

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas Carrera de Software

Aseguramiento de la calidad del software durante su desarrollo usando pruebas unitarias y de aceptación en los módulos de control de materia prima y recepción de pedidos para la Fábrica de Medias Neltex

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

Autor:

Kevin Brayan Lliguisupa Ponce

Director:

Ing. Cathy Pamela Guevara Vega, Ph.D.

Ibarra - Ecuador

2024



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

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:	1728848373		
APELLIDOS Y NOMBRES:	LLIGUISUPA PONCE KEVIN BRAYAN		
DIRECCIÓN:	CAYAMBE, FRANKLIN RIVADENEIRA Y MARCHENA		
EMAIL:	kbliguisupap@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0969796346

DATOS DE LA OBRA	
TÍTULO:	ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE DURANTE SU DESARROLLO USANDO PRUEBAS UNITARIAS Y DE ACEPTACIÓN EN LOS MÓDULOS DE CONTROL DE MATERIA PRIMA Y RECEPCIÓN DE PEDIDOS PARA LA FÁBRICA DE MEDIAS NELTEX
AUTOR (ES):	KEVIN BRAYAN LLIGUISUPA PONCE
FECHA:	30/07/2024
PROGRAMA:	PREGRADO
TITULO POR EL QUE OPTA:	INGENIERO DE SOFTWARE
DIRECTOR:	ING. CATHY GUEVARA VEGA, Ph.D.
ASESOR:	ING. MAURICIO REA PEÑAFIEL, MSc.

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 30 días del mes de julio de 2024

EL AUTOR:



.....
ESTUDIANTE
Kevin Brayan Lliguisupa Ponce
C.I: 1728848373

CERTIFICACIÓN DIRECTOR

Ibarra, 30 de julio del 2024

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE TITULACIÓN

Por medio del presente yo Ph.D. Cathy Guevara Vega, certifico que la Sr. Kevin Brayan Lliguisupa Ponce portador de la cedula de ciudadanía número 1728848373, ha trabajado en el desarrollo del proyecto de grado "Aseguramiento de la calidad del software durante su desarrollo usando pruebas unitarias y de aceptación en los módulos de control de materia prima y recepción de pedidos para la Fábrica de Medias Neltex", 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. Cathy Guevara Vega, Ph.D.
DIRECTOR DE TRABAJO DE GRADO

CERTIFICADO DE LA EMPRESA



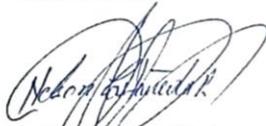
Otavallo, 19 de julio de 2024

Certificado

Siendo auspiciante del proyecto de titulación del señor Kevin Brayan Lliguisupa Ponce con C.I. 1728848373, quien realizo su proyecto con el tema: "Aseguramiento de la calidad del software durante su desarrollo usando pruebas unitarias y de aceptación en los módulos de control de materia prima y recepción de pedidos para la Fábrica de Medias Neltex", nos es grato contemplar que ha sido implantado correctamente en la empresa, cumpliendo a cabalidad con todos los requerimiento solicitados.

Es cuanto puedo certificar en honor a la verdad, facultando a la interesada hacer uso del presente documento como estime conveniente dentro del marco legal.

Atentamente,



Sr. Nelson Castañeda

Gerente Propietario Fábrica Neltex

NELTEX

RUC.: 1002118527001
TELF.. 2 635-022

Dirección: Ciudad Otavallo - Barrio Los Pinos (Via al parque el Cóndor) Parroquia Eugenio Espejo.
Teléfono: 062-635022 celular: 0997179084

DEDICATORIA

Dedico este trabajo de grado a quienes han sido mi roca y mi inspiración a lo largo de este viaje académico: mi querida madre y mis adorados hermanos.

Madre, tus sacrificios, tu amor incondicional y tu eterna dedicación han sido mi guía constante. Desde los días tempranos de mi educación hasta este momento trascendental, has sido mi ejemplo de fuerza, perseverancia y sabiduría. Tu apoyo inquebrantable ha sido mi sustento en los momentos difíciles y mi mayor alegría en los triunfos. Sin tus palabras alentadoras y tu presencia constante, este logro no habría sido posible.

A mis hermanos, ustedes son más que compañeros de vida, son mis confidentes, mis cómplices y mis mejores amigos. Su apoyo incondicional y su ánimo constante han sido un faro en los días oscuros. Este logro es tanto suyo como mío, y lo comparto con ustedes.

Kevin Lliguisupa

AGRADECIMIENTO

Quisiera expresar mi más sincero agradecimiento a todas las personas e instituciones que han contribuido de manera significativa al éxito de este trabajo de grado.

En primer lugar, quiero extender mi gratitud a la Universidad Técnica del Norte, por brindarme la oportunidad de formarme académicamente en sus aulas. Su compromiso con la excelencia y su apoyo continuo han sido fundamentales en mi desarrollo como profesional.

A la Ph.D. Cathy Guevara, mi tutora de tesis, le debo un profundo agradecimiento por su orientación experta, su paciencia infinita y su compromiso durante todo el proceso de investigación. Su sabiduría y guía hacia la culminación de este trabajo académico.

Asimismo, deseo expresar mi reconocimiento al MSc. Mauricio Rea, mi asesor de tesis, por su invaluable asistencia, sus consejos expertos y su disposición para compartir su conocimiento y experiencia.

No puedo dejar de mencionar a la empresa Neltex, por abrirme las puertas y permitirme realizar mi tesis en su entorno laboral. Su colaboración y facilitación de recursos han enriquecido enormemente brindado una perspectiva práctica invaluable a este trabajo.

Kevin Lliguisupa

INDICE DE CONTENIDOS

DEDICATORIA	VI
AGRADECIMIENTO	VII
INDICE DE CONTENIDOS	VIII
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS	XII
RESUMEN	XIV
ABSTRACT	XV
INTRODUCCIÓN.....	XVI
Tema	XVI
Problema	XVI
Antecedentes	XVI
Situación Actual	XVI
Prospectiva.....	XVII
Planteamiento del problema	XVII
Objetivos	XVIII
Objetivo General.....	XVIII
Objetivos Específicos.....	XVIII
Alcance	XIX
Metodología.....	XX
Justificación.....	XXII
Justificación Tecnológica.....	XXII
Justificación Económica	XXII
Justificación Social.....	XXIII
CAPÍTULO 1	1
Marco Teórico.....	1
1.1. Revisión de literatura	1
1.1.1. Enfoque temático.....	1
1.1.2. Cadenas de búsqueda	1
1.1.3. Criterios de selección	2
1.1.4. Búsqueda de artículos.....	2
1.1.5. Selección de artículos.....	3
1.2. Aseguramiento de la calidad del software.....	4
1.2.1. Antecedentes	4

1.2.2.	Importancia en el desarrollo de software.....	5
1.2.3.	Técnicas y herramientas.	7
1.2.4.	Normas y estándares para asegurar la calidad.	9
1.1.5	Desafíos en el aseguramiento de la calidad.....	11
1.3.	Pruebas de software.	12
1.3.1.	Proceso de pruebas.....	13
1.3.2.	Documentación de pruebas.....	16
1.3.3.	Niveles de prueba.	17
1.3.4.	Tipos de pruebas.....	20
1.3.5.	Pruebas por su ejecución.....	23
1.4.	Pruebas de software con metodologías ágiles.	24
1.4.1.	Introducción a las metodologías ágiles.....	24
1.4.2.	Pruebas ágiles.	25
1.4.3.	Pruebas ágiles vs pruebas tradicionales.	27
1.4.4.	Plan de pruebas para enfoques ágiles.	28
1.5.	Norma ISO/IEC 25023.	32
1.5.1.	Descripción y alcance.....	32
1.5.2.	Modelo de calidad del producto de software ISO/IEC 25010.....	34
1.4.3.	Característica de Adecuación funcional.....	34
CAPÍTULO 2		37
Desarrollo.....		37
2.1.	Descripción de los procesos a ser automatizados	37
2.1.1.	Solicitud y recepción de materiales	37
2.1.2.	Salida de Materiales.....	39
2.1.3.	Recepción de pedidos	40
2.2.	Preparación y planificación del proyecto.....	41
2.2.1.	Definición de roles y responsabilidades.....	41
2.2.2.	Historias de usuario.	42
2.2.3.	Priorización y organización del Product Backlog.	46
2.2.4.	Arquitectura de la aplicación.	47
2.2.5.	Diseño de base de datos	48
2.2.6.	Plan de pruebas.....	49
2.3.	Proceso de desarrollo Iterativo con Scrum	51
2.3.1.	Nomenclatura y organización de los sprints.....	51

2.3.2.	Planificación y desarrollo del sprint 1	52
2.3.3.	Planificación y desarrollo del sprint 2	58
2.3.4.	Planificación y desarrollo del sprint 3	65
2.3.5.	Planificación y desarrollo del sprint 4	71
2.4.	Cierre del proyecto	78
2.4.1.	Despliegue de la aplicación	78
2.4.2.	Cobertura de pruebas	81
CAPÍTULO 3		83
3.1.	Validación de la Adecuación Funcional según ISO/IEC 25023	83
3.1.1.	Evaluación de la Completitud Funcional	83
3.1.2.	Evaluación de la Corrección Funcional	84
3.1.3.	Pertenencia funcional	87
3.1.4.	Evaluación de adecuación funcional	88
3.1.5.	Interpretación de los resultados	88
DISCUSIÓN		90
	Análisis de hallazgos	90
	Implicaciones teóricas y prácticas	90
	Limitaciones	91
	Investigación futura	91
CONCLUSIONES		92
RECOMENDACIONES		93
BIBLIOGRAFÍA		94
ANEXOS		99

ÍNDICE DE FIGURAS

Figura 1	Árbol de problemas.....	XVIII
Figura 2	Diagrama de arquitectura y pruebas del software	XX
Figura 3	Metodología del proyecto.....	XXI
Figura 4	Fases de inyección de defectos de software.....	6
Figura 5	Relación entre “pruebas” y “aseguramiento de la calidad”	13
Figura 6	Actividades del proceso de pruebas	14
Figura 7	Principios del manifiesto ágil	24
Figura 8	Estrategia para pruebas ágiles	30
Figura 9	Ejemplo de plan de pruebas.....	31
Figura 10	Organización de las series de Normas SQuaRE.....	32
Figura 11	Modelo de calidad del producto de software.....	34
Figura 12	Proceso de solicitud y recepción de materiales.....	38
Figura 13	Proceso de salida de materiales.....	39
Figura 14	Proceso de recepción de pedidos	40
Figura 15	Arquitectura de la aplicación	48
Figura 16	Diseño de base de datos	49
Figura 17	Plan de pruebas.....	50
Figura 18	Pantalla de inicio de sesión	55
Figura 19	Pantalla de administración de usuarios	56
Figura 20	Pantalla de administración de materiales.....	56
Figura 21	Ejecución de pruebas unitarias en el backend sprint 1.....	57
Figura 22	Ejecución de pruebas unitarias en el frontend sprint 1.....	57
Figura 23	Creación de producto.....	62
Figura 24	Gestión de productos.....	62
Figura 25	Pantalla de recepción de pedidos	63
Figura 26	Ejecución de pruebas unitarias en el backend sprint 2.....	64
Figura 27	Ejecución de pruebas unitarias en el frontend sprint 2.....	64
Figura 28	Pantalla de gestión de solicitudes de compra de material	69
Figura 29	Pantalla de compra de materiales a partir de solicitud	69
Figura 30	Ejecución de pruebas unitarias en el backend sprint 3.....	70
Figura 31	Ejecución de pruebas unitarias en el frontend sprint 3	70
Figura 32	Pantalla de registro de movimiento de material	75
Figura 33	Pantalla de orden de producción.....	75
Figura 34	Pedido generado en PDF.....	76
Figura 35	Ejecución de pruebas unitarias en el backend sprint 4.....	76
Figura 36	Ejecución de pruebas unitarias en el frontend sprint 4.....	77
Figura 37	Repositorio de la aplicación	78
Figura 38	Despliegue de base de datos	79
Figura 39	Imágenes Docker de la aplicación en Amazon ECR.....	80
Figura 40	Funcionamiento de la aplicación en Amazon ECS.....	80
Figura 41	Aplicación en producción.....	81
Figura 42	Cobertura de código en el backend	81
Figura 43	Cobertura de código en el frontend.....	82

ÍNDICE DE TABLAS

Tabla 1 Artículos recuperados en la búsqueda y selección	2
Tabla 2 Artículos seleccionados para la revisión	3
Tabla 3 Relación entre artículos seleccionados y conceptos clave.....	4
Tabla 4 Objetivos y objetos por nivel de prueba	20
Tabla 5 Niveles de pruebas por cada tipo.....	22
Tabla 6 Principios de pruebas ágiles	26
Tabla 7 Comparación entre pruebas ágiles y tradicionales	28
Tabla 8 Subcaracterísticas de Adecuación funcional.....	35
Tabla 9 Métricas de calidad para Adecuación funcional	35
Tabla 10 Detalle del proceso de solicitud y recepción de materiales.....	38
Tabla 11 Detalle del proceso de salida de materiales	39
Tabla 12 Detalle del proceso de recepción de pedidos	41
Tabla 13 Roles y responsabilidades del equipo	42
Tabla 14 Historia de Usuario Nro. 1.....	42
Tabla 15 Historia de Usuario Nro. 2.....	43
Tabla 16 Historia de Usuario Nro. 3.....	43
Tabla 17 Historia de Usuario Nro. 4.....	44
Tabla 18 Historia de Usuario Nro. 5.....	44
Tabla 19 Historia de Usuario Nro. 6.....	45
Tabla 20 Historia de Usuario Nro. 7.....	45
Tabla 21 Historia de Usuario Nro. 8.....	46
Tabla 22 Product Backlog.....	46
Tabla 23 Organización de los sprints	51
Tabla 24 Planificación sprint 1	52
Tabla 25 Casos de pruebas unitarias HU-1 Login con roles de usuario	53
Tabla 26 Casos de pruebas unitarias HU-2 Gestión de materia prima	53
Tabla 27 Casos de pruebas de aceptación HU-1 Login con roles de usuario.....	54
Tabla 28 Casos de pruebas de aceptación HU-2 Gestión de materia prima	54
Tabla 29 Ejecución de pruebas de aceptación sprint 1.....	58
Tabla 30 Planificación sprint 2	59
Tabla 31 Casos de pruebas unitarias HU-6 Gestión de productos para producción	59
Tabla 32 Casos de pruebas unitarias HU-7 Recepción de pedidos para producción.....	60
Tabla 33 Casos de pruebas de aceptación HU-6 Gestión de productos para producción.....	61
Tabla 34 Casos de pruebas de aceptación HU-7 Recepción de pedidos para producción	61
Tabla 35 Ejecución de pruebas de aceptación sprint 2.....	65
Tabla 36 Planificación sprint 3	65
Tabla 37 Casos de pruebas unitarias HU-3 Creación de órdenes de compra.....	66
Tabla 38 Casos de pruebas unitarias HU-5 Registro de compras realizadas	67
Tabla 39 Casos de pruebas de aceptación HU-3 Creación de órdenes de compra	67
Tabla 40 Casos de pruebas de aceptación HU-5 Registro de compras realizadas.....	68
Tabla 41 Ejecución de pruebas de aceptación sprint 3.....	71
Tabla 42 Planificación sprint 4	71
Tabla 43 Casos de pruebas unitarias HU-4 Registro de movimiento de materia prima.....	72

Tabla 44 Casos de pruebas unitarias HU-8 Generar orden de producción	73
Tabla 45 Casos de pruebas de aceptación HU-4 Registro de movimiento de materia prima	73
Tabla 46 Casos de pruebas de aceptación HU-8 Generar orden de producción	74
Tabla 47 Ejecución de pruebas de aceptación sprint 4.....	77
Tabla 48 Evaluación de completitud funcional	84
Tabla 49 Evaluación de la corrección funcional	85
Tabla 50 Identificación de las funciones requeridas para el caso de uso.	87
Tabla 51 Grado de cumplimiento de las funciones entregadas.....	88
Tabla 52 Evaluación de la adecuación funcional	88

RESUMEN

Este documento detalla el proceso de aseguramiento de la calidad de software durante su proceso de desarrollo, utilizando pruebas unitarias y de aceptación en los módulos de control de materia prima y recepción de pedidos para la Fábrica de Medias Neltex. Inicialmente, se presenta el marco conceptual relacionado al aseguramiento de la calidad de software, los tipos y niveles de pruebas existentes. Posteriormente, se procede al desarrollo de una aplicación web encargada de automatizar los procesos mencionados, siguiendo la metodología ágil Scrum.

Se establece un plan de pruebas que incluye la creación y ejecución de casos de prueba unitarios y de aceptación. Finalmente, se valida el software desarrollado aplicando la norma ISO/IEC 25023, específicamente la característica de adecuación funcional, para evaluar cuantitativamente la calidad externa del producto.

Este documento se organiza en tres capítulos: el marco teórico, el desarrollo de la aplicación y la validación mediante la norma ISO/IEC 25023. Proporciona una visión detallada del proceso de obtención de un software de calidad, desde la conceptualización hasta la implementación final, garantizando la satisfacción de los requisitos y necesidades del usuario.

ABSTRACT

This document details the process of software quality assurance during its development process, using unit and acceptance testing in the raw material control and order reception modules for Neltex Socks Factory. Initially, the conceptual framework related to software quality assurance, and the existing types and levels of testing are presented. Subsequently, the development of a web application responsible for automating the mentioned processes is carried out, following the agile Scrum methodology.

A test plan is established, which includes the creation and execution of unit and acceptance test cases. Finally, the developed software is validated by applying the ISO/IEC 25023 standard, specifically the functional suitability characteristic, to quantitatively evaluate the external quality of the product.

This document is organized into three chapters: the theoretical framework, the application development, and the validation through the ISO/IEC 25023 standard. It provides a detailed view of the process of obtaining quality software, from conceptualization to final implementation, ensuring the satisfaction of user requirements and needs.

INTRODUCCIÓN

Tema

Aseguramiento de la calidad del software durante su desarrollo usando pruebas unitarias y de aceptación en los módulos de control de materia prima y recepción de pedidos para la Fábrica de Medias Neltex.

Problema

Antecedentes

Actualmente en Latinoamérica se puede ver una crisis de calidad de software, como lo dicen Nakai et al. (2016) solo el 28% de las empresas aplican el estándar ISO/IEC 25023 a sus productos de software, por otro lado, más del 70% de las empresas aplican sus propios modelos de calidad (p.410), mostrando que no todas las empresas desarrolladoras le dan la suficiente importancia a la calidad de su producto de software, así mismo The Standish Group (2015) con su reporte CHAOS indica que el 56% de los productos de software tuvieron problemas de presupuesto, el 60% de tiempo y el 44% no completaron la funcionalidad esperada (p.1), lo que puede causar que a posterior este software genere gastos mayores de mantenimiento e incluso no se llegue a usar.

Situación Actual

En este contexto, surge el caso de estudio, la Fábrica de Medias Neltex ubicada en el cantón Otavalo, la cual cuenta con 15 años de experiencia. Previamente esta empresa implementó un software para la automatización de su proceso de producción el cual, al no satisfacer sus necesidades y presentar fallos como pérdida de información, valores incorrectos o bloqueos de pantalla ocasionaron que optaran por no usarlo, regresando a su situación previa y cayendo en procesos de negocio deficientes, que al no ser resueltos causan costos adicionales y disminución de productividad a la empresa.

Prospectiva

El presente trabajo plantea ayudar a solventar estos problemas, garantizando la calidad del software durante su desarrollo mediante el uso de pruebas funcionales y de aceptación. De esta manera, se busca proporcionar una solución eficiente y confiable que permita a la Fábrica de Medias Neltex optimizar sus procesos de control de materia prima y recepción de pedidos.

Planteamiento del problema

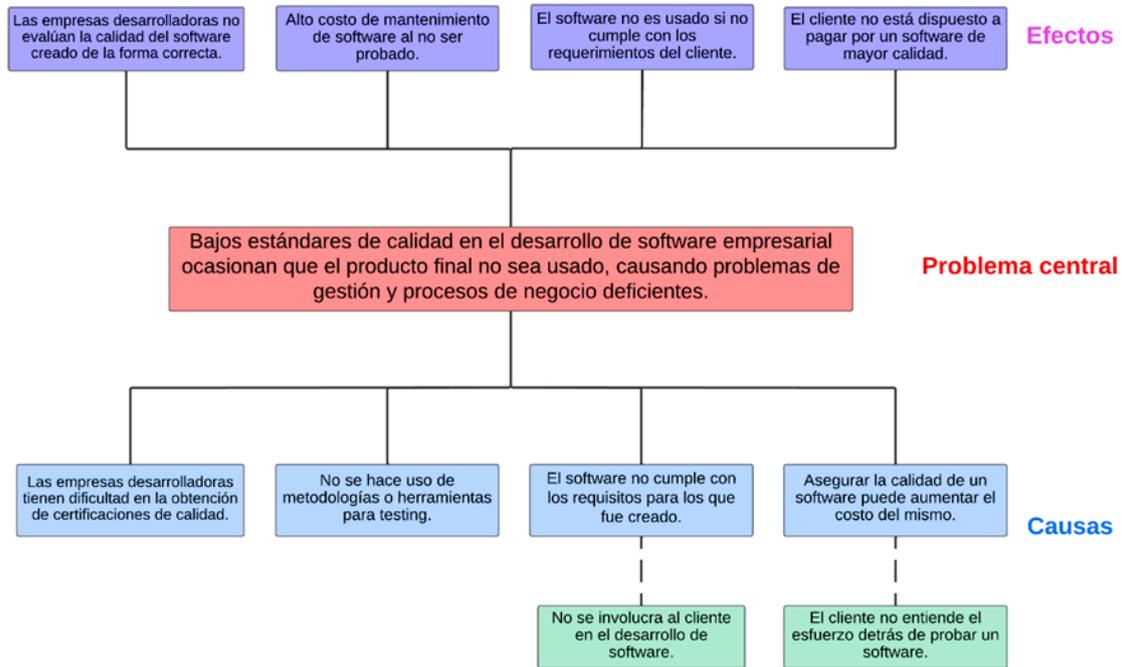
La falta en la calidad del software puede acarrear una serie de inconvenientes significativos en las empresas, como es el caso de la Fábrica de Medias Neltex. Esta fábrica presenta problemas en el proceso de control de ingreso y salida del material necesario para la producción, actualmente no existe un control documentado, simplemente se realiza la compra según lo que se espera utilizar mensualmente, tampoco se registra que material se utilizó por cada pedido o la cantidad de material que disponen.

La falta de control en el material causa otros problemas en el proceso de recepción de pedidos, los cuales no pueden ser estimados correctamente al no conocer de forma real cuanto material poseen en su almacén, es así como suelen tener contratiempos y aumentos en sus costos. La recepción de un pedido actualmente es realizada a papel y entregada a los encargados de producción, dejando posibilidad a que los documentos puedan extraviarse, destruirse o que no quede constancia de haber sido entregados al responsable.

En la Figura 1, se visualiza la representación de estas problemáticas a través de un gráfico de árbol de problemas, señalando las causas, efectos y la problemática principal.

Figura 1

Árbol de problemas



Nota: Elaboración propia.

Objetivos

Objetivo General

Asegurar la calidad del software durante su desarrollo, usando pruebas funcionales y de aceptación para proporcionar una aplicación web que ayude a solventar los problemas de control de ingreso y salida de materia prima y recepción de pedidos de la Fábrica de Medias Neltex.

Objetivos Específicos

- Establecer un marco conceptual referente al aseguramiento de la calidad del software y tipos de pruebas.
- Desarrollar una aplicación web encargada de automatizar los procesos de control de ingreso y salida de materia prima y recepción de pedidos, usando la metodología de desarrollo ágil Scrum.

- Elaborar un plan de pruebas y ejecutar los casos de pruebas para la aplicación web propuesta.
- Validar el software desarrollado aplicando la norma ISO/IEC 25023 mediante la característica de adecuación funcional.

Alcance

A través de la siguiente investigación se llevará a cabo el estudio y la aplicación pruebas unitarias “encargadas de probar el funcionamiento individual de los componentes del software” (Bourque & Fairley, 2014) y pruebas de aceptación, las cuales “se realizan para validar que el software cumple con los requisitos establecidos y está listo para ser desplegado” (Schmidt, 2013).

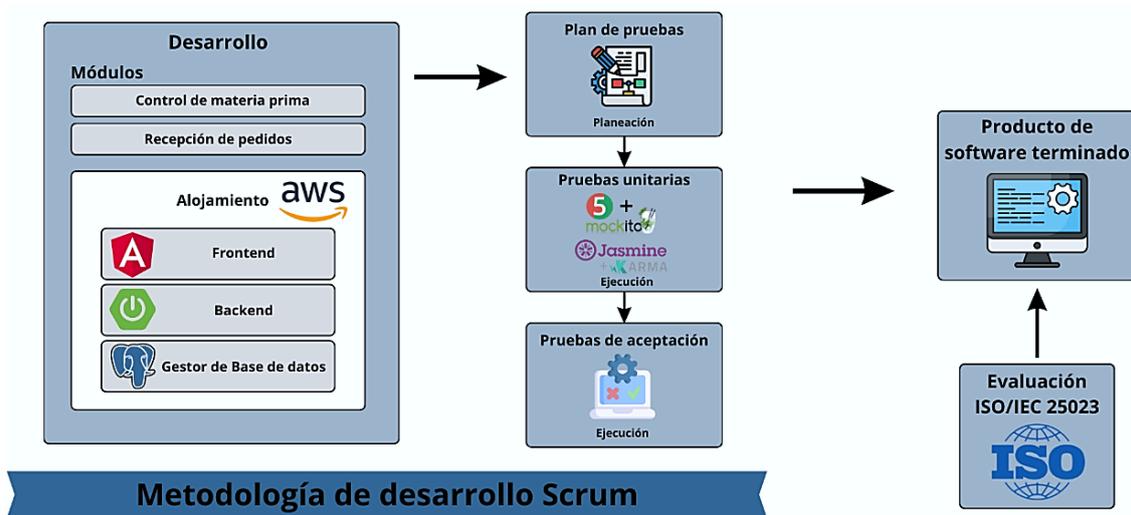
Dichas pruebas se aplicarán a un software para la Fábrica de medias Neltex que contará con módulos para el control de materia prima y recepción de pedidos, dicho software se desarrollará en un entorno web, usando: PostgreSQL para la gestión de base de datos; Spring Boot para la programación backend, al ser un framework “basado en Java y usar el Modelo Vista Controlador, hace fácil el desarrollo y despliegue de los servicios REST” (Haro et al., 2019); Angular para el frontend permitiendo “crear de manera eficiente aplicaciones web avanzadas de una sola página además que hace uso de TypeScript” (Marija, 2020) y todos estos componentes estarán alojados en la plataforma Amazon Web Services que “proporciona servicios como almacenamiento en la nube, servicio de base de datos, análisis, Internet de las cosas en red, informática móvil y servicios empresariales” (Gupta et al., 2021). Las herramientas seleccionadas para las pruebas unitarias del backend son Junit y Mockito que nos permitirán “ejecutar un conjunto de pruebas personalizado utilizando uno o más motores de prueba en la plataforma” (Bechtold et al., 2023). Jasmine y karma para pruebas en el frontend, ya que son herramientas que vienen integradas al instalar el CLI de angular.

Durante la realización de pruebas, se seguirá un plan de pruebas ágiles alineado con la metodología Scrum, es así como al finalizar cada una de las iteraciones se entregará un reporte correspondiente a las pruebas unitarias y de aceptación.

La validación del producto de software se llevará a cabo considerando la característica de adecuación funcional perteneciente a la norma ISO 25023 la cual “define medidas externas para evaluar cuantitativamente la calidad externa del software” (ISO/IEC 25000, 2014).

Figura 2

Diagrama de arquitectura y pruebas del software



Nota: Elaboración propia.

Metodología

Para alcanzar el primer objetivo, se realizará una revisión de literatura en cuanto al aseguramiento de la calidad del software y los diferentes tipos de pruebas para software. Esta revisión se fundamentará en la exploración de fuentes académicas como Google Scholar, IEEE y Scopus. Se utilizarán cadenas de búsqueda con los aspectos fundamentales de estos temas. Los criterios de selección se centrarán en artículos académicos publicados en los últimos 5 años y aquellos que sean más relevantes para el tema.

Para cumplir con el segundo objetivo se usará la documentación de los procesos de recepción de pedidos y control de ingreso y salida de materia prima con la que ya cuenta la empresa, de igual forma se realizará una elicitación de requisitos con el fin de entender que es lo que necesitará el usuario, se definirá cuál será la arquitectura que tendrá el proyecto, las herramientas usadas y la metodología. Para finalizar se procederá a desarrollar la aplicación en un marco de trabajo ágil y aplicando pruebas.

Para cumplir con el tercer objetivo se empleará un plan de pruebas “su propósito es describir los objetivos, el alcance, el enfoque y la orientación de las pruebas de software para las partes interesadas” (Crispin & Gregory, 2009), para su creación se usará una plantilla y se implementará tanto a las pruebas unitarias como de aceptación.

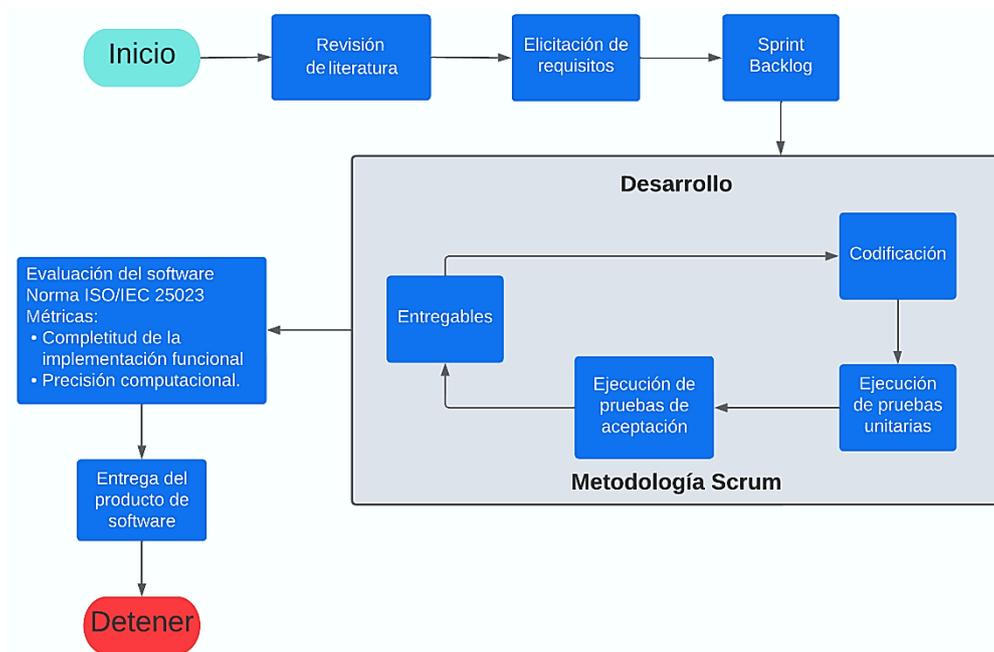
Al utilizar Scrum en el desarrollo del software, la planificación y ejecución de las pruebas se harán en cada uno de los Sprints, teniendo así los resultados de las pruebas, dando libertad para agregar cambios y mejoras para la siguiente iteración.

Para cumplir con el cuarto objetivo se evaluará el producto final de software mediante la característica de adecuación funcional. Para ello se usará métricas definidas en la Norma ISO/IEC 25023 en colaboración con dueño del software.

Una mejor representación del proceso a seguir se puede ver en la Figura 3 en la que se puede observar cada uno de los pasos mediante un diagrama de flujo.

Figura 3

Metodología del proyecto



Nota: Elaboración propia.

Justificación

Con el presente proyecto se pretende contribuir a uno de los Objetivos de Desarrollo Sostenible. Objetivo 9: La industria inclusiva, innovación e infraestructura pueden ser fuerzas que generen empleo e ingresos. Los avances tecnológicos son también esenciales para encontrar soluciones permanentes a los económicos (Naciones Unidas, 2019).

Coopera con las metas nacionales establecidas por el Plan de Creación de Oportunidades 2021-2025. Objetivo 3: “Fomentar la productividad y competitividad en los sectores, industrial, acuícola y pesquero, bajo el enfoque de la economía circular. Al mejorar la competitividad y productividad industrial, incentivando el acceso a infraestructura adecuada, insumos y uso de tecnologías modernas” (Secretaría Nacional de Planificación, 2021).

Además, se acopla con la visión de la Carrera de Software: “forma ingenieros competentes, críticos, humanistas, líderes y emprendedores con responsabilidad social; genera, fomenta y ejecuta procesos tecnológicos, de conocimientos científicos y de innovación en la gestión y desarrollo de software de calidad” (Universidad Técnica del Norte, 2017).

Justificación Tecnológica

El proyecto pretende mejorar los procesos para control de materia prima y recepción de pedidos en la Fábrica de Medias Neltex mediante un software de calidad probado tanto en código como en funcionalidad, además evaluado por la norma ISO/IEC 25023.

Justificación Económica

Este proyecto ayudará a esta y a cualquier empresa de características similares que desee implementar el software a mejorar sus procesos, lo que significa optimizar sus recursos, aprovechando de mejor manera todo el capital que se invierte en ella. También ayudará a mejorar el tiempo del proceso productivo con lo cual se finalizará los pedidos en menos tiempo mejorando las ganancias para la empresa.

Justificación Social

Actualmente en el cantón Otavalo existen empresas dedicadas a la fabricación de medias similar a la Fábrica de Medias Neltex, este proyecto puede ayudarlas a solucionar sus problemas mediante el uso de una herramienta tecnológica que permita automatizar y mejorar sus procesos.

CAPÍTULO 1

Marco Teórico

1.1. Revisión de literatura

Una revisión de literatura es un proceso preliminar que se realiza antes de iniciar una investigación. Permite explorar el conocimiento existente sobre un tema, identificando qué se sabe y qué aún es desconocido. En esta revisión se recopilan referencias bibliográficas relevantes al tema, autor, publicación o trabajo específico en un periodo determinado (Guirao Goris, 2015).

La revisión mostrada a continuación se centra en analizar y sintetizar la literatura académica más relevante y actualizada en el campo del aseguramiento de la calidad en el software y los tipos de pruebas de software. Enfocada en los últimos 5 años y utilizando fuentes en español e inglés, se busca identificar las tendencias, metodologías y mejores prácticas que impactan en la calidad y el desempeño del software.

1.1.1. Enfoque temático

El estado actual, tendencias y mejores prácticas en aseguramiento de calidad y pruebas en el ámbito de la ingeniería de software contemporánea.

1.1.2. Cadenas de búsqueda

Las cadenas de búsqueda en español e inglés se centran en términos clave relacionados con el aseguramiento de calidad y pruebas de software, abarcando el período de 2019 a 2024. Estas cadenas garantizan la recuperación de información actualizada y relevante en ambos idiomas para la investigación.

Cadena de búsqueda en español

```
("aseguramiento" AND "calidad" AND "software") OR ("calidad" AND "software") OR ("estándares" AND "calidad" AND "software") OR ("pruebas" AND "software") OR ("niveles" AND "pruebas" AND "software") OR ("tipos" AND "pruebas" AND "software") AND (year:[2019 TO 2024])
```

Cadena de búsqueda en inglés

```
("assurance" AND "quality" AND "software") OR ("quality" AND "software")  
OR ("standards" AND "quality" AND "software") OR ("testing" AND  
"software") OR ("levels" AND "testing" AND "software") OR ("types" AND  
"testing" AND "software") AND (year:[2019 TO 2024])
```

1.1.3. Criterios de selección

Criterios de inclusión

- Artículos publicados en revistas científicas o actas de conferencias indexadas.
- Artículos enfocados en aseguramiento de la calidad del software y pruebas de software.
- Artículos publicados entre el año 2019 y 2024.

Criterios de exclusión

- Artículos que no satisfacen los criterios de inclusión.
- Artículos no disponibles en texto completo.
- Filtrado adicional en los motores de búsqueda

1.1.4. Búsqueda de artículos

Para la búsqueda de artículos, se utilizan diversos motores de búsqueda científica con el propósito de recopilar información relevante para la revisión de literatura. Los resultados obtenidos corresponden a los más acercados al tema de la revisión y más relevantes dentro de cada motor de búsqueda, estos se presentan a continuación en la Tabla 1.

Tabla 1

Artículos recuperados en la búsqueda y selección

Motores de Búsqueda Científica	Resultado
Science Direct	40
Google Scholar	28
IEEE	18
Springer	32
Total	118

Nota: Elaboración propia.

Estos resultados muestran la cantidad de artículos recuperados de cada motor de búsqueda, lo que proporciona una amplia variedad de fuentes para la revisión de literatura.

1.1.5. Selección de artículos

La selección de artículos fue realizada según la relevancia de estos para la revisión de literatura. A continuación, se presenta una lista de artículos seleccionados junto con sus códigos y títulos.

Tabla 2

Artículos seleccionados para la revisión

Código	Título
A1	Ingeniería de Software: El aseguramiento de la calidad de los requisitos en la industria del software en el Eje Cafetero colombiano
A2	A Comparative Study of Dynamic Software Testing Techniques
A3	Planning, Management and Control of Software Quality
A4	SQAPlanner: Generating Data-Informed Software Quality Improvement Plans
A5	Aspects of software quality applied to the process of agile software development: a systematic literature review
A6	Comprehensive study of software testing: Categories, levels, techniques, and types
A7	Software Testing from an Agile and Traditional view
A8	A Study of the Organisational Behaviour of Software Test Engineers, Contributing to the Digital Transformation of Banks in the Irish Financial Sector
A9	Agile Test Management, Methods, and Techniques
A10	Exploring human factors of the agile software tester
A11	Methods of quality assurance of software development based on a systems approach
A12	A review on software testing and its methodology
A13	Desarrollo guiado por comportamiento: buenas prácticas para la calidad de software
A14	Modelo de medición y evaluación de calidad del software basado en la norma ISO/IEC 25000 para medir la usabilidad en productos de software académicos universitarios
A15	Calidad del Software una Perspectiva Continua

Nota: Elaboración propia.

La Tabla 3 que se presenta a continuación detalla los conceptos clave vinculados al aseguramiento de calidad, estándares de calidad de software, tipos de pruebas de software y gestión de pruebas abordados en los artículos proporcionados.

Tabla 3

Relación entre artículos seleccionados y conceptos clave

Código	Aseguramiento de Calidad	Estándares de Calidad de Software	Tipos de Pruebas de Software	Gestión de Pruebas
A1	X			
A2			X	
A3	X			X
A4	X			X
A5	X			
A6			X	
A7			X	
A8				
A9				X
A10				
A11	X	X		
A12	X			
A13	X			
A14	X	X		
A15	X			

Nota: Elaboración propia.

1.2. Aseguramiento de la calidad del software

1.2.1. Antecedentes.

Para comprender a qué se refiere el Aseguramiento de la Calidad en el Software, primero es necesario entender los conceptos que se relacionan con él.

Calidad de software:

En general, la Real Academia Española (2023) define la ‘calidad’ como la “propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.” Al hablar de calidad en un producto de software se suele usar el término “Calidad de software”, que como menciona Arcos-Medina & Mauricio (2019), refiere a la habilidad de satisfacer los requisitos funcionales y no funcionales explícitamente definidos por el cliente, así como a los estándares de desarrollo y los aspectos previstos para el software desarrollado de forma profesional.

Control de Calidad:

El término se refiere a la gestión de calidad que se centra a garantizar que se cumplan los requisitos y en evaluar la cantidad de defectos o errores, en caso de que los haya, en la aplicación. Su rol comprende una serie de procesos destinados a evaluar la aplicación desarrollada, como borradores de documentos o sistemas de desarrollo, para verificar el cumplimiento de los requisitos del cliente. El objetivo del control de calidad es identificar defectos y garantizar su corrección (Ushakova et al., 2021).

Definición de Aseguramiento de la calidad del software.

Una vez tratados los conceptos de “calidad de software” y “control de calidad”, se puede abordar el concepto de Aseguramiento de la Calidad del Software (SQA por sus siglas en inglés). Este consiste en un conjunto de métodos, herramientas y técnicas que definen y evalúan si los procesos de software son apropiados y cumplen con los requisitos necesarios para generar confianza en que se producirán productos de software de calidad adecuada para los propósitos previstos (Laporte & April, 2018).

Es decir, mientras el aseguramiento de la calidad está destinado a garantizar el desarrollo de software de alta calidad, el control de calidad captura y evalúa la calidad de una aplicación ya creada. Entonces, el control de calidad es un subsistema del sistema de aseguramiento de la calidad (Ushakova et al., 2021).

1.2.2. Importancia en el desarrollo de software.

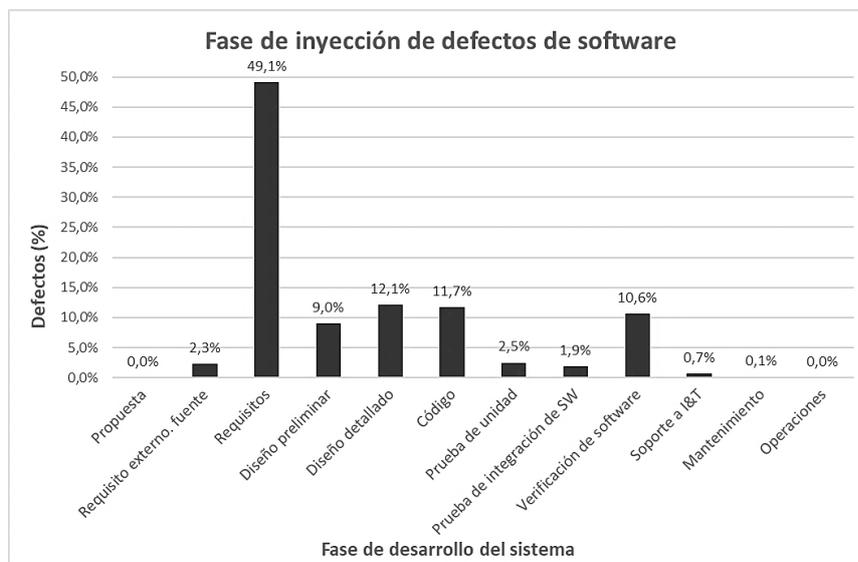
El Aseguramiento de la Calidad del Software (SQA) juega un papel destacado, ya que enfatiza que la calidad no debe considerarse solo al final, sino que exige acciones específicas a lo largo de todo el desarrollo del proyecto. Esto implica la integración constante de actividades de calidad, incluyendo estándares, administración de configuración, auditorías, revisiones, pruebas, capacitación, administración de riesgos, procesos de requisitos y procesos de diseño (Peláez et al., 2019).

Su alcance de responsabilidad en el desarrollo de software abarca diversos miembros de la organización, como ingenieros de software, desarrolladores, líderes de proyecto, clientes y personas encargadas del SQA (Carrizo & Alfaro, 2018).

Es bien sabido que el costo de los defectos de software aumenta significativamente si se descubren después del lanzamiento. Por lo tanto, la labor de los equipos de SQA es crucial, actuando como guardianes para evitar que los defectos lleguen a los usuarios finales (Rajapaksha et al., 2021), especialmente en sistemas de “alta confianza” o “alta integridad” donde las fallas pueden tener consecuencias extremadamente negativas. Su enfoque se extiende más allá de la funcionalidad del sistema, abarcando hardware, software y factores humanos (Ushakova et al., 2021).

Figura 4

Fases de inyección de defectos de software



Nota: Obtenido de (Laporte & April, 2018).

En la Figura 4 Laporte & April (2018) muestran el porcentaje de defectos según su fase durante el ciclo de vida de desarrollo en el software. Este estudio, muestra que aproximadamente el 50% de los defectos se originan en la fase de requerimientos. Si no se corrigen estos defectos en etapas tempranas, su solución en fases posteriores resultará mucho más costosa (Laporte & April, 2018).

1.2.3. Técnicas y herramientas.

Verificación y Validación (V&V).

La Verificación y Validación desempeñan un papel fundamental en el desarrollo de sistemas o componentes de software estas actividades se enfocan en garantizar que los requisitos establecidos se cumplan de manera adecuada y que los productos obtenidos en cada etapa del ciclo de vida del software satisfagan las condiciones impuestas por la fase anterior (Guevara-Vega, Patiño Chalacán, et al., 2022).

La verificación implica un proceso meticuloso de revisión para determinar si los requisitos iniciales se han abordado de manera completa y precisa. Estas prácticas rigurosas permiten detectar y corregir deficiencias o desviaciones de manera oportuna, minimizando el riesgo de errores o incumplimientos en etapas posteriores. En consecuencia, la verificación y validación contribuyen significativamente a la calidad, confiabilidad y éxito del producto final (Guevara-Vega, Patiño Chalacán, et al., 2022)

Pruebas de software.

Soraluz et al. (2023) hablan sobre un estudio realizado en Europa en 2019 el cual revela que el desarrollo de software dedica aproximadamente el 59,4% del esfuerzo a la implementación de pruebas para garantizar la calidad. Este enfoque es el factor principal considerado por las empresas para mejorar el rendimiento y lograr una mejora continua en la industria de desarrollo. La implementación de pruebas de software se considera un componente esencial para asegurar la estructura funcional y la calidad de los productos de software (Soraluz et al., 2023).

Las pruebas son reconocidas como herramientas útiles que proporcionan información detallada sobre el cumplimiento de los requisitos funcionales, no funcionales, estructurales y no estructurales a lo largo del ciclo de vida de un producto de software (Soraluz et al., 2023).

Plan de aseguramiento de la calidad del software.

El Plan de Aseguramiento de la Calidad del Software (SQAP) es el conjunto de actividades y tareas diseñadas para asegurar que el software desarrollado para un producto específico cumpla con los requisitos del proyecto y las necesidades del usuario, dentro de los límites de costo, cronograma y en función de los riesgos asociados (Bourque & Fairley, 2014).

El plan se inicia asegurando que los objetivos de calidad estén claramente definidos y comprendidos. También identifica medidas; técnicas estadísticas; procedimientos para la notificación de problemas y acciones correctivas; recursos como herramientas, técnicas y metodologías; seguridad para medios físicos; capacitación; e informes y documentación de SQA (Bourque & Fairley, 2014).

Normas y estándares de calidad.

Varios organismos especializados, como la Organización Internacional de Normalización (ISO) y el Instituto de Ingeniería de Software (SEI), han desarrollado diversos estándares, metodologías y buenas prácticas. Entre ellos, se destacan ISO/IEC 9126, enfocado en el producto de software, CMMI (Capability Maturity Model Integration), TSP (Team Software Process) y PSP (Personal Software Process), que buscan de igual forma asegurar la calidad del producto de software (Espejo et al., 2016). Estas normas y estándares serán tratados a mayor profundidad más adelante en el trabajo.

Otros enfoques

Es importante tener en cuenta que las formas mostradas no son las únicas existentes, otras variantes de metodologías o métodos también buscan asegurar la calidad del software, ajustándose a los requerimientos particulares de cada proyecto. Por ejemplo, el trabajo de Carrizo & Alfaro (2018) menciona dos enfoques alternativos:

El primero, que a pesar de no ser un método o plan de aseguramiento se basa en diversos estándares de calidad, que proporcionan una serie de características para guiar con éxito un desarrollo de software. Este marco tiene en cuenta las dificultades que enfrentan las

pymes al implementar métodos o estándares de calidad, debido a su complejidad, costos y limitaciones.

El segundo nos habla de un plan de aseguramiento de la calidad con diversas actividades de verificación con checklist para las etapas del desarrollo del proyecto. Su principal enfoque es controlar si se cumplen los requisitos al final de cada etapa del proyecto y, por lo tanto, no se presenta como una guía integral para aseguramiento de la calidad, sino más bien un control de calidad puntual.

1.2.4. Normas y estándares para asegurar la calidad.

El software, como producto intangible y resultado del conocimiento humano, presenta desafíos para mantener y asegurar su calidad. Para mejorar los procesos, productos y proyectos relacionados con el desarrollo y mantenimiento de software, las organizaciones utilizan estándares de calidad que las ayudan a asegurar y mantener la calidad del software (Arias et al., 2018). En este contexto, se presentan diversas normas y estándares que guían a los profesionales en la búsqueda de en el desarrollo de software.

IEEE 730 - Estándar para Aseguramiento de la Calidad del Software

Este estándar se enfoca en proporcionar requisitos uniformes y mínimamente aceptables para los procesos de Aseguramiento de la Calidad del Software (SQA) en proyectos de desarrollo o mantenimiento de software. Basado en el ciclo de vida del software de ISO/IEC/IEEE 12207 y los requisitos de contenido de información de ISO/IEC/IEEE 15289, el estándar guía a los profesionales de SQA en la producción y recopilación de evidencias que respalden la declaración de confianza en que el producto de software cumple con los requisitos establecidos (IEEE Computer Society, 2014).

Se definen 16 actividades de SQA, agrupadas en tres áreas principales: implementación del proceso, aseguramiento del producto y aseguramiento del proceso, el estándar también ofrece orientaciones para su aplicación en diferentes industrias, métodos ágiles, niveles de integridad y herramientas de software (IEEE Computer Society, 2014).

CMMI - Capability Maturity Model Integration

Es un modelo que orienta y se utiliza en el desarrollo de procesos. A diferencia de otros modelos, no prescribe procesos o descripciones específicas, ya que estos dependen de diversos factores, como el dominio de aplicación, la estructura y el tamaño de la organización, es ampliamente reconocido y se ha convertido en una exigencia en el ámbito global para la exportación de servicios de software (Castro et al., 2019).

CMMI ofrece una guía para la mejora de procesos de organizaciones dedicadas al desarrollo y mantenimiento de software, permitiendo crear estrategias de calidad y mejora continua, ofrece un programa de certificación para empresas privadas, sin ser una certificación ISO (Castro et al., 2019).

ISO/IEC 15504 (SPICE) - Software Process Improvement and Capability Determination

SPICE es un estándar internacional que permite evaluar la capacidad y madurez de los procesos de software de las organizaciones, su objetivo no es fijar cómo realizar los procesos, sino valorar su capacidad y proponer mejoras (Acosta et al., 2017).

Inicialmente estructurado en nueve apartados que ofrecen una base para implementar paso a paso el modelo con su correspondiente evaluación, con el tiempo se redujo el tamaño del estándar a cinco partes con el fin de consolidarlo. Su aplicación proporciona valores cuantitativos para determinar el nivel de calidad del producto software (Acosta et al., 2017).

PSP - Personal Software Process

Es un marco estructurado para desarrollo de software que ayuda a los ingenieros a controlar, gestionar y mejorar su forma de trabajo. Proporciona formas, pautas y procedimientos para estimar, planificar y cumplir con compromisos. La premisa es que la calidad del software depende del trabajo individual de los ingenieros, y PSP busca mejorar la productividad y la calidad a través de mediciones (Espejo et al., 2016).

Sus objetivos principales son maximizar la calidad del software, fomentar la mejora continua en el proceso de desarrollo y aumentar la productividad, utiliza métricas como

tamaño, tiempo y defectos para obtener datos útiles y reducir errores en el software (Espejo et al., 2016).

ISO/IEC 9126 - Estándar de Calidad del Software

El estándar ISO/IEC 9126, ahora sustituido por ISO/IEC 25000, se enfoca en las características y métricas de la calidad del software. Clasifica los atributos de calidad en seis características: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad (Sifuentes & Peralta, 2022). Además, las métricas de calidad en uso evalúan el nivel en que el software satisface las necesidades de los usuarios en un contexto específico (Arcos-Medina & Mauricio, 2019).

ISO 25000 - SQuaRE (System and Software Quality Requirements and Evaluation)

La familia ISO/IEC 25000, también conocida como SQuaRE, permite evaluar las características de calidad del producto software y obtener valores cuantitativos para determinar el nivel de calidad del producto (Sifuentes & Peralta, 2022). Esta familia de normas es el resultado de la evolución de normas anteriores, como ISO/IEC 9126 e ISO/IEC 14598, esta familia de normas aborda tanto el modelo de calidad como el proceso de evaluación de productos software (Sifuentes & Peralta, 2022).

ISO 12207 - Ciclo de Vida del Software

La norma ISO/IEC 12207 define un marco común para los procesos del ciclo de vida del software, se aplica a la adquisición, desarrollo, operación, mantenimiento y eliminación de productos de software (Aizprua et al., 2019). El estándar proporciona una descripción detallada de cada proceso, incluyendo el alcance, el propósito, los resultados esperados y las actividades recomendadas (Laporte & April, 2018; Aizprua et al., 2019).

1.1.5 Desafíos en el aseguramiento de la calidad.

En el contexto del aseguramiento de la calidad en el software, uno de los principales desafíos es su implementación en pequeñas empresas. Estas organizaciones requieren enfoques específicos, adaptados a su tamaño y tipo de negocio. A menudo, se enfrentan al

dilema de considerar los programas de mejora como simples versiones a escala de grandes compañías, lo cual no es suficiente para satisfacer sus necesidades (Carrizo & Alfaro, 2018).

Las empresas desarrolladoras deben lidiar con la creciente demanda de calidad por parte de los clientes, la calidad del producto y el cumplimiento de estándares son requisitos fundamentales para garantizar la satisfacción del cliente (Espejo et al., 2016). Sin embargo, el aseguramiento de la calidad en el software implica la implementación de métodos integrados de evaluación y el uso de indicadores de calidad individuales. Esta evaluación integrada debe llevarse a cabo en todas las etapas del ciclo de vida del software, lo que puede resultar en un desafío adicional para las organizaciones (Ushakova et al., 2021).

Otro de los obstáculos más frecuentes es la percepción de un alto costo, las organizaciones a menudo no cuentan con datos precisos sobre el costo de la no calidad en sus proyectos de desarrollo de software (Laporte & April, 2018). Esto puede llevar a la resistencia a implementar procesos y prácticas de aseguramiento de la calidad, ya que algunas empresas pueden considerar que es una inversión innecesaria.

Superar la percepción del alto costo asociado al aseguramiento de la calidad las empresas deben considerar beneficios a largo plazo. Evitar los costos adicionales por defectos y errores en el software desarrollado puede resultar en ahorros significativos y en la mejora de la reputación de la empresa (Laporte & April, 2018). Al enfrentar estos desafíos y adoptar prácticas eficientes de Ingeniería de Software, las organizaciones producirán software de calidad que respondan a las necesidades y fortalecerán su posición en el mercado.

1.3. Pruebas de software.

Las pruebas de software son un proceso crucial en la ingeniería de software, una parte que busca calidad y eficiencia de los productos creados. Como menciona Byrne et al. (2022), este proceso implica la ejecución de programas con datos que simulan las entradas del usuario final, lo que permite determinar si el comportamiento del software se ajusta a las

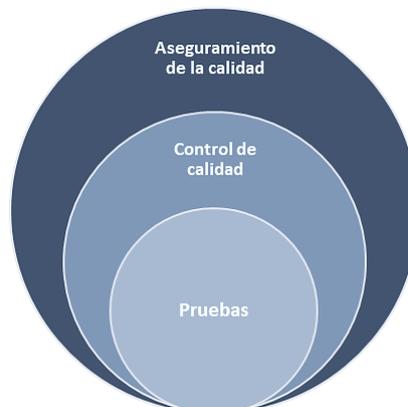
expectativas derivadas de los requisitos del proyecto. Sin embargo, vale la pena destacar que las pruebas no buscan únicamente demostrar que el software está libre de errores, sino que también se esfuerzan por ofrecer la confianza necesaria para su instalación, contribuyendo así a la calidad y fiabilidad del producto final (Umar & Zhanfang, 2020).

Desde sus inicios, las pruebas de software han evolucionado significativamente, en términos de eficiencia y recursos asignados. Según Ahmad et al. (2019), estas pruebas solían consumir una cantidad considerable de tiempo, lo que destacaba la necesidad de optimizar los métodos de prueba y reducir costos. En la actualidad, el presupuesto destinado a las pruebas de software puede representar una parte sustancial de los recursos, oscilando entre el 25% y el 50%, lo que refleja su relevancia en la obtención de software confiable y funcional.

Teniendo en cuenta los conceptos tratados en el capítulo anterior y como explica Ushakova et al. (2021), las “pruebas” son parte del “control de calidad”, y este último se encuentra contenido dentro del “aseguramiento de la calidad”, como se muestra la Figura 5.

Figura 5

Relación entre “pruebas” y “aseguramiento de la calidad”



Nota: Adaptado de (Ushakova et al., 2021).

1.3.1. Proceso de pruebas.

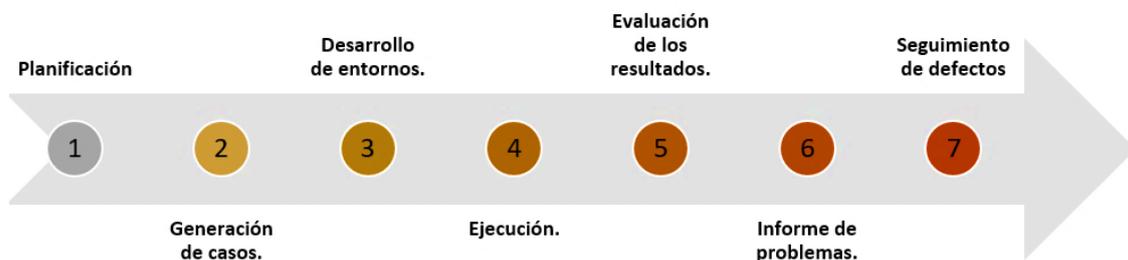
No existe un único proceso de prueba adecuado para todas las situaciones. Cada organización debe ajustar su enfoque de prueba según su contexto específico, considerando una variedad de factores. Estos incluyen el modelo de ciclo de vida de desarrollo de software

y las metodologías de proyecto utilizadas, la elección de los niveles y tipos de pruebas, los riesgos tanto del producto como del proyecto, entre otros (Graham et al., 2019).

Las actividades del proceso de pruebas descritas a continuación y en la Figura 6 son las presentadas en Software Engineering Body of Knowledge (SWEBOK) por Bourque & Fairley (2014), que recoge un conjunto de conocimientos en la ingeniería de software.

Figura 6

Actividades del proceso de pruebas



Nota: Elaboración propia.

Planificación.

Como con cualquier otro componente de la gestión de proyectos, es esencial realizar una planificación adecuada de las actividades de prueba. En esta planificación, es crucial coordinar el personal, asegurar la disponibilidad de las instalaciones y equipos necesarios para las pruebas, así como la creación y mantenimiento de toda la documentación relacionada con estos procesos. Además, se debe considerar la posibilidad de resultados no deseados y planificar en consecuencia. Si se mantienen múltiples versiones base del software, se vuelve esencial estimar el tiempo y esfuerzo requeridos para garantizar que el entorno de pruebas esté configurado de manera adecuada (Bourque & Fairley, 2014).

Generación de casos de prueba.

La creación de casos de prueba depende del tipo de prueba que se va a realizar y de las técnicas específicas de prueba utilizadas. Estos casos de prueba deben ser gestionados mediante el sistema de control de configuración del software y deben incluir los resultados

que se esperan obtener para cada prueba (Bourque & Fairley, 2014). En la elaboración de los casos de prueba también se llevará a cabo la identificación de los datos que serán empleados en las evaluaciones, y se realizará una verificación adicional para garantizar que dichos datos cumplan los requisitos previamente definidos (Ruliansyah et al., 2023)

Desarrollo de entornos de prueba.

Durante esta fase es esencial garantizar la compatibilidad del entorno con las herramientas empleadas y que dicho entorno facilite la creación y control de casos de prueba, así como la gestión de resultados esperados, scripts y otros materiales de prueba (Bourque & Fairley, 2014). Además, se encarga de la preparación de los entornos de prueba y los datos de prueba, lo que puede llevarse a cabo en paralelo con el proceso de generación de casos de prueba (Ruliansyah et al., 2023). Esta coordinación garantiza la adecuada configuración de los entornos de prueba para realizar evaluaciones efectivas del software en desarrollo.

Ejecución.

Esta fase implica ejecutar los planes y casos de prueba creados previamente. Si el caso de prueba se ejecuta según lo esperado se lo marcará como "aprobado". Por otro lado, si no cumple con las expectativas, se etiquetará como "fallido". En caso de detectar un error, se informará al equipo de desarrollo, se efectuarán las correcciones necesarias y una vez que se solucione el error, se repetirá la prueba (Ruliansyah et al., 2023).

Cualquier acción durante las pruebas debe llevarse a cabo y describirse con la suficiente claridad como para que alguien más pueda reproducir los resultados. Por lo tanto, las pruebas deben llevarse a cabo siguiendo procedimientos documentados utilizando una versión específica y bien definida del software que se está evaluando (Bourque & Fairley, 2014).

Evaluación de los resultados.

Es necesario analizar el resultado de las pruebas para determinar su éxito. Por lo general, se considera "exitoso" cuando el software funciona según lo esperado y no presenta resultados inesperados significativos. Sin embargo, no todos los resultados inesperados son

necesariamente fallos, a veces son solo interferencias menores. Para abordar un fallo, es crucial realizar un proceso de análisis y depuración para aislarlo, identificarlo y describirlo. Cuando los resultados de las pruebas son especialmente relevantes, puede ser necesario convocar una revisión formal para evaluarlos (Bourque & Fairley, 2014).

Informe de problemas (Registro de pruebas).

Las actividades de prueba se registran en un archivo de registro de pruebas, lo que permite identificar cuándo se realizó cada prueba, quién la llevó a cabo, la configuración de software utilizada y otra información relevante para su identificación. Los resultados inesperados o incorrectos de las pruebas se pueden anotar en un sistema de informes de problemas, cuyos datos sirven como base para abordar y resolver posteriormente los problemas que se identificaron.

Además, cualquier anomalía que no se clasifique inicialmente como una falla podría documentarse en caso de que posteriormente se revele como más grave de lo que se pensaba en un principio. Los informes de prueba también se utilizan como entradas para el proceso de solicitud de gestión de cambios (Bourque & Fairley, 2014).

Seguimiento de defectos.

Es posible rastrear y analizar los defectos para determinar cuándo se introdujeron en el software, las razones detrás de su creación (como requisitos mal definidos, declaraciones de variables incorrectas, fugas de memoria, etc.) y cuándo podrían haber sido identificados por primera vez en el software. La información obtenida del seguimiento de los defectos se utiliza para identificar áreas que requieren mejoras en las pruebas de software y otros procesos, además de evaluar la eficacia de enfoques previos (Bourque & Fairley, 2014).

1.3.2. Documentación de pruebas

Ya sea que se esté empleando metodologías tradicionales o ágiles, es esencial contar con una cantidad mínima de documentación que facilite la comprensión de lo que se va a evaluar. Esta documentación incluye formularios y plantillas para pruebas de software, las

cuales pueden estar disponibles para diversas organizaciones, proyectos particulares o actividades de prueba individuales (Guevara-Vega, Cárdenas-Hernández, et al., 2022).

Dentro de la documentación más común para pruebas se encuentra:

Plan de pruebas. Este documento formal, elaborado por el Líder de Pruebas, detalla la planificación de las actividades de prueba de software, incluyendo los tiempos y las personas responsables de su ejecución (Guevara-Vega, Cárdenas-Hernández, et al., 2022).

Especificación de diseño de prueba. El Líder de Pruebas elabora este documento que contiene el conjunto de características del software a evaluar, junto con su respectiva prioridad de ejecución (Guevara-Vega, Cárdenas-Hernández, et al., 2022).

Especificación del caso de prueba. Este documento detalla los escenarios específicos que se ejecutarán para cada característica del software identificada en el diseño de prueba.

Procedimiento de prueba. Este documento especifica el orden y la secuencia de ejecución de los casos de prueba, incluyendo restricciones y acciones previas para cada caso (Guevara-Vega, Cárdenas-Hernández, et al., 2022).

Informe de defectos. Se trata de un registro que documenta los casos de prueba ejecutados cuyo resultado difiere del esperado, indicando así la insatisfactoria conclusión de la prueba (Guevara-Vega, Cárdenas-Hernández, et al., 2022).

Informe de resultados. Un documento de gestión que resume de manera concisa los resultados obtenidos durante la ejecución de los casos de prueba, determinando la idoneidad del software para su implementación o la necesidad de corregir posibles errores (Guevara-Vega, Cárdenas-Hernández, et al., 2022).

1.3.3. Niveles de prueba.

El proceso de prueba incluye varios niveles en diversas etapas del desarrollo y mantenimiento de una aplicación. Estos niveles abordan aspectos como funcionalidad, seguridad y rendimiento. La clasificación mostrada es la propuesta por International Software

Testing Qualifications Board (ISTQB) que ofrece una estructura clara para comprender los niveles de prueba y su contribución a la calidad del software.

Pruebas de componentes (Unitarias).

La prueba de componentes, también llamada prueba de unidades o módulos busca detectar fallos y confirmar el funcionamiento de partes individuales del software, como módulos, programas u objetos. Estas pruebas se centran en los requisitos y especificaciones detalladas del componente y su código. Los elementos probados son diversos: componentes individuales, programas de conversión de datos para nuevas versiones, tablas de bases de datos, uniones, vistas, módulos, procedimientos, integridad referencial, restricciones de campo e incluso bases de datos completas (Graham et al., 2019).

Esta prueba es esencial para verificar la correcta operación del código más pequeño antes de la integración de las unidades en el software. Usualmente, el propio desarrollador realiza estas pruebas unitarias (Bele et al., 2022).

Pruebas de integración.

Involucran la verificación del funcionamiento conjunto de dos o más unidades, asegurando la coherencia de parámetros, su formato de archivos, entre otros aspectos. Esto valida la correcta conexión y la adecuada integración en el diseño global (Umar, 2020).

Estas pruebas evalúan las interfaces entre los componentes y las interacciones entre diferentes partes de un sistema, tales como un sistema operativo, un sistema de archivos, las conexiones con el hardware, o incluso las interfaces que conectan sistemas diferentes (Graham et al., 2019).

Pruebas de sistema.

La prueba del sistema implica la evaluación integral de todo el sistema como unidad, con el objetivo de comprobar su calidad. Aquí, los atributos tanto funcionales como no funcionales desempeñan un papel crucial y serán evaluados minuciosamente (Sneha & Malle, 2017). Esto puede involucrar pruebas basadas en análisis de riesgos, especificaciones de

requisitos del sistema, procesos comerciales, casos de uso y descripciones de alto nivel del comportamiento del sistema, incluso las interacciones con el sistema operativo y los recursos del sistema (Graham et al., 2019).

La prueba del sistema también es el lugar donde se evalúa la conformidad con requisitos legales, reglamentarios o estándares externos. Es esencial que el entorno de prueba sea lo más similar posible al entorno de producción final, para garantizar la validez de las pruebas (Graham et al., 2019).

Pruebas de aceptación.

El principal objetivo de las pruebas de aceptación es el de asegurar que el sistema cumpla con las expectativas del cliente y esté listo para su implementación en un entorno real (Umar, 2020). Es importante destacar que las pruebas de aceptación se realizan cuando el sistema está listo para ser entregado, marcando la fase previa a la producción (Hafner, 2020).

Una vez que la organización de desarrollo ha completado las pruebas del sistema, y posiblemente las pruebas de integración, el software es sometido a las pruebas de aceptación. Aunque esta fase puede revelar defectos, el enfoque principal está en validar la adecuación del sistema para su implementación en entornos de producción, satisfaciendo los requisitos definidos previamente. Además, se verifica el cumplimiento de normativas, regulaciones y estándares (Graham et al., 2019).

Características de los niveles de pruebas.

La Tabla 4 muestra los objetivos y objetos de prueba correspondientes a cada nivel de pruebas. Proporciona una clara comprensión de las metas específicas que se buscan alcanzar y las entidades o elementos que se someten a análisis en cada etapa del proceso.

Tabla 4

Objetivos y objetos por nivel de prueba

Característica	Pruebas de componentes	Pruebas de integración	Pruebas de sistema	Pruebas de aceptación
Objetivos	Verificar el comportamiento funcional Generar confianza en los componentes Encontrar defectos	Reducir el riesgo Generar confianza en las interfaces Prevenir defectos a niveles más altos	Validar la integridad, funciona como se esperaba Generar confianza en todo el sistema Prevenir defectos a niveles más altos	Establecer confianza en todo el sistema Validar que funciona como se esperaba Verificar el comportamiento funcional y no funcional
Objetos de prueba	Componentes, unidades, módulos Código Estructuras de datos Clases Modelos de bases de datos	Subsistemas Bases de datos Infraestructura Interfaces API Microservicios	Aplicaciones Hardware software Sistemas operativos Sistemas bajo pruebas Configuración y datos del sistema	Configuración del sistema Procesos de negocios Procesos de operación y mantenimiento Formularios Informes Datos de producción

Nota: Adaptado de (Graham et al., 2019).

1.3.4. Tipos de pruebas.

Las pruebas de software pueden ser clasificadas de diversas formas, dependiendo de los aspectos particulares que se deseen evaluar. Al igual que en la sección previa, la clasificación mostrada está respaldada por el ISTQB, ofreciendo una estructura sólida para comprender y aplicar una variedad de tipos de pruebas.

Pruebas Funcionales.

Las pruebas funcionales evalúan el comportamiento de un sistema según sus funciones descritas en documentos de requisitos o comprendidas por los evaluadores. Estas pruebas se pueden realizar en distintos niveles, como pruebas de componentes o pruebas de sistema. Pueden enfocarse en la idoneidad, la interoperabilidad, la seguridad y otros aspectos, incluyendo funciones no documentadas pero supuestas. Estas pruebas pueden ser basadas en requisitos o en procesos comerciales, utilizando especificaciones de requisitos o escenarios de uso para diseñar casos de prueba (Graham et al., 2019).

Pruebas No Funcionales.

Las pruebas no funcionales se centran en medir características de calidad y atributos del sistema, como el rendimiento, la usabilidad, la seguridad y la confiabilidad. A diferencia de las pruebas funcionales, se miden en una escala de medida y se aplican en todos los niveles de prueba (Graham et al., 2019).

Estas pruebas, como las pruebas de rendimiento, pruebas de carga y pruebas de seguridad, buscan evaluar cómo el sistema se comporta en condiciones específicas. A menudo, se requiere medir cuantitativamente el rendimiento y la eficacia del sistema en relación con sus características no funcionales (Graham et al., 2019).

Pruebas de Caja Blanca (White Box).

Se enfocan en analizar la estructura interna y la implementación del sistema o componente. También conocidas como pruebas estructurales, evalúan aspectos como el código, flujos de control, procesos comerciales y flujos de datos. Estas pruebas miden la cobertura de los elementos estructurales y pueden realizarse en varios niveles, incluyendo pruebas de componentes e integración. La cobertura del código es un aspecto importante, y herramientas de medición de cobertura se utilizan para evaluar cuántos elementos ejecutables han sido cubiertos por las pruebas (Graham et al., 2019).

Pruebas Relacionadas con Cambios.

Las pruebas relacionadas con cambios evalúan los efectos de modificaciones en el software, tanto en su funcionamiento como en su estructura. La prueba de confirmación, o reprobación, se realiza después de corregir un defecto y verifica si la solución fue exitosa. Las pruebas de regresión, en cambio, evalúan si las modificaciones introdujeron nuevos defectos en otras partes del sistema. Un conjunto de pruebas de regresión, diseñado para probar funciones clave, se ejecuta en diferentes niveles para asegurar que los cambios no causen efectos secundarios no deseados y que el software siga cumpliendo con sus requisitos (Graham et al., 2019).

Relación entre niveles y tipos de pruebas

Al hablar sobre los distintos tipos de pruebas, es esencial comprender que cada tipo de prueba es aplicable en cada nivel. Las pruebas serán diferentes dependiendo el nivel en el que se lo aplique (Graham et al., 2019).

Por ejemplo, en la Tabla 5, se presenta una aplicación financiera con diversas funcionalidades, algunas perceptibles para los usuarios y otras que operan detrás. Esta tabla ilustra mediante ejemplos cómo los elementos técnicos y detallados deben evaluarse en los niveles inferiores, mientras que las características orientadas al cliente deben ser tratadas en niveles superiores.

Tabla 5

Niveles de pruebas por cada tipo.

	Pruebas de componentes	Pruebas de integración	Pruebas de sistema	Pruebas de aceptación
Pruebas Funcionales	Cómo el componente debe calcular el interés compuesto.	Cómo se pasa la información de la interfaz de usuario a la lógica empresarial.	Cómo los titulares de cuentas pueden solicitar una línea de crédito.	Cómo maneja el banquero las aprobaciones o rechazos de créditos.
Pruebas no Funcionales	El tiempo o número de ciclos de para realizar un cálculo de interés complejo.	Comprobación de desbordamiento en el búfer de los datos pasados desde la interfaz a la lógica.	Pruebas de sí la capa de presentación funciona en navegadores y smartphones.	Pruebas de confiabilidad para evaluar la robustez del microservicio de calificación crediticia.
Pruebas de Caja Blanca	Logre una cobertura de declaraciones y decisiones del 100% para todos los componentes que realizan cálculos.	Cobertura de cómo cada pantalla en la interfaz pasa datos a la siguiente pantalla y a la lógica empresarial.	Cobertura de secuencias de páginas que pueden ocurrir durante la solicitud de una línea de crédito.	Cobertura de las estructuras de archivos admitidas y rangos de valores para transferencias de banco a banco.
Pruebas Relacionadas con Cambios	Las pruebas de regresión automatizadas para cada componente.	Las pruebas de confirmación de la interfaz se activan cuando se registran las correcciones.	Las pruebas de un flujo de trabajo se vuelven a ejecutar si alguna pantalla cambia.	Las pruebas fallidas se vuelven a ejecutar después de que se solucionen los defectos encontrados.

Nota: Elaborado con el contenido de (Graham et al., 2019).

1.3.5. Pruebas por su ejecución.

Pruebas Manuales.

Este es un enfoque de pruebas tradicional los evaluadores diseñan y siguen un plan de prueba, el cual proporciona una guía a lo largo de diversas etapas. Sin embargo, estas pruebas se enfrentan a desafíos notables, como la limitación en la reutilización de recursos y la ausencia de una función de secuencias de comandos. Esta carencia requiere un esfuerzo humano significativo y, en ocasiones, conduce a la omisión de errores (Abbas et al., 2017).

Estas pruebas abarcan diferentes fases, que van desde pruebas unitarias hasta pruebas de aceptación del usuario. En cada una de estas fases, los evaluadores asumen el rol del usuario final para identificar comportamientos inesperados o errores. Asimismo, las pruebas manuales incorporan un elemento de exploración, donde el evaluador escanea el software en busca de posibles fallos (Bhargava & Jain, 2018).

Pruebas Automáticas.

Las pruebas automatizadas se llevan a cabo a través de la creación de secuencias de comandos por parte de los evaluadores, los cuales se ejecutan en conjunto con otro software de evaluación. Este proceso implica la conversión de actividades manuales en acciones automatizadas y se emplean para repetir planes de evaluación, destacándose por su agilidad y repetibilidad (Bhargava & Jain, 2018).

Este enfoque amplía el alcance de las pruebas, aumenta su precisión y conduce a ahorros significativos de tiempo y recursos financieros. En esencia, las pruebas automatizadas representan un método más eficiente y efectivo para garantizar la calidad y el rendimiento de los productos de software (Bhargava & Jain, 2018). Con estas pruebas el evaluador tiene la capacidad de diseñar escenarios de prueba reutilizables y repetibles, con la flexibilidad de ejecutarlos con la frecuencia necesaria (Gamido & Gamido, 2019).

1.4. Pruebas de software con metodologías ágiles.

1.4.1. Introducción a las metodologías ágiles.

Las metodologías ágiles son enfoques de desarrollo de software que ponen énfasis en la entrega de valor al cliente de forma incremental e iterativa, en lugar de intentar lanzar un producto completo de una sola vez (Instituto Agile, 2021). Los métodos ágiles promueven el trabajo en equipo y la planificación adaptable en lugar de seguir un plan predeterminado. El objetivo es desarrollar software de calidad que satisfaga las necesidades cambiantes del cliente (Najihi et al., 2022).

A diferencia del enfoque tradicional, los métodos ágiles se orientan a sistemas complejos y dinámicos, donde no existen expectativas precisas ni información detallada sobre los desafíos tecnológicos al inicio del proyecto. Por ello, evitan las especificaciones exhaustivas y busca procesos adaptativos, iterativos y evolutivos (Brühl, 2022).

El enfoque ágil está fundamentado en los cuatro valores con doce principios que establece el manifiesto ágil. “Los valores son consideraciones generales en las que se debe enmarcar el desarrollo de software. Los principios se refieren a las características que diferencian entre un proceso ágil y uno tradicional y constituyen las ideas centrales del desarrollo ágil” (Quiña-Mera et al., 2021). Estos principios son presentados en la Figura 7.

Figura 7

Principios del manifiesto ágil



Nota: Obtenido de (Instituto Agile, 2021).

SCRUM

Es un marco rápido, iterativo e incremental para la mejora de la administración, el mantenimiento y el desarrollo de software basado en el manifiesto ágil (Guerrero et al., 2022), facilita la inspección rápida y repetida del software en funcionamiento, generalmente cada dos semanas o un mes. En este proceso, la empresa establece las prioridades, mientras que los equipos tienen la autonomía de autogestionarse (Jeya Mala & Pradeep Reynold, 2020).

Su ciclo de vida se divide en 3 fases:

- Pre juego (definición del sistema y actualización de requisitos): visión, historial de usuario, cartera de productos (Guerrero et al., 2022).
- Desarrollo (iteraciones y aumento de producto): Sprint, Reunión de planificación de Sprint, Backlog de Sprint, SCRUM diario, Lista de inconvenientes, Incremento, Reunión de revisión de Sprint y Reunión de retrospectiva (Guerrero et al., 2022).
- Post juego (entrega de producto): Reunión de cierre (Guerrero et al., 2022).

El equipo abarca diversas funciones, como el propietario del producto (product Owner), quien representa los intereses de los usuarios y asegurar la creación del producto apropiado. Asimismo, el Scrum Master, quien funge como entrenador del equipo, facilitando la comprensión del proceso y el equipo de desarrollo el cual es autoorganizado, y toma decisiones sobre la asignación de tareas entre sus miembros (O'Regan, 2019). En la Figura 8 se puede ver una representación del proceso de SCRUM y sus miembros.

1.4.2. Pruebas ágiles.

Las pruebas ágiles adoptan los principios ágiles, priorizando las pruebas iterativas desde la perspectiva del usuario sobre las rigurosas pruebas al final del ciclo, como en los enfoques tradicionales. De esta forma, las pruebas resuelven problemáticas como la adaptación poco flexible de los requisitos ante cambios y agilizan los ciclos de desarrollo (Yu, 2018).

Una diferencia clave es que las pruebas ágiles buscan validar soluciones que pasen las pruebas y confirmen historias de usuario, no invalidar la solución como ocurre en metodologías tradicionales. Para ello, los equipos de desarrollo y pruebas colaboran

estrechamente bajo los principios ágiles, superando la separación tradicional entre programadores y el equipo de pruebas, aunque respetando sus roles (Penmetsa, 2017).

Según Stray et al. (2022), la temprana participación del equipo de pruebas en cada iteración refuerza el enfoque en historias de usuario y arquitectura del sistema, mejorando el alcance de las pruebas. Además, el continuo intercambio de conocimiento reduce malentendidos y mejora el aprendizaje mutuo. Otra clave es la delegación de tareas de pruebas unitarias a los desarrolladores, permitiendo al equipo de pruebas concentrarse en técnicas como pruebas exploratorias y de usabilidad, incrementando la cobertura.

De forma similar a como se realizaba en las metodologías ágiles, las pruebas ágiles también tienen varios principios, mostrados en la Tabla 6.

Tabla 6

Principios de pruebas ágiles

Principio	Descripción
La prueba proporciona retroalimentación	La prueba se utiliza para proporcionar retroalimentación y visibilidad para avanzar en el proyecto.
Pruebas continuas	Las pruebas son una forma de vida en Agile y se realizan con frecuencia durante el sprint.
Pruebas por todo el equipo	Tanto los desarrolladores como los probadores ejecutan pruebas, con todo el equipo involucrándose para eliminar cuellos de botella en las pruebas.
Bucle de retroalimentación corto	Los equipos ágiles prueban temprano y con frecuencia para obtener retroalimentación rápida sobre cómo se comporta el software.
Código limpio	Los desarrolladores solucionan defectos genuinos a medida que se encuentran, manteniendo así el código limpio.
Documentación ligera	Los probadores utilizan listas de verificación reutilizables y herramientas de documentación ligeras.
"Hecho" significa "terminado"	Una característica no está completa hasta que ha sido completamente implementada y probada.
Desarrollo guiado por pruebas	Las pruebas se definen con los requisitos y se utilizan para guiar los esfuerzos de desarrollo.

Nota: Obtenido de (O'Regan, 2019).

1.4.3. Pruebas ágiles vs pruebas tradicionales.

Existen dos enfoques para las pruebas de software: el enfoque tradicional y el enfoque ágil. El enfoque tradicional se basa en procesos secuenciales y documentación exhaustiva, mientras que el enfoque ágil enfatiza la retroalimentación continua, la adaptación al cambio y la colaboración cercana entre roles (Najih et al., 2022).

En las metodologías tradicionales como cascada, las pruebas ocurren al final del ciclo de desarrollo. El equipo de pruebas recibe el software y se enfoca en la búsqueda y resolución de defectos a través de la ejecución de casos de prueba (Najih et al., 2022). La colaboración entre desarrolladores y probadores es mínima y se concentra en revisar documentos, en contraste con la interacción continua característica de las pruebas ágiles (Penmetsa, 2017).

De acuerdo con Najih et al. (2022), en los enfoques ágiles las pruebas no son una fase separada y los probadores están involucrados desde la recolección de requerimientos. Las pruebas son un proceso continuo, integrado en el ciclo incremental de desarrollo. Dado que el desarrollo ocurre en iteraciones cortas (típicamente de 2 a 3 semanas), el equipo de pruebas trabaja en ciclos junto con los desarrolladores y el gerente de producto (Najih et al., 2022).

La colaboración constante de todo el equipo asegura una comprensión mutua y una retroalimentación temprana. No obstante, las pruebas ágiles también plantean desafíos, como problemas de comunicación dentro y entre equipos, y la necesidad de manejar con precaución la implementación temprana. Además, lograr la adaptación de prácticas ágiles en entornos más amplios puede requerir esfuerzos significativos (Penmetsa, 2017).

Las diferencias entre estas metodologías de pruebas se pueden explicar de mejor forma en la Tabla 7, la cual muestra el comportamiento de ambos enfoques según ciertos factores.

Tabla 7

Comparación entre pruebas ágiles y tradicionales

Factor	Ágil	Tradicional
Equipo	Los probadores están integrados en el equipo de desarrollo.	Equipos de prueba separados y los probadores son muy hábiles en las pruebas.
Proceso	La preparación de las pruebas comienza con historias de usuario y las pruebas continúan con el desarrollo. Se enfoca en probar primero las áreas de alto valor.	Enfocado en la preparación de pruebas, ejecución y documentación. Los equipos de prueba están involucrados después de que el desarrollo está completo.
Comunicación	Comunicación continua con los interesados.	Comunicación infrecuente con el cliente, equipos de desarrollo e interesados.
Acomodación de Cambios	Adaptable a cambios. Los cambios se esperan como parte del proceso y causan retrasos mínimos.	Trabaja con un alcance predefinido, y los cambios en el alcance suelen causar retrasos en todo el calendario.
Planificación	Una planificación aproximada es suficiente.	Necesita una planificación muy formal.
Costo	Menos costos y costo de cambio mínimo.	Más costos y alto costo de cambio.
Fases de prueba	Depende del tamaño y tipo de los proyectos.	Múltiples fases de prueba.
Desarrollo	El enfoque está en la implementación de soluciones de extremo a extremo, incluyendo pruebas.	Las soluciones empresariales se dividen en múltiples piezas: presentación, persistencia, etc.
Habilidades del probador	Tienen conocimiento de pruebas y desarrollo, son receptivos a los cambios.	Experimentados y hábiles en pruebas, procesos de prueba.
Pruebas automatizadas	Fundamental	Opcional
Pruebas	Cada iteración (Pruebas continuas)	Después de completar la codificación.
Característica	Enfoque ágil	Enfoque tradicional
Requisitos	Se requiere aportación a lo largo del proceso de desarrollo.	Claramente definidos y revisados desde la fase inicial. La tasa de cambio es baja.

Nota: Obtenido de (Najih et al., 2022).

1.4.4. Plan de pruebas para enfoques ágiles.

Plan de pruebas.

La planificación de pruebas conlleva la delimitación del alcance de las pruebas a ejecutar; la especificación del entorno de prueba; la evaluación de la carga de trabajo necesaria para elaborar los escenarios y llevar a cabo las evaluaciones; la identificación de los recursos requeridos (personal, equipo, software y utilidades); la asignación de dichos

recursos; el establecimiento de la programación temporal; y la detección y gestión de posibles riesgos asociados a las pruebas (O'Regan, 2019).

Plan de pruebas en metodologías ágiles y tradicionales

Antes de abordar la planificación de pruebas en un entorno ágil, resulta valioso revisar de manera concisa los métodos de planificación de pruebas en entornos de proyectos tradicionales. En el desarrollo de software con metodologías tradicionales, la planificación de pruebas tiende a concebirse como una actividad independiente (Baumgartner et al., 2021).

En este enfoque se designa un responsable de gestión de pruebas que se encarga de iniciar la planificación de estas actividades. Dicha planificación conlleva una minuciosa inspección de documentos, por tanto, el gestor de pruebas tiende a comenzar buscando:

- Diagramas de arquitectura
- Documentos y especificaciones de requisitos
- Especificaciones sobre funcionalidad, servicios y máscaras
- Prototipos de GUI
- Guías de estilo (Baumgartner et al., 2021).

En los proyectos ágiles no existen tantas restricciones. En las pruebas ágiles se trabaja estrechamente con el equipo y si los programadores trabajan primero con las pruebas unitarias, se garantiza la revisión de todas las pequeñas partes. Luego, es posible colaborar con el cliente para definir las pruebas de aceptación. Se prueba cada historia de usuario a medida que el programador trabaja en ella (Crispin & Gregory, 2008).

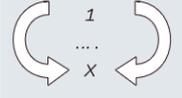
Estrategia de pruebas en entornos ágiles.

La transición de una metodología de pruebas tradicional a enfoques ágiles puede presentar desafíos, especialmente al diseñar el plan de pruebas y gestionar un proceso ágil. "A menudo se hace esta pregunta: ¿Pueden los modelos y procesos de calidad tradicionales coexistir con métodos de desarrollo ágiles? En teoría, no hay ninguna razón por la que no puedan hacerlo. En realidad, muchas veces no hay elección" (Crispin & Gregory, 2008)

Crispin & Gregory (2009) en su libro Agile Testing presentan una estrategia que se basa en la documentación mínima necesaria para satisfacer los requisitos de CMMI. Esta estrategia incluye un proceso destinado completamente a la documentación. Además, representa esta estrategia mediante diagramas como se puede ver en la Figura 8.

Figura 8

Estrategia para pruebas ágiles

<p>Iniciación del proyecto</p>	<p>Obtener una comprensión del proyecto.</p>
<p>Planificación de lanzamiento</p>	<p>Participar en la estimación de historias Crear planes de prueba</p>
<p>Cada iteración</p> 	<p>Participar en la planificación del sprint, estimando tareas Escribir y ejecutar pruebas de historias Realizar pruebas en pareja con otros probadores, desarrolladores Validación empresarial (clientes) Automatizar nuevos casos de prueba funcionales. Ejecutar casos de prueba de regresión automatizados Ejecutar pruebas de carga del proyecto</p>
<p>El juego Final (Prueba del sistema)</p>	<p>Gestión de lanzamientos prueba despliegue simulado en preproducción Prueba de humo en preproducción Realizar prueba de carga (si es necesario) Completar prueba de regresión Probadores de negocio realizan UAT (Prueba de Aceptación del Usuario) Participar en la preparación para el lanzamiento.</p>
<p>Lanzamiento a Producción/Soporte</p>	<p>Participar en el lanzamiento a producción Participar en retrospectivas</p>

Nota: Obtenido de (Crispin & Gregory, 2008).

Plan de pruebas ligero.

Existen alternativas a considerar en lugar de los tradicionales y detallados planes de prueba, que a veces pueden resultar complejos. El plan de pruebas debe beneficiar al equipo y al mismo tiempo satisfacer las expectativas del cliente e igual que otro documento técnico, deben tener un propósito claramente definido (Crispin & Gregory, 2008).

Los denominados "planes de prueba simplificados" son una opción cuando se requiere un plan formal, por ejemplo, para cumplir con requisitos regulatorios. El enfoque del plan pruebas debe ser ágil y ajustado a las necesidades reales, sin duplicar información (Crispin & Gregory, 2008). Un ejemplo de plan de pruebas podría ser el presentado en la Figura 9.

Figura 9

Ejemplo de plan de pruebas

Tester		% Comprometido
Janet		100%
Lisa		50%

Descripción de la Función	Profundidad de la Prueba
Agregar un nuevo interruptor para la selección de idioma en la página principal	Prueba de los 5 idiomas (Inglés, Español, Francés, Italiano y Alemán). Prueba de que somos capaces de cambiar de idioma dinámicamente.

#	Riesgo	Impacto	Plan de mitigación
1	Los usuarios no están listos para el UAT	Alto	

Nota: Obtenido de (Crispin & Gregory, 2008).

1.5. Norma ISO/IEC 25023.

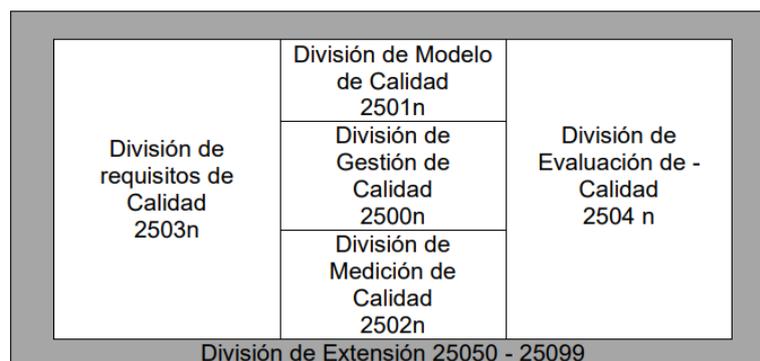
Las normas ISO/IEC 25010 y 25023, abordadas en esta sección, pertenecen a la familia de normas ISO/IEC 25000 SQuaRE (System and Software Quality Requirements and Evaluation).

El conjunto de normas SQuaRE busca guiar a aquellos que desarrollan y adquieren sistemas y productos de software en la definición y evaluación de los requisitos de calidad, estas normas brindan directrices para especificar los requisitos de calidad, así como para su medición y evaluación (ISO/IEC 25000, 2014).

La Figura 10 muestra la estructura de las series SQuaRE, que se dividen en cinco familias de normas, también referidas como divisiones.

Figura 10

Organización de las series de Normas SQuaRE



Nota: Obtenido de (NTE INEN-ISO/IEC 25010, 2015).

1.5.1. Descripción y alcance.

ISO/IEC 25010.

Esta norma se establece como un Modelo de Calidad. Su objetivo es describir el marco de calidad relacionado con el producto de software, tanto en sus aspectos internos, externos, y su calidad en uso. En relación con normativas previas, como ISO/IEC 9126-1 y 14598-1, esta norma especifica las características y subcaracterísticas que definen la calidad de los productos de software desde diversas perspectivas (ISO/IEC 25000, 2014).

Más concretamente, la ISO/IEC 25010 propone un modelo estructurado en ocho características. Estas características se subdividen a su vez en subcaracterísticas, proporcionando un panorama que engloba tanto las propiedades estáticas del software como las dinámicas del sistema informático (NTE INEN-ISO/IEC 25010, 2015).

Para concluir, es fundamental destacar que esta norma no opera de manera aislada, sino que está concebida para ser utilizada en coordinación con otras partes de la serie SQuaRE. Además, debe considerarse en relación con la ISO/IEC 14598 hasta que esta última sea reemplazada por las futuras normas ISO/IEC 2504n (NTE INEN-ISO/IEC 25010, 2015).

ISO/IEC 25023.

Está centrada en la medición de la calidad de productos de software y sistemas, establece medidas cuantitativas alineadas con las características y subcaracterísticas delineadas en ISO/IEC 25010 (ISO/IEC 25000, 2014). Aunque mantiene una estrecha relación con la ISO/IEC 25010, esta normativa también se integra con otras de la serie SQuaRE, incluyendo las ISO/IEC 25030, 25040 y 25041, conformando un marco integral para la especificación, medición y evaluación de la calidad del software y sistemas (ISO/IEC 25023, 2016).

Las medidas de calidad descritas por la ISO/IEC 25023 se organizan según su confiabilidad y, a pesar de proporcionar un marco robusto, se reconoce que no son absolutamente exhaustivas. De esta forma, se anima a los usuarios a adaptarlas según sus necesidades específicas. Además, es relevante indicar que la norma no asigna valores predeterminados a las medidas, sino que estos son establecidos considerando diversos factores, como la naturaleza del software y las demandas del usuario (ISO/IEC 25023, 2016).

Con la finalidad primordial de garantizar y potenciar la calidad de los productos de software y sistemas durante su evolución, la ISO/IEC 25023 está destinada principalmente a profesionales involucrados en la especificación y evaluación de calidad, sirviendo como referencia esencial para dichos especialistas (ISO/IEC 25023, 2016).

1.5.2. Modelo de calidad del producto de software ISO/IEC 25010.

El modelo de calidad actúa como elemento central en el sistema de evaluación de la calidad del producto software, en él se definen las características esenciales para valorar las propiedades del software (NTE INEN-ISO/IEC 25010, 2015).

El modelo de calidad del producto definido se encuentra compuesto por ocho características (adecuación funcional, confianza, eficiencia de rendimiento, usabilidad, seguridad, compatibilidad, capacidad de mantenimiento y portabilidad) y sus respectivas subcaracterísticas (NTE INEN-ISO/IEC 25010, 2015). Se muestran en la Figura 11.

Figura 11

Modelo de calidad del producto de software



Nota: Obtenido de (NTE INEN-ISO/IEC 25010, 2015).

El modelo de calidad de producto puede ser aplicado no solo al software, sino también a sistemas informáticos que incluyen software. Esto se debe a que muchas de las subcaracterísticas definidas en el modelo son relevantes tanto para evaluar la calidad del software como la de sistemas completos (NTE INEN-ISO/IEC 25010, 2015).

1.4.3. Característica de Adecuación funcional.

Se refiere al “grado en el cual un producto o sistema provee funciones que satisfacen las necesidades establecidas e implícitas cuando se las usa bajo condiciones específicas” (NTE INEN-ISO/IEC 25010, 2015). Sus tres subcaracterísticas son mostradas en la Tabla 8.

Tabla 8

Subcaracterísticas de Adecuación funcional

Subcaracterística	Definición
Compleitud funcional	Grado en el que el conjunto de funciones cubre todas las tareas especificadas y los objetivos del usuario.
Corrección funcional	Grado en el que un producto o sistema proporciona los resultados correctos con el grado de precisión necesario
Pertinencia funcional	Grado en el que las funciones facilitan el logro de tareas y objetivos específicos.

Nota: Obtenido de (Uyaguari et al., 2023).

Como se mencionó con anterioridad para medir las subcaracterísticas se usa las métricas definidas por la norma ISO/IEC 25023 las cuales nos presentan fórmulas matemáticas. Las métricas pertenecientes a la característica de “adecuación funcional” son las mostradas en la Tabla 9. El valor deseado al aplicar la fórmula es el más cercano a 1.

Tabla 9

Métricas de calidad para Adecuación funcional

Subcaracterística	Métrica	Propósito de la métrica	Método de aplicación	Fórmula
Compleitud funcional	Cobertura funcional	¿Cuán completa es la implementación de acuerdo con la especificación de requerimientos?	Contar el número de las funciones indicadas en la especificación de requerimientos y el número de funciones faltantes o incorrectas	$X = 1 - A/B$ A = Número de funciones perdidas B = Número de funciones especificadas
Corrección funcional	Corrección funcional	¿Cuál es el grado en el que las funciones proporcionadas producen resultados correctos?	Contar el número de funciones implementadas y el número de funciones que son incorrectas	$X = 1 - A/B$ A = Número de funciones incorrectas B = Número de funciones consideradas
Pertinencia funcional	Pertinencia funcional del objetivo de uso	¿Hasta qué punto el software cumple con su propósito y necesidades previstas?	Comparar las funciones proporcionadas por el software con las funciones requeridas para cumplir un objetivo de uso específico.	$X = A/B$ A = Número de funciones proporcionadas que cumplen con el objetivo de uso B = Número de funciones requeridas para el objetivo de uso

Pertinencia funcional del sistema	¿El software ofrece funciones relevantes y adecuadas para satisfacer los objetivos específicos del usuario?	Contar las funciones que el sistema proporciona con el conjunto de funciones que se esperaba o requería que el sistema proporcionara.	$X = A/B$ A = Número de funciones proporcionadas que son pertinentes y adecuadas B = Número total de funciones esperadas o requeridas
-----------------------------------	---	---	---

Nota: Adaptado de (Balseca, 2014; Perdomo Charry, 2019).

CAPÍTULO 2

Desarrollo

Este capítulo aborda el proceso llevado a cabo para desarrollar un software que automatice los procesos de control de materia prima y recepción de pedidos en la Fábrica de Medias Neltex, en adelante denominado: Gestor Neltex y el cual está dividido en varias etapas.

Primero se lleva a cabo un análisis de los procesos y necesidades para la aplicación web; seguido de la planificación del proyecto, donde se definen roles, historias de usuario, características prioritarias y un plan de pruebas general.

El desarrollo del software se realiza utilizando la metodología ágil Scrum de manera iterativa e incremental. Esta fase incluye la codificación y la ejecución de pruebas unitarias y de aceptación, asegurando la calidad del producto en cada etapa.

Finalmente, el software resultante es implementado en un entorno de producción, acompañado de un análisis del porcentaje de cobertura obtenido de las pruebas unitarias.

Los apartados siguientes profundizan en cada una de estas etapas, proporcionando una visión detallada del proceso de obtención del software.

2.1. Descripción de los procesos a ser automatizados

Los procesos de negocio por automatizar en el proyecto ya se encontraban documentados, sin embargo, estaban desactualizados, por lo que se realizó modificaciones para facilitar la automatización. A continuación, se detalla cada uno de ellos.

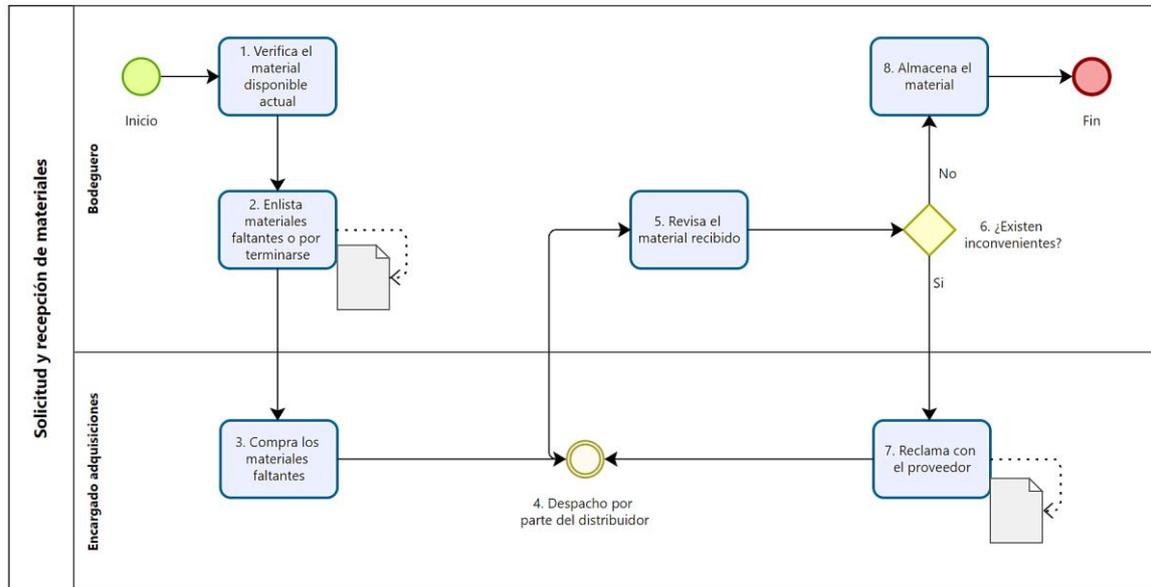
2.1.1. Solicitud y recepción de materiales

Diagrama del proceso.

El proceso presentado en la Figura 12 se enfoca en la gestión de materiales para asegurar la disponibilidad adecuada de insumos necesarios en la producción. Comprende una serie de pasos, desde la verificación de existencias en el almacén hasta la adquisición y almacenamiento de los materiales.

Figura 12

Proceso de solicitud y recepción de materiales



Nota: Elaboración propia.

Descripción del proceso.

La Tabla 10 describe las actividades involucradas durante el proceso.

Tabla 10

Detalle del proceso de solicitud y recepción de materiales

Nro.	Actividad	Descripción	Responsable
1	Verifica el material disponible actual	Busca en la bodega los materiales disponibles para producción y sus cantidades.	Bodeguero
2	Enlista materiales faltantes o por terminarse	Crea una lista con los materiales que no se disponen o que están prontos a terminarse.	Bodeguero
3	Compra los materiales faltantes	Realiza el pedido al proveedor, dependiendo del material y los precios ofrecidos.	Encargado de adquisiciones
4	Despacho por parte del distribuidor	Se espera a que el proveedor despache el material solicitado	Encargado de adquisiciones
5	Revisa el material recibido	Hace una revisión del material que se recibió comparándolo con la compra realizada	Bodeguero
6	¿Existen inconvenientes?	Si el material recibido presenta inconvenientes, va a la actividad 7. Caso contrario va a la actividad 8	Bodeguero
7	Reclama con el proveedor.	Se envía un reclamo al proveedor en el cual se indique el inconveniente y detalle el material	Encargado de adquisiciones
8	Almacena el material	Se almacena el material que se ha recibido	Bodeguero
9	Fin		

Nota: Elaboración propia.

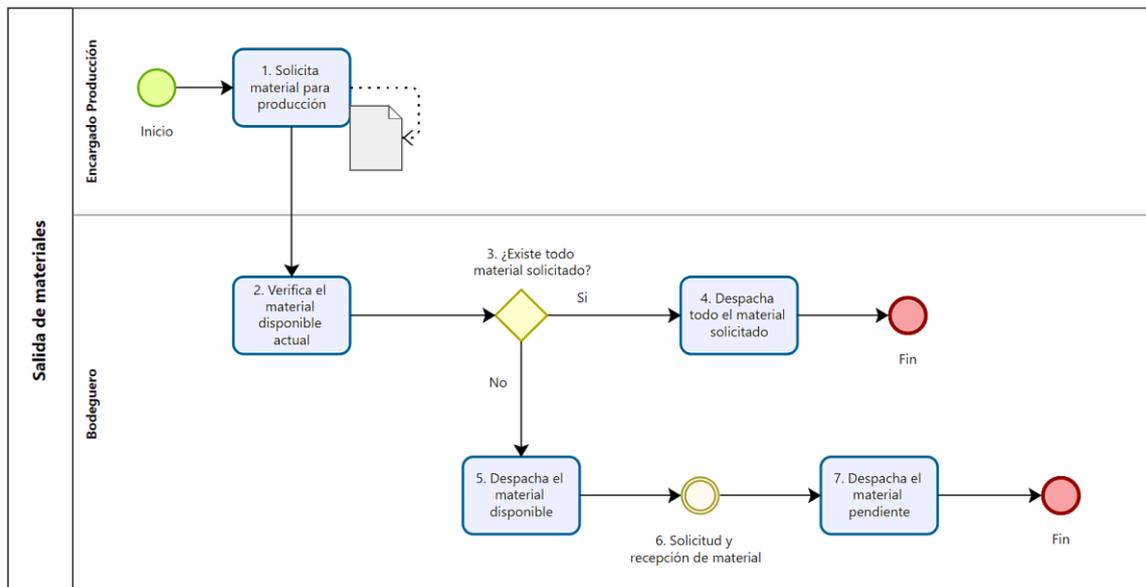
2.1.2. Salida de Materiales

Diagrama del proceso.

El proceso presentado en la Figura 13 es la distribución eficiente de los materiales necesarios para las órdenes de producción. Desde la solicitud inicial hasta la verificación y despacho, se busca asegurar un suministro constante para las necesidades de producción.

Figura 13

Proceso de salida de materiales



Nota: Elaboración propia.

Descripción del proceso.

La Tabla 11 describe las actividades involucradas en el proceso antes mostrado.

Tabla 11

Detalle del proceso de salida de materiales

Nro.	Actividad	Descripción	Responsable
1	Solicita material para producción	Pide el material necesario para una orden de producción	Encargado de producción
2	Verifica el material disponible actual	Busca en la bodega por los materiales disponibles para el pedido.	Bodeguero
3	¿Existe todo material solicitado?	Verifica si la cantidad de material disponible cubre lo necesario para el pedido. Sí cubre lo necesario ir a la actividad 4. Caso contrario ir a la actividad 5	Bodeguero

4	Despacha todo el material solicitado	Envía al departamento de producción el material indicado en la orden de producción.	Bodeguero
5	Despacha el material disponible	Despacha el material con el que se cuenta actualmente y registrar la cantidad entregada	Bodeguero
6	Solicitud y recepción de material	Empieza con el proceso de solicitud de material para el material faltante y para el que está pronto a terminarse.	Bodeguero
7	Despacha el material pendiente	Envía al departamento de producción el material faltante para las órdenes de producción pendientes	Bodeguero
8	Fin		

Nota: Elaboración propia.

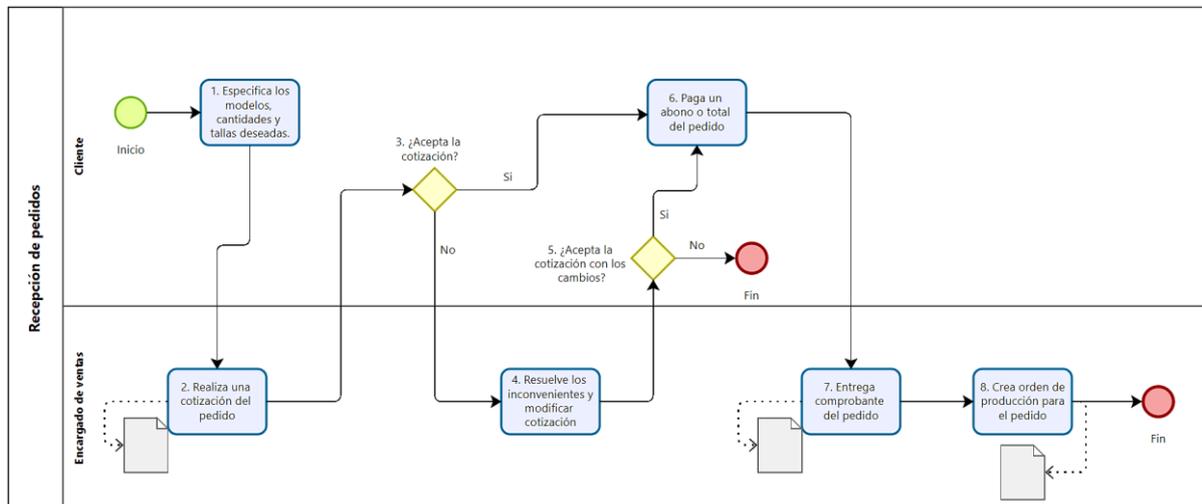
2.1.3. Recepción de pedidos

Diagrama del proceso.

El proceso presentado en la Figura 14 es la realización de pedidos de productos, desde la especificación inicial de modelos, cantidades y tallas por parte del cliente hasta la entrega de un comprobante detallado del pedido.

Figura 14

Proceso de recepción de pedidos



Nota: Elaboración propia.

Descripción del proceso

La Tabla 12 describe las actividades involucradas en el proceso antes mostrado.

Tabla 12

Detalle del proceso de recepción de pedidos

Nro.	Actividad	Descripción	Responsable
1	Especifica los modelos, cantidades, tallas deseadas.	Realiza el pedido de los productos deseados, brindando los detalles para su producción	Cliente
2	Realiza una cotización del pedido	Da el costo del pedido realizado por el cliente, detallando cada uno de los precios	Encargado de ventas
3	¿Acepta la cotización?	Evalúa el precio y tomar la decisión si aceptar o rechazar la cotización. En caso de rechazarla ir al paso 4. En caso de aceptarla ir al paso 6.	Cliente
4	Resolver los inconvenientes y modificar cotización	Evalúa cuáles fueron los inconvenientes que hicieron rechazar la cotización y solucionarlos en una nueva cotización.	Encargado de ventas
5	¿Acepta la cotización con los cambios?	Evalúa la nueva cotización. En caso de aceptarla, ir al paso 5. En caso de rechazarla, finalizar el pedido	Cliente
6	Paga un abono o total del pedido	Paga una parte del pedido mínima obligatoria o realizar el pago del total	Cliente
7	Entrega comprobante del pedido	Entrega un comprobante del pedido realizado donde se especifique el pago realizado, la fecha de entrega y los productos adquiridos.	Encargado de ventas
8	Crea orden de producción para el pedido	Crea una orden con los detalles necesarios para que pase a producción.	Encargado de ventas
9	Fin		

Nota: Elaboración propia.

2.2. Preparación y planificación del proyecto

La preparación y planificación del proyecto o también llamado “Sprint 0” constituye una etapa fundamental, ya que permite sentar las bases y estructurar adecuadamente el trabajo por realizar. En esta sección se abordan diversas actividades, necesarias para la construcción del software.

2.2.1. Definición de roles y responsabilidades.

En la Tabla 13 se define a cada uno de los involucrados en el proyecto, así como sus roles y responsabilidades en el marco de trabajo Scrum.

Tabla 13

Roles y responsabilidades del equipo

Rol	Nombre	Responsabilidad
Product Owner	Sr. Nelson Castañeda	Gerente de la Fábrica de medias Neltex. Encargado de proveer los requerimientos, verificar las funcionalidades y aceptar o rechazar los incrementos.
Scrum Master	Ph.D. Cathy Guevara Vega	Encargada del seguimiento de las diferentes actividades del proyecto y uso apropiado de la metodología SCRUM.
Equipo de desarrollo	Sr. Kevin Lliguisupa	Responsable de llevar a cabo el trabajo para la entrega de los incrementos del producto

Nota: Elaboración propia.**2.2.2. Historias de usuario.**

Constituyen una técnica comúnmente empleada en metodologías ágiles para definir los requerimientos de la aplicación desde la perspectiva del usuario y en las cuales no se usa un lenguaje técnico.

En el presente proyecto, las historias de usuario fueron levantadas mediante entrevistas con el gerente, quien desempeñó el papel de propietario del software (product Owner), junto con algunos empleados clave de la Fábrica. Esta estrategia posibilitó una comprensión más exhaustiva de las necesidades y expectativas, contribuyendo así a obtener historias de usuario más precisas y detalladas. Las historias de usuario obtenidas son presentadas en las tablas a continuación,

Tabla 14

Historia de Usuario Nro. 1

User Story			
Código:	HU-1	Usuario:	Gerente
Nombre:	Login con roles de usuario.		
Prioridad:	Alta	Dependencia:	N/A
Descripción:	Como gerente, deseo asignar roles a los empleados para controlar su acceso y permitirles realizar acciones adecuadas.		
Criterios de aceptación:	<ul style="list-style-type: none"> • Debe permitir el acceso a la aplicación solamente cuando las credenciales estén correctas y el usuario este activo. 		

- Debo poder ver una lista de todos los empleados registrados en la aplicación.
- Para cada empleado, debo poder asignar, eliminar o modificar su rol.
- Los roles disponibles deben incluir, al menos "Gerente", "Adquisición", "Vendedor" y "Bodeguero" con sus respectivos permisos.

Tabla 15

Historia de Usuario Nro. 2

User Story			
Código:	HU-2	Usuario:	Bodeguero
Nombre:	Gestión de materia prima.		
Prioridad:	Alta	Dependencia:	N/A
Descripción:			
Como bodeguero, quiero poder crear, visualizar, modificar y eliminar registros de materiales necesarios para la fabricación.			
Criterios de aceptación:			
<ul style="list-style-type: none"> • Debo poder agregar una nueva materia prima al inventario, especificando su nombre, cantidad inicial en stock, unidad de medida, proveedor, etc. • Debo poder ver una lista de todas las materias primas existentes en el inventario, con su información. • Debo poder editar la información de una materia prima existente. • Debo poder cambiar el estado de una materia prima a "Activa" o "Inactiva". 			

Tabla 16

Historia de Usuario Nro. 3

User Story			
Código:	HU-3	Usuario:	Bodeguero
Nombre:	Creación de órdenes de compra.		
Prioridad:	Baja	Dependencia:	HU-2
Descripción:			
Como bodeguero, quiero poder crear y visualizar órdenes de compra para adquirir materiales faltantes en la bodega.			
Criterios de aceptación:			
<ul style="list-style-type: none"> • Debo poder ver una lista de todas las órdenes de compra existentes, con detalles como número de orden, fecha de creación y estado (por ejemplo, pendiente, aprobada, entregada). • Debo poder crear una nueva orden de compra, especificando los materiales necesarios, las cantidades, los proveedores potenciales, etc. 			

Tabla 17

Historia de Usuario Nro. 4

User Story			
Código:	HU-4	Usuario:	Bodeguero
Nombre:	Registro de movimiento de materia prima.		
Prioridad:	Media	Dependencia:	HU-5, HU-8
Descripción:			
Como bodeguero, deseo registrar el ingreso y salida de materia prima, permitiendo crear salidas relacionadas con órdenes de producción y entradas con recepciones de productos comprados. Además, poder realizar modificaciones manuales en el stock de material.			
Criterios de aceptación:			
<ul style="list-style-type: none"> • Debo poder crear una salida de material relacionada con una orden de producción, especificando los materiales utilizados, las cantidades y la fecha de salida. • Debo poder crear una entrada de material relacionada con la recepción de productos comprados, especificando los materiales recibidos, las cantidades y la fecha. • Debo poder realizar modificaciones manuales en el stock de materiales cuando sea necesario, como ajustes por daños, pérdidas o cambios en las cantidades disponibles. • Debo poder visualizar un registro de todas las entradas y salidas de materiales, incluyendo detalles como fecha, tipo de transacción y materiales involucrados. 			

Tabla 18

Historia de Usuario Nro. 5

User Story			
Código:	HU-5	Usuario:	Adquisición
Nombre:	Registro de compras realizadas.		
Prioridad:	Baja	Dependencia:	HU-3
Descripción:			
Como encargado de compras, quiero poder registrar las compras realizadas a partir de las órdenes de compra recibidas por bodega.			
Criterios de aceptación:			
<ul style="list-style-type: none"> • Debo poder ver una lista de todas las órdenes de compra que han sido creadas por bodega y que están pendientes. • Debo poder seleccionar una orden de compra y registrar la compra correspondiente, especificando los detalles de la transacción, como la fecha de compra, el proveedor y los materiales o productos adquiridos. • La aplicación debe mantener un registro de todas las compras realizadas. 			

Tabla 19

Historia de Usuario Nro. 6

User Story			
Código:	HU-6	Usuario:	Vendedor, Gerente
Nombre:	Gestión de productos para producción.		
Prioridad:	Alta	Dependencia:	HU-2
Descripción:			
Como vendedor o gerente, quiero poder gestionar los productos disponibles para producir. Además, deseo que estos productos contengan información sobre los materiales necesarios, sus cantidades y precios.			
Criterios de aceptación:			
<ul style="list-style-type: none"> • Debo poder crear un nuevo producto, especificando su nombre, precio, estado, peso por docena y asociando los materiales necesarios, sus cantidades y precios unitarios. • Debo poder ver una lista de todos los productos, con su información incluyendo la lista de materiales necesarios, cantidades requeridas y precios unitarios. • Debo poder editar la información de un producto existente. • Debo poder cambiar el estado de un producto a "Disponible" o "No disponible". • La aplicación debe realizar cálculos automáticos para mostrar el costo total de los materiales utilizados en la producción de un producto. 			

Tabla 20

Historia de Usuario Nro. 7

User Story			
Código:	HU-7	Usuario:	Vendedor
Nombre:	Recepción de pedidos para producción.		
Prioridad:	Media	Dependencia:	HU-6
Descripción:			
Como vendedor, quiero poder recibir pedidos de producción, registrar la información del cliente y los detalles del pedido, que incluyen productos, cantidades, colores y otros detalles relevantes. Además, deseo que se genere un comprobante del pedido.			
Criterios de aceptación:			
<ul style="list-style-type: none"> • Debo poder recibir un nuevo pedido de un cliente, proporcionando información del cliente, como nombre, dirección y detalles de contacto. • Debo poder agregar productos al pedido, especificando el nombre del producto, cantidad, color, y otros detalles relevantes. • La aplicación debe calcular automáticamente el total del pedido en función de los productos y sus cantidades. • Debo poder guardar el pedido y generar un comprobante del pedido que incluya los detalles del cliente, los productos solicitados, el total y la fecha del pedido. • La aplicación debe permitir la visualización de los pedidos recibidos. 			

Tabla 21

Historia de Usuario Nro. 8

User Story			
Código:	HU-8	Usuario:	Vendedor
Nombre:	Generar orden de producción.		
Prioridad:	Baja	Dependencia:	HU-6
Descripción:	Como vendedor, quiero que una vez que cree un pedido automáticamente me genere una orden de producción, que pasara al departamento de producción.		
Criterios de aceptación:	<ul style="list-style-type: none"> • Una vez que el pedido sea creado, la aplicación debe generar automáticamente una orden de producción asociada. • La orden de producción debe incluir detalles como los productos solicitados, las cantidades, colores y otros detalles relevantes, de acuerdo con el pedido. • La orden de producción debe ser visible para el departamento de producción. • La orden de producción solo tendrá la información necesaria, evitando los precios. 		

2.2.3. Priorización y organización del Product Backlog.

El backlog de producto, mostrado en la tabla 22, ordena las historias de usuario según su relevancia para el avance general del proyecto, estimando el tiempo requerido para su implementación. Estas historias de usuario están categorizadas según su prioridad, en alta, media y baja.

Tabla 22

Product Backlog

Código HU	Historias de Usuario	Prioridad	Estimación (Horas)	Descripción
HU-1	Login con roles de usuario	Alta	75	Como administrador, asigno roles, controlo el acceso y acciones de empleados.
HU-2	Gestión de materia prima	Alta	45	Como bodeguero, creo, veo, modifico y elimino registros de materiales.
HU-3	Creación de órdenes de compra	Baja	40	Como bodeguero, creo y veo órdenes de compra para adquirir materiales.
HU-4	Registro de movimiento de materia prima	Media	45	Como bodeguero, registro ingreso/salida de materia prima asociado a órdenes de producción o compras.
HU-5	Registro compras realizadas.	Baja	35	Como encargado de compras, registro compras de órdenes recibidas desde bodega.

HU-6	Gestión de productos para producción	Alta	60	Como vendedor o gerente, gestiono productos para producción, detalles de materiales y precios.
HU-7	Recepción de pedidos para producción.	Media	60	Como vendedor, registro detalles del cliente registro información del pedido y genero comprobante.
HU-8	Generar orden de producción.	Baja	30	Como vendedor, creo una orden de producción automáticamente tras crear un pedido.

Nota: Elaboración propia.

2.2.4. Arquitectura de la aplicación.

El diagrama mostrado en la figura 15 exhibe los tres entornos creados para el desarrollo, pruebas y despliegue del software:

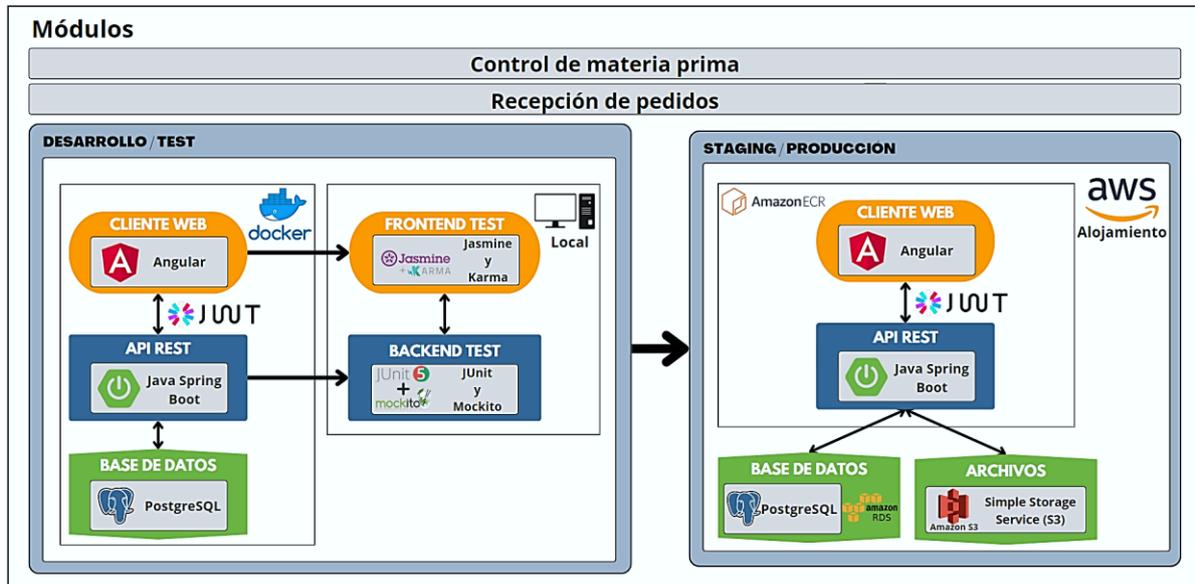
Desarrollo o codificación, junto con pruebas unitarias, aquí, se usó Docker para la ejecución de los componentes de la aplicación (base de datos, backend y frontend), mientras que las pruebas se ejecutaron de forma convencional usando las dependencias locales. Esto permitió ejecutar la aplicación y las pruebas simultáneamente sin inconvenientes.

La etapa de "Staging" representa una réplica del entorno de producción, donde el usuario realizó las pruebas de aceptación, dado que es muy similar a la aplicación en su entorno final, la única diferencia es que este entorno empleo datos de prueba.

Finalmente, el entorno de producción, este integra un componente adicional destinado al almacenamiento de archivos estáticos, como fotografías u otro tipo de documentos los componentes de la aplicación en producción se hospedan en la plataforma de Amazon Web Services, distribuidos en varios servicios. Estos incluyen Servicio de Bases de Datos Relacionales (RDS) para almacenar la base de datos Postgres, Amazon Elastic Container Registry (ECR) para almacenar los contenedores backend y frontend de la aplicación, por último, Amazon Simple Storage Service (S3) para el almacenamiento de archivos estáticos.

Figura 15

Arquitectura de la aplicación



Nota: Elaboración propia.

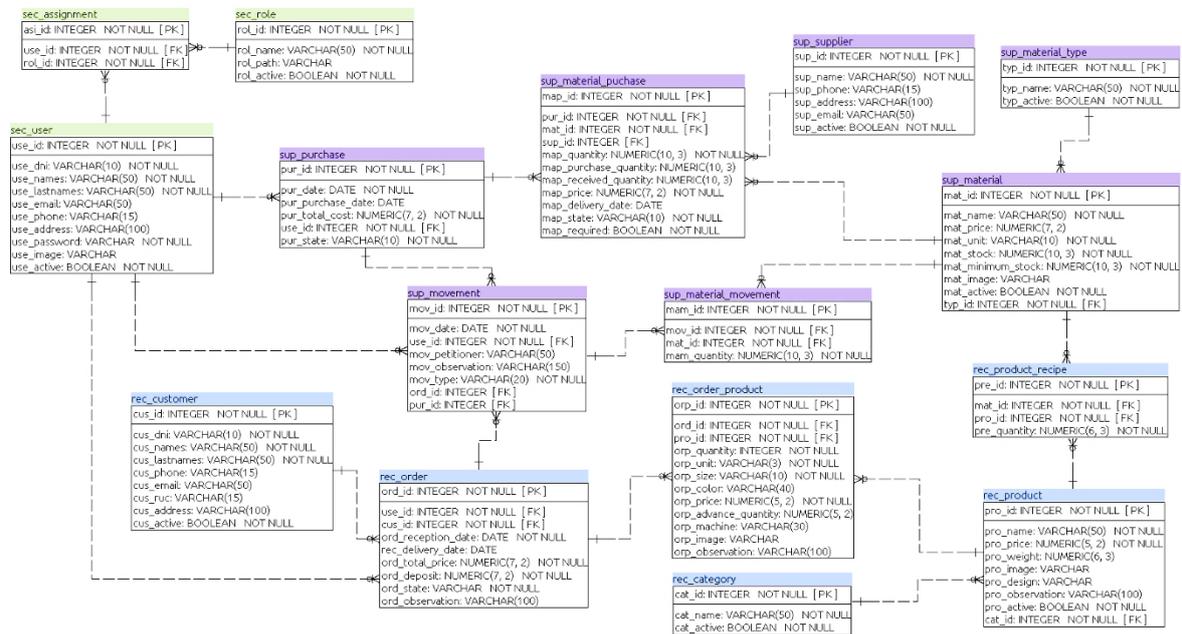
2.2.5. Diseño de base de datos

La Figura 16, presenta el diagrama de base de datos, modelado en consonancia con las especificaciones obtenidas en las historias de usuario. Este diseño aborda de manera específica tres módulos fundamentales: abastecimiento, seguridad y recepción de pedidos (ventas). Cada módulo se desglosa en una estructura coherente de tablas, llegando a 16 en conjunto.

En el módulo de abastecimiento, se incluyen tablas que registran proveedores, materiales, solicitudes de material, compras y movimientos, asegurando una gestión eficiente de los recursos. Por su parte, el módulo de seguridad está enfocado en la protección de la información sensible, con tablas que detallan permisos y roles de acceso. Finalmente, el módulo de recepción o ventas se estructura con tablas que capturan datos cruciales para el seguimiento de clientes, los productos disponibles y los pedidos recibidos. Este diseño integral garantiza una base de datos robusta y adaptable a las necesidades específicas de la aplicación.

Figura 16

Diseño de base de datos



Nota: Elaboración propia.

2.2.6. Plan de pruebas

Para el desarrollo de software, se optó por crear un plan de pruebas simplificado siguiendo los lineamientos propuestos por Crispin y Gregory (2008) y mostrados previamente en la Figura 9. Basándose en esta plantilla se procedió a crear un plan específico que incluyera los elementos esenciales como alcance, recursos, funcionalidades a probar, pruebas de aceptación del usuario (UAT), consideraciones de infraestructura, supuestos y riesgos.

El objetivo fue producir un documento formal y estratégico que sirva como guía efectiva para las actividades de prueba, centrándose en los aspectos indispensables para el contexto del proyecto y satisfaciendo los requerimientos pertinentes.

Figura 17

Plan de pruebas

Plan de Pruebas Gestor Neltex Preparado por: Kevin Lliguisupa		
Introducción		
El plan de pruebas se establece como una referencia para definir lo que está incluido y excluido del alcance de las pruebas, así como para identificar los riesgos y suposiciones.		
Recursos		
Nombre	Rol	% Comprometido
Kevin Lliguisupa	Tester	100%
Nelson Castañeda	Usuario final	25%
Dentro del Alcance		
Dentro del plan se contempla la ejecución de las pruebas unitarias y de aceptación según los casos planteados. En el contexto de las pruebas unitarias, se ejecutarán casos adicionales para los puntos críticos de la aplicación, tales como fragmentos de código propensos a fallos o de gran relevancia para el negocio. Además, se realizarán pruebas necesarias para alcanzar los objetivos de cobertura de código.		
Fuera del Alcance		
En este plan de pruebas solo se contempla la ejecución de pruebas unitarias y de aceptación. Cualquier otro tipo de pruebas quedan exentas de esta planificación.		
Funcionalidades		
Las siguientes funcionalidades serán desarrolladas para este proyecto.		
Descripción de la función	Profundidad de las pruebas	
Login con roles de usuario	Acceso con credenciales válidas e inválidas, gestión correcta de los usuarios y asignación precisa de roles.	
Gestión de materia prima	Gestión correcta de los materiales (creación, listado, modificación y cambio de estado).	
Creación de órdenes de compra	Correcta generación de órdenes, adición adecuada de materiales con sus cantidades y exactitud del costo.	
Registro de movimiento de materia prima	Correcta creación de los movimientos de material, exactitud en el aumento o reducción de las existencias de material.	
Registro compras realizadas.	Visualización adecuada de órdenes de compra y correcto registro de compras realizadas.	
Gestión de productos para producción	Gestión correcta de los productos, exactitud en la estimación del costo de su precio según sus materiales.	
Recepción de pedidos para producción.	Creación correcta del pedido, precisión tanto en los valores de los precios como en los montos a cancelar.	
Generar orden de producción.	Creación correcta de la orden para producción, visualización de las órdenes existentes y pendientes.	
UAT (Pruebas de Aceptación del Usuario)		
La ejecución de estas pruebas se coordinará con varios miembros de la empresa, quienes evaluarán las funcionalidades según sus responsabilidades. Además del gerente, quien evaluará plenamente el software.		
Las pruebas se llevarán a cabo en las instalaciones de la Fábrica de Medias Neltex al finalizar el desarrollo de cada sprint, su duración será de máximo una hora, previo a su ejecución se brindará una explicación del software.		

Consideraciones de Infraestructura

Las pruebas unitarias se las realizará en el ambiente de desarrollo, mientras que para las pruebas de aceptación se creará un ambiente lo más similar posible al de producción con datos de prueba.

Suposiciones

Las pruebas unitarias serán creadas y ejecutadas en la fase de desarrollo del sprint y se tendrán un porcentaje de cobertura de código superior al 60%.

Previo a las pruebas de aceptación se realizará una breve introducción de las funcionalidades a probar.

Riesgos

Se han identificado los siguientes riesgos y la acción adecuada para mitigar su impacto en el proyecto. El impacto (o gravedad) del riesgo se basa en cómo se vería impactado el proyecto si se materializa el riesgo.

#	Riesgo	Impacto	Plan de mitigación
1	Los usuarios no están listos para el UAT	Alto	Acordar previamente con los usuarios los días y horarios para la ejecución de las pruebas, además de una correcta socialización sobre las pruebas.

Nota: Adaptado de (Crispin & Gregory, 2008).

2.3. Proceso de desarrollo iterativo con Scrum

2.3.1. Nomenclatura y organización de los sprints.

Durante la planificación de los sprints, se asignaron historias de usuario a cada uno de los sprints, los cuales tienen una duración dependiendo de su complejidad entre dos y tres semanas. En la Tabla 23 se puede ver las historias de usuario involucradas, así como la fecha de inicio, finalización y duración en horas de cada sprint.

Tabla 23

Organización de los sprints

Nro. Sprint	Historias de Usuario	Inicio	Fin	Duración (Horas)
Sprint 1	HU-1, HU-2	30/10/2023	19/11/2023	120
Sprint 2	HU-6, HU-7	20/11/2023	10/12/2023	120
Sprint 3	HU-3, HU-5	11/12/2023	24/12/2023	80
Sprint 4	HU-4, HU-8	25/12/2023	07/01/2024	80

Nota: Elaboración propia.

2.3.2. Planificación y desarrollo del sprint 1.

Planificación del sprint 1

Esta iteración tomó en cuenta las funcionalidades de login con roles de usuario y gestión de materia prima. Además, se agregaron tareas para el desarrollo del sprint y preparación del proyecto, las cuales fueron tomadas en cuenta durante la planificación.

El Sprint Backlog resultante se muestra en la Tabla 24, detallando todas las actividades llevadas a cabo junto con la estimación del tiempo necesario para su finalización.

Tabla 24

Planificación sprint 1

Código HU	Nombre	Tarea	Horas
	Preparación del sprint y el proyecto	Diseño de casos de prueba	6
		Modificación de base de datos del proyecto	6
HU-1	Login con roles de usuario	Login de usuarios con token	8
		Control de permisos de usuario	9
		Listado y búsqueda de usuarios	9
		Visualización de la información de un usuario	8
		Creación y modificación de usuarios	8
		Listado de roles existentes y sus asignaciones	9
		Creación y eliminación de asignaciones	9
HU-2	Gestión de materia prima	Listado y búsqueda de materiales	8
		Visualización de la información de un material	8
		Creación y modificación de materiales	9
		Menú de la aplicación	7
		Ejecución de pruebas unitarias	9
		Ejecución de pruebas aceptación	7
Total			120

Nota: Elaboración propia.

Planificación de pruebas del sprint 1

Para llevar a cabo exitosamente tanto las pruebas unitarias como las de aceptación, es necesaria una planificación anticipada. Esta planificación ayuda a definir las pruebas a ejecutar mediante la planeación de los casos de prueba. Dichos casos consisten en una breve descripción de la prueba, las condiciones iniciales y resultados esperados.

los casos de prueba se derivan con respecto a los criterios de aceptación definidos por las historias de usuarios, lo que puede generar más de un caso por cada uno de los criterios.

Pruebas unitarias

Los casos de pruebas unitarias planteados son mostrados en la Tabla 25 y la Tabla 26, estos corresponden a las historias HU-1 y HU-2 respectivamente.

Tabla 25

Casos de pruebas unitarias HU-1 Login con roles de usuario

Id del Caso	Descripción	Condiciones de Entrada	Salida Esperada
CPU-1.1	Verificar el acceso con credenciales válidas	Credenciales válidas, usuario activo	Acceso permitido
CPU-1.2	Verificar el acceso con credenciales inválidas	Credenciales incorrectas, usuario activo	Acceso denegado y mensaje de error
CPU-1.3	Verificar Acceso Usuario Inactivo	Credenciales válidas, usuario inactivo	Mensaje de usuario inactivo
CPU-1.4	Verificar Acceso con Rol Correcto	Credenciales válidas, usuario activo, rol válido	Acceso permitido al recurso solicitado
CPU-1.5	Verificar Acceso con Rol Incorrecto	Credenciales válidas, usuario activo, rol inválido	Acceso denegado al recurso solicitado. Error 403
CPU-1.6	Obtener lista de empleados	N/A	Lista de empleados registrados en la aplicación.
CPU-1.7	Asignar rol a un empleado	Rol por asignar, empleado sin el rol.	Rol asignado con éxito
CPU-1.8	Eliminar rol de un empleado	Rol por eliminar, empleado con el rol.	Rol eliminado con éxito

Nota: Elaboración propia.

Tabla 26

Casos de pruebas unitarias HU-2 Gestión de materia prima

Id del Caso	Descripción	Condiciones de Entrada	Salida Esperada
CPU-2.1	Agregar materia prima al inventario	Datos de materia prima válidos	Materia prima creada con éxito
CPU-2.2	Ver lista de materias primas existentes	N/A	Lista de materias primas devuelta correctamente
CPU-2.3	Editar información de materia prima	Materia prima por editar, datos válidos	Información actualizada correctamente
CPU-2.4	Cambiar estado de material a "Activo"	Materia prima en estado "Inactivo"	Estado cambiado a "Activo" con éxito
CPU-2.5	Cambiar estado de material a " Inactivo "	Materia prima en estado "Activo"	Estado cambiado a " Inactivo " con éxito

Nota: Elaboración propia.

Pruebas de aceptación

Los casos correspondientes a las pruebas de aceptación se muestran en la Tabla 27 y Tabla 28. Como miembros para el desarrollo estuvieron el encargado de bodega y el gerente de la Fábrica de medias Neltex.

Tabla 27

Casos de pruebas de aceptación HU-1 Login con roles de usuario

Id Caso	Descripción	Precondición	Pasos por seguir	Resultado esperado
CPA-1.1	Verificar acceso con credenciales correctas	Número de cédula y contraseña válidos	<ol style="list-style-type: none"> 1. Ingresar número de cédula 2. Ingresar la contraseña 3. Hacer clic en "Ingresar" 	Inicio de sesión exitoso.
CPA-1.2	Verificar acceso con usuario inactivo	Credenciales válidas para usuario inactivo	<ol style="list-style-type: none"> 1. Ingresar número de cédula 2. Ingresar la contraseña 3. Hacer clic en "Ingresar" 	Error inicio de sesión incorrecta
CPA-1.3	Ver lista de empleados	N/A	<ol style="list-style-type: none"> 1. Clic a la opción de gestión de empleados 	Lista de empleados mostrada.
CPA-1.4	Asignar rol a empleado	Empleado nuevo o creado y rol a asignar	<ol style="list-style-type: none"> 1. Hacer clic en nuevo empleado / modificar 2. Seleccionar un rol 3. Guardar / crear 	Rol de usuario asignado correctamente
CPA-1.5	Eliminar rol de empleado	Empleado y roles asignados	<ol style="list-style-type: none"> 1. Seleccionar empleado 2. Eliminar rol de la lista de asignaciones 3. Guardar cambios 	Rol de usuario eliminado correctamente
CPA-1.6	Ver roles disponibles	N/A	<ol style="list-style-type: none"> 1. Acceder a la opción nuevo empleado / modificar 2. Revisar la lista de roles disponibles 	Lista incluye "Gerente," "Adquisición," "Vendedor," y "Bodeguero"

Nota: Elaboración propia.

Tabla 28

Casos de pruebas de aceptación HU-2 Gestión de materia prima

Id Caso	Descripción	Precondición	Pasos por seguir	Resultado esperado
CPA-2.1	Crear materia prima	Información de la materia prima	<ol style="list-style-type: none"> 1. Acceder a nuevo material 2. Ingresar la información 3. Hacer clic en "Crear" 	Materia prima creada con éxito.
CPA-2.2	Ver lista de materias primas	N/A	<ol style="list-style-type: none"> 1. Acceder a gestión de materiales 	Lista de materias primas mostrada.
CPA-2.3	Editar información de materia prima	Materia prima por editar	<ol style="list-style-type: none"> 1. Seleccionar materia prima 2. Editar la información deseada 3. Hacer clic en "Guardar" 	Información de materia prima editada correctamente.

CPA-2.4	Cambiar estado de materia prima a "Activa"	Materia prima por activar	<ol style="list-style-type: none"> 1. Hacer clic en la opción "Cambiar de estado" del material deseado 2. Aceptar 	Estado cambiado a "Activa" con éxito.
CPA-2.5	Cambiar estado de materia prima a "Inactiva"	Materia prima por desactivar	<ol style="list-style-type: none"> 1. Hacer clic en la opción "Cambiar de estado" del material deseado 2. Aceptar 	Estado cambiado a "Inactiva" con éxito.

Nota: Elaboración propia.

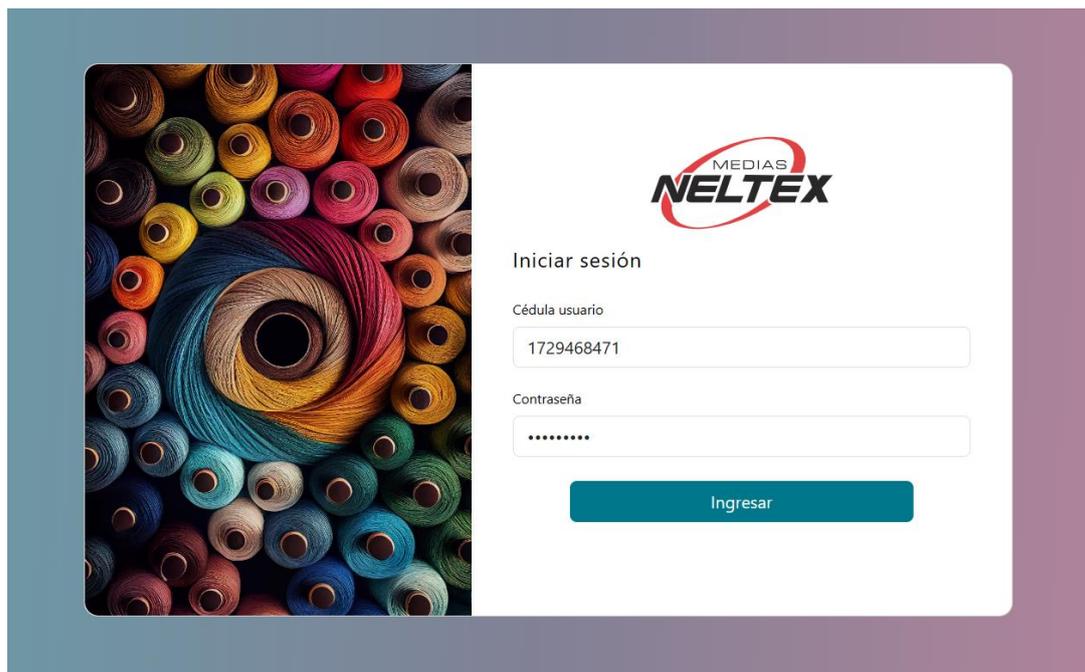
Incremento 1

Login con roles de usuario

La Figura 18 presenta la pantalla de login (iniciar sesión), donde el usuario ingresa sus credenciales y de ser válidas y tener un usuario activo este es redirigido a la pantalla de menú.

Figura 18

Pantalla de inicio de sesión

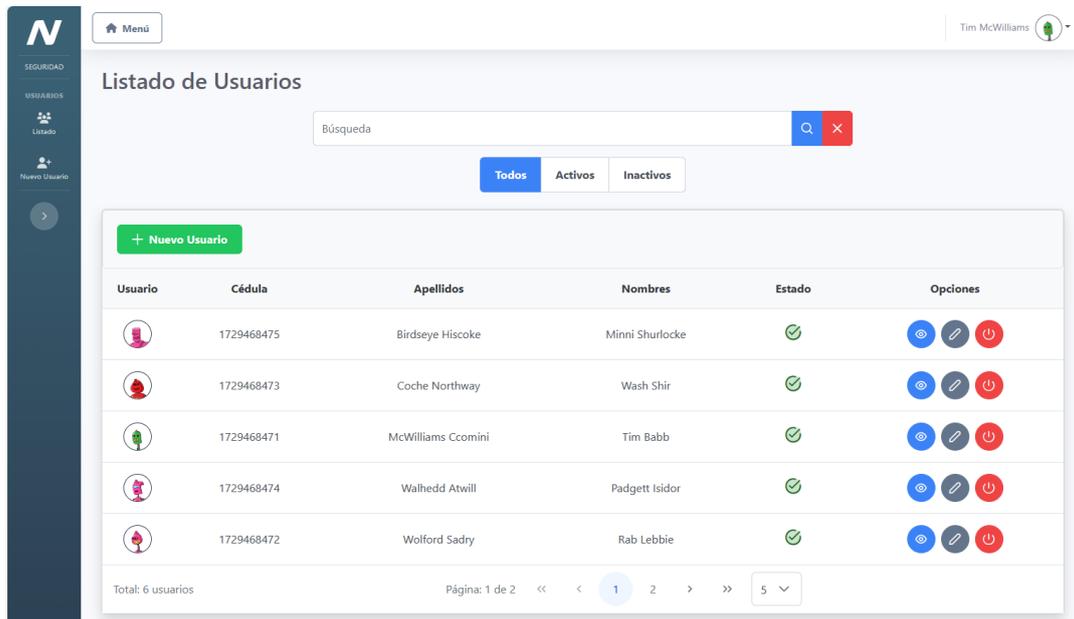


Nota: Elaboración propia.

La Figura 19 muestra la pantalla de administración de usuarios del sistema la cual solo puede ser accedida por usuarios con rol de "Seguridad". Desde aquí pueden registrar nuevos usuarios asignándoles roles, modificar usuarios existentes o desactivarlos.

Figura 19

Pantalla de administración de usuarios



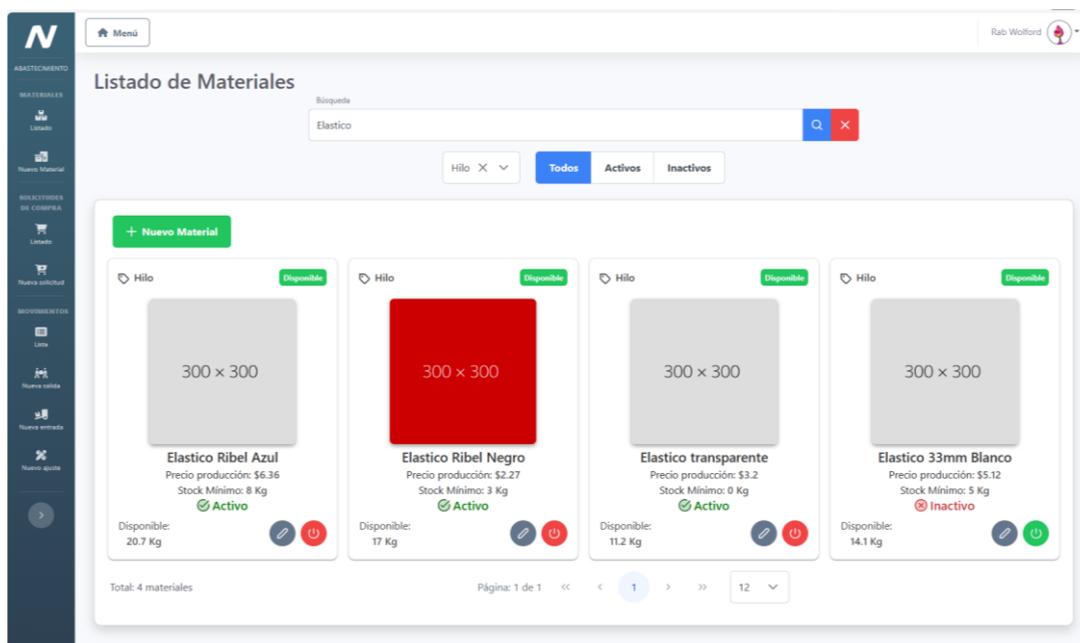
Nota: Elaboración propia.

Gestión de materia prima

La Figura 20 muestra el listado de materiales existentes, indicando la información más importante, además de su imagen. Esta pantalla se permite la visualización, edición, creación y desactivación del material.

Figura 20

Pantalla de administración de materiales



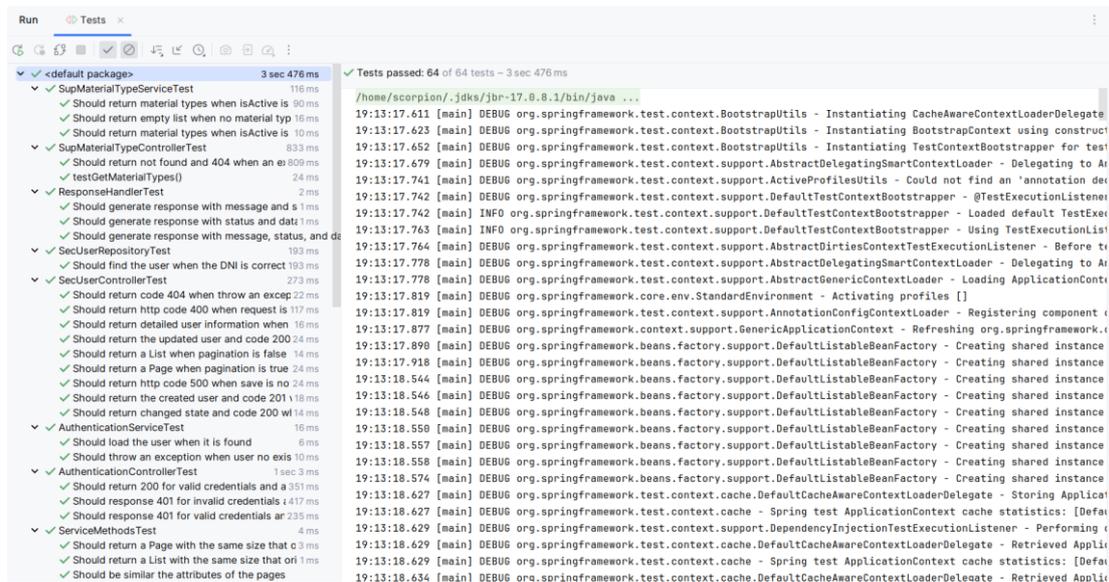
Ejecución de pruebas del sprint 1

Ejecución de pruebas unitarias

La ejecución de las pruebas unitarias, previamente propuestas, se presenta en la Figura 21 y Figura 22. Además de los casos inicialmente planteados, fueron creados más pruebas con el objetivo de incrementar el porcentaje de cobertura del código.

Figura 21

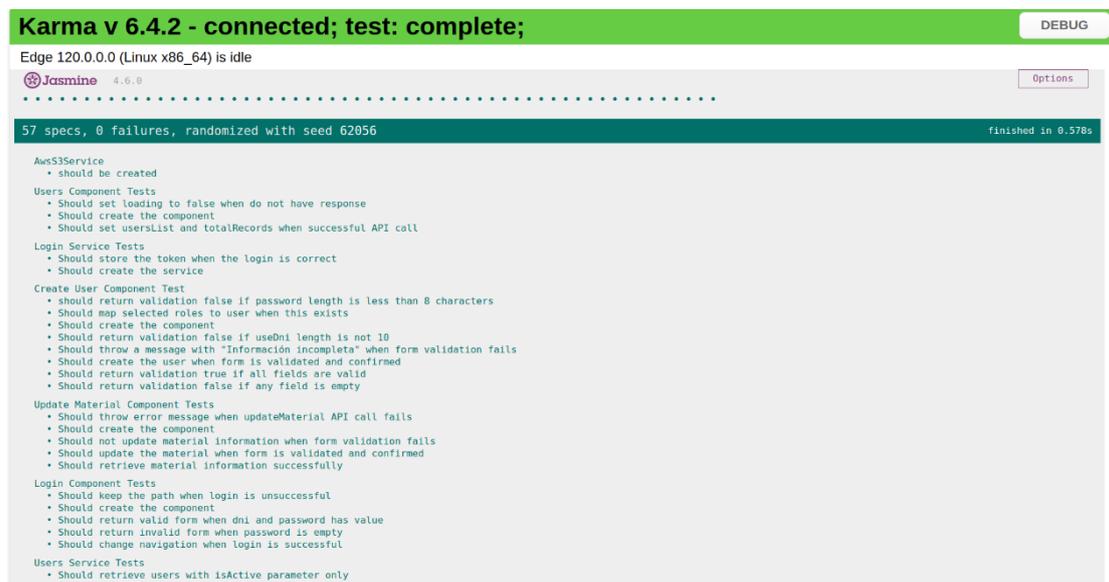
Ejecución de pruebas unitarias en el backend sprint 1



Nota: Elaboración propia.

Figura 22

Ejecución de pruebas unitarias en el frontend sprint 1



Nota: Elaboración propia.

Ejecución de pruebas de aceptación

Las pruebas de aceptación se llevaron a cabo en colaboración con los usuarios finales quienes fueron sometidos a los casos previamente planteados. Una visión más detallada de estas pruebas, se pueden consultar en el Anexo B. Los resultados obtenidos al ejecutar estas pruebas se muestran en la Tabla 29.

Tabla 29

Ejecución de pruebas de aceptación sprint 1

Id Caso	Descripción	Validación	Observaciones
CPA-1.1	Verificar acceso con credenciales correctas	Aprobado	
CPA-1.2	Verificar acceso con usuario inactivo	Aprobado	
CPA-1.3	Ver lista de empleados	Aprobado	Implementar búsqueda al presionar la tecla "Enter"
CPA-1.4	Asignar rol a empleado	Aprobado	Permite crear varios usuarios con la misma cédula
CPA-1.5	Eliminar rol de empleado	Aprobado	
CPA-1.6	Ver roles disponibles	Aprobado	
CPA-2.1	Crear materia prima	Rechazado	Presenta un mensaje de error al crear el material
CPA-2.2	Ver lista de materias primas	Aprobado	Implementar búsqueda al presionar la tecla "Enter"
CPA-2.3	Editar información de materia prima	Aprobado	
CPA-2.4	Cambiar estado de materia prima a "Activa"	Aprobado	
CPA-2.5	Cambiar estado de materia prima a "Inactiva"	Aprobado	

2.3.3. Planificación y desarrollo del sprint 2.

Planificación del sprint 2

Esta iteración consideró las funcionalidades de gestión de productos para producción y recepción de pedidos para producción. El Sprint Backlog resultante es mostrado en la Tabla 30 y detalla todas las actividades realizadas junto con su estimación del tiempo.

Tabla 30

Planificación sprint 2

Código HU	Nombre	Tarea	Horas
	Preparación del sprint y cambios	Diseño de casos de prueba	4
		Cambios y corrección de errores	6
HU-6	Gestión de productos para producción	Listado y búsqueda de productos	6
		Creación y modificación de productos	9
		Integración con materias primas	7
		Cálculos automáticos en productos	6
		Listado y búsqueda de materiales necesarios	7
		Cambio de estado de un producto	4
		Información detallada del producto	6
		Listado de clientes	6
		Registro de información de cliente	6
		Adición de productos al pedido	9
HU-7	Recepción de pedidos para producción	Creación de un pedido	8
		Listado de pedidos existentes	6
		Información detallada del pedido	8
		Generación de comprobante en PDF	6
		Ejecución pruebas unitarias	9
		Ejecución pruebas de aceptación	7
		Total	120

Nota: Elaboración propia.

Planificación de pruebas del sprint 2**Pruebas unitarias**

Las pruebas unitarias ideados se exhiben en la Tabla 31 y Tabla 32 las cuales corresponden a las historias HU-6 y HU-7 respectivamente.

Tabla 31

Casos de pruebas unitarias HU-6 Gestión de productos para producción

Id del Caso	Descripción	Condiciones de Entrada	Salida Esperada
CPU-3.1	Crear nuevo producto con datos válidos	Datos de nuevo producto con materiales asociados válidos	Producto creado en base de datos
CPU-3.2	Crear un producto sin asociar materiales	Datos de nuevo producto sin materiales asociados	Producto creado en BD
CPU-3.3	Ver lista de productos cuando no hay productos	Navegar a la lista de productos	Mensaje indicando que no hay productos disponibles

CPU-3.4	Editar información de un producto existente	Seleccionar producto y modificar información válida	Información del producto actualizada en la BD
CPU-3.5	Cambiar el estado de un producto	Seleccionar producto y cambiar estado	Actualización de estado del producto correcta
CPU-3.6	Calcular costo total de materiales para un producto sin materiales	Crear un producto sin materiales y verificar el costo de los materiales calculado	Costo total de materiales igual a cero
CPU-3.7	Crear un producto con cantidades negativas	Ingresar un producto con cantidades negativas	Mensaje de error indicando cantidades inválidas

Nota: Elaboración propia.

Tabla 32

Casos de pruebas unitarias HU-7 Recepción de pedidos para producción

Id del Caso	Descripción	Entrada	Salida Esperada
CPU-4.1	Crear nuevo pedido con datos válidos	Objeto de pedido con cliente y detalles válidos	Pedido registrado en la base de datos
CPU-4.2	Crear un pedido sin detalles de productos	Objeto de pedido con cliente y sin productos	Excepción indicando que se requieren productos en el pedido
CPU-4.3	Agregar productos al pedido con cantidades negativas	Objeto de pedido con productos y cantidades negativas	Excepción indicando cantidades inválidas
CPU-4.4	Guardar pedido sin detalles de cliente	Objeto de pedido sin cliente	Excepción indicando que se requieren detalles del cliente
CPU-4.5	Visualizar pedidos recibidos cuando no hay pedidos	Navegar a la lista de pedidos	Lista vacía de pedidos
CPU-4.6	Agregar producto al pedido con cantidad decimal	Objeto de pedido con producto y cantidad decimal	Producto agregado al pedido con cantidad redondeada
CPU-4.7	Calcular total del pedido con productos de diferentes precios	Objeto de pedido con productos de diferentes precios	Total, del pedido calculado correctamente considerando productos de diferentes precios

Nota: Elaboración propia.

Pruebas de aceptación

Los casos relativos a las pruebas de aceptación para este sprint son mostrados en la Tabla 33 y Tabla 34. Como miembros para la ejecución estuvieron el encargado de producción y el encargado de ventas de la Fábrica de medias Neltex.

Tabla 33

Casos de pruebas de aceptación HU-6 Gestión de productos para producción

Id Caso	Descripción	Precondición	Pasos por seguir	Resultado esperado
CPA-3.1	Crear nuevo producto.	Información del producto, materiales.	1. Acceder a nuevo producto 2. Ingresar información y materiales. 3. Guardar.	Mensaje: "Producto creado correctamente".
CPA-3.2	Crear producto sin materiales.	Información del producto.	1. Acceder a nuevo producto. 2. Ingresar información. 3. Guardar.	Mensaje: "Producto creado correctamente".
CPA-3.3	Editar información del producto.	Producto por editar y sus materiales	1. Acceder y editar producto. 2. Modificar información. 3. Guardar.	Mensaje: "Producto actualizado correctamente".
CPA-3.4	Cambiar estado del producto.	Producto por cambiar	1. Acceder y seleccionar producto 2. Cambiar estado.	Mensaje: "Producto modificado con éxito"
CPA-3.5	Cálculo de costo de materiales.	Nuevo producto con materiales	1. Crear nuevo producto con materiales.	Costo total de materiales correcto.
CPA-3.6	Ver materiales necesarios.	Producto con materiales	1. Acceder y seleccionar producto 2. Ver materiales asociados al producto.	Lista de materiales asociados mostrada correctamente.

Nota: Elaboración propia.**Tabla 34**

Casos de pruebas de aceptación HU-7 Recepción de pedidos para producción

Id Caso	Descripción	Precondición	Pasos por seguir	Resultado esperado
CPA-4.1	Recibir nuevo pedido.	Información del cliente, detalles del pedido.	1. Acceder a nuevo pedido. 2. Ingresar información 3. Guardar.	Mensaje: "Pedido creado correctamente".
CPA-4.2	Agregar productos al pedido.	Productos por agregar al pedido	1. Acceder a nuevo pedido. 2. Seleccionar productos. 3. Especificar detalles. 4. Guardar.	Productos agregados con éxito.
CPA-4.3	Calcular valor total.	Productos en el pedido	1. Verificar que el total del pedido se actualice automáticamente.	Total, del pedido calculado y correcto.
CPA-4.4	Guardar y generar comprobante	Pedido a guardar	1. Guardar pedido y generar comprobante. 2. Verificar detalles del comprobante.	Pedido guardado y comprobante generado con detalles.
CPA-4.5	Visualizar pedidos	N/A	1. Acceder y visualizar pedidos recibidos.	Lista de pedidos mostrada.
CPA-4.6	Eliminar producto del pedido	Producto en el pedido	1. Seleccionar y eliminar producto del pedido.	Producto eliminado del pedido correctamente.

Nota: Elaboración propia.

Incremento 2

Gestión de productos para producción

La Figura 23 y Figura 24 muestran la interfaz para administrar los productos disponibles para producir. Desde aquí se permite registrar un nuevo producto indicando los materiales necesarios, ver los productos existentes, editarlos o desactivarlos.

Figura 23

Creación de producto

← Atrás

Padgett Walbeed

Creación de producto

Información del producto

Nombre: Antideslizantes bebe

Categoría: Antideslizantes

Precio de venta: \$12.50

Diseño: + Seleccionar

Información del material: Costo de materiales: \$6.21, Peso por docena: 1.364 Kg

Antideslizante de talla de bebe

Materiales utilizados

#	Material	Tipo	%	Cantidad	Precio	Costo	Total
1	Algodon Peinado Rojo	Licra	24.93%	0.340 Kg	\$5.53	\$1.88	✖
2	Hilo Enkanyil Azul-Cielo	Sin tipo	11.00%	0.150 Kg	\$3.90	\$0.59	✖
3	Elastico Ribel Negro	Hilo	29.33%	0.400 Kg	\$2.27	\$0.91	✖
4	Hilo Enkaten Verde-Dixie	Sin tipo	34.75%	0.474 Kg	\$5.98	\$2.83	✖

+ Agregar

Crear Producto

Figura 24

Gestión de productos

Menú

Padgett Walbeed

Listado de Productos

Búsqueda

Filtrar por categoría

Todos Activos Inactivos

+ Nuevo Producto

- Antideslizantes bebe** - \$12.50 - Activo
- Babucha Algodon-normal** - \$18.47 - Activo
- Babucha Algodon-peinado** - \$19.19 - Activo
- Canillera Algodon-normal** - \$13.85 - Activo
- Canillera Algodon-peinado** - \$16.60 - Activo
- Canillera Antideslizante** - \$21.53 - Activo

Nota: Elaboración propia.

Recepción de pedidos para producción

La Figura 25 presenta la pantalla para generar un nuevo pedido. Se ingresa la información del cliente y los productos requeridos con sus cantidades. Al finalizar, se crea un comprobante con los detalles.

Figura 25

Pantalla de recepción de pedidos

Recepción de pedido

Fecha Recepción / Entrega: 07-11-2023 - 27-12-2023

Vendedor: Padgett Walhedd

Información del cliente

Nro. Cédula: 4082280251
Apellidos: Barthot Rothon
Nombres: Aubree Stanislas

RUC: 5105104943
Email: srothond@newsvine.com
Celular: 8329376206

Productos seleccionados

Producto	Cantidad	Color	Talla	Precio
Antideslizantes bebe Antideslizantes Precio por docena: \$12.5	2 Dozenas	Amarillo Fosforescente	0-3	\$25
Babucha Algodon-peinado Deportivas Precio por docena: \$19.19	8 Pares	Azul	9-12	\$12.79
Casual Economica Invisibles Precio por docena: \$21.51	1 Dozenas		10-12	\$21.51

Observaciones: Todas la medias deben llevar una franja de color rojo seguida de una azul en la punta.

Resumen
Valor total: \$59.3
Abono: 20
Valor pendiente: \$39.3

Crear Pedido

Nota: Elaboración propia.

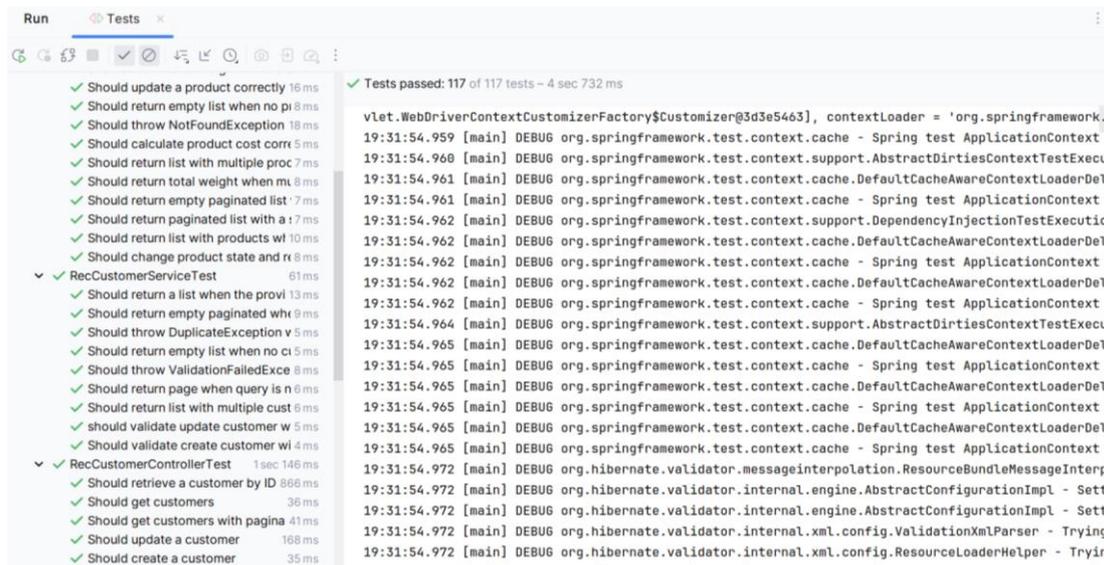
Ejecución de pruebas del sprint 2

Ejecución de pruebas unitarias

Los casos de pruebas unitarias planteados para este sprint se exhiben en la Figura 26 y Figura 27. Además, de las pruebas creadas para aumentar el porcentaje de cobertura del código.

Figura 26

Ejecución de pruebas unitarias en el backend sprint 2



Nota: Elaboración propia.

Figura 27

Ejecución de pruebas unitarias en el frontend sprint 2



Nota: Elaboración propia.

Ejecución de pruebas de aceptación

Las pruebas de aceptación se llevaron a cabo en colaboración con los usuarios finales quienes fueron sometidos a los casos previamente planteados. Una visión más detallada de

estas pruebas, se pueden consultar el Anexo C. Los resultados obtenidos al ejecutar estas pruebas se muestran en la Tabla 35.

Tabla 35

Ejecución de pruebas de aceptación sprint 2

Id Caso	Resultado esperado	Validación	Observaciones
CPA-3.1	Crear nuevo producto.	Aprobado	
CPA-3.2	Crear producto sin materiales.	Aprobado	
CPA-3.3	Editar información del producto.	Aprobado	
CPA-3.4	Cambiar estado del producto.	Aprobado	
CPA-3.5	Cálculo de costo de materiales.	Aprobado	
CPA-3.6	Ver materiales necesarios.	Aprobado	
CPA-4.1	Recibir nuevo pedido.	Aprobado	La fecha de recepción debe poder modificarse
CPA-4.2	Agregar productos al pedido.	Aprobado	Debe permitir ingresar el color y tamaño como opcional
CPA-4.3	Calcular valor total.	Aprobado	
CPA-4.4	Guardar y generar comprobante	Aprobado	Corrección de email de la empresa en el comprobante
CPA-4.5	Visualizar pedidos	Aprobado	
CPA-4.6	Eliminar producto del pedido	Aprobado	

Nota: Elaboración propia.

2.3.4. Planificación y desarrollo del sprint 3

Planificación del sprint 3

Esta iteración contempló las funcionalidades de creación de órdenes de compra y registro de compras realizadas, correspondientes a las historias de usuario HU-3 y HU-5 respectivamente, el Sprint Backlog resultante se muestra en la Tabla 36.

Tabla 36

Planificación sprint 3

Código HU	Nombre	Tarea	Horas
	Preparación del sprint y cambios	Diseño de casos de prueba	4
		Cambios y corrección de errores	6
HU-3	Creación de órdenes de compra.	Listado de solicitudes de compra	6
		Listado y búsqueda de materiales solicitados	5
		Creación de solicitud de compra	6
		Modificación de la solicitud de compra	4

		Información detallada de la solicitud de compra	5
		Eliminación de la solicitud de compra	3
		Listado de proveedores	3
		Creación de nuevo proveedor	4
		Listado de solicitudes de compra pendientes	3
		Compra a partir de una solicitud de compra	5
HU-5	Registro de compras realizadas.	Crear una nueva compra	5
		Cálculo del valor total de la compra	2
		Información detallada de la compra	5
		Listado de compras como bodeguero y compras	4
		Ejecución de pruebas unitarias	6
		Ejecución de pruebas aceptación	4
		Total	80

Nota: Elaboración propia.

Planificación de pruebas del sprint 3

Pruebas unitarias

Los casos de pruebas unitarias concebidos se exhiben en la Tabla 37 y Tabla 38, las cuales se corresponden con las historias HU-3 y HU-5 respectivamente.

Tabla 37

Casos de pruebas unitarias HU-3 Creación de órdenes de compra

Id del Caso	Descripción	Condiciones de Entrada	Salida Esperada
CPU-5.1	Ver lista de todas las órdenes de compra existentes	Navegar a la lista de órdenes de compra	Lista de órdenes de compra mostrada
CPU-5.2	Crear nueva orden de compra con datos válidos	Objeto de orden de compra con materiales y cantidades válidas	Orden de compra creada en la base de datos
CPU-5.3	Intentar crear una orden de compra sin materiales	Objeto de orden de compra sin materiales	Excepción indicando que se requieren materiales en la orden
CPU-5.4	Crear nueva orden de compra con cantidad decimal	Objeto de orden de compra con cantidades decimales	Orden de compra creada con cantidades redondeadas

Nota: Elaboración propia.

Tabla 38

Casos de pruebas unitarias HU-5 Registro de compras realizadas

Id del Caso	Descripción	Condiciones de Entrada	Salida Esperada
CPU-6.1	Ver lista de órdenes de compra pendientes	Navegar a la lista de órdenes de compra pendientes	Lista de órdenes de compra pendientes mostrada.
CPU-6.2	Seleccionar orden de compra y registrar compra con datos válidos	Orden de compra seleccionada y detalles de la transacción válidos	Compra registrada en la base de datos
CPU-6.3	Registrar compra sin seleccionar una orden de compra	Detalles de transacción sin orden de compra seleccionada	Compra registrada en la base de datos
CPU-6.4	Mantener registro de todas las compras realizadas	Ver lista de todas las compras realizadas	Lista de compras realizadas con detalles visibles
CPU-6.5	Registrar compra con detalles de transacción incompletos	Orden de compra seleccionada y detalles de transacción incompletos	Mensaje de error indicando que se requieren detalles completos
CPU-6.7	Cambiar estado de una orden de compra a "Comprado"	Registrar una compra y verificar su estado	Estado de la orden de compra actualizado a "Comprado"
CPU-6.8	Cálculo del valor total de la compra	Registro de varios detalles de la compra	El valor total debe actualizarse

Nota: Elaboración propia.**Pruebas de aceptación**

Los casos relativos a las pruebas de aceptación son mostrados en la Tabla 39 y Tabla 40. Como miembros para la ejecución estuvieron el encargado de compras y bodega junto al gerente de la Fábrica.

Tabla 39

Casos de pruebas de aceptación HU-3 Creación de órdenes de compra

Id Caso	Descripción	Precondición	Pasos por seguir	Resultado esperado
CPA-5.1	Ver lista de órdenes de compra	N/A	1. Acceder y ver lista de órdenes de compra.	Lista mostrada correctamente
CPA-5.2	Crear nueva orden de compra	Materiales por solicitar en la orden	1. Acceder a nueva orden. 2. Especificar materiales, cantidades, etc. 3. Guardar	Mensaje: "Orden de compra creada correctamente"
CPA-5.3	Ver detalles de orden de compra	Orden de compra existente	1. Seleccionar una orden de compra 2. Clic en ver detalles	Detalles de la orden de compra mostrados.

CPA-5.4	Eliminar orden de compra	Orden de compra existente	<ol style="list-style-type: none"> 1. Seleccionar una orden de compra 2. Eliminar la orden 	Orden de compra eliminada correctamente.
CPA-5.5	Filtrar órdenes por estado	Listado de órdenes de compra	<ol style="list-style-type: none"> 1. Acceder y aplicar filtro por estado en lista de órdenes. 	Lista mostrada con órdenes filtradas por estado.
CPA-5.6	Crear orden con materiales nuevos	Detalles de la orden	<ol style="list-style-type: none"> 1. Acceder a nueva orden. 2. Crear nuevo material 3. Especificar cantidad 4. Guardar. 	Mensaje: "Orden de compra creada correctamente"

Nota: Elaboración propia.

Tabla 40

Casos de pruebas de aceptación HU-5 Registro de compras realizadas

Id Caso	Descripción	Precondición	Pasos por seguir	Resultado esperado
CPA-6.1	Ver órdenes pendientes	N/A	<ol style="list-style-type: none"> 1. Acceder y ver lista de órdenes de compra pendientes. 	Lista de órdenes mostrada
CPA-6.2	Registrar compra desde orden	Orden de compra pendiente	<ol style="list-style-type: none"> 1. Seleccionar orden de compra pendiente. 2. Registrar compra con detalles. 3. Guardar. 	Compra registrada correctamente, estado actualizado
CPA-6.3	Ver registro de compras	N/A	<ol style="list-style-type: none"> 1. Acceder y ver lista de compras realizadas. 	Lista de compras realizadas
CPA-6.4	Registrar compra sin detalles	Orden de compra pendiente	<ol style="list-style-type: none"> 1. Seleccionar orden de compra pendiente. 2. Registrar compra sin especificar detalles. 3. Guardar. 	Mensaje de error al intentar registrar compra sin detalles.
CPA-6.5	Registrar compra sin orden asociada	Materiales comprados con sus detalles	<ol style="list-style-type: none"> 1. Acceder a nueva compra 2. Registrar los materiales comprados. 	Compra registrada correctamente.
CPA-6.6	Modificar estado de orden de compra	Orden de compra existente	<ol style="list-style-type: none"> 1. Registrar compra a partir de una orden. 2. Revisar estado actualizado. 	Estado de la orden de compra "Comprado"

Nota: Elaboración propia.

Incremento 3

Creación de órdenes de compra

La Figura 28 exhibe la interfaz para gestión de las solicitudes de compra de material, aquí se pueden identificar las solicitudes según varios estados en los que se encuentran, estas pueden ser solicitado, comprado o recibido.

Figura 28

Pantalla de gestión de solicitudes de compra de material

Nro.	Responsable	Fecha Creación	Valor Total	Estado	Opciones
11	Tim McWilliams	Jan 26, 2024	\$1.00	Comprado	
10	Tim McWilliams	Oct 25, 2023	\$374.09	Comprado	
9	Wash Coche	Oct 23, 2023	\$14.91	Comprado	
8	Wash Coche	Mar 16, 2023	\$63.89	Recibido	
7	Tim McWilliams	Aug 30, 2023	\$14.81	Solicitado	
6	Tim McWilliams	Feb 18, 2023	\$290.83	Recibido	
5	Tim McWilliams	Mar 17, 2023	\$244.29	Recibido	
4	Wash Coche	Jul 31, 2023	\$48.89	Solicitado	
3	Tim McWilliams	Mar 5, 2023	\$138.91	Comprado	
2	Wash Coche	Apr 8, 2023	\$330.68	Comprado	

Total: 11 solicitudes

1 of 2 << < 1 2 > >> 10

Nota: Elaboración propia.

Registro de compras realizadas

La Figura 29 muestra la interfaz de registro de la compra de una orden de materiales recibida desde bodega. Se indica la fecha de compra, proveedor y materiales adquiridos con sus cantidades.

Figura 29

Pantalla de compra de materiales a partir de solicitud

Material	solicitado	comprado	Costo	Entrega	Estado
Algodon Bonafil Rosado ↳ Licra Precio para creación: \$6.64	5 Kg	6 Kg	\$23.00 Elmore Chugg	Feb 8, 2024	Comprado
Elastico Ribel Negro ↳ Hilo Precio para creación: \$2.27	4 Kg	4 Kg	\$12.00	Sin fecha	Comprado
Licra Ribel ↳ Nylon Precio para creación: \$5.06	7 Kg/m2	9 Kg/m2	\$24.00 Aveline Domnick	Feb 9, 2024	Comprado
Malla Pol Verde ↳ Lino Precio para creación: \$3.48	2 Kg/m2		\$0.00	Sin fecha	Solicitado Comprar Anular

Resumen
Costo total: \$59.00 **Crear Compra**

Nota: Elaboración propia.

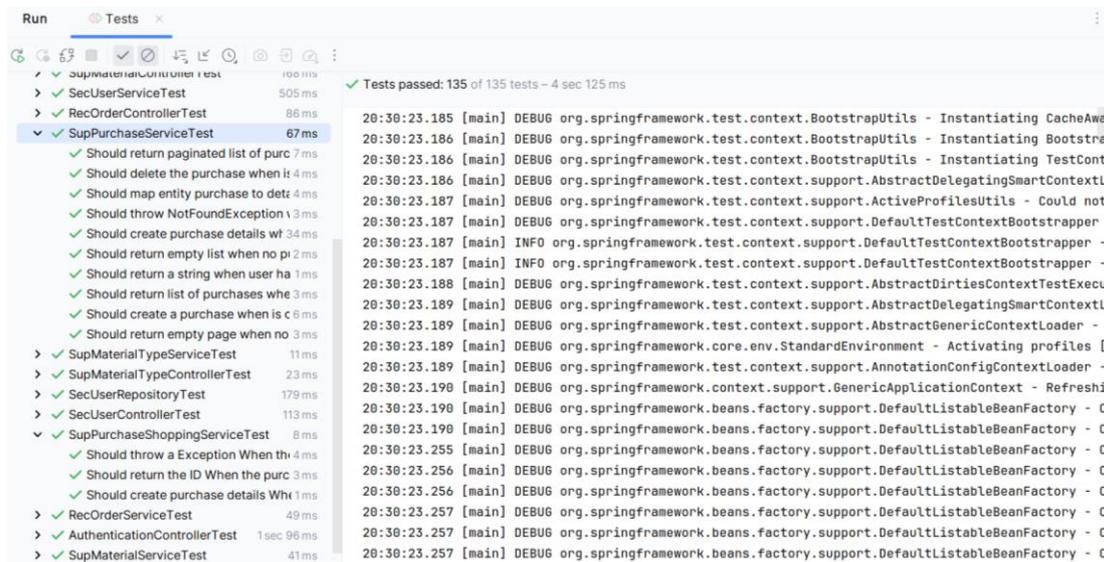
Ejecución de pruebas del sprint 3

Ejecución de pruebas unitarias

Las pruebas unitarias previamente planteadas, se presenta en la Figura 30 y Figura 31. Además, se crearon pruebas adicionales buscando aumentar el porcentaje de cobertura del código.

Figura 30

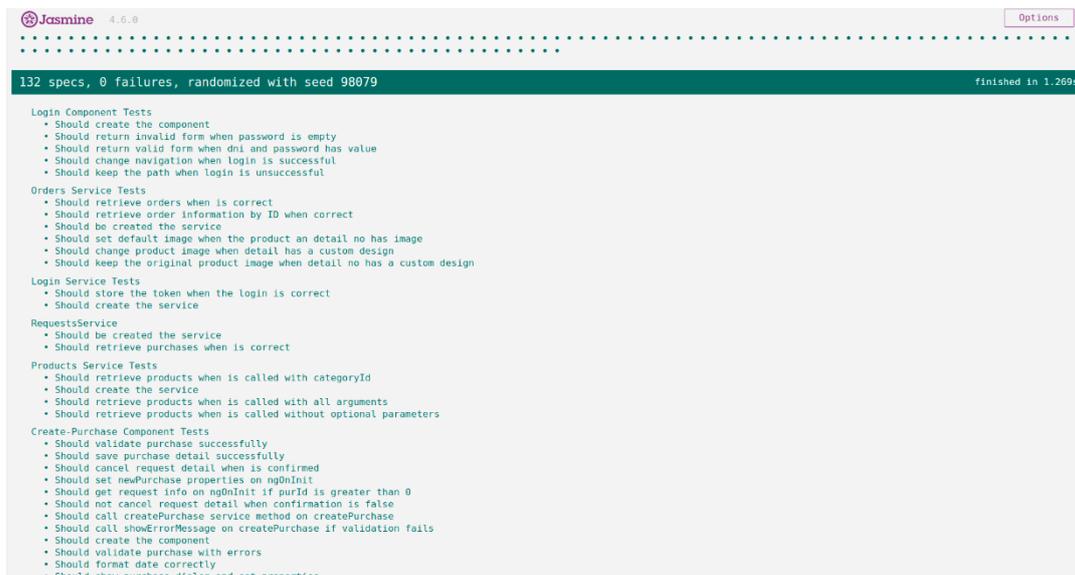
Ejecución de pruebas unitarias en el backend sprint 3



Nota: Elaboración propia.

Figura 31

Ejecución de pruebas unitarias en el frontend sprint 3



Nota: Elaboración propia.

Ejecución de pruebas de aceptación

Las pruebas de aceptación planteadas se efectuaron con los usuarios finales. Para obtener un panorama más detallado de la ejecución, se pueden ver el Anexo D. Los resultados conseguidos son mostrados en la Tabla 41.

Tabla 41

Ejecución de pruebas de aceptación sprint 3

Id Caso	Resultado esperado	Validación	Observaciones
CPA-5.1	Ver lista de órdenes de compra	Aprobado	Agregar el botón de compra en el perfil de Gerente
CPA-5.2	Crear nueva orden de compra	Aprobado	
CPA-5.3	Ver detalles de orden de compra	Aprobado	
CPA-5.4	Eliminar orden de compra	Aprobado	
CPA-5.5	Filtrar órdenes por estado	Aprobado	
CPA-5.6	Crear orden con materiales nuevos	Aprobado	Permitir crear un nuevo tipo de material
CPA-6.1	Ver órdenes pendientes	Aprobado	
CPA-6.2	Registrar compra desde orden	Aprobado	
CPA-6.3	Ver registro de compras	Aprobado	Ver registro de compras como bodeguero
CPA-6.4	Registrar compra sin detalles	Aprobado	
CPA-6.5	Registrar compra sin orden asociada	Aprobado	
CPA-6.6	Modificar estado de orden de compra	Aprobado	

Nota: Elaboración propia.

2.3.5. Planificación y desarrollo del sprint 4

Planificación del sprint 4

Esta iteración consideró las funcionalidades de registro de movimiento de materia prima y generar orden de producción. El Sprint Backlog resultante mostrado en la Tabla 42 precisa todas las labores realizadas junto con una valoración del tiempo necesario para culminarlas.

Tabla 42

Planificación sprint 4

Código HU	Nombre	Tarea	Horas
	Preparación del sprint y cambios	Diseño de casos de prueba	4
		Cambios y corrección de errores	6
HU-4	Registro de movimiento de materia prima	Creación de salida de material	7
		Listado de usuarios solicitantes	3

		Listado de órdenes de producción	3
		Creación de entrada de material	7
		Listado de compras realizadas	3
		Creación de ajuste de material	7
		Listado de movimientos de material	5
		Información detallada del movimiento	6
		Filtrado de movimientos por su tipo	3
HU-8	Generar orden de producción.	Listado de órdenes de producción	4
		Información detallada de la orden de producción	5
		Creación de PDF con la orden de producción	6
		Ejecución pruebas unitarias	6
		Ejecución pruebas de aceptación	5
		Total	80

Nota: Elaboración propia.

Planificación de pruebas del sprint 4

Pruebas unitarias

Los casos de pruebas unitarias ideados se muestran en la Tabla 43 y Tabla 44 las cuales corresponden a las historias HU-4 y HU-8 respectivamente.

Tabla 43

Casos de pruebas unitarias HU-4 Registro de movimiento de materia prima

Id del Caso	Descripción	Condiciones de Entrada	Salida Esperada
CPU-7.1	Crear salida de material relacionada con orden de producción	Objeto de salida con detalles de materiales, cantidades y fecha	Salida registrada en la base de datos
CPU-7.2	Intentar crear salida de material sin especificar materiales	Objeto de salida sin detalles de materiales	Excepción indicando que se requieren materiales en la salida
CPU-7.3	Crear entrada de material relacionada con recepción de productos comprados	Objeto de entrada con detalles de materiales, cantidades y fecha	Entrada registrada en la base de datos
CPU-7.4	Intentar crear entrada de material sin especificar materiales	Objeto de entrada sin detalles de materiales	Excepción indicando que se requieren materiales en la entrada
CPU-7.5	Realizar modificación manual en el stock de materiales	Objeto de modificación con detalles de material, cantidad y tipo de ajuste	Stock de materiales ajustado en la base de datos
CPU-7.6	Intentar realizar modificación manual sin especificar material	Objeto de modificación sin detalles de material	Excepción indicando que se requiere un material en la modificación

CPU-7.7	Visualizar registro de entradas y salidas de materiales	Navegar a la lista de movimientos de materiales	Lista de movimientos de materiales visible con detalles técnicos de fecha, tipo de transacción y materiales involucrados
CPU-7.8	Visualizar registro de movimientos de materiales cuando no hay registros	Navegar a la lista de movimientos de materiales	Lista de movimientos registrados vacía
CPU-7.9	Realizar modificación en el stock de los materiales	Objeto de modificación con detalles de material, cantidad y tipo de ajuste	Stock de los materiales seleccionados ajustado en la base de datos

Nota: Elaboración propia.

Tabla 44

Casos de pruebas unitarias HU-8 Generar orden de producción

Id del Caso	Descripción	Entrada	Salida Esperada
CPU-8.1	Generar orden de producción al crear un pedido	Objeto de pedido con detalles válidos	Orden de producción creada y asociada al pedido
CPU-8.2	Verificar detalles de la orden de producción generada	Navegar a la orden de producción asociada al pedido	Detalles de la orden de producción visibles, sin información de precios
CPU-8.3	Verificar visibilidad de la orden de producción para el departamento de producción	Acceder a órdenes de producción desde producción	Consulta de órdenes de producción correcta
CPU-8.4	Verificar que la orden de producción no incluya información de precios	Objetos de orden de producción con sus datos	No existe el campo de precio en el objeto
CPU-8.5	Verificar que la orden de producción incluya detalles correctos del pedido	Objeto de orden de producción seleccionado	Detalles de la orden de producción coinciden con los detalles del pedido

Nota: Elaboración propia.

Pruebas de aceptación

Los casos para las pruebas de aceptación se exhiben en la Tabla 45 y Tabla 46. Como miembros para la ejecución estuvieron el encargado de producción y bodega junto al gerente.

Tabla 45

Casos de pruebas de aceptación HU-4 Registro de movimiento de materia prima

Id Caso	Descripción	Precondición	Pasos por seguir	Resultado esperado
CPA-7.1	Crear Salida de Material	Orden de producción, detalles de salida	<ol style="list-style-type: none"> 1. Acceder a nueva salida 2. Especificar materiales, solicitante y orden. 3. Guardar. 	Salida de material creada con detalles especificados.

CPA-7.2	Crear Entrada de Material	Productos recibidos y detalles de entrada	<ol style="list-style-type: none"> 1. Acceder a nueva entrada. 2. Especificar materiales y compra 3. Guardar. 	Entrada de material creada con detalles especificados.
CPA-7.3	Modificar Stock Manualmente	Tipo de ajuste, detalles de modificación	<ol style="list-style-type: none"> 1. Acceder a nuevo ajuste 2. Especificar materiales y tipo de ajuste. 3. Guardar. 	Ajuste manual realizado correctamente.
CPA-7.4	Visualizar Registro de Movimientos	N/A	<ol style="list-style-type: none"> 1. Acceder al registro de movimientos 	Registro mostrado con su tipo.
CPA-7.5	Crear movimiento sin detalles	Orden de producción, detalles de salida	<ol style="list-style-type: none"> 1. Acceder a nueva salida 2. Guardar sin especificar detalles 	Mensaje de error al intentar crear salida sin detalles.
CPA-7.6	Modificar stock con cantidad negativa	Ajuste manual, cantidad negativa	<ol style="list-style-type: none"> 1. Acceder a nuevo ajuste 2. Ingresar una cantidad negativa en stock 3. Guardar. 	Mensaje de error stock con cantidad negativa.

Nota: Elaboración propia.

Tabla 46

Casos de pruebas de aceptación HU-8 Generar orden de producción

Id Caso	Descripción	Precondición	Pasos por seguir	Resultado esperado
CPA-8.1	Generar orden de producción automáticamente	Pedido creado	<ol style="list-style-type: none"> 1. Verificar que, al crear un pedido, se genere automáticamente una orden de producción asociada. 	Orden de producción generada automáticamente
CPA-8.2	Ver Detalles de Orden de Producción	Orden de producción generada	<ol style="list-style-type: none"> 1. Acceder y ver detalles de la orden de producción generada. 	Detalles de la orden de producción mostrados correctamente.
CPA-8.3	Visualizar Orden de Producción en Departamento de Producción	N/A	<ol style="list-style-type: none"> 1. Verificar que la orden de producción sea visible para el departamento de producción. 	Orden de producción visible para producción.
CPA-8.4	Verificar Contenido de Orden de Producción	Detalles de la orden de producción	<ol style="list-style-type: none"> 1. Acceder a la orden de producción deseada y verificar sus detalles 	Contenido de la orden de producción y sin información de precios.

Nota: Elaboración propia.

Incremento 4

Registro de movimiento de materia prima

La Figura 32 muestra la interfaz para registrar entradas y salidas de materiales de bodega, permitiendo relacionarlas a órdenes de compra o producción según corresponda.

Figura 32

Pantalla de registro de movimiento de material

Nueva Salida de Material

Fecha del registro: Jan 30, 2024. Responsable: Rab Wolford

Información del solicitante: 1729408475 - Minni Birdseye. Nro. Cédula: 1729468475. Apellidos: Birdseye Hiscoke. Nro. empleado: 5. Nombres: Minni Shurlocke

Orden de producción: Nro. Orden: 11. Recepción: 2023-11-07. Estado: Recibido. Cédula Cliente: 4082280251. Apellidos: Barthot Rothon. Nombres: Aubree Stanislas

Materiales seleccionados:

Material	Cantidad	Opciones
Malla poliester Blanca Lino Stock actual: 4.15 Kg	3 Kg	Remove
Malla Pol Verde Lino Stock actual: 17.2 Kg/m2	3.5 Kg/m2	Remove

Observaciones:

Registrar Salida

Nota: Elaboración propia.

Generar orden de producción

La Figura 33 presenta la pantalla para visualizar las órdenes de producción generadas automáticamente a partir de la creación de un pedido de un cliente en el sistema.

Figura 33

Pantalla de orden de producción

Orden de Producción

Nro: ORD-11. Fecha Recepción / Entrega: 2023-11-07 / 2023-12-27. Vendedor: Padgett Walhedd

Información del cliente: Nro. Cédula: 4082280251. Apellidos: Barthot Rothon. Nombres: Aubree Stanislas. RUC: 5105104943. Email: srothond@newsvine.com. Celular: 8329376206

Productos seleccionados:

Producto	Cantidad	Color	Talla	Diseño
Antideslizantes bebe Antideslizantes	2 Doc	Amarillo Fosforescente	0-3	
Babucha Algodon-peinado Deportivas	8 Par	Azul	9-12	
Casual Economica Invisibles	1 Doc		10-12	

Observaciones: Todas la medias deben llevar una franja de color rojo seguida de una azul en la punta.

Comprobante

Nota: Elaboración propia.

Figura 34

Pedido generado en PDF



Orden de producción

Barrio los Pinos de Eugenio Espejo, 100206
(+593) 0997179084
email@example.com
info@example.com
fabrica-de-medias-neltex.negocio.site

Información del cliente

Lattin Northbridge Ozzy Waylon

Cédula: 5864353469
Telf/Cel: 9135883863
Email: wnorthbridge7@shutterfly.com

Pedido Nro: 9

Fecha de recepción: 2023-10-20
Fecha de entrega: 2023-10-30

#	Producto	Categoría	Color	Talla	Cantidad
2	Zapatilla Algodon-peinado	Locuras	Crimson	5	4 Doc
3	Zapatilla Algodon-peinado	Locuras	Purple	2	6 Doc
4	Canillera Algodon-normal	Tobilleras	Mauv	5	4 Doc
5	Invisible Algodon-peinado	Tobilleras	Puce	7	8 Doc
6	Casual Locura	Invisibles	Turquoise	8	2 Doc
7	Canillera Algodon-normal	Tobilleras	Fuscia	1	9 Doc

Observaciones:

Cras non velit nec nisi vulputate nonummy. Maecenas.

Nota: Elaboración propia.

Ejecución de pruebas del sprint 4

Ejecución de pruebas unitarias

La realización de los casos de prueba unitarias inicialmente planteados, se exhibe en la Figura 35 y Figura 36. También se crearon pruebas adicionales con el objetivo de incrementar el porcentaje de cobertura del código.

Figura 35

Ejecución de pruebas unitarias en el backend sprint 4

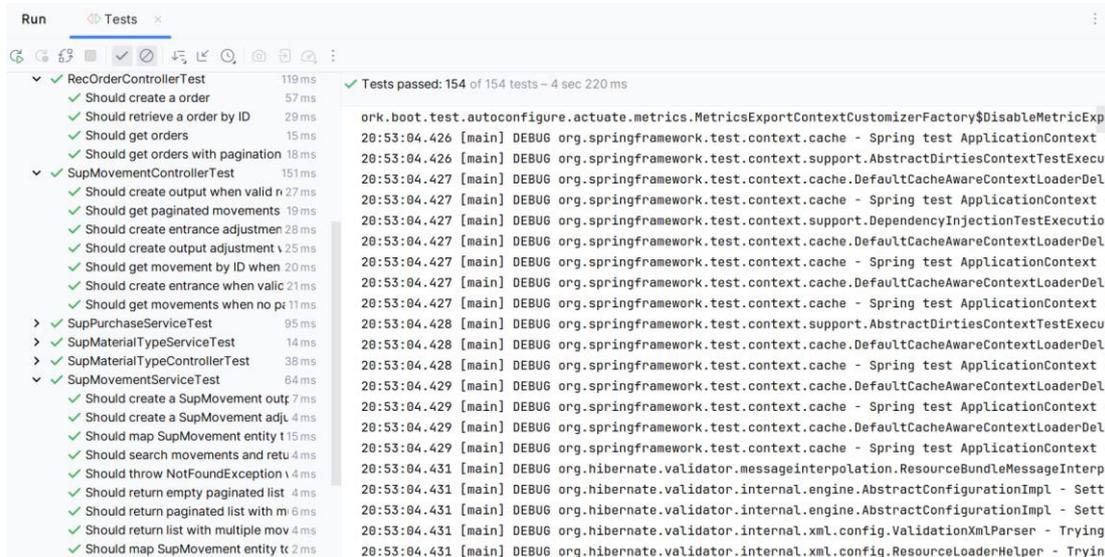


Figura 36

Ejecución de pruebas unitarias en el frontend sprint 4



Ejecución de pruebas de aceptación

La ejecución de las pruebas de aceptación se efectuó con los usuarios finales, sometiéndolos a los casos previamente concebidos. Para obtener un panorama más detallado de la ejecución, se pueden consultar el Anexo E. Los resultados alcanzados se muestran en la Tabla 47.

Tabla 47

Ejecución de pruebas de aceptación sprint 4

Id Caso	Resultado esperado	Validación	Observaciones
CPA-7.1	Crear Salida de Material	Aprobado	
CPA-7.2	Crear Entrada de Material	Aprobado	
CPA-7.3	Modificar Stock Manualmente	Aprobado	Mostrar el stock actual del material
CPA-7.4	Visualizar Registro de Movimientos	Aprobado	
CPA-7.5	Crear movimiento sin detalles	Aprobado	
CPA-7.6	Modificar stock con cantidad negativa	Aprobado	
CPA-8.1	Generar orden de producción automáticamente	Aprobado	Generar orden de producción en PDF
CPA-8.2	Ver Detalles de Orden de Producción	Aprobado	
CPA-8.3	Visualizar Orden de Producción en Departamento de Producción	Aprobado	
CPA-8.4	Verificar Contenido de Orden de Producción	Aprobado	

Nota: Elaboración propia.

2.4. Cierre del proyecto

Tras concluir el desarrollo de la aplicación y ejecutar todas las pruebas planificadas, se logra un software plenamente funcional que satisface las especificaciones del usuario final y está listo para su despliegue.

En esta sección, se expone la implementación de la aplicación, la cual, como se mencionó previamente, se desplegará utilizando Amazon Web Services (AWS). Asimismo, se presentará el porcentaje de cobertura de pruebas alcanzado en la ejecución de los casos de pruebas unitarias.

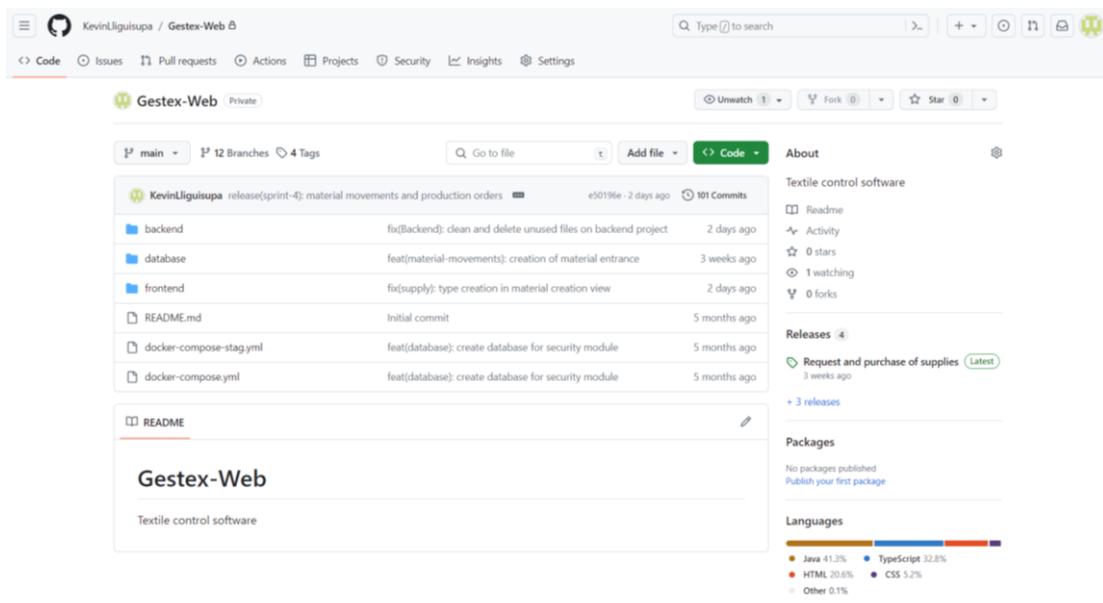
2.4.1. Despliegue de la aplicación

Repositorio de código

El código de toda la aplicación backend, frontend e incluso el diagrama de base de datos, se encuentra alojado en el repositorio GitHub, en la que cada una de las ramas de “feature” representa al desarrollo de una historia de usuario, el repositorio de muestra en la Figura 37.

Figura 37

Repositorio de la aplicación



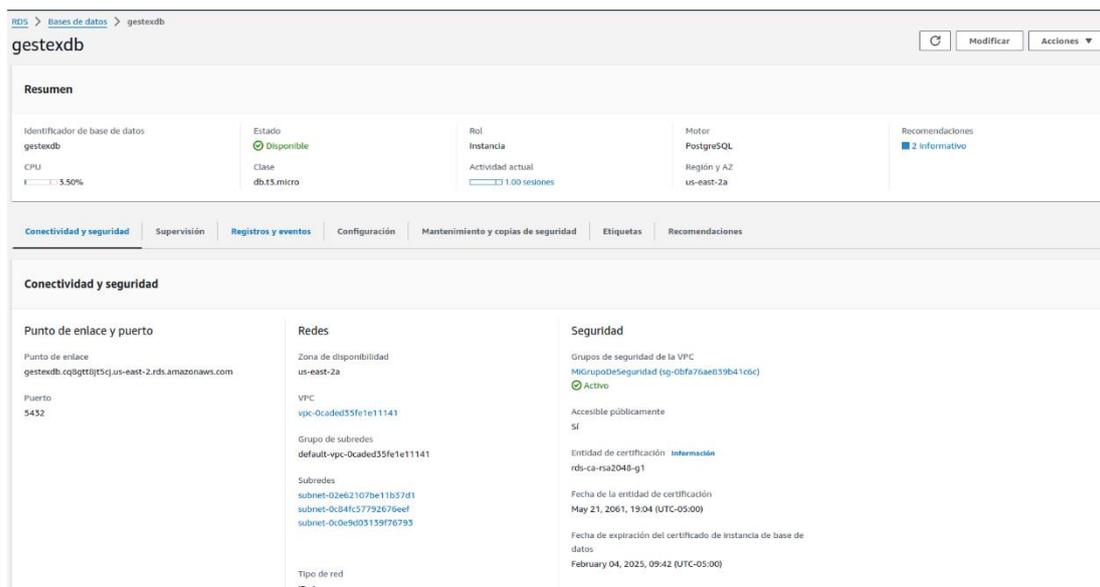
Nota: Elaboración propia.

Funcionamiento en Amazon

En la Figura 38 se puede ver el despliegue de la base de datos PostgreSQL, el cual se realizó en el servicio de bases de datos relacionales (RDS) de Amazon Web Services.

Figura 38

Despliegue de base de datos



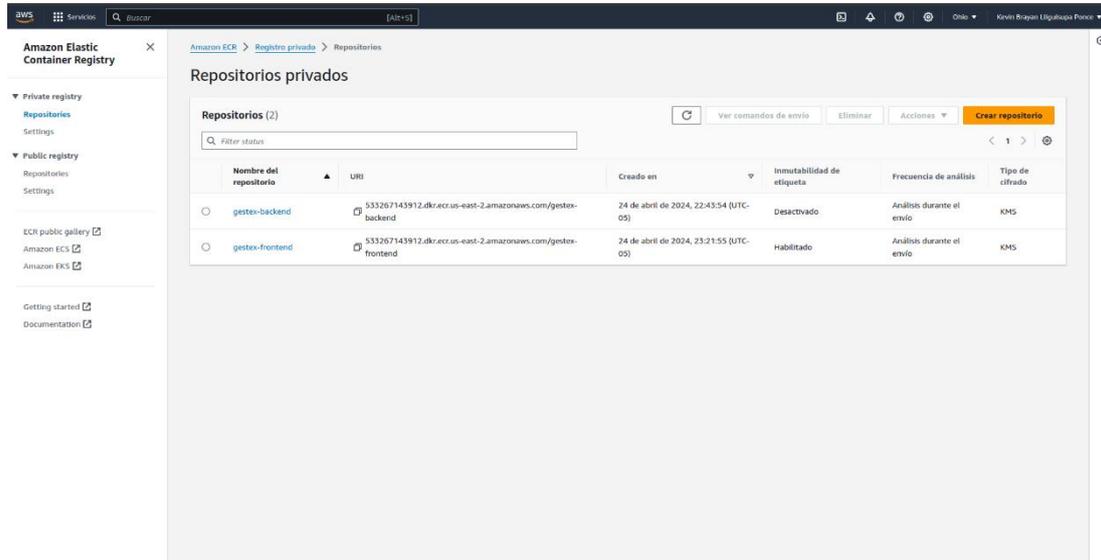
Nota: Elaboración propia.

El despliegue de la aplicación se realizó usando imágenes Docker previamente usadas en la etapa de staging, las cuales habían sido preparadas con una configuración muy similar a la cual sería usada en producción.

Cada una de las imágenes Docker de la aplicación fueron almacenadas en Amazon Elastic Container Registry (ECR), las cuales posteriormente serían administradas por Amazon Elastic Container Service (ECS), de forma que la primera se encarga de almacenar las imágenes y la segunda se funciona como un orquestador de contenedores que permite que tanto backend y frontend se comuniquen correctamente. Ambos pueden ver en la Figura 39 y Figura 40.

Figura 39

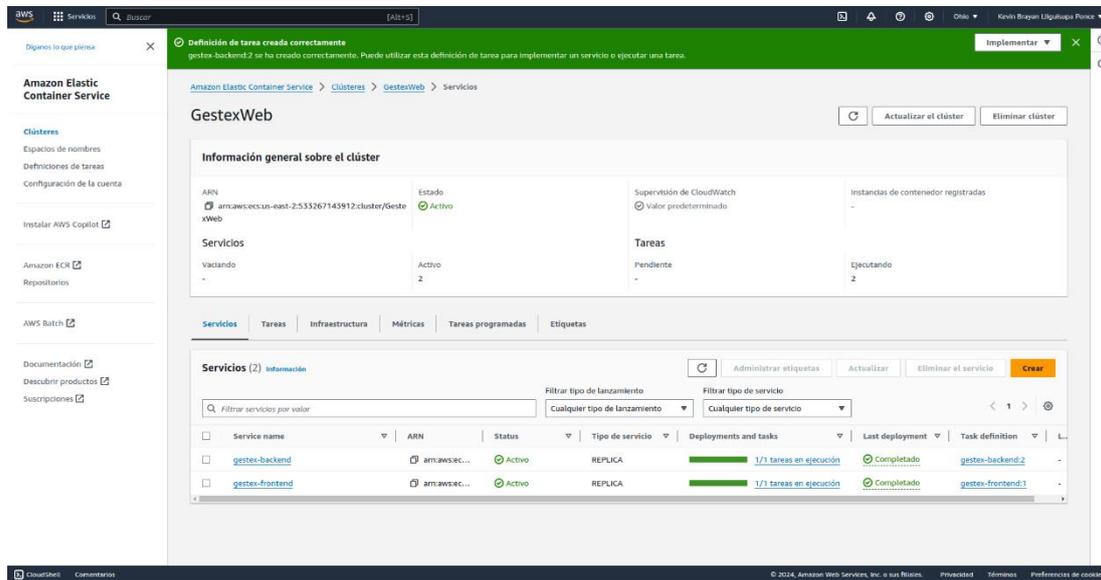
Imágenes Docker de la aplicación en Amazon ECR



Nota: Elaboración propia.

Figura 40

Funcionamiento de la aplicación en Amazon ECS



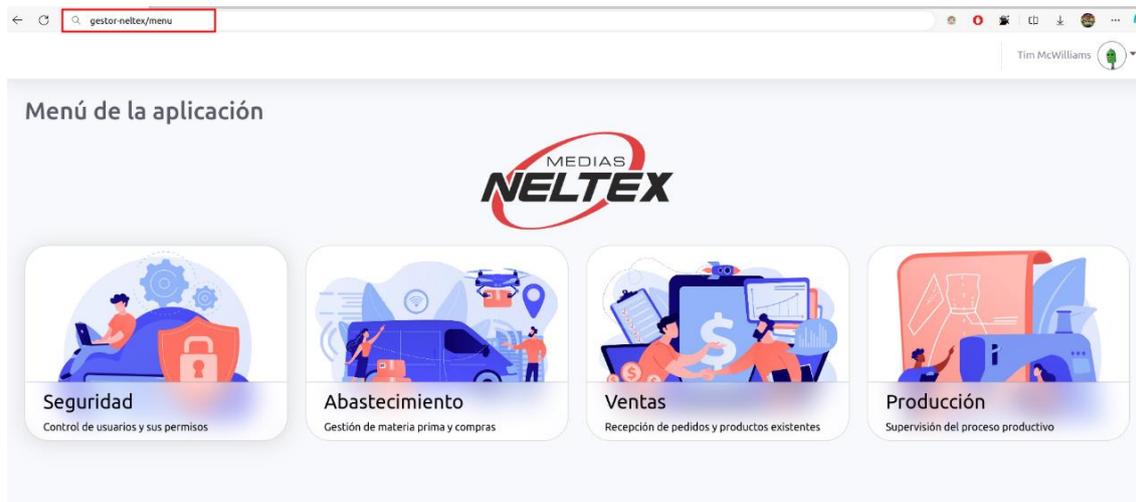
Nota: Elaboración propia.

Aplicación en producción

En la Figura 41 se puede ver la pantalla de menú, en la que se ve que ya se está haciendo uso de un nombre de dominio y un certificado SSL.

Figura 41

Aplicación en producción



Nota: Elaboración propia.

2.4.2. Cobertura de pruebas

Una vez ejecutado los casos de prueba planteados y los casos adicionales, se ha llegado a cumplir con el objetivo del plan de pruebas, el cual era un porcentaje de cobertura de los métodos mínimo del 60%.

En la Figura 42 se ve el porcentaje de cobertura del código en la aplicación creada usando Spring Boot (Backend), llegando a un porcentaje de cobertura en métodos del 63%.

Figura 42

Cobertura de código en el backend

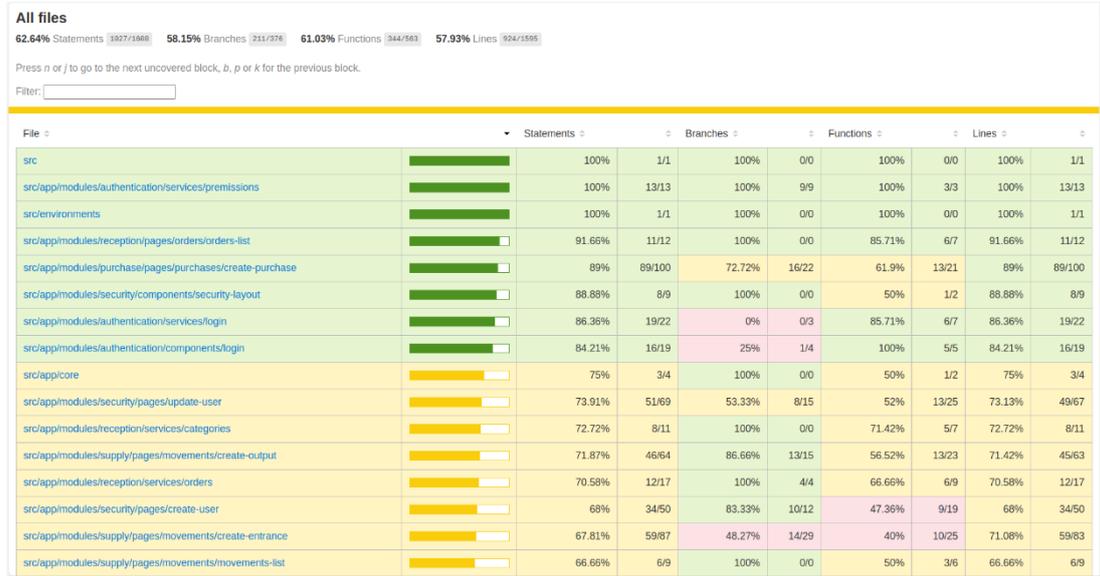
Element	Class, %	Method, %	Line, %
com	81% (109/133)	63% (573/908)	60% (1177/1961)
restapi	81% (109/133)	63% (573/908)	60% (1177/1961)
gestexbackend	81% (109/133)	63% (573/908)	60% (1177/1961)
GestexBackendApplication	100% (2/2)	75% (3/4)	90% (9/10)
supply	74% (40/54)	53% (199/370)	50% (424/835)
presentation	100% (11/11)	52% (26/50)	49% (66/134)
persistence	81% (9/11)	52% (42/80)	60% (63/105)
bussiness	62% (20/32)	54% (131/240)	49% (295/596)
security	91% (22/24)	75% (108/144)	62% (232/374)
presentation	88% (8/9)	64% (20/31)	43% (77/178)
persistence	100% (4/4)	72% (21/29)	73% (28/38)
business	90% (10/11)	79% (67/84)	80% (127/158)
reception	83% (31/37)	73% (214/292)	69% (385/565)
presentation	100% (4/4)	80% (16/20)	80% (38/47)
persistence	90% (9/10)	76% (59/77)	79% (79/99)
business	78% (18/23)	71% (139/195)	66% (278/419)
production	100% (5/5)	64% (25/39)	81% (60/74)
presentation	100% (1/1)	100% (3/3)	100% (8/8)
bussiness	100% (4/4)	61% (22/36)	78% (52/66)
control	50% (2/4)	10% (3/30)	25% (9/36)
common	100% (7/7)	72% (21/29)	71% (46/67)
service	100% (1/1)	100% (4/4)	100% (19/19)
handler	100% (1/1)	100% (3/3)	100% (14/14)
exception	100% (3/3)	100% (3/3)	100% (3/3)
dto	100% (1/1)	100% (10/10)	100% (10/10)
controller	100% (1/1)	11% (1/9)	9% (2/21)

Nota: Elaboración propia.

La Figura 43 muestra el resumen de cobertura de código en la aplicación creada usando Angular (frontend), llegando a un total de 61.03% cobertura en funciones.

Figura 43

Cobertura de código en el frontend



Nota: Elaboración propia.

CAPÍTULO 3

3.1. Validación de la Adecuación Funcional según ISO/IEC 25023

En el marco del presente trabajo, se validó la adecuación funcional del software Gestor Neltex, empleando una metodología estructurada basada en la Norma ISO/IEC 25023 y con la colaboración activa del Product Owner del proyecto. Los objetivos principales fueron evaluar la presencia de las funciones principales definidas en las historias de usuario, así como identificar cualquier funcionalidad faltante o errónea que pudiera afectar su adecuación funcional.

La característica Adecuación Funcional de la Norma ISO/IEC 25023 hace referencia al grado en el que un producto o sistema satisface las necesidades establecidas e implícitas bajo ciertas condiciones (ISO/IEC 25023, 2016). A continuación, se muestra la evaluación del producto de software resultante bajo los criterios que establece esta característica.

3.1.1. Evaluación de la Completitud Funcional

Esta subcaracterística tiene como propósito evaluar cuán completa es la implementación con respecto a los requerimientos.

Evaluación

Para su evaluación, se utilizó una checklist diseñado específicamente para valorar la subcaracterística de Completitud Funcional del software, en conformidad con la Norma ISO/IEC 25023. Dicha checklist se puede ver en el Anexo F.

La evaluación consistió en revisar la presencia de las funciones principales del software, correspondientes a las historias de usuario, en el producto final. El Product Owner proporcionó retroalimentación y marcó las funciones como presentes o faltantes, según lo observado durante la revisión detallada. Los resultados de esta valoración se pueden ver en la Tabla 48.

Se emplearon herramientas de colaboración online para compartir la checklist con el Product Owner y registrar los resultados de manera estructurada y trazable.

Tabla 48

Evaluación de completitud funcional

Nro.	Código HU	Descripción	Presente	Faltante
1	HU-1	Como administrador, asigno roles, controlo el acceso y acciones de empleados.	X	
2	HU-2	Como bodeguero, creo, veo, modifico y elimino registros de materiales.	X	
3	HU-3	Como bodeguero, creo y veo órdenes de compra para adquirir materiales.	X	
4	HU-4	Como bodeguero, registro ingreso/salida de materia prima asociado a órdenes de producción o compras.	X	
5	HU-5	Como encargado de compras, registro compras de órdenes recibidas desde bodega.	X	
6	HU-6	Como vendedor o gerente, gestiono productos para producción, detalles de materiales y precios.	X	
7	HU-7	Como vendedor, registro detalles del cliente registro información del pedido y genero comprobante.	X	
8	HU-8	Como vendedor, creo una orden de producción automáticamente tras crear un pedido.	X	

Nota: Elaboración propia.**Resultado de la evaluación de completitud funcional**

- A: Nro. funciones perdidas = 0
- B: Nro. funciones especificadas = 8
- X: Completitud funcional = $1 - A/B \rightarrow 1 - 0/8 \rightarrow 1$

3.1.2. Evaluación de la Corrección Funcional

En esta sección, se evalúa el grado en el que las funciones proporcionadas por el software producen resultados correctos, centrándose en la precisión y satisfacción de los requerimientos del usuario.

Evaluación

Para su evaluación, se hizo uso de los criterios de aceptación definidos en las historias de usuario, los cuales se usaron para ejecutar las pruebas de aceptación mostradas en el capítulo anterior. Estos criterios permitieron evaluar si los resultados obtenidos por el software son precisos y satisfacen los requerimientos de los usuarios.

Los criterios marcados como "Incorrecto" indican que el software no logro superar la prueba de aceptación para este requerimiento específico, por lo que fue necesario realizar correcciones o serán necesarias a futuro.

La Tabla 49 presenta los resultados de la evaluación de corrección funcional, indicando si cada criterio de aceptación fue cumplido correctamente o no.

Tabla 49

Evaluación de la corrección funcional

Nro.	Código HU	Criterios	Correcto	Incorrecto
1	HU-1	Debe permitir el acceso a la aplicación solamente cuando las credenciales estén correctas y el usuario este activo.	X	
		Debo poder ver una lista de todos los empleados registrados en la aplicación.	X	
		Para cada empleado, debo poder asignar, eliminar o modificar su rol.	X	
		Los roles disponibles deben incluir, "Gerente", "Adquisición", "Vendedor" y "Bodeguero" con sus respectivos permisos.		X
2	HU-2	Debo poder agregar una nueva materia prima al inventario, especificando su nombre, cantidad inicial en stock, unidad de medida, proveedor, etc.	X	X
		Debo poder ver una lista de todas las materias primas existentes en el inventario, con su información.	X	
		Debo poder editar la información de una materia prima existente.	X	
		Debo poder cambiar el estado de una materia prima a "Activa" o "Inactiva".	X	
3	HU-3	Debo poder ver una lista de todas las órdenes de compra existentes, con detalles como número de orden, fecha de creación y estado (por ejemplo, pendiente, aprobada, entregada).	X	
		Debo poder crear una nueva orden de compra, especificando los materiales necesarios, las cantidades, los proveedores potenciales, etc.	X	
4	HU-4	Debo poder crear una salida de material relacionada con una orden de producción, especificando los materiales utilizados, las cantidades y la fecha de salida.	X	
		Debo poder crear una entrada de material relacionada con la recepción de productos comprados, especificando los materiales recibidos, las cantidades y la fecha.	X	
		Debo poder realizar modificaciones manuales en el stock de materiales cuando sea necesario, como ajustes por daños, pérdidas o cambios en las cantidades disponibles.	X	
		Debo poder visualizar un registro de todas las entradas y salidas de materiales, incluyendo detalles como fecha, tipo de transacción y materiales involucrados.	X	

		Debo poder ver una lista de todas las órdenes de compra que han sido creadas por bodega y que están pendientes.	X	
5	HU-5	Debo poder seleccionar una orden de compra y registrar la compra correspondiente, especificando los detalles de la transacción, como la fecha de compra, el proveedor y los materiales o productos adquiridos.	X	
		La aplicación debe mantener un registro de todas las compras realizadas.	X	
		Debo poder crear un nuevo producto, especificando su nombre, precio, estado, peso por docena y asociando los materiales necesarios, sus cantidades y precios unitarios.	X	
6	HU-6	Debo poder ver una lista de todos los productos, con su información, incluyendo la lista de materiales necesarios, cantidades requeridas y precios unitarios.	X	
		Debo poder editar la información de un producto existente.	X	
		Debo poder cambiar el estado de un producto a "Disponible" o "No disponible".		X
		La aplicación debe realizar cálculos automáticos para mostrar el costo total de los materiales utilizados en la producción de un producto.	X	
		Debo poder recibir un nuevo pedido de un cliente, información del cliente, como nombre, dirección y detalles de contacto.	X	
7	HU-7	Debo poder agregar productos al pedido, especificando el nombre del producto, cantidad, color, y otros detalles relevantes.	X	
		La aplicación debe calcular automáticamente el total del pedido en función de los productos y sus cantidades.	X	
		Debo poder guardar el pedido y generar un comprobante del pedido que incluya los detalles del cliente, los productos solicitados, el total y la fecha del pedido.	X	
		La aplicación debe permitir la visualización de los pedidos recibidos.	X	
		Una vez que el pedido sea creado, la aplicación debe generar automáticamente una orden de producción asociada.	X	
8	HU-8	La orden de producción debe incluir detalles como los productos solicitados, las cantidades, colores y otros detalles relevantes, de acuerdo con el pedido.	X	
		La orden de producción debe ser visible para el departamento de producción.	X	
		La orden de producción solo tendrá la información necesaria, evitando los precios.	X	

Nota: Elaboración propia.

Resultado de la evaluación de corrección funcional

- A: Nro. de funciones incorrectas = 3
- B: Nro. de funciones consideradas = 31
- X: Corrección funcional = $1 - A/B \rightarrow 1 - 3/31 \rightarrow 0.90$

3.1.3. Pertenencia funcional

En la evaluación de esta subcaracterística, se tomó en cuenta solamente la métrica de "Pertenencia Funcional del Objetivo de Uso", la cual evalúa en qué medida el software cumple el propósito y necesidades previstas.

Evaluación

Se definió el siguiente objetivo de uso del software: "Gestión integral de inventario de materia prima y recepción de pedidos de fabricación"

Para la evaluación de esta métrica, se utilizaron las historias de usuario que cumplían con el objetivo de uso, mostradas en la Tabla 50, y se evaluó cuáles de las funciones correspondientes fueron cumplidas satisfactoriamente según las pruebas de aceptación.

Tabla 50

Identificación de las funciones requeridas para el caso de uso.

Nro.	Código HU	Descripción
1	HU-2	Como bodeguero, creo, veo, modifico y elimino registros de materiales.
2	HU-3	Como bodeguero, creo y veo órdenes de compra para adquirir materiales.
3	HU-4	Como bodeguero, registro ingreso/salida de materia prima asociado a órdenes de producción o compras.
4	HU-5	Como encargado de compras, registro compras de órdenes recibidas desde bodega.
5	HU-6	Como vendedor o gerente, gestiono productos para producción, detalles de materiales y precios.
6	HU-7	Como vendedor, registro detalles del cliente registro información del pedido y genero comprobante.
7	HU-8	Como vendedor, creo una orden de producción automáticamente tras crear un pedido.

Nota: Elaboración propia.

Se determinaron como funciones que cumplieron parcialmente aquellas que tuvieron fallos al pasar sus pruebas de aceptación. Estos casos se pueden ver en la Tabla 51.

Tabla 51

Grado de cumplimiento de las funciones entregadas

Nro.	Código HU	Cumplimiento
1	HU-2	Cumple parcialmente
2	HU-3	Cumple
3	HU-4	Cumple
4	HU-5	Cumple
5	HU-6	Cumple parcialmente
6	HU-7	Cumple
7	HU-8	Cumple

Nota: Elaboración propia.

Resultado de la evaluación de completitud funcional

- A: Nro. de funciones proporcionadas que cumplen con el objetivo de uso = 5
- B: Nro. de funciones requeridas para el objetivo de uso = 7
- X: Completitud funcional = $A/B \rightarrow 5/7 \rightarrow 0.71$

3.1.4. Evaluación de adecuación funcional

Una vez evaluadas sus tres subcaracterísticas es posible evaluar la adecuación funcional del software obteniendo un promedio en base a la completitud funcional, corrección funcional y pertenencia funcional el detalle se muestra en la Tabla 52.

Tabla 52

Evaluación de la adecuación funcional

Característica	Valor
Completitud funcional	1.00
Corrección funcional	0.90
Pertenencia funcional	0.71
Total	0.87

Nota: Elaboración propia.

3.1.5. Interpretación de los resultados

Según la evaluación presentada, se pueden hacer las siguientes interpretaciones sobre la adecuación funcional del software:

1. Completitud funcional: Se obtuvo un valor de 1.00, lo que indica que todas las funciones especificadas en los requerimientos (historias de usuario) fueron implementadas en el producto final, es decir, no falta ninguna función.
2. Corrección funcional: Se obtuvo un valor de 0.90, lo que significa que el 90% de las funciones implementadas producen resultados correctos de acuerdo con los criterios de aceptación. Hubo 3 funciones que presentaron resultados incorrectos sobre un total de 31 funciones evaluadas.
3. Pertenencia funcional: Se obtuvo un valor de 0.71, indicando que gran parte de las funciones que cumplen con el objetivo de uso del software son correctas, sin embargo, existen funciones que presentan fallos.
4. Adecuación funcional: Considerando el promedio de las 3 subcaracterísticas, se obtuvo un valor de 0.87. Lo que significa que en términos generales el software presenta una buena adecuación funcional, ya que cumple en gran medida con los requerimientos funcionales, entrega funciones completas y pertinentes, y la mayoría de sus resultados son correctos.

El análisis muestra que el software desarrollado tiene una alta adecuación funcional, cumpliendo de gran manera con entregar las funciones esperadas por los usuarios de manera precisa. En el futuro es necesario corregir las 3 funciones que presentaron problemas en sus resultados para mejorar la corrección funcional.

DISCUSIÓN

Análisis de hallazgos

Los resultados alcanzados mediante la aplicación de la metodología propuesta para asegurar la calidad de software en proyectos ágiles demuestran su efectividad. En primer lugar, se logró una cobertura de pruebas satisfactoria, con un 63% de cobertura de métodos en el backend y un 61.03% en el frontend, superando el objetivo planteado del 60% (ver sección 2.4.2 Cobertura de pruebas).

Además, se obtuvieron muy buenos resultados en la evaluación de la Adecuación Funcional según la norma ISO/IEC 25010 (ver sección 3.1.4 Evaluación de adecuación funcional). Se alcanzó una Completitud Funcional de (1.00), reflejando que no faltó ningún requerimiento por implementar. La Corrección Funcional indica que el 90% de las funciones produjeron resultados correctos. La Pertenencia Funcional de 0.71 demostró que, aunque algunas funciones presentaron fallas menores, la gran mayoría fueron pertinentes y alineadas al objetivo de uso del software. Esto dio un promedio general de 0.87 para la Adecuación Funcional general.

Estos hallazgos concuerdan con la literatura previa que resalta los beneficios de adoptar un enfoque ágil centrado en la calidad mostrados por Arcos-Medina & Mauricio, (2019) y Umar, (2020). Adicionalmente, los resultados sobre la importancia de combinar prácticas ágiles con estrategias de medición y evaluación de la calidad son convergentes con Ushakova et al., (2021). Sin embargo, el presente estudio propone un modelo específico detallado con técnicas, métricas y estándares concretos como la ISO 25010 aplicados para gestionar la calidad en proyectos ágiles.

Implicaciones teóricas y prácticas

Desde la teoría, este estudio extiende los modelos existentes no solo integrando diferentes aspectos, sino definiendo un marco de trabajo estructurado con actividades, roles, artefactos y mecanismos de evaluación clave para materializar los principios generales de

calidad del software en un contexto ágil. Esta perspectiva operativa y el uso de estándares reconocidos como las normas ISO complementan los aportes teóricos previos.

En la práctica, el modelo brinda una guía accionable para incorporar la gestión de la calidad desde las primeras etapas de un proyecto ágil, alineando las actividades con los criterios de calidad establecidos en estándares internacionales. Esto facilitaría su adopción en empresas al brindar un enfoque estructurado que combina agilidad con prácticas de calidad ampliamente reconocidas.

Limitaciones

Aunque se aplicaron buenas prácticas y estándares reconocidos, una limitación fue el enfoque en proyectos de software a la medida. Se requiere análisis adicional para determinar la aplicabilidad total del modelo a otros contextos como desarrollo de productos empaquetados.

Investigación futura

Además de las líneas sugeridas previamente, convendría realizar estudios comparativos que cuantifiquen los beneficios del modelo propuesto frente a otros enfoques para la gestión de la calidad en proyectos ágiles, considerando distintas características de calidad relevantes además de la funcionalidad, como usabilidad, eficiencia, velocidad, etc. Asimismo, sería valioso explorar mecanismos para personalizar y adaptar aún más el modelo a diferentes contextos, dominios y estándares de calidad según las necesidades específicas.

CONCLUSIONES

Se estableció un marco teórico sólido, que incluyó conceptos clave como aseguramiento de la calidad, tipos de pruebas, metodologías ágiles y estándares de calidad, Este marco fue fundamental para abordar el proyecto de manera efectiva, sentando las bases teóricas y metodológicas necesarias para diseñar y ejecutar un plan integral de pruebas durante el desarrollo de la aplicación web.

La aplicación web se desarrolló satisfactoriamente mediante la implementación de la metodología ágil Scrum y el uso de las tecnologías planteadas. Esta solución automatiza los procesos de control de materia prima y recepción de pedidos, brindando una aplicación escalable diseñada para mejorar estas operaciones en la empresa. Además de agilizar las operaciones diarias, el software desarrollado sienta las bases para un control más preciso y eficiente de los inventarios y pedidos, lo que impacta positivamente en la productividad y competitividad del negocio.

Al adoptar la metodología ágil Scrum, se elaboró y ejecutó de manera iterativa e incremental un plan de pruebas en cada Sprint del ciclo de desarrollo. Esta estrategia permitió identificar, reportar y corregir defectos y no conformidades de manera temprana, garantizando que el producto de software final cumpliera con todos los requerimientos funcionales y no funcionales establecidos para estar listo para su despliegue y operación exitosa.

Se validó el producto final de forma rigurosa utilizando los criterios y el método de la norma internacional ISO/IEC 25023, centrada especialmente en la característica de adecuación funcional. Los resultados cuantitativos obtenidos demuestran de manera concluyente que el enfoque sistemático de aseguramiento de la calidad implementado ha generado un software confiable, robusto y eficiente.

RECOMENDACIONES

Además de las pruebas unitarias y de aceptación, se recomienda llevar a cabo pruebas de carga y rendimiento para evaluar la capacidad del sistema ante cargas típicas y pico. Esto asegurará un rendimiento óptimo, incluso bajo alta demanda, garantizando la robustez y confiabilidad del software entregado a los usuarios finales.

Para ampliar la utilidad de los resultados obtenidos, se sugiere realizar análisis adicionales sobre la aplicabilidad del modelo propuesto en diferentes contextos. Esto incluye investigar cómo el enfoque de aseguramiento de calidad puede adaptarse a proyectos de desarrollo de productos empaquetados u otros entornos específicos. Al explorar nuevas aplicaciones y escenarios, se podrá validar la efectividad y la versatilidad del enfoque propuesto, ampliando así su impacto en la industria del desarrollo de software.

Para una más completa evaluación de la calidad, se recomienda investigar y medir métricas relacionadas con la usabilidad y eficiencia del software. Estas métricas pueden incluir la facilidad de uso, la velocidad de respuesta y el consumo de recursos del sistema. Al comprender mejor la experiencia del usuario y la eficiencia operativa del software, será posible identificar áreas de mejora y optimizar la satisfacción del usuario final.

Se recomienda investigar y desarrollar mejoras para la metodología aplicada, usando diferentes estándares de calidad (como CMMI, SPICE o normas específicas de dominio), cambiando las técnicas de pruebas (como rendimiento, carga, seguridad, usabilidad, etc.) para una evaluación más integral, explorando su combinación con otras metodologías ágiles además de Scrum (XP, Kanban, Lean, etc.), e identificando las mejores prácticas posibles.

BIBLIOGRAFÍA

- Abbas, R., Sultan, Z., & Bhatti, S. N. (2017). Comparative Analysis of Automated Load Testing Tools: Apache JMeter, Microsoft Visual Studio (TFS), LoadRunner, Siege. *International Conference on Communication Technologies (ComTech)*.
- Acosta, N. J., Espinel, L. A., & García, J. L. (2017). Estándares para la calidad de software. *Tecnología, Investigación y Academia TIA*. <http://revistas.udistrital.edu.co/ojs/index.php/tia/issue/archive>
- Ahmad, J., Abu ul, H., Naqv, T., & Mubeen, T. (2019). A review on software testing and its methodology. *I-Manager's Journal on Software Engineering*, 13, 32–38.
- Aizprua, S., Ortega Amable, & Von Chong, L. (2019). Calidad del Software una Perspectiva Continua Software. *CENTROS*. <http://miar.ub.edu/issn/2304->
- Arcos-Medina, G., & Mauricio, D. (2019). Aspects of software quality applied to the process of agile software development: a systematic literature review. In *International Journal of System Assurance Engineering and Management* (Vol. 10, Issue 5, pp. 867–897). Springer. <https://doi.org/10.1007/s13198-019-00840-7>
- Arias, D. I., Domínguez, J. F., & Milián, I. (2018). Actividades para el aseguramiento de la calidad en la administración de la configuración del software. *TLATEMOANI*, 27.
- Balseca, E. A. (2014). *Evaluación de calidad de productos software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000*. Escuela Politécnica Nacional.
- Baumgartner, M., Klöckl, M., Mastnak, C., Pichler, H., Seidl, R., & Tanczos, S. (2021). Agile Test Management, Methods, and Techniques. In M. Baumgartner, M. Klöckl, C. Mastnak, H. Pichler, R. Seidl, & S. Tanczos (Eds.), *Agile Testing: The Agile Way to Quality* (pp. 113–141). Springer International Publishing. https://doi.org/10.1007/978-3-030-73209-7_5
- Bechtold, S., Brannen, S., Link, J., Merdes, M., & Philipp, M. (2023). *JUnit 5 User Guide*. <https://junit.org/junit5/docs/current/user-guide/>
- Bele, P. S. B., Harinkhede, A. A., & Bhatti, V. M. (2022). Software Testing Using White Box. *International Research Journal of Innovations in Engineering and Technology (IRJIET)*, 6(10), 142–144. <https://doi.org/10.47001/IRJIET/2022.610028>
- Bhargava, S., & Jain, P. Bhatewara. (2018). Testing connect automated technologies. *I-Manager's Journal on Software Engineering*, 13.
- Bourque, P., & Fairley, R. E. (2014). *Guide to the Software Engineering Body of Knowledge* (Version 3.0). IEEE Computer Society.
- Brühl, V. (2022). Agile methods in the German banking sector: some evidence on expectations, experiences and success factors. *Journal of Business Economics*, 92(8), 1337–1372. <https://doi.org/10.1007/s11573-022-01102-y>

- Byrne, D., Tuite, A., & Organ, J. (2022). A Study of the Organisational Behaviour of Software Test Engineers, Contributing to the Digital Transformation of Banks in the Irish Financial Sector. *IFAC-PapersOnLine*, 55(39), 259–264. <https://doi.org/10.1016/j.ifacol.2022.12.031>
- Carrizo, D., & Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. In *Revista chilena de ingeniería* (Vol. 26, Issue 1).
- Castro, Y. V., Solarte, G., & Muñoz, L. E. (2019). Planning, Management and Control of Software Quality. *Scientia Et Technica*, 24.
- Crispin, L., & Gregory, J. (2008). *Agile Testing* (1st ed.). Addison-Wesley Professional.
- Crispin, L., & Gregory, J. (2009). *Agile Testing*. www.XProgramming.com
- Espejo, A., Bayona, S., & Pastor, C. (2016). Aseguramiento de la calidad en el proceso de desarrollo de software utilizando CMMI, TSP y PSP. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, 2016(20), 62–77. <https://doi.org/10.17013/risti.20.62-77>
- Gamido, H. V., & Gamido, M. V. (2019). Comparative review of the features of automated software testing tools. *International Journal of Electrical and Computer Engineering*, 9(5), 4473–4478. <https://doi.org/10.11591/ijece.v9i5.pp4473-4478>
- Graham, D., Veenendaal, E. Van, & Black, R. (2019). *Foundations of Software Testing: ISTQB Certification*. Cengage, Andover, 2019.
- Guerrero, G., Guevara, A., Quiña-Mera, J. A., Guevara-Vega, C. P., & García-Santillán, I. (2022). Software Project Management Integrating CMMI-DEV and SCRUM. In M. Botto-Tobar, S. Montes León, P. Torres-Carrión, M. Zambrano Vizuetete, & B. Durakovic (Eds.), *Applied Technologies* (pp. 538–551). Springer International Publishing.
- Guevara-Vega, C. P., Cárdenas-Hernández, W. A., Landeta, P. A., Rea-Peñañiel, X. M., & Quiña-Mera, J. A. (2022). Software Test Management to Improve Software Product Quality. In M. Botto-Tobar, S. Montes León, P. Torres-Carrión, M. Zambrano Vizuetete, & B. Durakovic (Eds.), *Applied Technologies* (pp. 426–440). Springer International Publishing.
- Guevara-Vega, C. P., Patiño Chalcán, R. A., Landeta, P. A., Rea-Peñañiel, X. M., & Quiña-Mera, J. A. (2022). Verification and Validation Plan in a Software Product: A Practical Study. In M. Botto-Tobar, S. Montes León, P. Torres-Carrión, M. Zambrano Vizuetete, & B. Durakovic (Eds.), *Applied Technologies* (pp. 525–537). Springer International Publishing.
- Guirao Goris, S. J. A. (2015). Utilidad y tipos de revisión de literatura. *Ene*, 9, 0.
- Gupta, B., Mittal, P., & Mufti, T. (2021, March 17). *A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services*. <https://doi.org/10.4108/eai.27-2-2020.2303255>
- Hafner, C. (2020). *Enhancing Page Object Maintainability in Modern Automated Software Testing Life Cycles DIPLOMARBEIT zur Erlangung des akademischen Grades Software Engineering and Internet Computing eingereicht von*. www.tuwien.at

- Haro, E., Guarda, T., Zambrano, A. O., & Ninahualpa, G. (2019). *Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot*.
- IEEE Computer Society. (2014). IEEE Standard for Software Quality Assurance Processes. In *IEEE Std 730-2014 (Revision of IEEE Std 730-2002)*. <https://doi.org/10.1109/IEEESTD.2014.6835311>
- Instituto Agile. (2021). *Manifiesto ágil, 12 principios de la agilidad*.
<https://www.institutoagile.com/post/12-principios-del-manifiesto-%C3%A1gil>
- ISO/IEC 25000. (2014). *ISO/IEC 25000:2014 - Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-Guide to SQuaRE*.
- ISO/IEC 25023. (2016). *Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-Measurement of system and software product quality*.
www.iso.org
- Jeya Mala, D., & Pradeep Reynold, A. (2020). Towards Green Software Testing in Agile and DevOps Using Cloud Virtualization for Environmental Protection. In M. Ramachandran & Z. Mahmood (Eds.), *Software Engineering in the Era of Cloud Computing* (pp. 277–297). Springer International Publishing. https://doi.org/10.1007/978-3-030-33624-0_11
- Laporte, C. Y., & April, A. (2018). *Software Quality Assurance*. <http://www.computer.org>.
- Marija, J. (2020). Comparison: Angular vs. React vs. Vue. Which framework is the best choice? *Information Society of Serbia*.
- Naciones Unidas. (2019). *Informe de los Objetivos de Desarrollo Sostenible*.
- Najihi, S., Elhadi, S., Abdelouahid, R. A., & Marzak, A. (2022). Software Testing from an Agile and Traditional view. *Procedia Computer Science*, 203, 775–782.
<https://doi.org/https://doi.org/10.1016/j.procs.2022.07.116>
- Nakai, H., Tsuda, N., Honda, K., Washizaki, H., & Fukazawa, Y. (2016). Initial Framework for Software Quality Evaluation Based on ISO/IEC 25022 and ISO/IEC 25023. *Proceedings - 2016 IEEE International Conference on Software Quality, Reliability and Security-Companion, QRS-C 2016*, 410–411. <https://doi.org/10.1109/QRS-C.2016.66>
- NTE INEN-ISO/IEC 25010. (2015). *Sistemas e ingeniería de software-Requisitos y Evaluación de Sistemas y Calidad de software (SQuaRE)-Modelos de calidad del sistema y software*.
- O'Regan, G. (2019). Testing in the Agile World. In G. O'Regan (Ed.), *Concise Guide to Software Testing* (pp. 221–233). Springer International Publishing. https://doi.org/10.1007/978-3-030-28494-7_12
- Peláez, L. E., Toro, A., Arias, J. L., & Rodríguez, D. E. (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. <https://doi.org/10.17981/ingecuc.15.2.2019.11>
- Penmetsa, J. (2017). Agile Testing. In *Trends in Software Testing* (pp. 19–33).
https://doi.org/10.1007/978-981-10-1415-4_2

- Perdomo Charry, W. (2019). *Modelo para la medición del progreso del alfa sistema de software del núcleo de Semat con base en las normas ISO/IEC 2502n e ISO/IEC 2504n*. Universidad Nacional de Colombia.
- Quiña-Mera, A., Chamorro Andrade, L., Montaluisa Yugla, J., Chicaiza Angamarca, D., & Guevara-Vega, C. P. (2021). Improving Software Project Management by Applying Agile Methodologies: A Case Study. In M. Botto-Tobar, S. Montes León, O. Camacho, D. Chávez, P. Torres-Carrión, & M. Zambrano Vizuete (Eds.), *Applied Technologies* (pp. 672–685). Springer International Publishing.
- Rajapaksha, D., Tantithamthavorn, C., Jiarpakdee, J., Bergmeir, C., Grundy, J., & Buntine, W. (2021). SQAPlanner: Generating Data-Informed Software Quality Improvement Plans. *IEEE Transactions on Software Engineering*, 48(8), 2814–2835. <https://doi.org/10.1109/TSE.2021.3070559>
- Real Academia Española. (2023). *calidad | Diccionario esencial de la lengua española | RAE - ASALE*. <https://www.rae.es/desen/calidad>
- Ruliansyah, Tukino, Huda, B., & Hananto, A. L. (2023). Penerapan Software Testing Life Cycle Pada Pengujian Otomatisasi Platform Dzikra Application of Software Testing Life Cycle in Automated Testing of Dzikra Platform. *Computer Science Research and Its Development Journal*, 15(1), 1–11. <https://doi.org/10.22303/csrid.15.1.2023.01-11>
- Schmidt, R. F. (2013). Software Acceptance Testing. In *Software Engineering* (pp. 335–341). Elsevier. <https://doi.org/10.1016/b978-0-12-407768-3.00020-3>
- Secretaría Nacional de Planificación. (2021). *Plan de Creación de Oportunidades 2021-2025*.
- Sifuentes, Y. M., & Peralta, J. L. (2022). Modelo de medición y evaluación de calidad del software basado en la norma ISO/IEC 25000 para medir la usabilidad en productos de software académicos universitarios. *TecnoHumanismo*.
- Sneha, K., & Malle, G. M. (2017). Research on Software Testing Techniques and Software Automation Testing Tools. *International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*.
- Soraluz, A. E., Valles, M. Á., & Levano, D. (2023). Desarrollo guiado por comportamiento: buenas prácticas para la calidad de software. *Ingeniería y Desarrollo*, 39(01), 190–204. <https://doi.org/10.14482/inde.39.1.005.3>
- Stray, V., Florea, R., & Paruch, L. (2022). Exploring human factors of the agile software tester. *Software Quality Journal*, 30(2), 455–481. <https://doi.org/10.1007/s11219-021-09561-2>
- The Standish Group. (2015). *CHAOS REPORT 2015*.
- Umar, M. A. (2020). Comprehensive study of software testing: Categories, levels, techniques, and types. *TechRxiv*. <https://doi.org/10.36227/techrxiv.12578714.v2>

- Umar, M. A., & Zhanfang, C. (2020). A Comparative Study of Dynamic Software Testing Techniques. *Int. J. Advanced Networking and Applications*, 4575–4584.
- Universidad Técnica del Norte. (2017). *Misión y Visión*.
<https://software.utn.edu.ec/informacion/mision-y-vision/>
- Ushakova, I., Skorin, Y., Shcherbakov, A., & Kharkiv, S. K. (2021). Methods of quality assurance of software development based on a systems approach. *III International Scientific And Practical Conference "Information Security And Information Technologies."*
- Uyaguari, F., Guevara-Vega, C., Quiña-Mera, A., Uyaguari, A., & Acosta, C. (2023). Effect of requirements specification using native language on external software quality. *IET Software*, 17(3), 287–300. <https://doi.org/https://doi.org/10.1049/sfw2.12124>
- Yu, J. (2018). Design and Application of a Testing Framework of Online Course Based on Agile. *IOP Conference Series. Materials Science and Engineering*, 394(3).
<https://doi.org/https://doi.org/10.1088/1757-899X/394/3/032099>

ANEXOS

Anexo A

Situación inicial de los procesos a ser automatizados en la empresa

BABUCHA DON CRISTIAN. 12/11/2021

Acilico 7.60
 Liana 7.60
 Nylon 7.60
 January 7.60

Sábado Saturday 1-364

1 PVP 15

BABUCHA SR. ALBORNOS. ALGODON PEINADO

PESO 0.950 x 10 = 9.50

Algodon Peinado + 3
 Liana + 2
 Elastico + 8
 Nylon 22.50

AT 24 PVP 28.25

Babucha Sr. Albornos Peinado SR. Albornos 15.50 14

Domingo Sunday **2**

BABUCHA DON CRISTIAN. ALGODON NORMAL 16/11

Algodon Benaflex 0.562 x 5 = 2.81
 Liana Nylon Rebel 0.108 x 9.50 = 1.03
 Nylon Entender 0.129 x 12 = 1.548
 Elastico Rebel 0.015 x 12.80 = 0.19

PESO 10-12 700 grams } + 5.57 Material
 + CORTE 100 } + 2 produccion individual
 Total 800 } 9.57 / 13.50 AP

13.57 P.V.P. 25

Sin help

Don Marco
 Febrero February 2021

2.000000 ALGODON NORMAL

PESO 0.500

Algodon 0.430 x 5.60 = 2.39
 Liana 0.040 x 9.50 = 0.38
 Elastico 0.020 x 11.70 = 0.23
 Nylon 0.010 x 11.70 = 0.12

Lunes Monday 3-362

Material 2.43
 Mano de obra 2.13

AT 10% 10.4

5.13 P.V.P. 11.70

3.000000 ALGODON NORMAL 13/11/2021

PESO 0.580

Algodon 0.430 x 5.60 = 2.39
 Liana 0.040 x 9.50 = 0.38
 Elastico 0.020 x 9.87 = 0.19
 Nylon 0.010 x 11. = 0.11

Anita 10.96 + 11.55

5.54 PVP 10.30

4.000000 CAMILLERA ANTIDESLIZANTE ALGODON PEINADO

PESO 0.690 Wilson C. Juan 16. Peso Real 1.56 puntos Canada

Nylon
 Elastico
 Liana

PVP 18

AT 26 Mico 18 puntos Canada

5.000000 CAMILLERA ALGODON NORMAL

PESO 0.640

Algodon 0.430 x 5.60 = 2.41
 Liana 0.080 x 9.87 = 0.79
 Elastico 0.020 x 11. = 0.22
 Nylon 0.010 x 11. = 0.11

6.15 PVP 12.10

Enero January

SEM L M M J V S D

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Don Luis.
 16/10/2023
 Fútbol (Nylon) ~~Bebera~~ 18doc.

Malla Don Luis
 16/10/2023
 18doc. ▶
 (92731 Anulado 165 Mallas)

Etafashion 73403
 Ingresa el lunes.

Etafashion leopardo - Blanco 6x Talla

R.M. Mallas → 13 de Noviembre

Fecha de Recepción: _____ Fecha de Entrega: _____
 Cliente: Sr. Cristian Ramos Teléfono: _____
 Dirección: 05/10/2023
 Material: _____

MODELO	TALLA	CANTIDAD
Polaina con Pie		
	Negro	5
	Azul Marino	3
	Azul Jean	3
	Gris oscuro	3
	Cafe	3
	Vino claro	3
	Vino bajo	3
Polaina Sin Pie con puño		
	Negro	3
	Azul Marino	3
	Azul Jean	3
	Gris claro	3
	Gris oscuro	3
	Cafe	3
	Vino claro	3
	Vino bajo	3
	Habano	3
	Resaca bebe	3

Observaciones: _____

SR. WILSON CIFUENTES

TALLA 10-12 CANILLERA		
NEGRO	BLANCO	3
BLANCO	NEGRO	3



SR. WILSON CIFUENTES

CANIKE 2. CU TALLA 10-12 CANILLERA NIKE CLÁSICO		
CELESTE (PLANTA CELESTE)	NEGRO	3 □
AZUL MARINO (PLANTA AZUL MARINO)	BLANCO	6 □
BLANCO (PLANTA BLANCO)	TURQUESA	3 □
CANLANA. CU TALLA 10-12 CANILLERA LLANAS		
NEGRO (PLANTA NEGRO)		8 □ T
CAISLA. CU TALLA 10-12 CANILLERA ISLAS		
NEGRO (PLANTA NEGRO)	ROJO	3 □
CANIKE 3. CU TALLA 1-12 CANILLERA NIKE MANCHAS		
FUCSIA FUERTE (PLANTA NEGRO)	NEGRO	3
TURQUESA (PLANTA NEGRO)	NEGRO	3 □
AMARILLO (PLANTA NEGRO)	NEGRO	3 □
AMARILLO FOSFORESCENTE (PLANTA NEGRO)	NEGRO	3 □
ROJO (PLANTA NEGRO)	NEGRO	3 □
TURQUESA (PLANTA BLANCO)	BLANCO	3 1

Anexo B

Ejecución de las pruebas de aceptación para el Sprint 1

GESTOR NELTEX
Prueba de Aceptación de Usuarios (UAT)
Identificador: UAT - 01

Nombre de usuario	Cargo	Fecha
<i>Wilson Rodríguez P.</i>	<i>Gerente</i>	<i>18 noviembre 2023</i>

Id Caso	Descripción del caso	Precondición	Pasos por seguir	Resultado esperado	Validación		Observaciones
					Aceptado	Rechazado	
CPA-1.1	Verificar acceso con credenciales correctas	Número de cédula y contraseña válidos	1. Ingresar número de cédula 2. Ingresar la contraseña 3. Hacer clic en "Ingresar"	Inicio de sesión exitoso.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-1.2	Verificar acceso con usuario inactivo	Credenciales válidas para usuario inactivo	1. Ingresar número de cédula 2. Ingresar la contraseña 3. Hacer clic en "Ingresar"	Error inicio de sesión incorrecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-1.3	Ver lista de empleados	N/A	1. Clic a la opción de gestión de empleados	Lista de empleados mostrada.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>Implementar búsqueda con tecla</i>
CPA-1.4	Asignar rol a empleado	Empleado nuevo o creado y rol a asignar	1. Hacer clic en nuevo empleado / modificar 2. Seleccionar un rol 3. Guardar / crear	Rol de usuario asignado correctamente	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>No debe permitir crear varios usuarios con la misma cédula</i>
CPA-1.5	Eliminar rol de empleado	Empleado y roles asignados	1. Seleccionar empleado 2. Eliminar rol de la lista de asignaciones 3. Guardar cambios	Rol de usuario eliminado correctamente	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-1.6	Ver roles disponibles	N/A	1. Acceder a la opción nuevo empleado / modificar 2. Revisar la lista de roles disponibles	Lista incluye "Gerente," "Adquisición," "Vendedor," y "Bodeguero"	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Acceptado	Rechazado	Firma del usuario
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>Wilson Rodríguez P.</i>

GESTOR NELTEX
Prueba de Aceptación de Usuarios (UAT)
Identificador: UAT - 02

Nombre de usuario	Cargo	Fecha
<i>Juan Rodríguez</i>	<i>Bodeguero</i>	<i>18 noviembre 2023</i>

Id Caso	Descripción del caso	Precondición	Pasos por seguir	Resultado esperado	Validación		Observaciones
					Aceptado	Rechazado	
CPA-2.1	Crear materia prima	Información de la materia prima	1. Acceder a nuevo material 2. Ingresar la información requerida 3. Hacer clic en "Crear"	Materia prima creada con éxito.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>Presenta un código de error al crear un material</i>
CPA-2.2	Ver lista de materias primas	N/A	1. Acceder a gestión de materiales	Lista de materias primas mostrada.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>Implementar búsqueda con tecla</i>
CPA-2.3	Editar información de materia prima	Materia prima por editar	1. Seleccionar materia prima 2. Editar la información deseada 3. Hacer clic en "Guardar"	Información de materia prima editada correctamente.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-2.4	Cambiar estado de materia prima a "Activa"	Materia prima por activar	1. Hacer clic en la opción "Cambiar de estado" del material deseado 2. Aceptar	Estado cambiado a "Activa" con éxito.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-2.5	Cambiar estado de materia prima a "Inactiva"	Materia prima por desactivar	1. Hacer clic en la opción "Cambiar de estado" del material deseado 2. Aceptar	Estado cambiado a "Inactiva" con éxito.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Acceptado	Rechazado	Firma del usuario
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>Juan Rodríguez</i>

Anexo C

Ejecución de las pruebas de aceptación para el Sprint 2

GESTOR NELTEX
Prueba de Aceptación de Usuarios (UAT)
Identificador: UAT – 03

Nombre de usuario	Cargo	Fecha
Maria Jesus Andronago	Vendedor	09 diciembre 2023

Id Caso	Descripción del caso	Precondición	Pasos por seguir	Resultado esperado	Validación		Observaciones
					Aceptado	Rechazado	
CPA-3.1	Crear nuevo producto.	Información del producto, materiales.	1. Acceder a nuevo producto. 2. Ingresar información y materiales. 3. Guardar.	Mensaje: "Producto creado correctamente".	X	—	
CPA-3.2	Crear producto sin materiales.	Información del producto.	1. Acceder a nuevo producto. 2. Ingresar información. 3. Guardar.	Mensaje: "Producto creado correctamente".	X	—	
CPA-3.3	Editar información del producto y sus materiales.	Producto por editar y sus materiales	1. Acceder y editar producto. 2. Modificar información. 3. Guardar.	Mensaje: "Producto actualizado correctamente".	X	—	
CPA-3.4	Cambiar estado del producto.	Producto por cambiar	1. Acceder y seleccionar producto 2. Cambiar estado.	Mensaje: "Producto modificado con éxito"	X	—	
CPA-3.5	Cálculo de costo de materiales.	Nuevo producto con materiales	1. Crear nuevo producto con materiales.	Costo total de materiales correcto.	X	—	
CPA-3.6	Ver materiales necesarios.	Producto con materiales	1. Acceder y seleccionar producto 2. Ver materiales asociados al producto.	Lista de materiales asociados mostrada correctamente.	X	—	

Aceptado	Rechazado	Firma del usuario
X	—	

GESTOR NELTEX
Prueba de Aceptación de Usuarios (UAT)
Identificador: UAT – 04

Nombre de usuario	Cargo	Fecha
Maria Jesus Andronago	Vendedor	09 diciembre 2023

Id Caso	Descripción del caso	Precondición	Pasos por seguir	Resultado esperado	Validación		Observaciones
					Aceptado	Rechazado	
CPA-4.1	Recibir nuevo pedido.	Información del cliente, detalles del pedido.	1. Acceder a nuevo pedido. 2. Ingresar información 3. Guardar.	Mensaje: "Pedido creado correctamente".	X	—	La fecha de recepción debe poder modificarse
CPA-4.2	Agregar productos al pedido.	Productos por agregar al pedido	1. Acceder a nuevo pedido. 2. Seleccionar productos. 3. Especificar detalles. 4. Guardar.	Productos agregados con éxito.	X	—	Debe permitir ingresar el color y tamaño opcional
CPA-4.3	Calcular valor total.	Productos en el pedido	1. Verificar que el total del pedido se actualice automáticamente.	Total, del pedido calculado y correcto.	X	—	
CPA-4.4	Guardar y generar comprobante	Pedido a guardar	1. Guardar pedido y generar comprobante. 2. Verificar detalles del comprobante.	Pedido guardado y comprobante generado con detalles.	X	—	Corrección de email de la empresa en el comprobante
CPA-4.5	Visualizar pedidos	N/A	1. Acceder y visualizar pedidos recibidos.	Lista de pedidos mostrada.	X	—	
CPA-4.6	Eliminar producto del pedido	Producto en el pedido	1. Seleccionar y eliminar producto del pedido.	Producto eliminado del pedido correctamente.	X	—	

Aceptado	Rechazado	Firma del usuario
X	—	

Anexo D

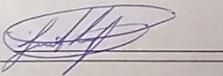
Ejecución de las pruebas de aceptación para el Sprint 3

GESTOR NELTEX
Prueba de Aceptación de Usuarios (UAT)
Identificador: UAT - 05

Nombre de usuario	Cargo	Fecha
Juan Rodríguez	Bodeguero	23 diciembre 2023

Creación de órdenes de compra (HU-3)

Id Caso	Descripción del caso	Precondición	Pasos por seguir	Resultado esperado	Validación		Observaciones
					Aceptado	Rechazado	
CPA-5.1	Ver lista de órdenes de compra	N/A	1. Acceder y ver lista de órdenes de compra.	Lista mostrada correctamente	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Agregar el botón de compra
CPA-5.2	Crear nueva orden de compra	Materiales por solicitar en la orden	1. Acceder a nueva orden. 2. Especificar materiales, cantidades, etc. 3. Guardar	Mensaje: "Orden de compra creada correctamente"	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-5.3	Ver detalles de orden de compra	Orden de compra existente	1. Seleccionar una orden de compra 2. Clic en ver detalles	Detalles de la orden de compra mostrados.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-5.4	Eliminar orden de compra	Orden de compra existente	1. Seleccionar una orden de compra 2. Eliminar la orden	Orden de compra eliminada correctamente.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-5.5	Filtrar órdenes por estado	Listado de órdenes de compra	1. Acceder y aplicar filtro por estado en lista de órdenes.	Lista mostrada con órdenes filtradas por estado.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-5.6	Crear orden con materiales nuevos	Detalles de la orden	1. Acceder a nueva orden. 2. Crear nuevo material 3. Especificar cantidad 4. Guardar.	Mensaje: "Orden de compra creada correctamente"	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Permitir crear un nuevo tipo de material

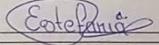
Aceptado	Rechazado	Firma del usuario
<input checked="" type="checkbox"/>	<input type="checkbox"/>	

GESTOR NELTEX
Prueba de Aceptación de Usuarios (UAT)
Identificador: UAT - 06

Nombre de usuario	Cargo	Fecha
Catalina Tupe	Adquisición	22 diciembre 2023

Registro de compras realizadas (HU-5)

Id Caso	Descripción del caso	Precondición	Pasos por seguir	Resultado esperado	Validación		Observaciones
					Aceptado	Rechazado	
CPA-6.1	Ver órdenes pendientes	N/A	1. Acceder y ver lista de órdenes de compra pendientes.	Lista de ordenes mostrada	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-6.2	Registrar compra desde orden	Orden de compra pendiente	1. Seleccionar orden de compra pendiente. 2. Registrar compra con detalles. 3. Guardar.	Compra registrada correctamente, estado actualizado	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-6.3	Ver registro de compras	N/A	1. Acceder y ver lista de compras realizadas.	Lista de compras realizadas	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Ver registro de compra como bodeguero
CPA-6.4	Registrar compra sin detalles	Orden de compra pendiente	1. Seleccionar orden de compra pendiente. 2. Registrar compra sin especificar detalles. 3. Guardar.	Mensaje de error al intentar registrar compra sin detalles.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-6.5	Registrar compra sin orden asociada	Materiales comprados con sus detalles	1. Acceder a nueva compra comprados con sus detalles. 2. Registrar los materiales comprados.	Compra registrada correctamente.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CPA-6.6	Modificar estado de orden de compra	Orden de compra existente	1. Registrar compra a partir de una orden. 2. Revisar estado actualizado.	Estado de la orden de compra "Comprado"	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Aceptado	Rechazado	Firma del usuario
<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Anexo E

Ejecución de las pruebas de aceptación para el Sprint 4

GESTOR NELTEX
Prueba de Aceptación de Usuarios (UAT)
Identificador: UAT - 07

Nombre de usuario	Cargo	Fecha
Juan Rodriguez	Bodegaero	06 enero 2024

Registro de movimiento de materia prima (HU-4)

Id Caso	Descripción del caso	Precondición	Pasos por seguir	Resultado esperado	Validación		Observaciones
					Aceptado	Rechazado	
CPA-7.1	Crear Salida de Material	Orden de producción, detalles de salida	1. Acceder a nueva salida 2. Especificar materiales, solicitante y orden. 3. Guardar.	Salida de material creada con detalles especificados.	X	—	
CPA-7.2	Crear Entrada de Material	Productos recibidos y detalles de entrada	1. Acceder a nueva entrada. 2. Especificar materiales y compra 3. Guardar.	Entrada de material creada con detalles especificados.	X	—	
CPA-7.3	Modificar Stock Manualmente	Tipo de ajuste, detalles de modificación	1. Acceder a nuevo ajuste 2. Especificar materiales y tipo de ajuste. 3. Guardar.	Ajuste manual realizado correctamente.	X	—	Mostrar el stock actual del material
CPA-7.4	Visualizar Registro de Movimientos	N/A	1. Acceder al registro de movimientos	Registro mostrado con su tipo.	X	—	
CPA-7.5	Crear movimiento sin detalles	Orden de producción, detalles de salida	1. Acceder a nueva salida 2. Guardar sin especificar detalles	Mensaje de error al intentar crear salida sin detalles.	X	—	
CPA-7.6	Modificar stock con cantidad negativa	Ajuste manual, cantidad negativa	1. Acceder a nuevo ajuste 2. Ingresar una cantidad negativa en stock 3. Guardar.	Mensaje de error stock con cantidad negativa.	X	—	

Aceptado	Rechazado	Firma del usuario
X	—	

GESTOR NELTEX
Prueba de Aceptación de Usuarios (UAT)
Identificador: UAT - 08

Nombre de usuario	Cargo	Fecha
Eduardo Andrade	Producción	06 enero 2024

Generar orden de producción (HU-8)

Id Caso	Descripción del caso	Precondición	Pasos por seguir	Resultado esperado	Validación		Observaciones
					Aceptado	Rechazado	
CPA-8.1	Generar orden de producción automáticamente	Pedido creado	1. Verificar que, al crear un pedido, se genere automáticamente una orden de producción asociada.	Orden de producción generada automáticamente	X	—	Generar orden de producción en PDF
CPA-8.2	Ver Detalles de Orden de Producción	Orden de producción generada	1. Acceder y ver detalles de la orden de producción generada.	Detalles de la orden de producción mostrados correctamente.	X	—	
CPA-8.3	Visualizar Orden de Producción en Departamento de Producción	N/A	1. Verificar que la orden de producción sea visible para el departamento de producción.	Orden de producción visible para producción.	X	—	
CPA-8.4	Verificar Contenido de Orden de Producción	Detalles de la orden de producción	1. Acceder a la orden de producción deseada y verificar sus detalles	Contenido de la orden de producción y sin información de precios.	X	—	

Aceptado	Rechazado	Firma del usuario
X	—	—

Anexo F

Checklist usado para evaluar la característica de Adecuación Funcional

Evaluación de Completitud Funcional del Software según ISO/IEC 25023

El siguiente checklist fue diseñado para valorar la Completitud Funcional del software, siguiendo los estándares de la Norma ISO/IEC 25023.

Utilice el checklist proporcionado para guiar su evaluación y asegurarse de cubrir todos los aspectos requeridos en el software.

Una vez completada la evaluación, proporcione retroalimentación detallada sobre cada función marcada, destacando cualquier observación relevante que pueda contribuir a mejorar la Completitud

1. Revise las funciones principales del software, cargue cada una como "presente" o "faltante" según lo que observe durante la revisión detallada del producto final.

	Presente	Faltante
Como administrador, asigna roles, controla acceso y acciones de empleados.	<input type="radio"/>	<input type="radio"/>
Como bodeguero, crear, ver, modificar y eliminar registros de materiales.	<input type="radio"/>	<input type="radio"/>
Como bodeguero, crear y ver órdenes de compra para adquirir materiales.	<input type="radio"/>	<input type="radio"/>
Como bodeguero, registrar ingreso/salida de materia prima asociado a órdenes de producción o compras.	<input type="radio"/>	<input type="radio"/>
Como encargado de compras, registrar compras de órdenes recibidas desde bodega.	<input type="radio"/>	<input type="radio"/>
Como vendedor o gerente, gestionar productos para producción, detalles de materiales y precios.	<input type="radio"/>	<input type="radio"/>
Como vendedor, recibir pedidos, registrar detalles del cliente y pedido, generar comprobante	<input type="radio"/>	<input type="radio"/>

Resumen de resultados

Ver resultados ...

1. Revise las funciones principales del software, cargue cada una como "presente" o "faltante" según lo que observe durante la revisión detallada del producto final.

[Más detalles](#)

■ Presente ■ Faltante

