cobertura de los requisitos documentados. Se diseña un conjunto de pruebas para verificar que todos los requisitos del sistema han sido implementados correctamente.

Prueba: es el proceso de ejecutar un sistema o componente para verificar si cumple con los requisitos especificados y si funciona según lo esperado.

Caso de prueba: es un conjunto específico de condiciones bajo las cuales se verificará el correcto funcionamiento del sistema.

Entorno de prueba: es la configuración de hardware, software, red y otros elementos donde se ejecutan las pruebas. Debe ser representativo del entorno real de producción.

Elemento de prueba: es un componente específico del software que debe ser probado. Puede ser un módulo, función, característica o componente.

Objetivo de la prueba: define el propósito y el enfoque de las pruebas que se van a realizar.

Plan de prueba: es un documento que describe el enfoque, recursos y cronograma para las actividades de prueba. Incluye los objetivos de las pruebas, el alcance, los elementos a ser probados y los criterios de éxito.

Proceso de prueba: incluye todas las actividades involucradas en la planificación, diseño, ejecución y seguimiento de las pruebas.

Resultado de prueba: es la observación obtenida después de ejecutar un caso de prueba, que puede coincidir o no con el resultado esperado.

Métricas de pruebas: son medidas cuantitativas que se utilizan para evaluar el progreso y la calidad de las pruebas.

Criterios de suspensión y reanudación: los criterios de suspensión definen cuándo se deben detener las pruebas, y los criterios de reanudación describen las condiciones bajo las cuales se pueden reiniciar las actividades de prueba.

Pruebas de regresión: se realizan para verificar que el software siga funcionando correctamente después de haber sido modificado. Este tipo de pruebas asegura que las

correcciones de errores, mejoras o cambios realizados en el código no hayan introducido nuevos problemas en áreas previamente probadas y funcionales.

Pruebas de confirmación: también conocidas como pruebas de re-testing, se llevan a cabo para verificar que los errores específicos que fueron detectados en una ejecución anterior de pruebas han sido corregidos correctamente.

2.4.4. Roles generales

Se establece los siguientes roles y responsabilidades, para el equipo asignado para el proceso de ejecución de pruebas y puesta en producción:

Tabla 26 Roles y responsabilidades para la fase de pruebas.

Rol	Responsabilidades
Administrador de desarrollo	Evaluar y mantener la guía metodológica
Líder de QA	Planificar, construir y ejecutar las pruebas

Nota: Fuente: Elaboración propia.

Tabla 27 Roles y responsabilidades para la fase de puesta en producción.

Rol	Responsabilidades
Administrador de desarrollo	Evaluar y mantener la guía metodológica Coordinar y supervisar las actividades de desarrollo y puesta en producción.
Equipo de desarrollo	Gestionar los requisitos de configuración Implementar y validar los cambios Participar en la planificación y procedimiento del despliegue.

Nota: Fuente: Elaboración propia.

Fase de pruebas

2.4.5. Planificación de pruebas

La planificación de pruebas comprende las siguientes actividades:

- a. Identificador único del documento

Identifica de forma uniforme la documentación o el registro de la prueba.

El identificador único puede incluir el título de la documentación de ensayo, la fecha de emisión, la versión y/o el estado de la documentación (por ejemplo, borrador, revisado, corregido, final).

b. Introducción

En esta sección se estableció el marco general para la planificación de las pruebas, como:

- Alcance, se detalla lo que cubre el plan de pruebas, especificando los componentes del software o sistema que serán probados.
- Referencias, se lista las normativas, estándares y documentos previos que sirven como base para la planificación de las pruebas.
- Glosario, se define los términos técnicos relevantes que se utilizarán a lo largo del documento, facilitando la comprensión uniforme de los mismos.

c. Contexto de las pruebas

En este apartado se detalla el entorno en el que se llevarán a cabo las pruebas, como:

- Proyecto a evaluar, se detalla el sistema en general, así como las partes específicas que serán abordadas durante el tipo de pruebas a aplicarse.
- Elementos de prueba, se describe los componentes del software o hardware que serán objeto de evaluación.
- Alcance de la prueba se refiere a los límites del sistema o componentes que serán cubiertos por las pruebas
- Suposiciones y restricciones, se enumera las limitaciones del proceso de pruebas, como recursos disponibles o plazos.

- Partes interesadas, se identifica las partes clave como desarrolladores, clientes o usuarios que tienen algún interés o participación en el proceso de prueba.

d. Comunicación de las pruebas

En esta sección se aborda los canales y métodos mediante los cuales se comunicará el progreso y los resultados de las pruebas. La planificación de cómo, cuándo y a quién se deben enviar los informes sobre los resultados de las pruebas es esencial para garantizar la claridad y transparencia en todo el proceso.

e. Registro de riesgos

El registro de riesgos abarca la identificación de posibles riesgos que podrían afectar la ejecución de las pruebas, la descripción de estos riesgos y las acciones que se tomarán para mitigarlos. Aquí, se destaca la importancia de prever problemas potenciales, como fallos en el sistema o incompatibilidades, y preparar estrategias para abordarlos. Además, para establecer un nivel de severidad de cada riesgo, primero se determina el impacto potencial que tendría si ocurrieran, así como la probabilidad de cada riesgo en una escala del 1 al 5, siendo 5 el punto más alto. Por último, se multiplica el impacto por la probabilidad de cada riesgo para priorizarlos y en consecuencia decidir las acciones de mitigación.

f. Estrategia de prueba

En esta sección se describe cómo se ejecutarán las pruebas, abarcando todos los aspectos estratégicos, como:

- Tipos de prueba, se detalla las distintas fases o tipos de pruebas.

- Entregables de prueba, se detalla los productos que resultan de las pruebas, como informes o gráficos.
- Técnicas de diseño de prueba, se explica el enfoque metodológico, ya sea caja negra, caja blanca u otra técnica específica.
- Criterios de finalización de prueba, se determina los indicadores que definirán cuándo se consideran completadas las pruebas.
- Métricas, se describe cómo se medirá el rendimiento de las pruebas, como la cantidad de casos probados y defectos encontrados.
- Requisitos del entorno de pruebas, se describe tanto el ambiente de pruebas como las herramientas de pruebas y los recursos técnicos necesarios.
- Re-testing y regresión, se realiza al menos 3 ciclos de pruebas para verificar que los errores identificados se han solucionado correctamente y que la funcionalidad afectada ahora funciona como se esperaba.
- Criterios de suspensión y reanudación, que determinan cuándo detener o reiniciar las pruebas según las circunstancias

g. Actividades de prueba

En esta sección se detalla las actividades específicas que serán realizadas durante las pruebas para asegurar que todas las tareas necesarias están incluidas y programadas de manera eficiente.

h. Personal

El personal es un recurso clave en el proceso de pruebas, por ello se define lo siguiente:

- Roles, actividades y responsabilidades, se asigna claramente las funciones de cada miembro del equipo de pruebas.
- Necesidades de contratación, se indica si se requerirá personal adicional.
- Necesidades de entrenamiento para asegurar que todos los involucrados tengan las habilidades necesarias.

i. Cronograma

Finalmente, se presenta un cronograma detallado con las fechas clave para cada fase del proceso de pruebas, asegurando que todas las partes interesadas conozcan el progreso y los plazos a cumplir. Este cronograma debe ser realista y ajustado a los recursos disponibles.

2.4.6. Diseño de pruebas

La información para los casos de prueba se especifica a continuación, además el diseño de los casos de pruebas se puede describir en listas o tablas.

a. Identificador único del documento

Identifica de forma uniforme la documentación o el diseño de caso de prueba.

El identificador único puede incluir el título de la documentación de ensayo, la fecha de emisión, la versión y/o el estado de la documentación (por ejemplo, borrador, revisado, corregido, final).

b. Alcance

Identifica el alcance de la cobertura de cada uno de los casos de prueba diseñados y describe cualquier inclusión, exclusión, suposición y/o limitación.

c. Referencias

Enumera otros documentos o registros referenciados e identifica los repositorios de información sobre sistemas, software y pruebas. Las referencias pueden separarse en referencias «externas» que se imponen desde fuera de la organización y referencias «internas» que se imponen desde dentro de la organización.

d. Identificador único del caso de prueba

El identificador único de un caso de prueba es un número o código que lo diferencia de todos los demás. Este identificador puede ser creado de forma automática con una herramienta o de manera manual, siguiendo un esquema específico. Generalmente, este identificador no se modifica a lo largo del tiempo, ya que sirve para hacer seguimiento y mantener el control del caso de prueba.

e. Objetivo

Señala y explica de manera simple el propósito o la meta principal del caso de prueba. Esto generalmente se muestra como un título breve.

f. Prioridad

Establece la importancia de los casos de pruebas, si es necesario. Los casos de prueba con mayor prioridad se realizarán primero, siempre que sea posible. Para ello se realiza un análisis de impacto el cual considera factores como:

- Gravedad del fallo: clasificado en alta, media y baja. Esta clasificación nos permite comprender que tan probable es encontrar defectos durante la ejecución de pruebas en un requisito específico.

- Frecuencia de uso: clasificado en alta, media y baja. Esta clasificación nos permite conocer con qué frecuencia se hace uso del requisito analizado y si pudiese tener un impacto significativo en la experiencia del usuario.

Por último, para asignar la prioridad se determina el valor de estos factores de acuerdo con la siguiente tabla 28.

Tabla 28 Prioridad de pruebas

Prioridad	Frecuencia de uso			
	Alta	Media	Baja	
Probabilidad de falla	Alta	Alta	Alta	Media
	Media	Alta	Media	Baja
	Baja	Media	Baja	Baja

Nota: Fuente: Elaboración propia.

g. Condiciones previas

Describe el estado requerido del entorno de prueba y de los datos de prueba, así como cualquier restricción especial relativa a la ejecución del caso de prueba.

h. Entradas

Especifica cada una de las acciones necesarias para llevar el elemento de ensayo a un estado en el que el resultado esperado pueda compararse con los resultados reales. El nivel de detalle de las descripciones debe adaptarse a los conocimientos de los ejecutores de las pruebas.

i. Resultados esperados

Especifica los resultados esperados y el comportamiento requerido del elemento de ensayo en respuesta a las entradas que se dan al elemento de ensayo cuando está en su

estado de precondition. Proporciona los valores esperados (con tolerancias, si procede) para cada salida requerida.

2.4.7. Ejecución de pruebas

a. Identificador único del documento

Identifica de forma uniforme la documentación de la ejecución de las pruebas.

El identificador único puede incluir el título de la documentación de ensayo, la fecha de emisión, la versión y/o el estado de la documentación (por ejemplo, borrador, revisado, corregido, final).

b. Alcance

Identifica el alcance de la ejecución de cada uno de los casos de pruebas planificados y describe las inclusiones, exclusiones, supuestos y limitaciones de las pruebas.

c. Referencias

Enumera otros documentos o registros referenciados e identifica los repositorios de información sobre sistemas, software y pruebas. Las referencias pueden separarse en referencias «externas» que se imponen desde fuera de la organización y referencias «internas» que se imponen desde dentro de la organización.

d. Descripción

Describe lo ocurrido durante cada evento. Puede incluir una referencia al procedimiento de prueba y al caso de prueba que se estaba ejecutando cuando se produjo el evento, si procede.

2.4.8. Seguimiento y control de pruebas

a. Identificador único del documento

Identifica de forma uniforme la documentación, puede incluir el título de la documentación de ensayo, la fecha de emisión, la versión y/o el estado de la documentación (por ejemplo, borrador, revisado, corregido, final).

b. Alcance

Identifica el alcance de la cobertura de las actividades de seguimiento y control durante la fase de pruebas. Se especificó las áreas de prueba que están siendo monitoreadas y controladas, y se describió las inclusiones, exclusiones, suposiciones y/o limitaciones que aplican en esta etapa del proceso de prueba. Esto incluye la identificación de los casos de prueba que fueron supervisados, los defectos encontrados, los recursos involucrados y los criterios de éxito para determinar si las pruebas están progresando según lo planificado.

c. Referencias

Enumera otros documentos o registros referenciados e identifica los repositorios de información sobre sistemas, software y pruebas. Las referencias pueden separarse en referencias «externas» que se imponen desde fuera de la organización y referencias «internas» que se imponen desde dentro de la organización.

d. Estado de la prueba

Proporciona información sobre el estado de las pruebas que incluye información clave sobre el progreso en la ejecución de los casos de prueba, destacando si se han alcanzado los objetivos de prueba, y señalando los casos que están completados, en proceso o pendientes.

e. Período de referencia

Especifica el periodo de tiempo cubierto por el informe, que permite contextualizar los datos presentados y entender a qué fase o ciclo de pruebas pertenecen para un seguimiento efectivo, ya que proporciona claridad sobre cuándo se ejecutarán las pruebas y comparación de resultados.

f. Progreso respecto al plan de pruebas

Describe los progresos realizados con respecto al plan de pruebas. Destaca cualquier desviación notable del plan, con explicación de las razones de la desviación, descripciones de las medidas correctoras, una relación de los efectos y una consideración de las implicaciones con respecto a los objetivos previstos del proyecto.

g. Factores que bloquean el progreso

Identifica los factores que impidieron el progreso durante el periodo del informe y las correspondientes soluciones que se aplicaron para resolverlos. Deben registrarse los problemas pendientes (sin resolver) que siguen impidiendo el progreso e identificarse las posibles soluciones.

h. Medidas de prueba

Presenta las medidas de prueba recopiladas en relación con el periodo del informe. Las medidas de prueba hacen referencia a las métricas y datos cuantitativos recopilados durante el período de prueba. Estas pueden incluir, por ejemplo, el número de casos de prueba ejecutados, el número de defectos encontrados, la cobertura de prueba alcanzada, y el porcentaje de éxito o fallo en los casos de prueba ejecutados.

i. Riesgos nuevos y modificados

Enumera los nuevos riesgos que se han identificado como resultado de las pruebas, así como los cambios en los riesgos existentes durante el periodo del informe.

2.4.9. Documentación

- Planificación: Plan de Pruebas. Ver Anexo B.
- Diseño: Especificación de Casos de Prueba. Ver Anexo C.
- Ejecución, seguimiento y control: Informes de Estado de las Pruebas. Ver Anexo D.
- Evaluación: Informe de Finalización de las pruebas. Ver Anexo E.

2.4.10. Evaluación de resultados de las pruebas

Al finalizar el proceso de ejecución de pruebas, se procede a analizar los resultados de cada una de las pruebas realizadas, para ello se establecen métricas con la finalidad de medir la calidad del software evaluado.

a. Identificador único del documento

Identifica de forma uniforme la documentación de los resultados obtenidos de las pruebas ejecutadas. El identificador único puede incluir el título de la documentación de ensayo, la fecha de emisión, la versión y/o el estado de la documentación (por ejemplo, borrador, revisado, corregido, final).

b. Alcance

Identifica el alcance de los resultados obtenidos durante la ejecución de las pruebas planificadas, enfocándose en la evaluación de la cobertura de pruebas, la madurez de las pruebas, y la densidad de defectos encontrados en los submódulos evaluados. Se

documentan los casos de prueba ejecutados, los casos con resultados satisfactorios y los defectos identificados, proporcionando un análisis detallado de la efectividad del esfuerzo de prueba.

c. Referencias

Enumera otros documentos o registros referenciados e identifica los repositorios de información sobre sistemas, software y pruebas. Las referencias pueden separarse en referencias «externas» que se imponen desde fuera de la organización y referencias «internas» que se imponen desde dentro de la organización.

d. Cobertura de pruebas

La cobertura de pruebas permite medir la efectividad general del esfuerzo de prueba. Un porcentaje de cobertura de prueba más alto generalmente indica que se ha ejecutado un conjunto de pruebas más completo, lo que conduce a una mejor calidad del software. Para establecer ese porcentaje se recomienda la siguiente fórmula:

$$CP = CPE/ CPP$$

- CP es el valor de la cobertura de pruebas.
- CPE es el número de Casos de Prueba que han sido ejecutados.
- CPP es el número total de Casos de Prueba planificados.

Representa la proporción de casos de prueba planificados que se han ejecutado durante el proceso de pruebas.

Alto valor de CP (cercano a 1): Indica que una gran parte de los casos de prueba planificados se han ejecutado, lo que sugiere un esfuerzo de prueba completo y una mayor probabilidad de detectar errores.

Bajo valor de CP (cercano a 0): Indica que solo se ha ejecutado una pequeña fracción de los casos de prueba planificados, lo que sugiere un esfuerzo de prueba incompleto y un mayor riesgo de que queden errores sin detectar.

e. Madurez de pruebas

La madurez de las pruebas permite conocer si las pruebas detectan y previenen errores en el software, para establecer un porcentaje específico se recomienda la siguiente fórmula:

$$MP = CPS/CPR$$

- MP es el valor de madurez de la prueba.
- CPS es el número de casos de prueba con un resultado satisfactorio.
- CPR es el número de casos de prueba diseñados para todos los requisitos.

Esta fórmula calcula la madurez de la prueba (MP) en función de la proporción de casos de prueba con resultados satisfactorios (CPS) en comparación con el número total de casos de prueba diseñados para todos los requisitos (CPR).

Madurez de prueba alta (MP cercana a 1): Indica que una parte importante de los casos de prueba se han ejecutado y han producido resultados satisfactorios. Esto sugiere que el esfuerzo de prueba ha sido exhaustivo y ha cubierto eficazmente los requisitos especificados.

Madurez de prueba baja (MP cercano a 0): Indica que se ha ejecutado una proporción menor de los casos de prueba o que un número mayor de casos de prueba han fallado. Esto sugiere que el esfuerzo de prueba puede no haber sido tan completo o que puede haber problemas con los requisitos o la implementación del software.

f. Densidad de defectos

La siguiente fórmula representa la cantidad de defectos encontrados por cada elemento de especificación revisado durante el proceso de pruebas.

$$DD = TD/CER$$

- DD es la densidad de defectos.
- TD es el número total de defectos encontrados durante las pruebas.
- CER es el número de elementos de especificación revisados.

Alta densidad de defectos (DD alto): Indica que se ha encontrado un número significativo de defectos en relación con la cantidad de especificaciones revisadas. Esto sugiere que el software puede tener problemas de calidad y requiere un mayor esfuerzo de prueba.

Baja densidad de defectos (DD bajo): Indica que se han encontrado pocos defectos en relación con la cantidad de especificaciones revisadas. Esto sugiere que el software tiene una mejor calidad y puede estar listo para su lanzamiento.

Fase de puesta en producción

2.4.11. Planificación de despliegue

La planificación de la puesta en producción comprende las siguientes actividades:

a. Introducción

- Alcance, se detalla los componentes que se van a desplegar y en que entorno se implementaron.

b. Requisitos previos

Antes de que el sistema pueda desplegarse en el entorno de producción, es fundamental detallar lo siguiente:

- Configuraciones en el hardware y software para garantizar un rendimiento adecuado y estable del sistema.
- Base de datos previamente configuradas
- Permisos de acceso establecidos tanto para usuarios como para administradores.
- Dependencias, es decir otros módulos, servicios o componentes externos con los que el sistema necesita interactuar o que deben estar operativos para que el despliegue se complete sin inconvenientes.

c. Equipo de trabajo requerido

El equipo de trabajo es un recurso clave en la puesta en producción, por ello se definió lo siguiente:

- Roles, actividades y responsabilidades, se asigna claramente las funciones de cada miembro del equipo de trabajo.

d. Riesgos

Identificar y documentar los riesgos asociados al proceso de puesta en producción es un paso fundamental para brindar la continuidad y estabilidad del servicio, a continuación, se describe como abordar este aspecto:

- Análisis de impacto, se analiza antes del despliegue, el impacto potencial que cualquier interrupción o fallo del despliegue podría tener en el sistema. Esto incluye entender las dependencias entre servicios y la criticidad del sistema en producción.
- Tipo de riesgo, en el registro de riesgos hallados se los debe clasificar como técnicos, operacionales, seguridad, dependencias, tiempo y disponibilidad.
- Evaluación de riesgos, a cada riesgo registrado se asigna un nivel de probabilidad (alto, medio, bajo) según la frecuencia con la que se ha observado en despliegues previos o la probabilidad de que ocurra en el entorno actual. De la misma forma se califica el impacto (bajo, medio, alto) que tendrá cada riesgo considerando el potencial de interrupciones y afectaciones al sistema.
- Estrategias de mitigación, se detalla las acciones específicas para reducir la probabilidad de que el riesgo ocurra.

e. Cronograma

Describir un cronograma que cubra todas las etapas y actividades necesarias para implementar el sistema en el entorno de producción sin interrupciones significativas en el servicio. El cronograma debe especificar fechas y tiempos exactos para cada actividad, asignando responsabilidades claras y puntos de revisión donde se valide el avance del proceso.

2.4.12. Control de cambios

Cualquier cambio en la puesta en producción debe ser gestionado a través de un control de cambios formal, el cual se lo puede registrar en listas o tablas y debe incluir:

a. Historial de cambios

- Naturaleza del cambio, se describe el tipo de cambio que se realizó, es decir una actualización, modificación o implementación de una nueva funcionalidad.
- Impacto, se explica cómo se espera que el cambio afecte al sistema o a los usuarios.
- Fecha de aprobación, se detalla la fecha en la que el cambio fue aprobado para realizarse. Esto es importante para rastrear el momento en que se permitió el cambio, lo cual puede ser relevante para auditorías o revisiones posteriores.

2.4.13. Procedimientos de despliegue

El procedimiento de despliegue debe incluir una serie de pasos que aseguran la implementación ordenada y controlada del software en el entorno de producción, detallados a continuación:

a. Ejecución de despliegue

Especificar los pasos para la transferencia del software al entorno de producción, incluyendo la configuración de bases de datos, la integración de servicios y las conexiones necesarias con otros sistemas. Esto debe hacerse de forma que se minimicen las interrupciones a los usuarios.

b. Plan de retroceso

Como parte del procedimiento, debe existir un plan para revertir los cambios en caso de que el despliegue no sea exitoso. Este plan debe incluir pasos claros para restaurar el sistema a su estado anterior sin pérdida de datos ni impacto significativo en el usuario final.

2.4.14. Cierre de despliegue

El cierre de despliegue comprende las siguientes actividades:

a. Informe de cierre de despliegue

- Resultados del despliegue, se realiza un resumen de los logros del despliegue, indicando si se cumplieron todos los objetivos planteados y cualquier desviación del plan original.
- Incidencias, se registra cualquier problema o incidencia que surgió durante el despliegue, así como las acciones tomadas para resolverlas. Este registro es esencial para prevenir problemas similares en futuros despliegues.

b. Lecciones aprendidas y recomendaciones

- Identificación de obstáculos, se detalla los obstáculos que dificultaron el despliegue y sugerencias sobre cómo abordarlos en el futuro.
- Practicas exitosas, se menciona las prácticas que podrían convertirse en estándares para futuros despliegues,
- Recomendaciones sobre la documentación adicional o ajustes al plan de despliegue para hacerlo más eficiente y menos propenso a problemas.

CAPÍTULO 3

Propuesta operativa

Para garantizar la correcta aplicación de la guía metodológica en el desarrollo de software del DDTI, se han definido y estructurado las actividades esenciales que deben seguirse. Estas incluyen tanto su integración en proyectos específicos como la posibilidad de automatizar el proceso formal de pruebas, promoviendo así una mayor eficiencia y consistencia. Además, se han detallado los insumos necesarios para cada etapa del proceso de ejecución de pruebas, asegurando un enfoque sistemático y alineado con los objetivos de calidad establecidos.

En el anexo A, se presentan las acciones llevadas a cabo para identificar los principales hallazgos en los portafolios docente y estudiante. Estos hallazgos se traducen en recomendaciones específicas, las cuales se proponen como base para su implementación en futuros trabajos de investigación. De esta manera, se busca contribuir al desarrollo continuo y la mejora de los procesos relacionados con la calidad del software en el contexto académico.

A continuación, se describe el procedimiento de ejecución de pruebas y puesta en producción de calidad de software para ser aplicado en el DDTI:

3.1. Planificación de pruebas

- Identificador único del documento

Tabla 29 Identificación de documentación - plan de pruebas

Versión	Autores	Estado del documento (Borrador/Revisado/Corregido/Final)	Fecha

Nota: Fuente: Elaboración propia

- Introducción

Alcance: [Describa lo que cubre el plan de pruebas, incluyendo componentes específicos del software o sistema.]

Referencias: [Liste las normativas, estándares y documentación previa relevante.]

Glosario: [Defina términos técnicos clave para garantizar una comprensión uniforme.]

- Contexto de las pruebas

Proyecto a evaluar: [Describa el sistema en general y las partes específicas que serán evaluadas.]

Elementos de prueba: [Enumere los componentes específicos del software/hardware objeto de evaluación.]

Tabla 30 Formato para listar los requisitos funcionales

Id	RF-001
Nombre	
Descripción	
Entrada	
Salida	
Precondición	
Postcondición	

Nota: Fuente: Elaboración propia

Alcance de las pruebas: [Especifique los límites que serán objeto de evaluación del sistema y de los componentes incluidos.]

Suposiciones y restricciones: [Enumere las limitaciones que puedan afectar el proceso de pruebas, como recursos o plazos. Podrían ser reconocidas en aspectos operativos, técnicos, económicos e infraestructura]

Partes interesadas: [Identifique desarrolladores, clientes o usuarios clave involucrados.]

Tabla 31 Partes interesadas

Partes interesadas (director, Líder de pruebas, equipo de desarrollo, usuarios finales, equipo de soporte técnico)	Responsabilidades

Nota: Fuente: Elaboración propia

- Comunicación de las pruebas

En la comunicación de pruebas se describe el tipo de reunión para intercambiar información entre los participantes clave, con un propósito clave, utilizando medios adecuados y con frecuencia preestablecida.

Tabla 32 Comunicación de las pruebas

Punto de Comunicación	Propósito	Frecuencia	Medios	Responsable	Audiencia

Nota: Fuente: Elaboración propia

- Registro de riesgos

Tabla 33 Registro de riesgos

Riesgos	Probabilidad (1-5)	Impacto (1-5)	Severidad (Probabilidad * Impacto)	Plan de Mitigación

Nota: Fuente: Elaboración propia

- Estrategia de pruebas

Tipos de prueba: [Describa los tipos de pruebas, entre ellas las que establece la ISO 29119, pruebas dinámicas o estáticas]

Entregables de prueba: [Liste la documentación esperada de las pruebas, puede ser plan de pruebas, especificación de casos de prueba, informe de ejecución de pruebas, informe de evaluación de resultados]

Técnicas de diseño de prueba: [Detalle las técnicas de pruebas para el diseño de casos de prueba a aplicar según la ISO 29119.]

Criterios de finalización: [Defina los criterios para considerar las pruebas como completadas.]

Tabla 34 Criterios de finalización

Criterios (casos de prueba, defectos, ciclos de re-testing y regresión)	Estado de la prueba (Completas/Incompletas)

Nota: Fuente: Elaboración propia

Métricas: [Especifique las métricas a evaluar, como número de casos probados o defectos encontrados.]

Tabla 35 Métricas

Métricas (Cobertura, Defectos, Efectividad, Calidad)	Dimensión (% de requisitos funcionales cubiertos, número de defectos encontrados, % de ejecución de casos de prueba, % de detección de defectos)

Nota: Fuente: Elaboración propia

Requisitos del entorno: [Describa el ambiente, herramientas y recursos necesarios para las pruebas.]

Tabla 36 Ambiente de pruebas

Componente (Sistema Operativo, Navegadores, Velocidad de internet)	Requisitos

Nota: Fuente: Elaboración propia

Tabla 37 Herramientas

Herramientas	Función	Uso específico

Nota: Fuente: Elaboración propia

Re-testing y regresión: [Explique el proceso de repetición de pruebas en caso de errores.]

Tabla 38 Proceso de re-testing y regresión

Casos de prueba	Ciclo 1 (Ejecuta casos de prueba: detecta y registra errores.)	Ciclo 2 (pruebas de confirmación: verificar que los errores hallados se han solucionado correctamente)	Ciclo 3 (pruebas de regresión: confirmar si los cambios no afectaron al sistema)

Nota: Fuente: Elaboración propia

Criterios de suspensión y reanudación: [Especifique las condiciones para detener o reiniciar las pruebas.]

Tabla 39 Criterios de suspensión y reanudación

Criterios	Descripción
Suspensión	Defectos críticos

	Ambiente de pruebas inestable No disponibilidad de datos de prueba Falta de recursos Superación del límite de tiempo o presupuesto
Reanudación	Resolución de defectos críticos Estabilidad del entorno de pruebas Disponibilidad de datos de prueba Reasignación de recursos clave Restauración del cronograma

Nota: Fuente: Elaboración propia

- Actividades de prueba

Tabla 40 Actividades de prueba

Actividad	Descripción	Responsable	Fecha de inicio y fecha de fin	Estado

Nota: Fuente: Elaboración propia

- Personal

Roles y responsabilidades: [Describa las funciones asignadas a cada miembro del equipo.]

Necesidades de contratación: [Indique si se requiere personal adicional.]

Necesidades de entrenamiento: [Detalle las capacitaciones necesarias.]

- Cronograma

Tabla 41 Cronograma

Fase	Actividad	Responsable	Fecha de inicio y fecha de fin	Comentarios

Nota: Fuente: Elaboración propia

3.2. Diseño de pruebas

- Identificador único del documento

Tabla 42 Identificación de documento - Diseño de casos de prueba

Versión	Autores	Estado del documento (Borrador/Revisado/Corregido/Final)	Fecha

Nota: Fuente: Elaboración propia

- Alcance

Cobertura de las pruebas: [Describa los aspectos incluidos y excluidos para el diseño de los casos de pruebas, así como las suposiciones y limitaciones existentes para su diseño.]

Tabla 43 Criterios de suposiciones y limitaciones - casos de prueba

Criterios	Descripción
Suposiciones	Disponibilidad de requisitos claros Ambiente de pruebas estable Datos de prueba adecuados Disponibilidad del equipo de trabajo Acceso a módulos o componentes del sistema
Limitaciones	Restricciones de tiempo Alcance limitado de pruebas Dependencias externas Falta de documentación completa Recursos limitados

Nota: Fuente: Elaboración propia

- Referencias

Referencias internas: [Documentos y políticas organizacionales relevantes.]

Referencias externas: [Normas o documentos externos utilizados como base.]

- Lista de casos de pruebas

Tabla 44 Formato para listar los casos de prueba

Caso de Prueba	CP-001	Prioridad Requisito funcional	
Título		Objetivo	
Tipo de prueba			
Descripción			
Entradas			
Condiciones			
Resultados esperados			

Nota: Fuente: Elaboración propia

Identificador único del caso de prueba: [Ejemplo: TC001]

Objetivo: [Describa el propósito o meta principal de este caso de prueba.]

Prioridad: [Para asignar la prioridad al caso de prueba, utilizar la tabla 28 de la guía metodológica.]

Condiciones previas: [Indique el estado del entorno, datos de prueba, y restricciones.]

Entradas: [Especifique cada una de las acciones necesarias para preparar el caso, es decir los valores específicos que se ingresan para validar una funcionalidad]

Resultados esperados: [Detalle los valores esperados para cada salida, con tolerancias si corresponde.]

3.3. Ejecución de pruebas

- Identificador único del documento

Tabla 45 Identificación de documento - ejecución de pruebas

Versión	Autores	Estado del documento	Fecha
---------	---------	----------------------	-------

		(Borrador/Revisado/Corregido/Final)	

Nota: Fuente: Elaboración propia

- Alcance

Cobertura de la ejecución de pruebas: [Describa las pruebas incluidas y excluidas en la ejecución.]

Suposiciones: [Enumere las suposiciones relacionadas con la ejecución.]

Limitaciones: [Mencione cualquier restricción que afecte la ejecución de las pruebas.]

Tabla 46 Criterios de suposiciones y limitaciones - ejecución de pruebas

Criterios	Descripción
Suposiciones	Disponibilidad del entorno de prueba Integridad de los datos de prueba Disponibilidad del sistema Soporte técnico disponible
Limitaciones	Restricciones de tiempo Recursos técnicos insuficientes Inestabilidad del sistema Dependencias externas Personal limitado

Nota: Fuente: Elaboración propia

- Referencias

Referencias internas: [Documentos y procedimientos organizacionales relacionados.]

Referencias externas: [Normas, guías o documentación externa utilizada.]

- Registro de ejecución de pruebas

En esta sección se registra cada evento de la ejecución, haciendo referencia a los casos de prueba y procedimientos aplicados.

Tabla 47 Registro de ejecución de pruebas

Identificador del caso de prueba	Descripción del evento	Resultado obtenido	Estado (Aprobado/Fallido)	Observaciones

Nota: Fuente: Elaboración propia

- Resumen general de la ejecución de pruebas

Tabla 48 Casos de prueba ejecutados

Total, de casos ejecutados	Aprobados	Fallidos	En revisión
Ciclo 1			
Ciclo 2			
Ciclo 3			

Nota: Fuente: Elaboración propia

3.4. Seguimiento y control de pruebas

- Identificador único del documento

Tabla 49 Identificador de documento - seguimiento y control de pruebas

Versión	Autores	Estado del documento (Borrador/Revisado/Corregido/Final)	Fecha

Nota: Fuente: Elaboración propia

- Alcance

Cobertura: [Especifique las áreas de prueba monitoreadas y controladas.]

Casos supervisados: [Detalle los casos de prueba monitoreados y sus características.]

Defectos encontrados: [Proporcione una lista de defectos detectados durante la supervisión.]

Recursos involucrados: [Identifique los recursos clave para esta fase.]

Criterios de éxito: [Defina los criterios utilizados para evaluar el progreso y éxito de las pruebas.]

Tabla 50 Criterios de éxito

Criterios	Descripción
Éxito	% de cobertura de casos de prueba planificada alcanzada % de defectos altos resueltos % de éxito de casos de prueba ejecutados que cumplen con los resultados esperados

Nota: Fuente: Elaboración propia

- Referencias

Referencias internas: [Documentos, procedimientos o estándares organizacionales relacionados.]

Referencias externas: [Normas, estándares o guías externas utilizadas.]

- Estado de la prueba

Tabla 51 Estado de las pruebas

Caso de prueba	Estado (Completo/En proceso, pendiente)	Observaciones

Nota: Fuente: Elaboración propia

- Período de la prueba

Fecha de inicio: [Especifique el inicio del periodo de referencia.]

Fecha de fin: [Especifique el fin del periodo de referencia.]

Fase/ciclo: [Indique la fase o ciclo de pruebas que corresponde.]

- Progreso respecto al plan de pruebas

Tabla 52 Progreso del plan de pruebas

Actividad planeada	Progreso %	Desviación (SI/NO)	Causa de la desviación	Medidas correctivas

Nota: Fuente: Elaboración propia

- Factores que bloquean el progreso

Tabla 53 Lista de factores

Bloqueo	Impacto	Acción correctiva aplicada	Estado actual (Resuelto/en proceso)

Nota: Fuente: Elaboración propia

- Medidas de prueba

Tabla 54 Métricas para la ejecución de pruebas

Métricas	Valor
Total, de casos de prueba ejecutados	[Número]
Total de defectos encontrados	[Número]
Cobertura de pruebas alcanzada	[Porcentaje]
Porcentaje de fallos	[Porcentaje]
Porcentaje de éxito	[Porcentaje]

Nota: Fuente: Elaboración propia

- Riesgos nuevos y modificados

Tabla 55 Lista de nuevos riesgos

Riesgo	Descripción	Estado (Nuevo/Modificado)	Impacto (Alto/Medio/Bajo)	Acción de mitigación

Nota: Fuente: Elaboración propia

3.5. Planificación de despliegue

- Alcance

Componentes a desplegar: [Detalle los módulos, servicios o componentes del sistema.]

Entorno de implementación: [Describa el entorno donde se realizará el despliegue, por ejemplo, producción, pruebas, etc.]

- Requisitos previos

Tabla 56 Lista de requisitos previos para el despliegue

Requisito	Descripción	Estado (Cumplido/En proceso)
Configuración de hardware	[Describa las configuraciones específicas necesarias para el hardware.]	[Estado]
Configuración de software	[Especifique las configuraciones del software requeridas.]	[Estado]
Base de datos	[Indique que las bases de datos están configuradas y listas para el despliegue.]	[Estado]
Permisos de acceso	[Detalle los permisos establecidos para usuarios y administradores.]	[Estado]
Dependencias	[Enumere los servicios, módulos o componentes externos necesarios.]	[Estado]

Nota: Fuente: Elaboración propia

- Equipo de trabajo requerido

Tabla 57 Lista de los miembros del equipo

Rol	Responsabilidades	Nombre del responsable
[Ejemplo: Administrador]	[Definir configuraciones del servidor]	[Nombre]
[Ejemplo: Desarrollador]	[Solucionar problemas técnicos]	[Nombre]

[Ejemplo: Tester]	[Validar el correcto funcionamiento]	[Nombre]
-------------------	--------------------------------------	----------

Nota: Fuente: Elaboración propia

- Análisis de riesgos

Tabla 58 Lista de riesgos - planificación de despliegue

Riesgo	Tipo	Probabilidad (Alto/Medio/Bajo)	Impacto (Alto/Medio/Bajo)	Estrategia de mitigación
[Ejemplo: Falla de servidor]	Técnico	Alto	Alto	[Revisión previa del hardware]
[Ejemplo: Permisos incorrectos]	Seguridad	Medio	Medio	[Verificación de configuraciones]

Nota: Fuente: Elaboración propia

- Cronograma

Tabla 59 Cronograma - planificación de despliegue

Actividad	Descripción	Fecha y hora	Responsable	Estado (Pendiente/En curso/Completado)
Preparación del entorno	Configuración de hardware y software necesarios	[Fecha y hora]	[Nombre]	[Estado]
Validación de dependencias	Comprobación de servicios o módulos relacionados	[Fecha y hora]	[Nombre]	[Estado]
Despliegue inicial	Implementación del sistema en el entorno objetivo	[Fecha y hora]	[Nombre]	[Estado]
Validación post-despliegue	Pruebas finales de funcionalidad	[Fecha y hora]	[Nombre]	[Estado]
Documentación final	Registro de resultados del despliegue	[Fecha y hora]	[Nombre]	[Estado]

Nota: Fuente: Elaboración propia

3.6. Control de cambios

- Identificador único del documento

Tabla 60 Identificación del documento - control de cambios

Versión	Autores	Estado del documento (Borrador/Revisado/Corregido/Final)	Fecha

Nota: Fuente: Elaboración propia

- Historial de cambios

Tabla 61 Lista del historial de cambios

ID del cambio	Naturaleza del cambio (Actualización/Modificación/Nueva funcionalidad)	Impacto en el sistema/usuarios	Fecha de aprobación	Responsable

Nota: Fuente: Elaboración propia

ID del Cambio: [Ejemplo: CHG-001]

Naturaleza del Cambio: [Describa si el cambio es una actualización, modificación o nueva funcionalidad.]

Impacto: [Explique cómo se espera que el cambio afecte al sistema o a los usuarios, incluyendo cualquier posible riesgo.]

Fecha de Aprobación: [Indique la fecha en que el cambio fue aprobado para implementación.]

Responsable: [Nombre del responsable de gestionar o implementar el cambio.]

- Resumen de cambios totales

Tabla 62 Lista del estado de los cambios

Total, de cambios registrados	Cambios completados	Cambios pendientes	Cambios rechazados

Nota: Fuente: Elaboración propia

3.7. Procedimientos de despliegue

- Identificador único del documento

Tabla 63 Identificación del documento - procedimiento de despliegue

Versión	Autores	Estado del documento (Borrador/Revisado/Corregido/Final)	Fecha

Nota: Fuente: Elaboración propia

- Ejecución de despliegue

Pasos para la transferencia del software al entorno de producción:

Tabla 64 Pasos para la ejecución de despliegue

Paso	Descripción	Responsable	Estado (Pendiente/En curso/ Completado)
1	Configuración de bases de datos	[Nombre responsable] del	[Estado]
2	Integración de servicios	[Nombre responsable] del	[Estado]
3	Configuración de conexiones con sistemas externos	[Nombre responsable] del	[Estado]
4	Validación del entorno	[Nombre responsable] del	[Estado]
5	Despliegue del software	[Nombre responsable] del	[Estado]
6	Pruebas iniciales post-despliegue	[Nombre responsable] del	[Estado]

Nota: Fuente: Elaboración propia

- Plan de retroceso

Pasos para revertir el despliegue en caso de fallos, tomar en cuenta lo siguiente: si los problemas detectados están relacionados con funcionalidades no probadas adecuadamente, es posible retroceder hasta la fase de pruebas para realizar evaluaciones adicionales y corregir los defectos o si el despliegue fallido afecta el sistema en producción, se ejecuta una restauración basada en configuraciones, datos o versiones previas.

Tabla 65 Pasos para seguir el plan de retroceso

Paso	Descripción	Responsable	Estado (Pendiente/En curso/ Completado)
1	Identificar y documentar el fallo	[Nombre responsable] del	[Estado]
2	Restaurar las configuraciones anteriores	[Nombre responsable] del	[Estado]
3	Reconectar sistemas y servicios	[Nombre responsable] del	[Estado]
4	Reconectar bases de datos al estado previo	[Nombre responsable] del	[Estado]
5	Validar que el sistema este operativo nuevamente	[Nombre responsable] del	[Estado]
6	Documentar el incidente y realizar análisis posterior	[Nombre responsable] del	[Estado]

Nota: Fuente: Elaboración propia

3.8. Cierre de despliegue

- Identificador único del documento

Tabla 66 Identificador de documento - cierre de despliegue

Versión	Autores	Estado del documento (Borrador/Revisado/Corregido/Final)	Fecha

Nota: Fuente: Elaboración propia

- Informe de cierre de despliegue

Tabla 67 Resultados del Despliegue

Objetivo (Detallar si todos los objetivos planteados se cumplieron)	Estado (Cumplido/No cumplido)	Observaciones

Nota: Fuente: Elaboración propia

Tabla 68 Incidencias en el despliegue

Incidencias (Problemas de funcionalidad, errores en la configuración del entorno, errores de integración, interrupciones del servicio)	Descripción	Acciones tomadas	Estado (Resuelto/Pendiente)

Nota: Fuente: Elaboración propia

- Lecciones aprendidas y recomendaciones

Tabla 69 Identificación de obstáculos

Obstáculo	Descripción	Sugerencia para mitigación futura

Nota: Fuente: Elaboración propia

Tabla 70 Practicas exitosas

Practica exitosa	Descripción

Nota: Fuente: Elaboración propia

Recomendaciones

[Recomendación 1 sobre ajustes al plan de despliegue.]

[Recomendación 2 sobre documentación adicional necesaria.]

CAPÍTULO 4

Validación

4.1. Prueba de concepto

La prueba de concepto que se aplicó es una validación teórica de la guía metodológica de calidad de software. Esta prueba se enfoca en recoger la opinión experta de profesionales en ingeniería de software y calidad, con el fin de evaluar si la guía cumple con los estándares y prácticas necesarias para su implementación en la Dirección de Desarrollo Tecnológico e Informático (DDTI) de la UTN.

Este tipo de prueba de concepto busca principalmente verificar si la guía propuesta es clara, aplicable y alineada con las normas ISO (ISO 29119 para pruebas de software e ISO 20000-1 para puesta en producción), según la evaluación de los expertos. Aunque no se ejecuta un proceso práctico o técnico en un ambiente de desarrollo real, se utiliza el conocimiento especializado de los docentes para validar la solidez y viabilidad de la guía en su contexto.

Objetivo de esta prueba de concepto:

- Evaluar la viabilidad teórica de la guía metodológica.
- Recoger recomendaciones y retroalimentación.
- Validar la aplicabilidad de las normas ISO mencionadas.
- Determinar si la guía puede mejorar los procesos de calidad de software en el DDTI.

4.2. Revisión por pares

Para validar la calidad y aplicabilidad de la guía metodológica, se utilizó la revisión por pares que consiste en un proceso crítico de evaluación en el que expertos en un campo específico

revisan y critican un trabajo, proyecto o propuesta antes de su publicación o implementación. Este proceso es fundamental en muchas disciplinas, especialmente en la investigación académica.

La revisión por pares sugiere los siguientes pasos para su aplicación, detallados en la tabla 71

Tabla 71 Revisión por pares

Revisión por pares	
Paso 1: Selección de Expertos	<p>Criterios de Selección</p> <p>Experiencia mínima de 5 años en calidad de software o ingeniería de software. Conocimiento práctico en la aplicación de estándares ISO relevantes. Experiencia en educación o formación en ingeniería de software.</p> <hr/> <p>Identificación de Expertos</p> <p>Contactar a docentes de la Facultad de Ingeniería y Ciencias Aplicadas de la UTN. Incluir profesionales de la industria con experiencia en proyectos de software.</p> <hr/> <p>Número de Expertos</p> <p>Seleccionar entre 3 y 7 expertos para equilibrar la diversidad de opiniones y la manejabilidad del proceso.</p>
Paso 2: Preparación de la Revisión	<p>Documentación</p> <p>Preparar la guía metodológica en un formato claro y accesible ya sea impreso o digital. Incluir un resumen de los objetivos de la guía, los principales procesos y las normas ISO aplicadas.</p> <hr/> <p>Formato de Evaluación</p> <p>Crear un cuestionario estructurado que incluya preguntas específicas sobre la claridad, aplicabilidad, y efectividad de la guía. Incluir preguntas abiertas para recibir comentarios detallados.</p>
Paso 3: Presentación a los Expertos	<p>Organización de la Sesión</p> <p>Programar una reunión presencial o virtual (por ejemplo, a través de Zoom) para presentar la guía.</p>

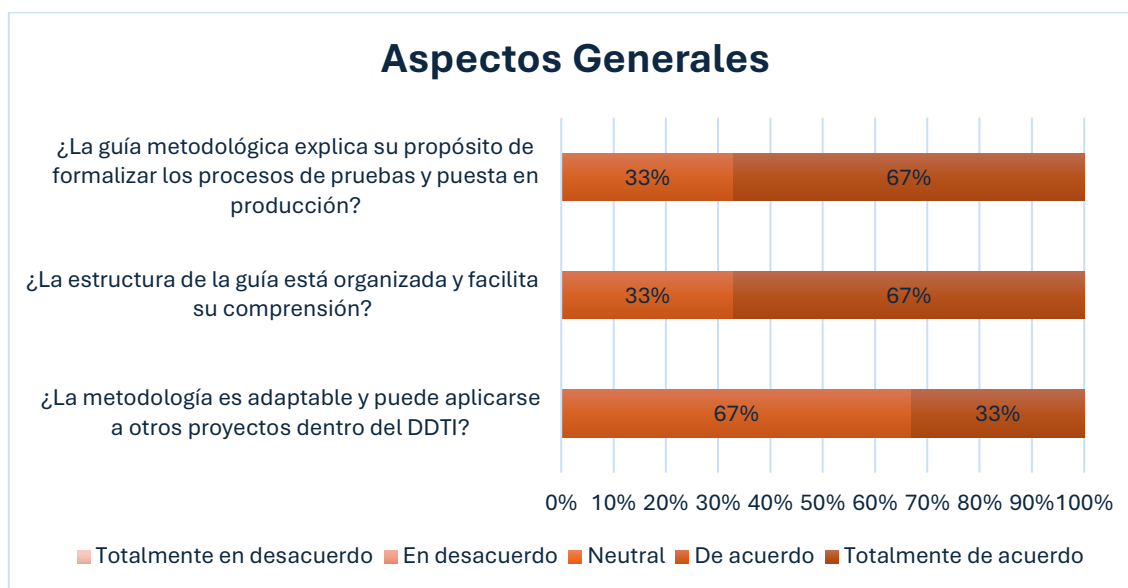
		Duración de la presentación: 30-45 minutos, seguida de una sesión de preguntas y respuestas.
	Presentación	Resumir los objetivos de la guía y las ISO utilizadas. Explicar el proceso de revisión y la importancia del feedback de los expertos.
	Distribución del Cuestionario	Enviar el cuestionario a los expertos después de la presentación, con un plazo de 15 minutos para completarlo. Incluir instrucciones claras sobre cómo completar el cuestionario.
Paso 4: Recolección de Feedback	Formato de Cuestionario	Preguntas de Escala de Likert (1-5): Las preguntas deben estar en un orden específico y por bloques. Es fundamental iniciar con preguntas sencillas y gratas. Luego, avanzar con preguntas más complejas, las cuales suelen ser las más importantes y reveladoras. Por último, añadir preguntas abiertas para obtener un mejor feedback de los expertos seleccionados.
	Compilación de Respuestas	Reunir las respuestas del cuestionario y clasificarlas por tipo (escala y abiertas).
Paso 5: Análisis de Resultados	Análisis Cuantitativo	Calcular promedios y porcentajes para las respuestas de la escala de Likert. Identificar tendencias o áreas problemáticas.
	Análisis Cualitativo	Revisar las respuestas abiertas para extraer temas recurrentes y sugerencias de mejora.
Paso 6: Presentación de Resultados	Informe de Revisión	Elaborar un informe que resuma los hallazgos y el feedback recibido. o Incluir recomendaciones específicas basadas en las opiniones de los expertos.

Nota: Fuente:

4.3. Resultados de la prueba de concepto

4.3.1. Análisis de resultados

Figura 25 Gráfico de barras apiladas - Aspectos generales



Nota: fuente: Elaboración propia

Pregunta 1. ¿La guía metodológica explica su propósito de formalizar los procesos de pruebas y puesta en producción?

El 67% de los encuestados indicó estar totalmente de acuerdo en que la guía metodológica explica claramente su propósito de formalizar los procesos de pruebas y puesta en producción, mientras que el 33% restante señaló estar de acuerdo. Esto refleja que la guía logra transmitir de forma adecuada su objetivo principal, evidenciando que su diseño y redacción son comprensibles y alineados con las expectativas de los evaluadores.

Pregunta 2. ¿La estructura de la guía está organizada y facilita su comprensión?

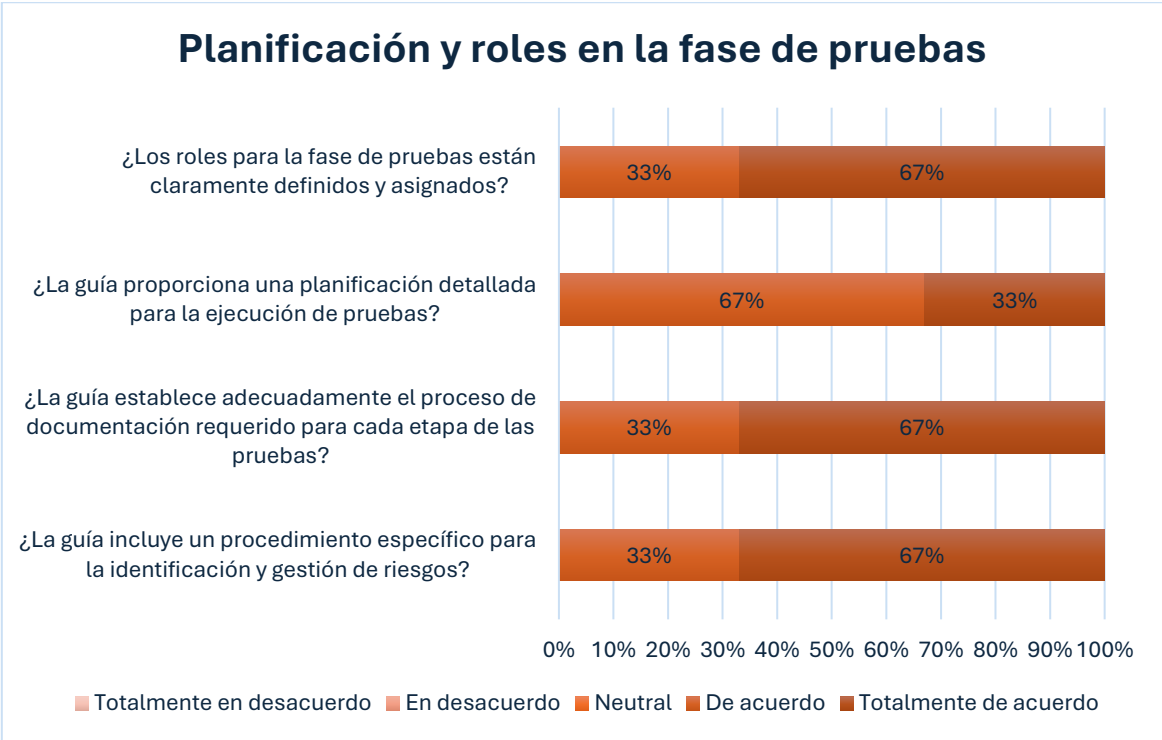
Los resultados muestran que el 67% de los evaluadores están totalmente de acuerdo con que la guía está organizada de forma que facilita su comprensión, mientras que el 33% se encuentra de acuerdo. Este nivel de aceptación total sugiere que la estructura de la guía ha

sido diseñada de manera lógica y fluida, permitiendo a los usuarios acceder al contenido sin dificultades.

Pregunta 3. ¿La metodología es adaptable y puede aplicarse a otros proyectos dentro del DDTI?

En cuanto a la adaptabilidad de la metodología a otros proyectos, el 67% de los encuestados se mostró de acuerdo, y el 33% indicó estar totalmente de acuerdo. Aunque estos resultados reflejan una valoración positiva, el hecho de que un porcentaje mayoritario no haya seleccionado la opción de "totalmente de acuerdo" podría señalar que algunos evaluadores perciben limitaciones en la flexibilidad de la metodología para ajustarse a otros contextos o proyectos del DDTI.

Figura 26 Gráfico de barras apiladas - Planificación y roles de pruebas



Nota: fuente: Elaboración propia

Pregunta 4. ¿Los roles para la fase de pruebas están claramente definidos y asignados?

El 67% de los encuestados indicó estar totalmente de acuerdo en que los roles para la fase de pruebas están claramente definidos y asignados, mientras que el 33% está de acuerdo. Esto refleja que la guía metodológica aborda de manera efectiva la asignación de responsabilidades, lo que facilita la ejecución de las actividades de prueba.

Pregunta 5. ¿La guía proporciona una planificación detallada para la ejecución de pruebas?

En este caso, el 67% de los evaluadores se mostró de acuerdo, mientras que el 33% indicó estar totalmente de acuerdo. Si bien la mayoría reconoce que la guía incluye una planificación detallada, el porcentaje menor que seleccionó "totalmente de acuerdo" podría indicar que aún hay margen para hacer más explícitos ciertos elementos de la planificación, como cronogramas, hitos clave o recursos necesarios.

Pregunta 6. ¿La guía establece adecuadamente el proceso de documentación requerido para cada etapa de las pruebas?

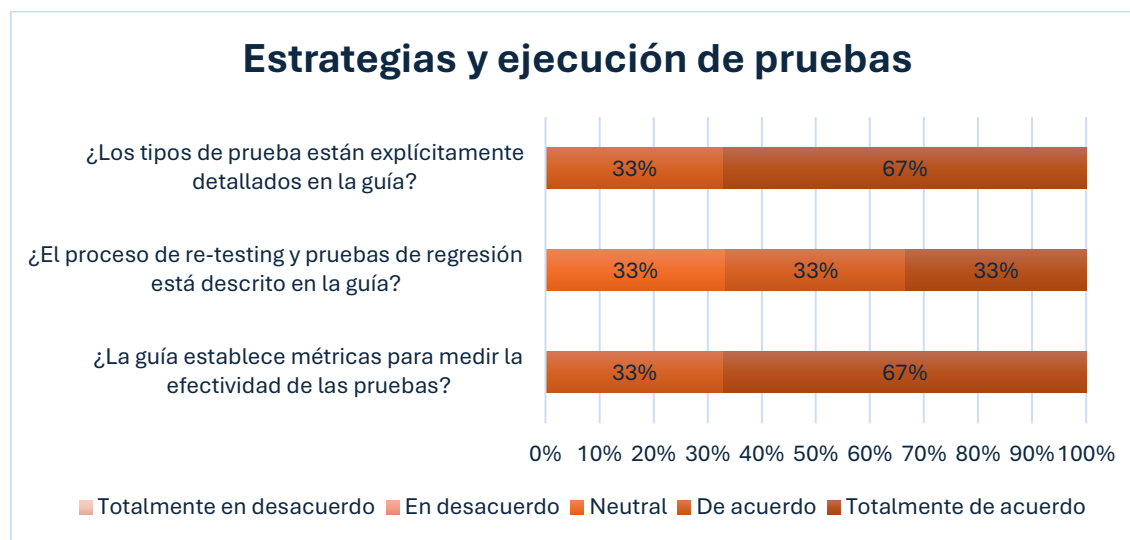
El 67% de los encuestados manifestó estar de acuerdo con que la guía define de manera adecuada el proceso de documentación en cada etapa de las pruebas, mientras que el 33% está totalmente de acuerdo. Aunque los resultados son positivos, el menor porcentaje en la categoría de "totalmente de acuerdo" podría sugerir que algunos aspectos del proceso de documentación no están completamente desarrollados o claros.

Pregunta 7. ¿La guía incluye un procedimiento específico para la identificación y gestión de riesgos?

El 67% de los participantes se encuentra de acuerdo en que la guía incluye un procedimiento específico para la gestión de riesgos, mientras que el 33% está totalmente de acuerdo. Este resultado

sugiere que, aunque la guía aborda la gestión de riesgos, algunos evaluadores pueden percibir que la información proporcionada no es lo suficientemente detallada o práctica.

Figura 27 Gráfico de barras apiladas - Estrategia y ejecución de pruebas



Nota: Fuente: Elaboración propia

Pregunta 8. ¿Los tipos de prueba están explícitamente detallados en la guía?

El 67% de los encuestados está de acuerdo en que los tipos de prueba están explícitamente detallados en la guía, mientras que el 33% indicó estar totalmente de acuerdo. Estos resultados reflejan que la guía aborda de manera efectiva los distintos tipos de pruebas requeridos en el proceso de validación de software.

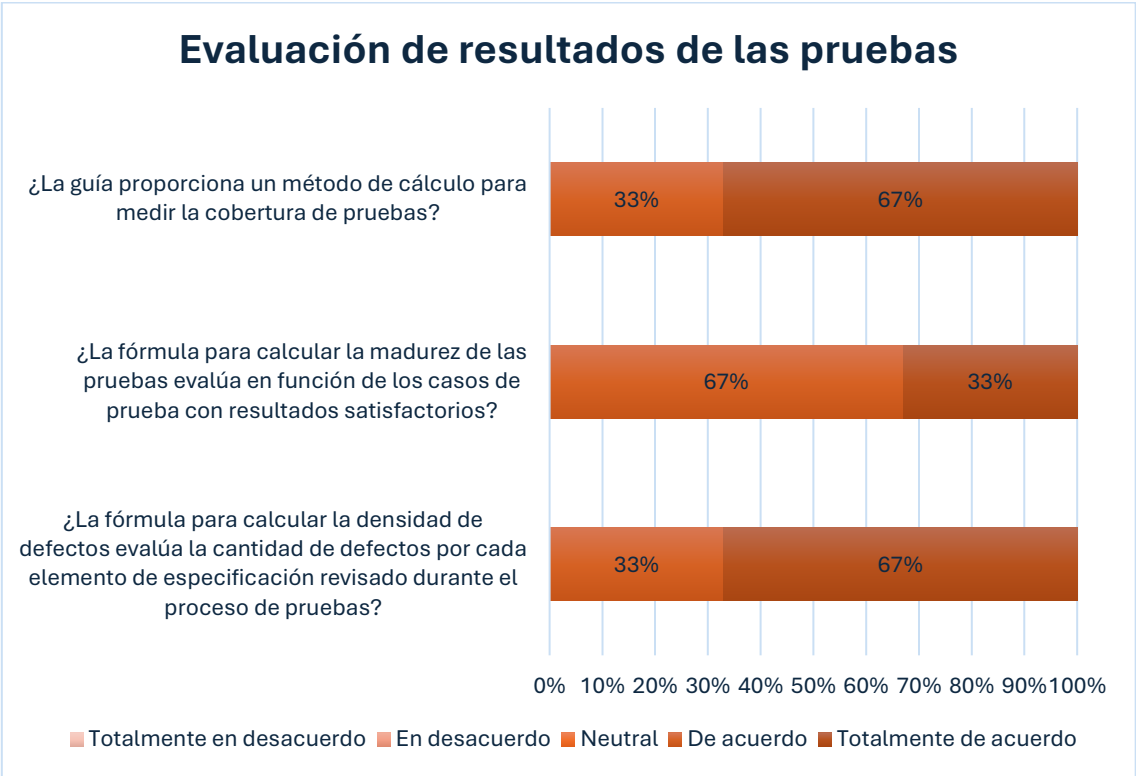
Pregunta 9. ¿El proceso de re-testing y pruebas de regresión está descrito en la guía?

En este caso, el 33% de los evaluadores se encuentra de acuerdo y otro 33% totalmente de acuerdo, mientras que un 33% permaneció neutral. Este equilibrio entre respuestas positivas y neutrales indica que, aunque el proceso de re-testing y pruebas de regresión está incluido en la guía, no todos los usuarios perciben que está completamente desarrollado o explicado.

Pregunta 10. ¿La guía establece métricas para medir la efectividad de las pruebas?

El 67% de los encuestados indicó estar de acuerdo con que la guía establece métricas para medir la efectividad de las pruebas, mientras que el 33% está totalmente de acuerdo. Esto evidencia que la guía incluye indicadores útiles para evaluar la efectividad del proceso de pruebas, pero hay espacio para mejorar.

Figura 28 Gráfico de barras apiladas - Evaluación de resultados de las pruebas



Nota: Fuente: Elaboración propia

Pregunta 11. ¿La guía proporciona un método de cálculo para medir la cobertura de pruebas?

El 67% de los encuestados indicó estar de acuerdo con que la guía incluye un método de cálculo para medir la cobertura de pruebas, mientras que el 33% expresó estar totalmente de acuerdo. Estos resultados demuestran que la guía satisface, en gran medida, las expectativas relacionadas con este aspecto.

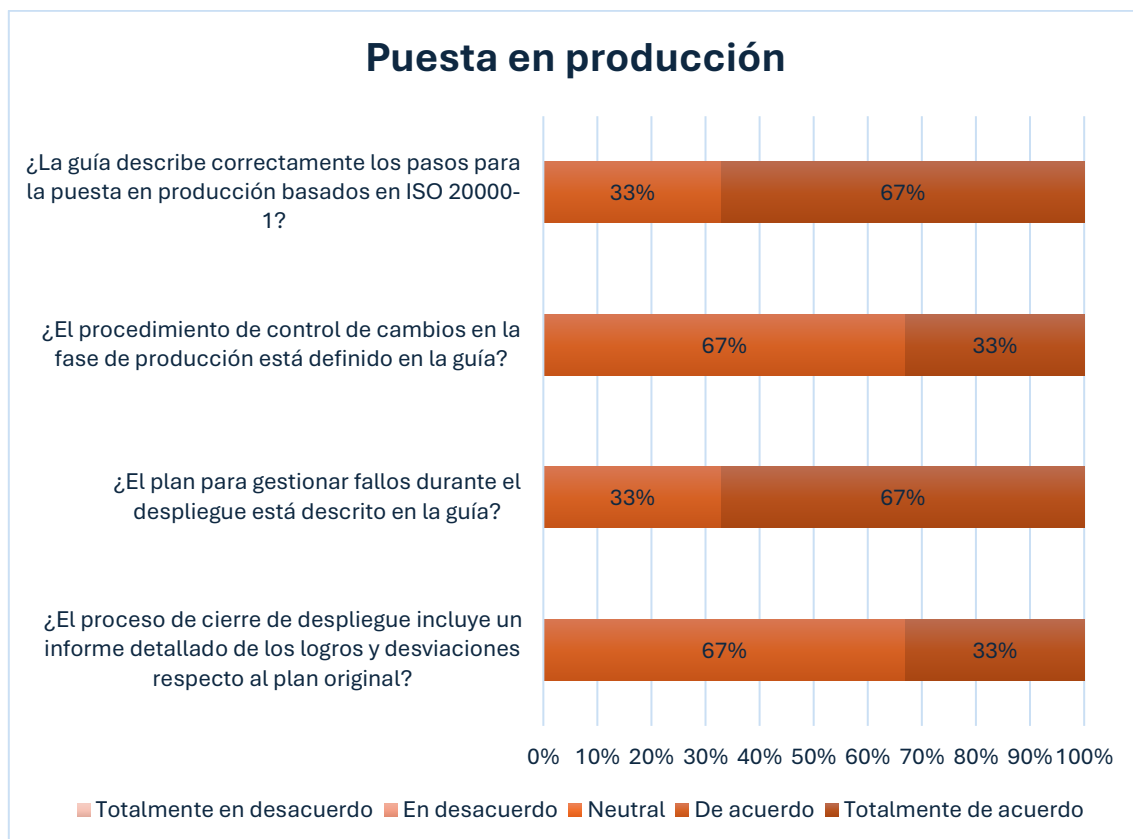
Pregunta 12. ¿La fórmula para calcular la madurez de las pruebas evalúa en función de los casos de prueba con resultados satisfactorios?

Un 67% de los participantes está de acuerdo en que la guía incluye una fórmula adecuada para calcular la madurez de las pruebas basada en los resultados satisfactorios, mientras que el 33% está totalmente de acuerdo. Esto indica una percepción favorable respecto a la utilidad de esta fórmula.

Pregunta 13. ¿La fórmula para calcular la densidad de defectos evalúa la cantidad de defectos por cada elemento de especificación revisado durante el proceso de pruebas?

Los resultados reflejan un 67% de acuerdo y un 33% totalmente de acuerdo, lo que demuestra que la fórmula incluida en la guía para medir la densidad de defectos es bien recibida.

Figura 29 Gráfico de Barras Apiladas - Puesta en producción



Nota: Fuente: Elaboración propia

Pregunta 14. ¿La guía describe correctamente los pasos para la puesta en producción basados en ISO 20000-1?

Los resultados muestran que un 67% de los encuestados considera que la guía describe correctamente los pasos para la puesta en producción basados en la norma ISO 20000-1. Esto indica una alta conformidad percibida entre la guía y los estándares internacionales de gestión de servicios. Sin embargo, un 33% de los encuestados no está totalmente de acuerdo, lo que sugiere que podrían existir áreas de mejora en la alineación de la guía con la norma ISO 20000-1.

Pregunta 15. ¿El procedimiento de control de cambios en la fase de producción está definido en la guía?

En este caso, el 67% de los encuestados está de acuerdo con que el procedimiento de control de cambios está claramente definido en la guía. Esta respuesta refleja un nivel positivo de confianza en la guía, pues la gestión de cambios es una parte crucial en la fase de producción. No obstante, el 33% en desacuerdo podría sugerir que algunos consideran que el procedimiento no cubre completamente las necesidades o que falta especificación.

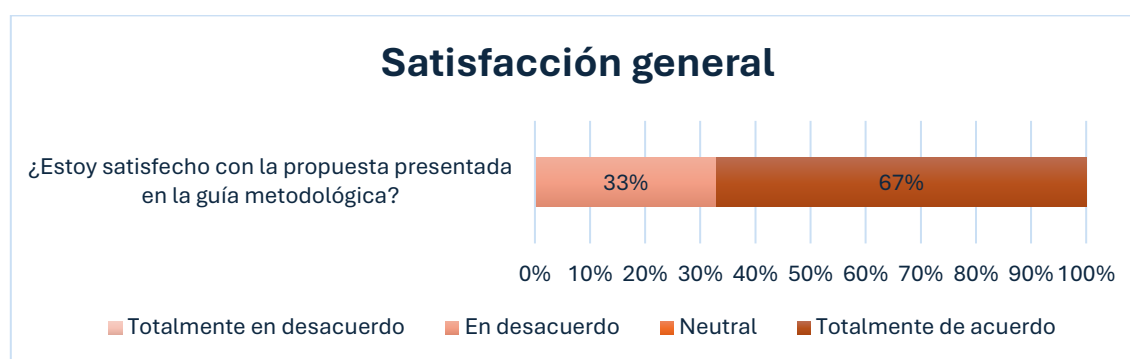
Pregunta 16. ¿El plan para gestionar fallos durante el despliegue está descrito en la guía?

En cuanto al plan de gestión de fallos, los resultados también son mayoritariamente positivos, con un 67% de los encuestados de acuerdo en que la guía incluye una descripción clara del plan para gestionar fallos durante el despliegue. Sin embargo, el 33% en desacuerdo indica que hay incertidumbre o áreas que no se cubren suficientemente, lo que podría llevar a confusión o falta de preparación en la práctica.

Pregunta 17. ¿El proceso de cierre de despliegue incluye un informe detallado de los logros y desviaciones respecto al plan original?

Por último, en relación al proceso de cierre del despliegue, los resultados son preocupantes, ya que el 67% de los encuestados no está de acuerdo con la afirmación de que la guía incluya un informe detallado de los logros y desviaciones. Este es un aspecto crítico, ya que la documentación de los resultados del despliegue es esencial para la evaluación del proceso y para la mejora continua. El 33% que está de acuerdo puede estar interpretando que la guía cubre algunos aspectos del cierre, pero claramente hay deficiencias que deben ser abordadas.

Figura 30 Gráfico de Barras Apiladas - Satisfacción general



Nota: Fuente: Elaboración propia

Pregunta 18. ¿Estoy satisfecho con la propuesta presentada en la guía metodológica?

Los resultados de la encuesta revelan una satisfacción general positiva con la propuesta presentada en la guía metodológica. Un contundente 67% de los encuestados expresó estar totalmente de acuerdo o de acuerdo con la propuesta, lo cual indica una percepción favorable hacia la guía como herramienta de trabajo.

Sin embargo, es importante destacar que un 33% de los participantes se mostró en desacuerdo. Este porcentaje, aunque menor, sugiere la existencia de áreas de oportunidad para mejorar la guía y adaptarla a las necesidades específicas de todos los usuarios.

A continuación, en la tabla 72, se calcula el promedio por cada pregunta, esto nos permitió medir la efectividad general de la guía para saber si cumple con los objetivos según la percepción de los

expertos. Además, se pudo realizar una comparación entre cada pregunta para conocer las áreas que necesitan una mejora.

Tabla 72 Promedio por pregunta

Expertos	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18
Experto 1	5	5	5	5	5	4	5	5	5	5	5	5	5	5	4	5	5	5
Experto 2	4	4	4	4	4	5	5	4	3	4	4	4	4	4	4	4	4	4
Experto 3	5	5	4	5	4	5	4	5	4	5	5	4	5	5	5	5	4	5
Promedio	4,67	4,67	4,33	4,67	4,33	4,67	4,67	4,67	4,00	4,67	4,67	4,33	4,67	4,67	4,33	4,67	4,33	4,67

Nota: Fuente: Elaboración propia

En una escala de 1 a 5, el puntaje promedio general de la guía es 4.5, lo que indica que la guía es muy efectiva y bien recibida por los expertos. La guía tiene una base sólida y se considera satisfactoria, con solo algunas pequeñas áreas que podrían mejorarse.

Calificación obtenida de la efectividad de la guía metodológica es de: 4.5/5

La tabla 73, detalla la opinión y sugerencias de cada uno de los expertos sobre las mejoras que se pueden aplicar en la guía metodológica.

Tabla 73 Comentarios de expertos

Expertos	Comentarios
Experto 1	Creo que la guía se encuentra completa y útil para la evaluación de la calidad del software, es necesario implementarla en otros módulos de software que se encuentran en desarrollo y posiblemente se encuentre la necesidad de actualizar ciertos parámetros
Experto 2	Pienso que la guía resumió de manera práctica la parte metodológica de las ISO. Además, pienso que se debería considerar el cálculo de escenarios y número de pruebas necesarios para atender a los requisitos.
Experto 3	En lo posterior realizar un caso de estudio donde aplique la guía.

Nota: Fuente: Elaboración propia

Las opiniones de los expertos destacan diversos aspectos clave sobre la efectividad de la guía metodológica. En primer lugar, se reconoce que la guía es completa y útil para evaluar la

calidad del software, con el potencial de ser implementada en otros módulos, aunque podría requerir actualizaciones para adaptarse a contextos específicos (Experto 1). También se valora positivamente su capacidad para sintetizar de manera práctica las metodologías descritas en las normas ISO, sugiriendo, sin embargo, que se incluya el cálculo de escenarios y pruebas necesarios para garantizar el cumplimiento de los requisitos (Experto 2). Finalmente, se resalta la importancia de realizar un caso de estudio como paso futuro, lo cual permitiría validar y optimizar la guía en un contexto práctico (Experto 3). En conjunto, las observaciones apuntan a que la guía es sólida en su enfoque actual, pero puede fortalecerse mediante adaptaciones específicas y validaciones adicionales.

CONCLUSIONES

La norma ISO/IEC/IEEE 29119 establece un marco integral para las pruebas de software que abarca desde la planificación hasta la evaluación de resultados. Su enfoque basado en procesos organizativos, de gestión y dinámicos permite a las organizaciones estructurar de manera eficiente las actividades de prueba. Este enfoque es altamente adaptable al contexto del DDTI, donde la formalización y estandarización de los procesos de prueba pueden mejorar significativamente la calidad del software desarrollado y desplegado.

Los resultados de la entrevista y la encuesta revelan una diferencia en la valoración de la documentación del proceso de pruebas y puesta en producción en el DDTI. Según el subdirector, esta documentación es específica y detallada, principalmente a través de actas de entrega y recepción de resultados finales. Sin embargo, el equipo de desarrollo considera que dicha documentación no es suficiente para garantizar un registro completo y accesible del proceso. Esta variación en las percepciones sugiere que, aunque existen esfuerzos por formalizar la documentación, el contenido actual no cubre todas las necesidades operativas del equipo, lo que podría limitar la efectividad y el seguimiento del proceso de pruebas.

La evaluación de la calidad del software en el DDTI muestra enfoques distintos entre el subdirector y el equipo de desarrollo. El subdirector enfatiza las encuestas de satisfacción del usuario como principal herramienta de medición, sin referirse a métricas técnicas específicas. Sin embargo, el equipo de desarrollo percibe que estas métricas no son suficientes para una evaluación completa y precisa de la calidad, sugiriendo la necesidad de indicadores más técnicos que permitan una valoración integral y objetiva. Esta diferencia de perspectivas evidencia una posible limitación en la evaluación actual, que podría beneficiarse de un

enfoque que combine la satisfacción del usuario con métricas técnicas concretas para un análisis más exhaustivo.

En conclusión, existe una visión contrastante respecto a la capacitación en gestión de riesgos en el DDTI. Por un lado, el subdirector hace hincapié en estrategias de seguridad como la separación de ambientes, sin referirse específicamente a la preparación del equipo en esta área. Por otro lado, algunos miembros del equipo consideran que sus habilidades en la gestión de riesgos requieren mayor fortalecimiento. Esta divergencia indica que, si bien se implementan ciertas prácticas de seguridad, existe un interés dentro del equipo por mejorar su preparación en gestión de riesgos para asegurar una mayor calidad y seguridad en el software desarrollado.

La aplicación teórica de la guía metodológica permite estructurar y formalizar el proceso de pruebas en los submódulos de portafolio docente y estudiante. A través de una planificación detallada, que incluye la definición de objetivos, identificación de riesgos, y asignación clara de roles, se asegura que las actividades estén alineadas con los requerimientos específicos del software y las expectativas del usuario final. Este enfoque contribuye a la optimización de los recursos y a la reducción de fallos en la etapa de producción.

La guía metodológica incluye un componente robusto de documentación que facilita la trazabilidad y el control del proceso de pruebas. La generación de informes detallados sobre el estado de los casos de prueba, los defectos encontrados, y las métricas de evaluación, no solo asegura un seguimiento eficiente, sino que también aporta información valiosa para futuras mejoras. Este nivel de formalización es esencial para garantizar la calidad del software en entornos institucionales.

La validación de la guía metodológica a través de una prueba de concepto permitió evaluar su viabilidad teórica mediante la opinión de expertos en calidad e ingeniería de software. Los resultados indicaron que la guía cumple con los estándares internacionales, como la ISO/IEC 29119 e ISO 20000-1, y es clara en sus objetivos. Sin embargo, algunos expertos señalaron áreas de mejora, como la necesidad de mayor detalle en ciertos aspectos operativos, lo que resalta la importancia de ajustes para garantizar una implementación efectiva.

RECOMENDACIONES

Se recomienda que el DDTI implemente un programa de capacitación continuo para su equipo de desarrollo, enfocado en los principios y lineamientos de la norma ISO/IEC/IEEE 29119. Esto permitirá que los desarrolladores comprendan y apliquen de manera efectiva los estándares de pruebas de software, incrementando la calidad y consistencia de los procesos de desarrollo.

Se recomienda que dentro de la estructura organizativa del DDTI se incorpore un área específica dedicada a la calidad de software. Esta área tendría como responsabilidad principal garantizar el cumplimiento de estándares internacionales, como la ISO 29119 para las fases de ejecución de pruebas, y la ISO 20000-1 para la puesta en producción. Su creación permitiría formalizar y centralizar los procesos relacionados con la validación de requisitos, diseño y ejecución de pruebas, gestión de defectos, y la planificación y control del despliegue del software. La creación de esta área no solo fortalecería la capacidad del DDTI para gestionar la calidad del software, sino que también alinearía los procesos internos con las mejores prácticas internacionales, asegurando un ciclo de vida del software más eficiente y confiable.

Se recomienda incorporar un proceso formal de gestión de pruebas que utilice herramientas de automatización para cubrir todas las etapas del ciclo de pruebas, desde la planificación y diseño de casos hasta la ejecución, seguimiento de defectos y generación de reportes. Este proceso debe integrarse con herramientas como Selenium, TestLink o JIRA, que permitan centralizar la documentación y mejorar la trazabilidad entre los requisitos, los casos de prueba y los resultados. Además, esta integración asegurará que las pruebas funcionales y de usabilidad realizadas en los submódulos "portafolio docente" y "portafolio estudiante" sean

más eficientes, consistentes y alineadas con los estándares de la ISO 29119. La automatización reducirá el tiempo y esfuerzo manual, mejorará la cobertura de pruebas y facilitará la realización de pruebas de regresión ante cambios en los módulos.

Se recomienda desarrollar un sistema que automatice la gestión de pruebas en todas sus etapas, desde la planificación y diseño hasta la ejecución, monitoreo y análisis de resultados. Esto optimizará el tiempo, reducirá errores humanos y garantizará una mayor cobertura de pruebas, alineándose con estándares internacionales como la ISO 29119. Además, centralizará la documentación y facilitará la generación de métricas para la toma de decisiones. La automatización también permitirá validar cambios en la puesta en producción, asegurando la estabilidad del software y mejorando la capacidad del DDTI para ofrecer soluciones tecnológicas de calidad.

Para la fase de puesta en producción, se recomienda establecer un entorno de preproducción que replique fielmente las condiciones del entorno real. Esto incluye configuraciones específicas de hardware, software y bases de datos que permitan realizar pruebas finales antes del despliegue. Además, se debe implementar un protocolo de despliegue controlado, documentado y alineado con la ISO 20000, que incluya actividades como validaciones post-despliegue, monitoreo continuo y planes de contingencia ante fallos. Este enfoque garantizará que la transición al entorno de producción sea segura y sin interrupciones para los usuarios finales, minimizando riesgos y asegurando la calidad del sistema en su conjunto.

BIBLIOGRAFÍA

- Alonso, A. F., Martínez Normand, L., & Segovia Pérez, J. (2005). *Introducción a la Ingeniería del Software: modelos de desarrollo de programas*. Delta Publicaciones.
<https://elibro.net/es/lc/utnorte/titulos/170188>
- Ardila, C. (n.d.). *Calidad de Software*. Retrieved November 22, 2024, from
http://artemisa.unicauca.edu.co/~cardila/CS__Cap_1__Introduccion_Calidad_Software.pdf
- Bruegge, B., & Dutoit, A. H. (2002). *Ingeniería de software orientado a objetos* (1st ed.).
- Camisón, C., Cruz, S., & González, T. (2007). *Gestión de calidad: conceptos, enfoques, modelos y sistemas*. Pearson/Prentice Hall.
https://www.academia.edu/33042332/Gesti%C3%B3n_de_la_calidad_Conceptos_enfoques_modelos_y_sistemas
- Cárdenas, W. (2020). *Elaboración de un marco de trabajo para pruebas de software, basado en el estándar ISO/IEC/IEEE 29119 y su impacto en el proceso de evaluación del software* [Universidad Técnica del Norte].
<http://repositorio.utn.edu.ec/handle/123456789/9758>
- García, F. J. (2022, March 7). *Los métodos de revisión sistemática de literatura*.
<https://repositorio.grial.eu/handle/grial/2543>
- IEEE Computer Society. (2014). *Guide to the Software Engineering Body of Knowledge SWEBOK. IEEE*.
- ISO 9000. (2015). *ISO 9000:2015 Sistemas de gestión de calidad - Fundamentos y vocabulario*. www.iso.org
- ISO 10005. (2018). *Gestión de la calidad-Directrices para los planes de la calidad*. www.iso.org
- ISO/IEC 20000-1: - Gestión del servicio: Requisitos. (2018). *ISO/IEC 20000-1:2018 (E)*.
- ISO/IEC/IEEE 29119 - Parte 1: conceptos generales. (2022). *ISO/IEC/IEEE 29119-1: 2022(E)*, 1–60.
- ISO/IEC/IEEE 29119 - Parte 2: procesos de prueba. (2021). *ISO/IEC/IEEE 29119-2: 2021(E)*, 1–64.
- ISO/IEC/IEEE 29119 - Parte 3: documentación de pruebas. (2021). *ISO/IEC/IEEE 29119-3: 2021(E)*, 1–98.
- ISO/IEC/IEEE 29119 - Parte 4: técnicas de prueba. (2021). *ISO/IEC/IEEE 29119-4: 2021(E)*, 1–148.
- ISO/IEC/IEEE 29119 - Parte 5: pruebas basadas en palabras clave. (2016). *ISO/IEC/IEEE 29119-5: 2016 (E)*, 1–69.

- ISTQB. (n.d.). *Who we are*. Retrieved October 25, 2023, from <https://www.istqb.org/about-us/who-we-are>
- Laos, A. (2020). *Calidad del proceso de QA y satisfacción del usuario interno del software UXPOS en la empresa de Retail SODIMAC - Lima 2019* [Universidad Nacional José Faustino Sánchez Carrión]. <http://repositorio.unjfsc.edu.pe/handle/20.500.14067/3868>
- López Echeverry, A. M., Valencia Ayala, L. E., & Cabrera, C. (2008). Introducción a la calidad de software. *Scientia et Technica, ISSN 0122-1701, Vol. 2, No. 39, 2008, Págs. 326-331, 2(39), 326–331.*
<https://dialnet.unirioja.es/servlet/articulo?codigo=4745899&info=resumen&idioma=EN>
G
- Mejía Trejo, J. (2024). *Principios de aseguramiento de calidad para el diseño de software: innovación de procesos en las tecnologías de información*. Academia Mexicana de Investigación y Docencia en Innovación (AMIDI).
<https://elibro.net/es/lc/utnorte/titulos/250101>
- Naciones Unidas. (n.d.). *Objetivos y Metas de desarrollo sostenible*. Retrieved November 22, 2024, from <https://www.un.org/sustainabledevelopment/es/sustainable-development-goals/>
- Napoleão, B. M. and P. F. and H. S. (2020). Open Source Software Development Process: A Systematic Review. *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*, 135–144.
- Norma ISO/IEC 20000-1. (n.d.). Retrieved November 24, 2024, from <https://static1.squarespace.com/static/5ff7814fc11799276a34a6c6/t/600aedc5c86806372868d534/1611328967451/ISO-20000-1-Norma-Espanol-V102018A-1.pdf>
- Páez, J. (2013). *Diseño de un modelo para evaluación/pruebas del software en base a ingeniería de pruebas aplicando el estándar ISO/IEC 29119 en la empresa OMNISOFTE de la ciudad de Quito*. [Universidad de las Fuerzas Armadas ESPE].
<http://repositorio.espe.edu.ec/handle/21000/6385>
- Pantaleo, G. (2016). *Calidad en el desarrollo de software* (2nd ed.). Alfaomega.
- Piattini, M., García, F., García, I., & Pino, F. (2018). *Calidad de Sistemas de Información* (4th ed.). RA-MA Editorial.
- Pressman, R. S. (2010). *Ingeniería del software: un enfoque práctico* (7th ed.). McGraw-Hill.
- Redrován, F., Loja, N., Correa, K., & Piña, J. (2017). Estado del arte: Métricas de calidad para el desarrollo de aplicaciones web. *3C Tecnología_Glosas de Innovación Aplicadas a La Pyme*, 6, 1–12.
- Reid, S. (2023). *ISO/IEC/IEEE 29119 SOFTWARE TESTING STANDARDS - A Practitioner's Guide* (2nd ed.). kindle.

- Robles, M. T. (2017). *Guía Metodológica. ¿Qué es? ¿Cómo se realiza? 1. Definición de objetivo, alcance y audiencia APROBACIÓN DIFUSIÓN EDICIÓN Y DISEÑO*.
<https://docplayer.es/user/38697302/>
- Shyamal, D. K. K., Asanka, P. P. G. D., & Wickramaarachchi, D. (2023). A Comprehensive Approach to Evaluating Software Code Quality Through a Flexible Quality Model. *2023 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, 6, 1–8. <https://doi.org/10.1109/SCSE59836.2023.10215004>
- SSTQB. (2023, April 21). *Probador Certificado de ISTQB Programa de Estudio Nivel Básico Versión Es V01.02*. HASTB.
- Steidl, D., Deissenboeck, F., Poehlmann, M., Heinke, R., & Uhink-Mergenthaler, B. (2014). Continuous Software Quality Control in Practice. *2014 IEEE International Conference on Software Maintenance and Evolution*, 561–564. <https://doi.org/10.1109/ICSME.2014.95>
- Terán-Ávila, D. L. (2021). *Proyectos de desarrollo de software y su potencial para subvención de capital semilla, Ecuador*. <https://api.semanticscholar.org/CorpusID:236749403>
- Terrera, G. (2023, March 11). *ISTQB CTFL v2018. 1. Fundamentos – 7 principios de la prueba. Breve explicación y ejemplos – TestingBaires*. <https://testingbaires.com/istqb-ctfl-v2018-1-fundamentos-7-principios-de-la-prueba-breve-explicacion-y-ejemplos/>
- Valencia, L. E. P., Lazo, A. T., Vargas, J. L. A., & Franco, D. E. R. (2019). Ingeniería de Software: El aseguramiento de la calidad de los requisitos en la industria del software en el Eje Cafetero colombiano. *INGE CUC*, 15(2), 110–122.
- Veenendaal, E. (2016). TMMi and ISO 29119: Friends or Foes? *White Paper TMMi Foundation*.
- Ventura, M. T. (n.d.). *Costo de la Calidad en el Desarrollo de Software*. Retrieved October 26, 2023, from http://paginaspersonales.unam.mx/app/webroot/files/69/Publica_20110622215614.pdf



ANEXOS

ANEXO A

INFORME DE RESULTADOS

ENCUESTA A DOCENTES Y

ESTUDIANTES

Contenido

1.	Introducción.....	2
2.	Objetivo	2
3.	Contexto.....	2
4.	Población.....	3
5.	Muestra	3
6.	Método de tabulación.....	7
7.	Análisis general.....	9
8.	Conclusión.....	15
9.	Anexos	16

1. Introducción

Las encuestas aplicadas permitieron recopilar opiniones y sugerencias sobre las áreas de mejora percibidas en el uso cotidiano de cada portafolio, tanto en términos de funcionalidad como de usabilidad. Los resultados obtenidos proporcionan una visión clara sobre las expectativas y necesidades de los usuarios, las cuales guiarán la toma de decisiones para optimizar la experiencia de uso de ambas herramientas.

Este informe presenta un análisis detallado de los resultados, organizados en tablas y gráficos para facilitar la comprensión de los puntos críticos identificados. Asimismo, se destacan las principales áreas de mejora propuestas por los usuarios, proporcionando una base sólida para futuras modificaciones que busquen mejorar la calidad y el rendimiento de los portafolios docente y estudiante.

2. Objetivo

Analizar y documentar los resultados de las encuestas aplicadas a los usuarios finales de los submódulos de portafolio docente y portafolio estudiante, con el fin de identificar áreas de mejora en términos de funcionalidad y usabilidad, proporcionando información clave que guíe la optimización y adecuación de ambos portafolios a las necesidades de docentes y estudiantes.

3. Contexto

En el marco de la mejora continua de los submódulos de portafolio docente y portafolio estudiante, pertenecientes al Sistema de Información Integrado Universitario, se ha identificado la necesidad de ajustar estas herramientas para que respondan mejor a las necesidades y expectativas de los usuarios. Ambos submódulos, utilizados de manera cotidiana por docentes y estudiantes, deben adaptarse no solo en términos de funcionalidad, sino también en cuanto a su usabilidad, con el fin de maximizar la eficiencia y satisfacción de sus usuarios.

Para este propósito, se implementó un levantamiento de información a través de encuestas dirigidas a los usuarios finales. Este levantamiento tuvo una doble finalidad: por un lado, recopilar las opiniones de los usuarios sobre el rendimiento y la experiencia de uso de los portafolios y, por otro, levantar nuevos requisitos funcionales que permitan adecuar las funcionalidades actuales a las necesidades reales de docentes y estudiantes.

Los resultados obtenidos constituyen una base fundamental para la planificación de mejoras en los portafolios, permitiendo priorizar aquellos aspectos críticos que inciden en la satisfacción y en la operatividad de estas herramientas. Este informe presenta un análisis estructurado de los datos recopilados y las principales áreas de mejora, así como una lista preliminar de nuevos requisitos funcionales que guiarán el desarrollo futuro de los submódulos.

4. Población

El primer paso para definir la población es identificar el número total de estudiantes y docentes que utilizan el portafolio electrónico universitario. Esto es crucial para garantizar que la muestra seleccionada sea representativa de toda la población y que los resultados obtenidos sean válidos y aplicables a la totalidad de usuarios del sistema.

En este caso, la población objetivo se compone de:

- Docentes: El número total de docentes que utilizan el SIIU – Portafolio Docente es de 646. Los docentes pertenecen a distintas facultades y están incluidos los docentes por nombramiento, contrato, contrato por factura y postgrado, lo que también implica una variedad de expectativas y requerimientos funcionales que deben ser capturados.
- Estudiantes: El número total de estudiantes que utilizan el SIIU – Portafolio Estudiante es de 11613 (datos del periodo marzo – agosto 2024). Estos estudiantes están distribuidos en diferentes semestres, carreras y facultades lo que garantiza una diversidad de experiencias y necesidades que deben ser consideradas en el levantamiento de requisitos.

5. Muestra

Para definir la muestra de docentes y estudiantes, se empleará lo siguiente:

- Fórmula de Cochran para Poblaciones Finitas

La Fórmula de Cochran es una metodología estadística utilizada para determinar el tamaño de la muestra necesaria para un estudio, asegurando que los resultados sean representativos de la población total.

$$n = \frac{Z^2 * p * q * N}{(N * e^2) + (Z^2 * p * q)}$$

Donde:

$Z = 1.96$ (Nivel de confianza del 95%)

$e = 0.05$ (Error de estimación)

$p = 0.5$ (Probabilidad de éxito)

$q = 0.5$ (Probabilidad de fracaso)

$N = ?$ (Tamaño de la población)

- Muestreo estratificado

El muestreo probabilístico estratificado es una técnica de muestreo que divide a la población en subgrupos homogéneos (estratos) y selecciona muestras aleatorias de cada estrato.

Esta técnica asegura que todas las subpoblaciones estén representadas proporcionalmente en la muestra, capturando la diversidad de experiencias y necesidades. Permite obtener resultados específicos para cada subgrupo, lo que es crucial para identificar requisitos funcionales precisos y relevantes para distintos segmentos de la población.

Por lo tanto, la muestra para docentes y estudiantes es la siguiente:

- Docentes:

$$n = \frac{Z^2 * p * q * N}{(N * e^2) + (Z^2 * p * q)}$$
$$n = \frac{(1.96)^2 * 0.5 * 0.5 * 646}{(646 * (0.05)^2) + ((1.96)^2 * 0.5 * 0.5)}$$
$$n = \frac{620.7436}{2.5754}$$
$$n = 241$$

- Estudiantes:

$$n = \frac{Z^2 * p * q * N}{(N * e^2) + (Z^2 * p * q)}$$
$$n = \frac{(1.96)^2 * 0.5 * 0.5 * 11613}{(11613 * (0.05)^2) + ((1.96)^2 * 0.5 * 0.5)}$$

$$n = \frac{11164.64}{29.9929}$$

$$n = 372$$

Continuamos con el muestreo probabilístico estratificado para docentes, en donde distribuimos la muestra total entre los estratos (en este caso, las facultades), detallado en la tabla 1, de manera proporcional al tamaño de cada estrato en la población total.

Tabla 1 *Docentes por cada facultad*

Facultad	Docentes
FACAE	92
FCCSS	90
FECYT	151
FICA	85
FICAYA	67
POSGRADO	161

Nota: Fuente: elaboración propia

A continuación, se obtiene la muestra estratificada de docentes por cada facultad, detallado en la tabla 2

Tabla 2 *Cálculo para obtener la muestra de docentes por cada facultad*

Facultad	Docentes
FACAE	$P_{FACAE} = \frac{92}{646} = 0.1424$
FCCSS	$P_{FCCSS} = \frac{90}{646} = 0.1393$
FECYT	$P_{FECYT} = \frac{151}{646} = 0.2337$
FICA	$P_{FICA} = \frac{85}{646} = 0.1316$
FICAYA	$P_{FICAYA} = \frac{67}{646} = 0.1037$
POSTGRADO	$P_{POSTGRADO} = \frac{161}{646} = 0.2492$

Nota: Fuente: elaboración propia

Por último, se multiplica el tamaño de la muestra total por la proporción de cada estrato, detallado en la tabla 3

Tabla 3 *Muestra de docentes por cada facultad.*

Facultad	Proporción	Cálculos	Docentes
FACAE	$n_{FACAE} = n * P_{FACAE}$	$n_{FACAE} = 241 * 0.1424$	34
FCCSS	$n_{FCCSS} = n * P_{FCCSS}$	$n_{FCCSS} = 241 * 0.1393$	34
FECYT	$n_{FECYT} = n * P_{FECYT}$	$n_{FECYT} = 241 * 0.2337$	56
FICA	$n_{FICA} = n * P_{FICA}$	$n_{FICA} = 241 * 0.1316$	32

FICAYA	$n_{FICAYA} = n * P_{FICAYA}$	$n_{FICAYA} = 241 * 0.1037$	25
POSTGRADO	$n_{POSTGRADO} = n * P_{POSTGRADO}$	$n_{POSTGRADO} = 241 * 0.2492$	60

Nota: Fuente: elaboración propia

Ahora realizamos el muestreo probabilístico estratificado para estudiantes, en donde distribuimos la muestra total entre los estratos (en este caso, las facultades), detallado en la tabla 4, de manera proporcional al tamaño de cada estrato en la población total.

Tabla 4 *Estudiantes por cada facultad*

Facultad	Estudiantes
FACAE	2554
FCCSS	1329
FECYT	4745
FICA	1732
FICAYA	1253

Nota: Fuente: elaboración propia

A continuación, se obtiene la muestra estratificada de estudiantes por cada facultad, detallado en la tabla 5

Tabla 5 *Cálculo para obtener la muestra de estudiantes por cada facultad*

Facultad	Estudiantes
FACAE	$P_{FACAE} = \frac{2554}{11620} = 0.2198$
FCCSS	$P_{FCCSS} = \frac{1329}{11620} = 0.1144$
FECYT	$P_{FECYT} = \frac{4745}{11620} = 0.4083$
FICA	$P_{FICA} = \frac{1732}{11620} = 0.1491$
FICAYA	$P_{FICAYA} = \frac{1253}{11620} = 0.1078$

Nota: Fuente: elaboración propia

Por último, se multiplica el tamaño de la muestra total por la proporción de cada estrato, detallado en la tabla 6

Tabla 6 *Muestra de estudiantes por cada facultad*

Facultad	Proporción	Cálculos	Estudiantes
FACAE	$n_{FACAE} = n * P_{FACAE}$	$n_{FACAE} = 372 * 0.2198$	82
FCCSS	$n_{FCCSS} = n * P_{FCCSS}$	$n_{FCCSS} = 372 * 0.1144$	43
FECYT	$n_{FECYT} = n * P_{FECYT}$	$n_{FECYT} = 372 * 0.4083$	152
FICA	$n_{FICA} = n * P_{FICA}$	$n_{FICA} = 372 * 0.1491$	55
FICAYA	$n_{FICAYA} = n * P_{FICAYA}$	$n_{FICAYA} = 372 * 0.1078$	40

6. Método de tabulación

En los cuestionarios se utilizó una escala de Likert para medir el nivel de acuerdo o satisfacción de los encuestados con respecto a varias afirmaciones o preguntas, detallado en la tabla 7.

Tabla 7 Escala de Likert

Nivel/Puntos	1	2	3	4	5
Nivel de acuerdo	Totalmente en Desacuerdo	En Desacuerdo	Neutral	De Acuerdo	Totalmente de Acuerdo
Nivel de Satisfacción	Muy Insatisfecho	Insatisfecho	Neutral	Satisfecho	Muy Satisfecho

Nota: Fuente: elaboración propia

Además, se realizó una agrupación de preguntas por categorías, en la que se dividen las preguntas en tres áreas específicas, detallado en la tabla 8.

- Usabilidad: relacionada con la facilidad de uso y la experiencia del usuario.
- Funcionalidad: se refiere a cómo las funciones del sistema o software cumplen con las necesidades de los usuarios.
- Satisfacción de usuario: Se refiere a la satisfacción general que los usuarios sienten respecto al uso del software.

Tabla 8 Agrupación de preguntas por categorías

Agrupación de preguntas por categorías					
Usabilidad		Funcionalidad		Satisfacción de usuario	
Docentes	Estudiantes	Docentes	Estudiantes	Docentes	Estudiantes
P1	P1	P9	P7	P19	P10
P2	P2	P10	P8	P20	P11
P3	P3	P11	P9	P21	P12
P4	P4	P12		P22	P13
P5	P5	P13		P23	P14
P6	P6	P14			P15
P7		P15			
P8		P16			
		P17			
		P18			

Nota: Fuente: elaboración propia

Desde la tabla 9 hasta la tabla 11 se calcula el número de respuestas por opción, es decir se obtiene las frecuencias absolutas para cada pregunta en el cuestionario para docentes.

Tabla 9 Usabilidad - Docentes

Usabilidad - Nivel de acuerdo	P1	P2	P3	P4	P5	P6	P7	P8
Totalmente en Desacuerdo	3	3	6	8	7	10	5	7
En Descuerdo	3	11	23	17	20	12	17	13
Neutral	31	55	44	54	49	49	39	33
De Acuerdo	89	97	93	86	97	94	100	106

Totalmente de Acuerdo	115	75	75	76	68	76	80	82
-----------------------	-----	----	----	----	----	----	----	----

Nota: Fuente: elaboración propia

Tabla 10 *Funcionalidad - Docentes*

Funcionalidad - Nivel de acuerdo	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18
Totalmente en Desacuerdo	8	10	14	7	8	9	7	8	15	12
En Descuerdo	8	20	24	21	27	21	18	19	29	23
Neutral	44	57	66	45	66	63	58	50	62	68
De Acuerdo	107	91	82	96	78	82	91	98	74	74
Totalmente de Acuerdo	74	63	55	72	62	66	67	66	61	64

Nota: Fuente: elaboración propia

Tabla 11 *Satisfacción de usuario - Docentes*

Satisfacción de usuario - Nivel de Satisfacción	P19	P20	P21	P22	P23
Muy Insatisfecho	6	5	8	13	5
Insatisfecho	7	15	11	17	17
Neutral	56	55	49	62	54
Satisfecho	102	99	102	86	103
Muy Satisfecho	70	67	71	63	62

Nota: Fuente: elaboración propia

Desde la tabla 12 hasta la tabla 14 se calcula el número de respuestas por opción, es decir se obtiene las frecuencias absolutas para cada pregunta en el cuestionario para estudiantes.

Tabla 12 *Usabilidad - Estudiantes*

Usabilidad - Nivel de Acuerdo	P1	P2	P3	P4	P5	P6
Totalmente en desacuerdo	16	15	19	20	19	17
En desacuerdo	9	19	16	37	41	28
Neutral	66	80	73	123	106	88
De acuerdo	125	144	137	126	145	145
Totalmente de acuerdo	156	114	127	66	61	94

Nota: Fuente: elaboración propia

Tabla 13 *Funcionalidad - Estudiantes*

Funcionalidad - Nivel de Acuerdo	P7	P8	P9
Totalmente en desacuerdo	11	13	30
En desacuerdo	16	12	30
Neutral	82	72	84
De acuerdo	156	144	138

Totalmente de acuerdo	107	131	90
-----------------------	-----	-----	----

Nota: Fuente: elaboración propia

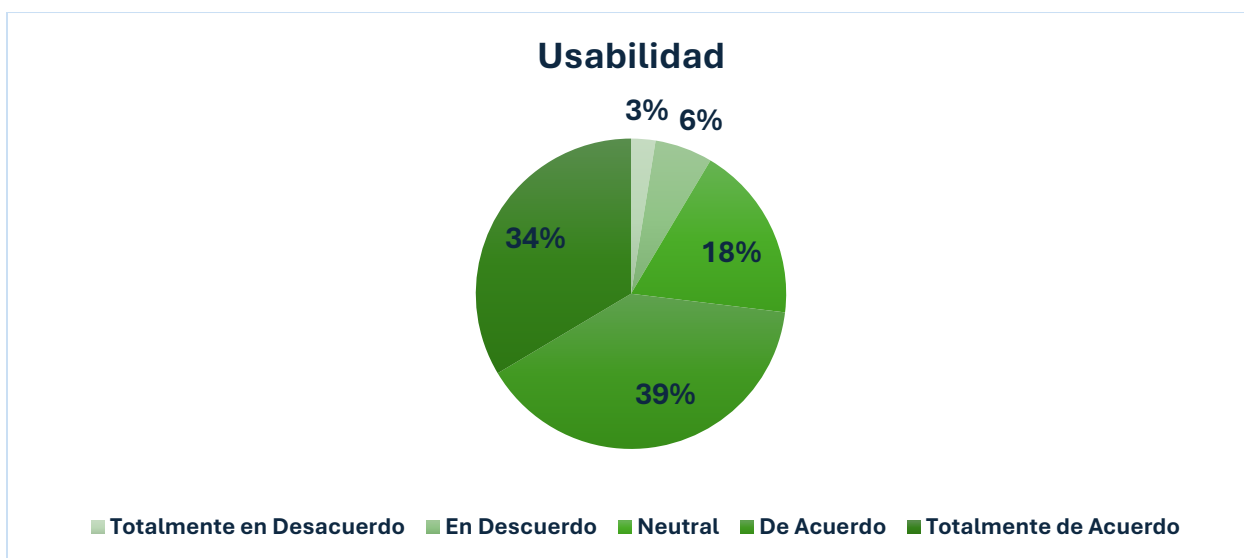
Tabla 14 Satisfacción de usuario - Estudiantes

Satisfacción - Nivel de satisfacción	P10	P11	P12	P13	P14	P15
Muy Insatisfecho	11	12	15	17	13	13
Insatisfecho	15	13	32	30	17	16
Neutral	91	89	108	104	91	98
Satisfecho	155	163	146	142	160	163
Muy Satisfecho	100	95	71	79	91	82

Nota: Fuente: elaboración propia

7. Análisis general

Figura 1 Usabilidad - Docentes



Nota: Fuente: elaboración propia

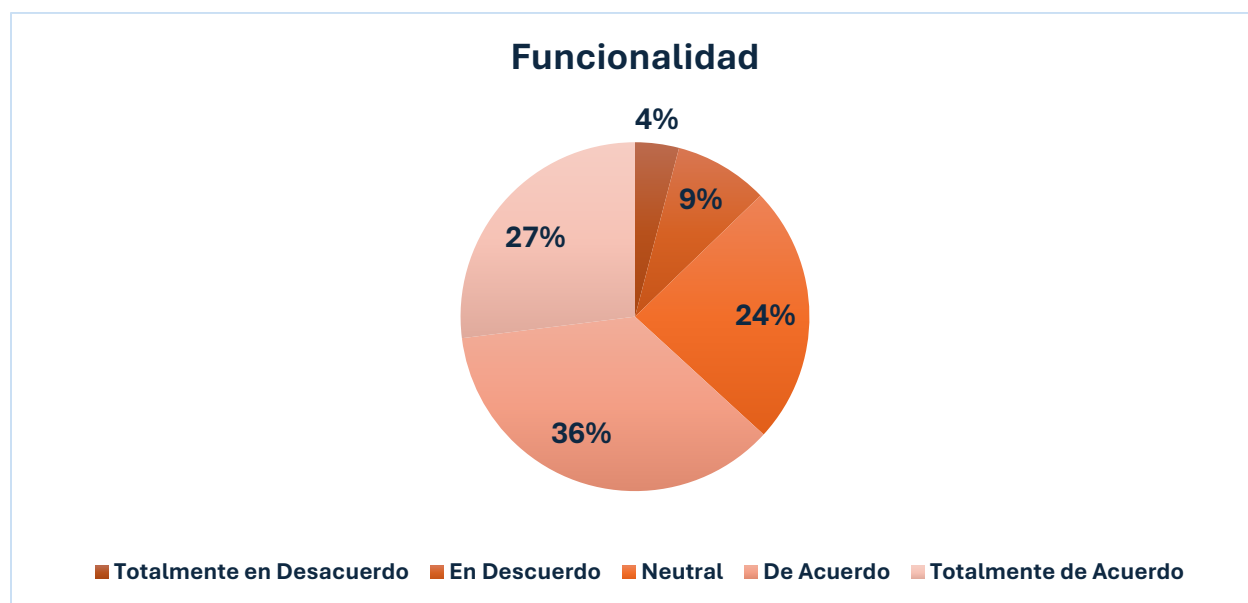
Los resultados de la encuesta sobre la usabilidad del portafolio docente, realizada a 646 docentes, reflejan una percepción mayoritariamente positiva respecto al uso y la experiencia del sistema. El 34% de los encuestados está totalmente de acuerdo con las preguntas relacionadas con la usabilidad, lo que indica un fuerte grado de satisfacción. Este alto porcentaje refleja que una parte significativa de los usuarios considera que el portafolio es fácil de usar, intuitivo y adecuado en términos de tiempos de respuesta y navegación.

A pesar de este alto nivel de satisfacción, el 39% de los docentes está de acuerdo, lo que sugiere una respuesta favorable, pero algo más moderada. Esto podría indicar que aunque la mayoría de los usuarios tienen una experiencia positiva, aún podrían existir áreas de oportunidad para mejorar la usabilidad y optimizar ciertas funciones del portafolio.

El 18% de respuestas neutrales muestra que una fracción de los encuestados no tiene una opinión clara sobre la usabilidad del portafolio. Este grupo podría estar experimentando una experiencia de usuario promedio, sin destacar características ni positivas ni negativas, lo que podría ser un indicativo de que el sistema cumple con sus funciones básicas, pero no genera un alto nivel de entusiasmo.

Por otro lado, el 6% de respuestas en desacuerdo y el 3% de totalmente en desacuerdo sugieren que un pequeño pero relevante porcentaje de los usuarios encuentra dificultades en el uso del portafolio. Esto podría estar relacionado con problemas en la interfaz, tiempos de carga o navegación, aspectos que deben ser considerados para realizar mejoras específicas y minimizar la frustración de estos usuarios.

Figura 2 *Funcionalidad - Docentes*



Nota: Fuente: elaboración propia

Un 27% de los encuestados están totalmente de acuerdo con las preguntas sobre la funcionalidad del portafolio, lo que refleja un alto nivel de satisfacción con las herramientas y características que ofrece el sistema. Estos docentes consideran que el portafolio es eficaz para gestionar recursos, evaluar actividades, comunicar con estudiantes y realizar tareas académicas en general.

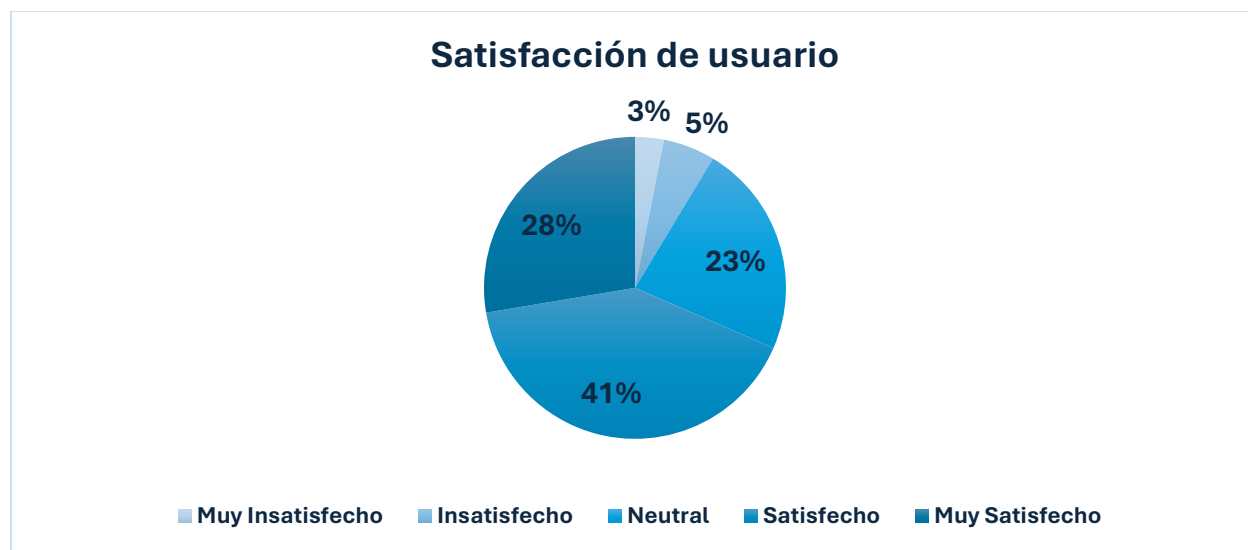
El 36% de los encuestados están de acuerdo con las afirmaciones, lo que también es positivo, aunque menos rotundo. Esto sugiere que, si bien los docentes consideran que el portafolio cumple su propósito, podría haber aspectos o funcionalidades que no alcanzan su nivel de expectativas en términos de eficiencia o facilidad de uso.

El 24% de los encuestados se encuentran en desacuerdo, lo que indica que una parte significativa de los usuarios no está completamente conforme con las funcionalidades del portafolio. Este grupo puede estar experimentando dificultades en áreas clave como la gestión de calificaciones, las herramientas de evaluación o la integración de herramientas externas. Es importante identificar y abordar estas áreas de insatisfacción para mejorar la experiencia del usuario.

El 9% de los encuestados tiene una postura neutral, lo que sugiere que, aunque no presentan una insatisfacción clara, tampoco consideran que el sistema se destaque significativamente. Este grupo podría estar utilizando el sistema de manera funcional, pero no encuentra características particularmente destacables o innovadoras en la plataforma.

Finalmente, el 4% de los encuestados están totalmente en desacuerdo con las afirmaciones sobre la funcionalidad del portafolio, lo que destaca a un pequeño grupo de usuarios que enfrentan problemas significativos al utilizar el sistema. Este grupo podría estar experimentando dificultades críticas en el uso de herramientas esenciales, como la creación de foros o la gestión de calificaciones, lo que afecta gravemente su experiencia.

Figura 3 Satisfacción de usuario - Docentes



Nota: Fuente: elaboración propia

Un 41% de los encuestados están satisfechos, lo que sugiere que la mayoría de los usuarios experimenta una interacción razonablemente positiva con el portafolio docente. Este porcentaje refleja una buena aceptación en cuanto a la facilidad de uso, la interfaz de usuario y la funcionalidad en general. Sin embargo, aunque los usuarios están satisfechos, este resultado también sugiere que hay margen para una mayor optimización para alcanzar niveles más altos de satisfacción.

El 23% de los encuestados están muy satisfechos, lo que indica que una fracción significativa de usuarios tiene una experiencia realmente positiva. Este grupo percibe el portafolio como una

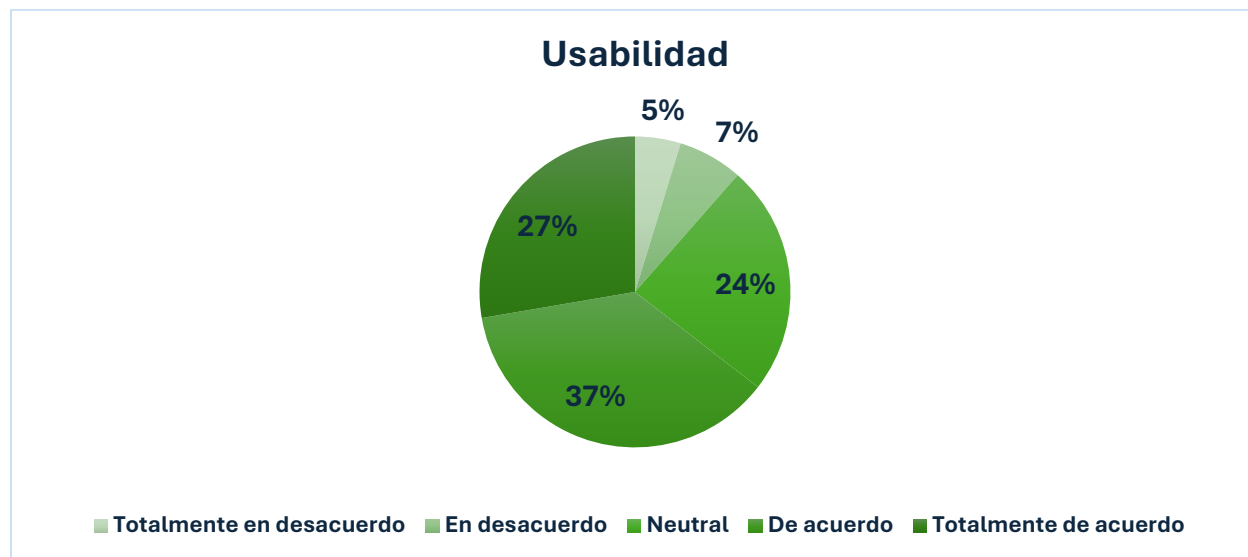
herramienta eficiente, fácil de usar y que cumple con sus expectativas, lo que es un buen indicador de la calidad del sistema desde la perspectiva de estos usuarios.

El 28% de los encuestados se encuentra insatisfecho, lo que es un porcentaje considerablemente alto. Este grupo podría estar enfrentando dificultades con aspectos del portafolio como la gestión de calificaciones, la funcionalidad de la interfaz o la calidad del soporte técnico. Este nivel de insatisfacción destaca la necesidad de abordar áreas críticas que están afectando la experiencia de los usuarios.

El 5% de los encuestados están muy insatisfechos, lo que señala que un pequeño pero relevante porcentaje de usuarios tiene problemas significativos con el sistema, posiblemente relacionados con la usabilidad, los tiempos de respuesta o la falta de soporte adecuado. Estos usuarios probablemente enfrentan barreras serias que afectan su desempeño y percepción del portafolio, lo que requiere atención urgente para evitar una mayor frustración.

Finalmente, el 3% de los encuestados se muestra neutral, lo que sugiere que, aunque no tienen una opinión negativa, no consideran que el sistema sea excepcionalmente satisfactorio. Este grupo podría estar experimentando una interacción promedio, sin características destacables ni problemas graves.

Figura 4 Usabilidad - Estudiantes



Nota: Fuente: elaboración propia

El 37% de los encuestados están de acuerdo con las preguntas relacionadas con la usabilidad del portafolio. Esto indica que más de un tercio de los estudiantes consideran que el portafolio es funcional y cumple con sus expectativas en términos de navegación, interfaz y facilidad de uso. Este grupo valora positivamente la experiencia general con el sistema, aunque aún podrían existir aspectos a mejorar.

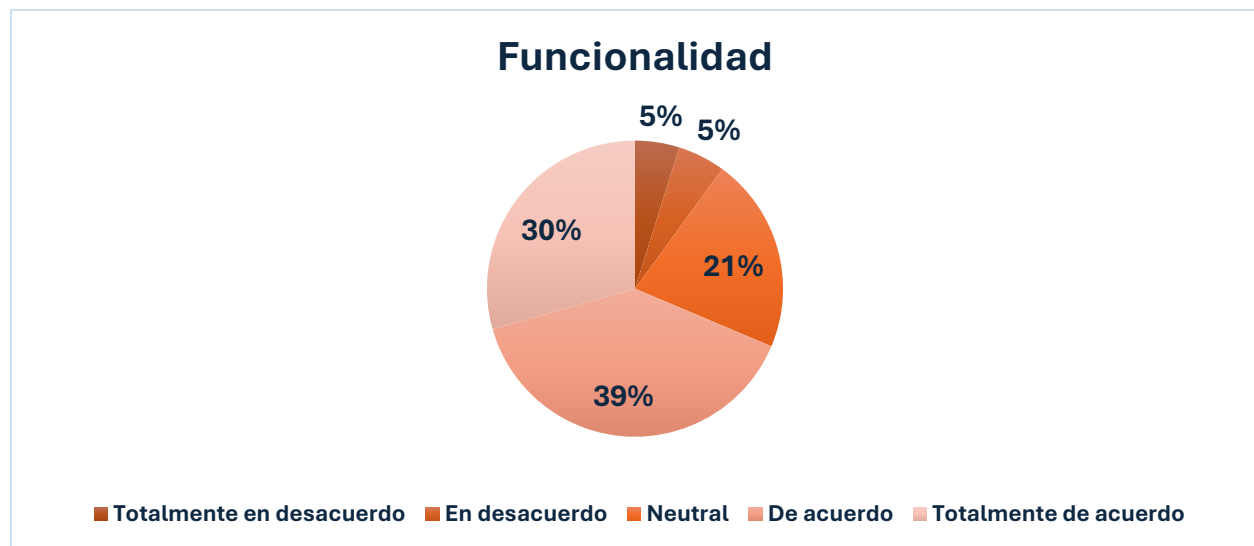
El 27% de los encuestados están totalmente de acuerdo, lo que sugiere que una proporción considerable de estudiantes tiene una experiencia excelente con el portafolio. Para este grupo, el portafolio es fácil de usar, intuitivo y cumple con sus necesidades, lo que es un indicador positivo de la efectividad del sistema.

El 24% de los encuestados se muestran neutrales, lo que significa que no tienen una opinión fuerte ni positiva ni negativa sobre la usabilidad del portafolio. Esto podría indicar que estos estudiantes no encuentran grandes dificultades en su uso, pero tampoco consideran que el sistema tenga características excepcionales que mejoren significativamente su experiencia.

El 5% de los encuestados están en desacuerdo con la usabilidad del portafolio, lo que revela que algunos estudiantes encuentran problemas con la interfaz, la navegación o las funcionalidades del sistema. Este grupo podría estar experimentando dificultades en áreas como la carga de documentos o la organización del contenido, lo que afecta su percepción del portafolio.

Finalmente, el 7% de los encuestados están totalmente en desacuerdo con las afirmaciones relacionadas con la usabilidad. Este pequeño porcentaje de estudiantes enfrenta problemas significativos que podrían estar relacionados con una experiencia muy frustrante con el portafolio, como dificultades graves con la interfaz o tiempos de respuesta inadecuados.

Figura 5 *Funcionalidad - Estudiantes*



Nota: Fuente: elaboración propia

El 39% de los encuestados están de acuerdo con las funcionalidades del portafolio, lo que indica que una mayoría significativa considera que el portafolio ofrece las herramientas necesarias para gestionar sus actividades académicas, como las calificaciones y las notificaciones. Este porcentaje refleja que el portafolio cubre las expectativas de la mayoría de los estudiantes en términos de las funcionalidades que ofrece para el seguimiento de su proceso académico.

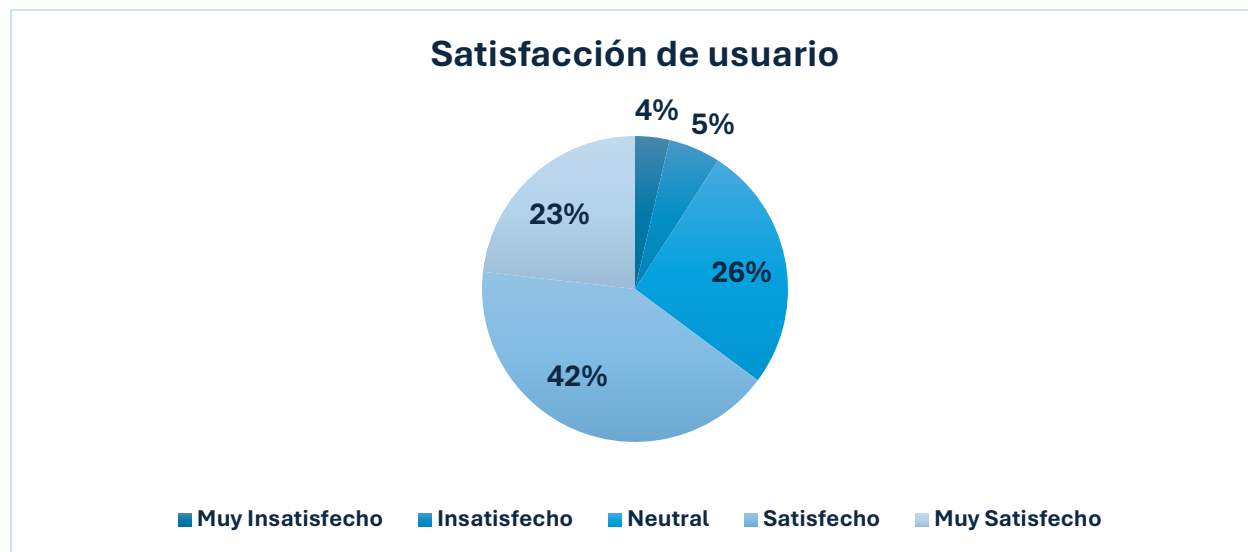
El 30% de los encuestados están totalmente de acuerdo, lo que sugiere que un grupo considerable de estudiantes tiene una experiencia extremadamente positiva con las funcionalidades del portafolio. Estos estudiantes encuentran que las herramientas disponibles son adecuadas y eficientes, permitiéndoles gestionar de manera efectiva sus asignaturas y estar informados sobre los eventos universitarios.

El 21% de los encuestados se muestra neutral en cuanto a las funcionalidades del portafolio. Este grupo probablemente no tiene una opinión definida sobre el sistema, lo que puede indicar que el portafolio cumple su función básica, pero no sobresale ni genera una satisfacción destacada. Es posible que estos estudiantes utilicen el portafolio sin problemas graves, pero tampoco lo consideran excepcional o innovador.

El 5% de los encuestados están en desacuerdo, lo que refleja que una pequeña proporción de usuarios experimenta dificultades con algunas de las funcionalidades del portafolio. Este grupo podría estar enfrentando limitaciones en las herramientas disponibles, como la gestión de calificaciones o las notificaciones, lo que afecta su experiencia en la plataforma.

Finalmente, el 5% de los encuestados están totalmente en desacuerdo con las funcionalidades del portafolio, lo que sugiere que este grupo tiene una experiencia significativamente negativa. Los estudiantes que se encuentran en esta categoría podrían estar enfrentando serias deficiencias en la funcionalidad del sistema, como la falta de herramientas necesarias para su desarrollo académico o problemas con la recepción de notificaciones importantes.

Figura 6 Satisfacción de usuario - Estudiantes



Nota: Fuente: elaboración propia

El 42% de los encuestados están satisfechos con el portafolio del estudiante. Este porcentaje es indicativo de que una gran parte de los usuarios considera que el portafolio cumple con sus expectativas en términos de funciones, información, tiempos de respuesta y gestión académica.

Sin embargo, este grupo también sugiere que, aunque el sistema es funcional, hay margen para mejorar algunos aspectos de la experiencia.

El 23% de los encuestados están muy satisfechos, lo que representa un grupo significativo de usuarios que valoran altamente el portafolio. Este grupo considera que el sistema es eficaz, rápido y cumple con sus necesidades de manera sobresaliente, lo que es una señal positiva de la funcionalidad general del portafolio para este segmento de estudiantes.

El 26% de los encuestados se muestra neutral en relación con la satisfacción con el portafolio. Este grupo probablemente no tiene una opinión fuerte sobre el sistema, lo que puede indicar que el portafolio satisface sus necesidades básicas, pero no genera una experiencia destacada ni negativa. Estos estudiantes podrían estar usando el sistema sin mayores inconvenientes, pero tampoco lo consideran excepcional.

El 5% de los encuestados están insatisfechos, lo que revela que una pequeña proporción de los usuarios enfrenta dificultades con el sistema, ya sea en términos de la carga de documentos, los tiempos de respuesta o la gestión de las actividades académicas. Este grupo podría estar experimentando problemas técnicos que afectan su interacción con el portafolio.

Finalmente, el 4% de los encuestados están muy insatisfechos, lo que indica que este pequeño grupo tiene una experiencia significativamente negativa. Estos estudiantes probablemente enfrentan problemas graves con el portafolio, como dificultades para gestionar calificaciones, recibir notificaciones o acceder a funcionalidades clave.

8. Conclusión

Después del análisis realizado a los resultados de las encuestas se concluye que tanto los docentes como estudiantes se sienten en su mayoría satisfechos con la usabilidad de los portafolios, aunque existen áreas de oportunidad importantes. Un porcentaje significativo de docentes y estudiantes se mantiene neutral, lo que sugiere que el sistema es funcional, pero no completamente intuitivo o fácil de usar para todos. Es necesario mejorar la navegación y la claridad en las instrucciones para maximizar la satisfacción y la eficiencia de la plataforma.

La mayoría de los usuarios está de acuerdo con las funcionalidades de los portafolios, aunque hay un grupo considerable que se muestra neutral. Este resultado implica que el sistema cumple con las funciones esenciales, pero algunas características podrían no estar completamente optimizadas o faltar funcionalidades adicionales que responderían mejor a las necesidades de los usuarios. Las mejoras en este sentido podrían enfocarse en una mejor integración de herramientas y funciones específicas que los usuarios perciben como insuficientes.

El análisis de los resultados demuestra que tanto los docentes como los estudiantes tienen necesidades específicas que deben ser consideradas al realizar mejoras en los portafolios. Los docentes, por ejemplo, podrían necesitar una mayor claridad en las funcionalidades de evaluación y gestión de contenidos, mientras que los estudiantes podrían beneficiarse de ajustes en la usabilidad y tiempos de respuesta. La segmentación de la mejora en base a estos diferentes grupos

de usuarios permitirá que los portafolios se ajusten más eficazmente a sus expectativas y necesidades particulares.

9. Anexos

Anexo A. Hallazgos de la encuesta a docentes

Tabla 1 Requisito funcional N.º 1

Id	RF-001
Nombre	Indicadores visuales de progreso proactivos
Descripción	Utilizar colores y marcas visuales en la interfaz del sistema para indicar claramente el progreso de los docentes en el registro de avances y los temas pendientes, facilitando una rápida identificación de las áreas que requieren atención.
Entrada	Datos de avances
Salida	Información sobre los temas pendientes
Precondición	Indicadores visuales en la interfaz del sistema que muestren el progreso de los docentes y los temas pendientes.
Postcondición	Visualización de barras de progreso, colores y marcas distintivas para resaltar el estado de los avances.

Nota: Fuente: Elaboración propia.

Tabla 2 Requisito funcional N.º 2

Id	RF-002
Nombre	Buzón de inquietudes
Descripción	Implementar un buzón de inquietudes accesible desde el portafolio para que los docentes puedan reportar problemas o solicitar ayuda fácilmente. Este sistema debe permitir el seguimiento y la respuesta rápida a las consultas y solicitudes enviadas.
Entrada	Consultas y solicitudes de ayuda enviadas por los docentes, que incluyen descripción del problema, prioridad, y datos de contacto. Posibilidad de adjuntar documentos o capturas de pantalla que ayuden a describir mejor el problema.
Salida	Respuestas y soluciones a las consultas y solicitudes de ayuda enviadas por los docentes. Notificaciones a los docentes con el estado y la resolución de su solicitud.
Precondición	El buzón de inquietudes debe estar implementado, configurado y accesible desde el portafolio docente. Los docentes deben estar autenticados en el sistema para enviar consultas.
Postcondición	Las consultas y solicitudes de ayuda son monitoreadas y respondidas rápidamente por el equipo de soporte. Se mantiene un registro de todas las consultas y sus soluciones para referencia futura y mejora continua del sistema.

Nota: Fuente: Elaboración propia.

Tabla 3 Requisito funcional N.º 3

Id	RF-003
Nombre	Guía Interactiva para registro de avances y banco de preguntas
Descripción	Implementar una guía interactiva (video tutorial) dentro del portafolio docente que detalle las funcionalidades existentes en el registro de avances y banco de preguntas. Esta guía debe ser accesible para todos los docentes, permitiéndoles elegir entre ver el tutorial u omitirlo.
Entrada	Materiales de capacitación (videos e imágenes tutoriales). Información sobre las nuevas funcionalidades.
Salida	Docentes capacitados en el registro de avances y uso del banco de preguntas a través de la guía interactiva. Interfaz de usuario con indicaciones visuales y tutoriales paso a paso.
Precondición	El portafolio docente debe estar configurado para mostrar la guía interactiva.
Postcondición	Los docentes pueden acceder a la guía interactiva en cualquier momento y recibir capacitación adecuada y oportuna. Los docentes tienen la opción de ver u omitir los tutoriales paso a paso.
Observación	El requisito funcional no puede ser implementado, ya que implica la participación del personal del área de desarrollo en tareas de programación, lo que resultaría en un incremento en el tiempo necesario para su ejecución.

Nota: Fuente: Elaboración propia.

Tabla 4 Requisito funcional N.º 4

Id	RF-004
Nombre	Aumento del tiempo de inactividad de la sesión
Descripción	Extender el tiempo de inactividad, antes de que el sistema bloquee automáticamente la sesión del usuario, e implementar notificaciones de advertencia que alerten al usuario antes de que la sesión caduque.
Entrada	Parámetros de tiempo de sesión (tiempo máximo de inactividad). Actividad del usuario (movimiento del ratón, pulsaciones del teclado).
Salida	Sesión extendida. Notificaciones de advertencia enviadas al usuario.
Precondición	El usuario debe estar autenticado en el sistema. El sistema debe tener capacidad para monitorear la actividad del usuario.
Postcondición	El tiempo de navegación se extiende, reduciendo interrupciones por bloqueos de sesión. Los usuarios reciben notificaciones de advertencia antes de que se cierre la sesión por inactividad.

Nota: Fuente: Elaboración propia.

Tabla 5 Requisito funcional N.º 5

Id	RF-005
Nombre	Rediseño del Banco de Preguntas
Descripción	Rediseñar la interfaz del banco de preguntas para que sea más amigable y eficiente, añadiendo un botón que permita la creación de preguntas sin necesidad de regresar al banco principal.

Entrada	Datos de preguntas (texto de la pregunta, opciones de respuesta, respuestas correctas, etc.). Datos de secciones/categorías (nombre de la sección, descripción, etc.).
Salida	Interfaz rediseñada y funcional para la gestión de preguntas. Datos de preguntas organizados en secciones o categorías.
Precondición	Los datos de preguntas existentes deben estar disponibles. La estructura del sistema debe permitir la modificación de la interfaz y la organización de datos.
Postcondición	Los docentes pueden gestionar y crear preguntas de manera más eficiente. Las preguntas están organizadas en secciones o categorías, facilitando su manejo y acceso.
Observación	El requisito funcional no puede ser implementado, ya que implica la participación del personal del área de desarrollo en tareas de programación, lo que resultaría en un incremento en el tiempo necesario para su ejecución

Nota: Fuente: Elaboración propia.

Tabla 6 Requisito funcional N.º 6

Id	RF-006 pendiente
Nombre	Editor de Ecuaciones y Herramientas de Matemáticas
Descripción	Implementar un editor de ecuaciones y herramientas específicas para crear evaluaciones de Matemáticas, permitiendo a los docentes insertar y editar ecuaciones matemáticas de manera sencilla y eficiente.
Entrada	Datos de preguntas (texto de la pregunta, opciones de respuesta). Ecuaciones matemáticas (utilizando notaciones matemáticas estándar).
Salida	Evaluaciones con ecuaciones y contenido matemático correctamente formateados y visualizados.
Precondición	Interfaz de usuario adecuada y accesible para el editor de ecuaciones. El sistema debe soportar notaciones matemáticas estándar y renderizar ecuaciones correctamente.
Postcondición	Los docentes pueden crear evaluaciones de Matemáticas con ecuaciones y contenido matemático de manera eficiente. Las ecuaciones se visualizan correctamente en las evaluaciones.

Nota: Fuente: Elaboración propia.

Tabla 7 Requisito funcional N.º 7

Id	RF-007
Nombre	Navegación entre Preguntas
Descripción	Permitir la navegación entre preguntas durante la creación y configuración de evaluaciones, proporcionando a los docentes la opción de configurar cómo los estudiantes pueden moverse a través de las preguntas y permitiendo un diseño de evaluaciones personalizado sin necesidad de regresar al banco de preguntas.
Entrada	Datos de preguntas (texto de la pregunta, opciones de respuesta, respuestas correctas). Configuraciones de navegación (permitir o restringir la navegación entre preguntas).
Salida	Evaluaciones configuradas con opciones de navegación entre preguntas.

Precondición	El sistema debe estar configurado para aceptar y procesar las configuraciones de navegación.
Postcondición	Las evaluaciones se configuran de acuerdo con las necesidades del docente, permitiendo o restringiendo la navegación entre preguntas según se haya configurado, y se facilita el diseño de evaluaciones personalizadas sin necesidad de regresar al banco de preguntas.

Nota: Fuente: Elaboración propia.

Tabla 8 Requisito funcional N.º 8

Id	RF-008
Nombre	Sección de Prácticas Preprofesionales
Descripción	Implementar una sección específica en el portafolio docente para gestionar las prácticas preprofesionales de los estudiantes, incluyendo registro de actividades, calificaciones y evidencias.
Entrada	Datos de actividades de prácticas (descripción, fechas, horas). Calificaciones y comentarios. Evidencias (documentos, fotos, etc.).
Salida	Registro de actividades de prácticas. Calificaciones y comentarios de las prácticas. Evidencias asociadas a las prácticas.
Precondición	Interfaz de usuario adecuada para registrar y gestionar prácticas preprofesionales. Datos de actividades y evidencias disponibles para subir.
Postcondición	Las prácticas preprofesionales se registran y gestionan de manera eficiente en el portafolio.
Observación	El requisito funcional no puede ser implementado, ya que implica la participación del personal del área de desarrollo en tareas de programación, lo que resultaría en un incremento en el tiempo necesario para su ejecución

Nota: Fuente: Elaboración propia.

Tabla 9 Requisito funcional N.º 9

Id	RF-009
Nombre	Sección de Tutorías Académicas y de Titulación
Descripción	Implementar una sección específica en el portafolio docente para gestionar tutorías académicas y de titulación, incluyendo registro de sesiones, avances de tesis, evidencias y calificaciones.
Entrada	Datos de sesiones de tutoría (fecha, hora, duración, temas tratados). Avances de tesis. Evidencias (documentos, fotos, etc.). Calificaciones y comentarios.
Salida	Registro de sesiones de tutoría. Registro de avances de tesis. Evidencias asociadas a las tutorías. Calificaciones y comentarios de las tutorías.
Precondición	Interfaz de usuario adecuada para registrar y gestionar tutorías académicas y de titulación. Datos de sesiones, avances y evidencias disponibles para subir.

Postcondición	Las tutorías académicas y de titulación se registran y gestionan de manera eficiente en el portafolio.
Observación	El requisito funcional no puede ser implementado, ya que implica la participación del personal del área de desarrollo en tareas de programación, lo que resultaría en un incremento en el tiempo necesario para su ejecución

Nota: Fuente: Elaboración propia.

Tabla 10 Requisito funcional N.º 10

Id	RF-010
Nombre	Visualización de Archivos en Ventanas Emergentes
Descripción	Permitir que los archivos adjuntos de los estudiantes (PDF, Word, Excel) se abran en ventanas emergentes, y que, al cerrar estas ventanas, no se cierre la sesión del docente, permitiendo continuar con la calificación y visualización de otros archivos.
Entrada	Solicitud de visualización de archivo. Datos de los archivos adjuntos.
Salida	Archivos visualizados en ventanas emergentes.
Precondición	Los archivos adjuntos deben estar correctamente almacenados y accesibles. El sistema debe estar configurado para abrir archivos en ventanas emergentes.
Postcondición	Los docentes pueden visualizar los archivos adjuntos en ventanas emergentes sin interrumpir su sesión de trabajo.

Nota: Fuente: Elaboración propia.

Tabla 11 Requisito funcional N.º 11

Id	RF-011
Nombre	Descarga Masiva de Archivos Adjuntos
Descripción	Implementar una funcionalidad que permita la descarga masiva de los archivos adjuntos de los estudiantes con un solo botón, facilitando la gestión de tareas y evaluaciones.
Entrada	Solicitud de descarga masiva de archivos. Datos de los archivos adjuntos (ubicación y metadatos).
Salida	Archivos adjuntos descargados en un solo archivo comprimido (por ejemplo, .zip).
Precondición	Los archivos adjuntos deben estar correctamente almacenados en el sistema. El sistema debe tener capacidad para generar y descargar archivos comprimidos.
Postcondición	Los docentes pueden descargar todos los archivos adjuntos de los estudiantes de una sola vez, ahorrando tiempo y esfuerzo.

Nota: Fuente: Elaboración propia.

Anexo B. Hallazgos de la encuesta a estudiantes

Tabla 12 *Requisito funcional N.º 1*

Id	RF-001
Nombre	Guía Interactiva
Descripción	Implementar una guía interactiva (video tutorial) dentro del portafolio estudiante que detalle las principales funcionalidades. La guía debe ser accesible para todos los estudiantes, permitiéndoles elegir entre ver el tutorial u omitirlo.
Entrada	Materiales de capacitación (videos e imágenes tutoriales). Información sobre las principales funcionalidades.
Salida	Visualización del video tutorial.
Precondición	El portafolio estudiante debe estar configurado para mostrar la guía interactiva (video tutorial).
Postcondición	Los estudiantes pueden acceder a la guía interactiva en cualquier momento y recibir capacitación adecuada y oportuna. Los estudiantes tienen la opción de ver u omitir los tutoriales paso a paso.
Observación	El requisito funcional no puede ser implementado, ya que implica la participación del personal del área de desarrollo en tareas de programación, lo que resultaría en un incremento en el tiempo necesario para su ejecución

Nota: Fuente: Elaboración propia.

Tabla 13 *Requisito funcional N.º 2*

Id	RF-002
Nombre	Incluir una sección FAQ
Descripción	El sistema debe incluir una sección de preguntas frecuentes (FAQ) para guiar a los estudiantes en la búsqueda de información y resolución de problemas comunes. La sección de FAQ debe abordar las preguntas más comunes de los estudiantes y ofrecer soluciones claras y concisas.
Entrada	Selección de la opción de "Ayuda" o "FAQ" en el menú
Salida	Visualización de páginas de ayuda o sección FAQ
Precondición	El estudiante necesita información o solución a un problema relacionado con el uso del portafolio
Postcondición	El estudiante encuentra la información necesaria o solución a su problema, mejorando su experiencia de uso del portafolio
Observación	El requisito funcional no puede ser implementado, ya que implica la participación del personal del área de desarrollo en tareas de programación, lo que resultaría en un incremento en el tiempo necesario para su ejecución

Nota: Fuente: Elaboración propia.

Tabla 14 *Requisito funcional N.º 3*

Id	RF-003
Nombre	Incorporar indicadores de carga
Descripción	El sistema debe proporcionar indicadores de carga claros y visibles, como barras de progreso o íconos de carga, para que los estudiantes sepan el estado actual de carga en términos de porcentaje completado y tiempo

	estimado restante. Estos indicadores deben aparecer en todas las secciones del portafolio donde el contenido pueda tardar en cargarse, como al acceder a materiales de clase, visualizar notas, o cargar documentos.
Entrada	Acción que inicia una carga de contenido (ej., clic en una sección, solicitud de descarga)
Salida	Visualización de indicadores de carga (barras de progreso, íconos de carga)
Precondición	El estudiante realiza una acción que requiere carga de contenido
Postcondición	El estudiante es informado visualmente de que el contenido está cargando y espera adecuadamente

Nota: Fuente: Elaboración propia.

Tabla 15 Requisito funcional N.º 4

Id	RF-004
Nombre	Indicadores en actividades de Foros.
Descripción	El sistema debe marcar las actividades en los foros como completadas (verde) una vez que el estudiante haya participado. Esto incluye la creación de nuevos hilos, responder a hilos existentes y participar en discusiones de grupo. Cada foro debe mostrar un indicador visual (color verde) para señalar que el estudiante ha cumplido con la participación requerida.
Entrada	Participación en un foro (crear nuevo hilo, responder a un hilo existente, o participar en una discusión de grupo)
Salida	Foro marcado como completado con un indicador verde
Precondición	El estudiante participa en un foro dentro del portafolio
Postcondición	El foro aparece marcado como completado (color verde) para indicar que el estudiante ha cumplido con la participación requerida

Nota: Fuente: Elaboración propia.

Tabla 17 Requisito funcional N.º 6

Id	RF-006
Nombre	Visualización de Respuestas Correctas e Incorrectas
Descripción	El sistema debe permitir a los estudiantes ver las respuestas correctas e incorrectas después de completar las evaluaciones. Esto ayudará a los estudiantes a identificar sus errores y mejorar su comprensión en la asignatura.
Entrada	Finalización de una evaluación
Salida	Visualización de las respuestas correctas e incorrectas
Precondición	El estudiante ha completado una evaluación
Postcondición	El estudiante puede ver qué respuestas fueron correctas o incorrectas, mejorando su aprendizaje

Nota: Fuente: Elaboración propia.

Tabla 18 Requisito funcional N.º 7

Id	RF-007
Nombre	Notificaciones de Mantenimiento del Sistema

Descripción	El sistema debe enviar notificaciones a los estudiantes cuando el portafolio vaya a estar en mantenimiento, indicando la fecha, hora de inicio y finalización del mantenimiento, y los posibles efectos sobre la disponibilidad del sistema. Las notificaciones deben enviarse con suficiente antelación para que los estudiantes puedan planificar sus actividades
Entrada	Programación de mantenimiento del sistema
Salida	Notificación enviada al estudiante por el portafolio y correo electrónico
Precondición	Se programa un mantenimiento del sistema
Postcondición	El estudiante recibe una notificación detallada sobre el mantenimiento programado y su impacto

Nota: Fuente: Elaboración propia.

Tabla 19 Requisito funcional N.º 8

Id	RF-008
Nombre	Extensión del Tiempo de Sesión
Descripción	El sistema debe extender el tiempo de sesión predeterminado para los estudiantes, evitando cierres prematuros mientras realizan actividades en el portafolio.
Entrada	Actividad del usuario (movimiento del ratón, pulsaciones del teclado).
Salida	Sesión extendida. Notificaciones de advertencia enviadas al usuario.
Precondición	El usuario debe estar autenticado en el sistema.
Postcondición	El tiempo de navegación se extiende, reduciendo interrupciones por cierre de sesión. Los usuarios reciben notificaciones de advertencia antes de que se cierre la sesión por inactividad.

Nota: Fuente: Elaboración propia.

Tabla 20 Requisito funcional N.º 9

Id	RF-009
Nombre	Mejora de la Seguridad de las Claves y Autenticación de Sesión
Descripción	El sistema debe implementar medidas adicionales de seguridad para proteger las claves de los portafolios. Esto incluye un proceso de autenticación de sesión que dirige al estudiante a una ventana de verificación de código enviado al correo institucional antes de permitir el acceso al portafolio.
Entrada	Ingreso de usuario y contraseña inicial, ingreso del código de sesión
Salida	Acceso seguro al portafolio
Precondición	El estudiante inicia sesión con las credenciales iniciales
Postcondición	El estudiante accede al portafolio después de verificar el código de sesión enviado a su correo institucional

Nota: Fuente: Elaboración propia.