



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

**Proceso de testing basado en ISTQB para optimizar los
tiempos de pruebas en aplicaciones bancarias**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniera de Sistemas

AUTOR

Karent Minoshca CHECCLLO AGUIRRE

ASESOR

Luis Alberto ALARCÓN LOAYZA

Lima, Perú

2024



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Checcllo, K. (2024). *Proceso de testing basado en ISTQB para optimizar los tiempos de pruebas en aplicaciones bancarias*. [Trabajo de Suficiencia Profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	KARENT MINOSHCA CHECCLO AGUIRRE
Tipo de documento de identidad	DNI
Número de documento de identidad	76018627
URL de ORCID	https://orcid.org/0009-0001-4172-2849
Datos de asesor	
Nombres y apellidos	Luis Alberto Alarcón Loayza
Tipo de documento de identidad	DNI
Número de documento de identidad	00456684
URL de ORCID	https://orcid.org/0000-0001-8340-7161
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Jorge Santiago Pantoja Collantes
Tipo de documento	DNI
Número de documento de identidad	06254022
Miembro del jurado 1	
Nombres y apellidos	José César Piedra Isusqui
Tipo de documento	DNI
Número de documento de identidad	25628915
Datos de investigación	
Línea de investigación	No aplica
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima

	Distrito: Cercado de Lima Jr. Carlos Amezaga No. 375 Universidad Nacional Mayor de San Marcos Latitud: -12.0564232 Longitud: -77.0843327
Año o rango de años en que se realizó la investigación	2023
URL de disciplinas OCDE	2.02.04 -- Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Sistemas

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 11:05 horas del día 27 de enero del año 2024, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Lic. Pantoja Collantes Jorge Santiago (Presidente), Mg. Piedra Isusqui José César (Miembro) y el Lic. Alarcón Loayza Luis Alberto (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/byr-ntdg-dks?hs=224>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“PROCESO DE TESTING BASADO EN ISTQB PARA OPTIMIZAR LOS TIEMPOS DE PRUEBAS EN APLICACIONES BANCARIAS”**, por la Bachiller **Checcllo Aguirre Karent Minoshca**; para obtener el Título Profesional de Ingeniera de Sistemas.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó a la Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

La Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiendo efectuado la calificación correspondiente por los miembros del Jurado, la Bachiller obtuvo la nota de **18 (DIECIOCHO)**.

A continuación el Presidente de Jurados el Lic. Pantoja Collantes Jorge Santiago, declara a la Bachiller **Ingeniera de Sistemas**.

Siendo las 12:05 horas, se levantó la sesión.

Presidente

Lic. Pantoja Collantes Jorge Santiago

Miembro

Mg. Piedra Isusqui José César

Miembro Asesor

Lic. Alarcón Loayza Luis Alberto



CERTIFICADO DE SIMILITUD

Yo LUIS ALBERTO ALARCÓN LOAYZA en mi condición de asesor acreditado con la Resolución Decanal N° 001246-2023-D-FISI y Oficio N° 001-PTTSP-FISI-2023 del Trabajo de Suficiencia Profesional cuyo título es PROCESO DE TESTING BASADO EN ISTQB PARA OPTIMIZAR LOS TIEMPOS DE PRUEBAS EN APLICACIONES BANCARIAS, presentado por el bachiller CHECCLO AGUIRRE KARENT MINOSHCA para optar el título de Ingeniero de Sistemas CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante el software de similitud textual, el documento evaluado cuenta con el porcentaje de 14% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del título correspondiente.

Firma del Asesor

DNI: 00456684

Nombres y apellidos del asesor: LUIS A. ALARCÓN LOAYZA



DEDICATORIA

Dedico el presente trabajo principalmente a Dios por ser la guía en mi vida y por todos los logros obtenidos; a mis padres, familia y amigos cercanos por acompañarme en cada etapa y ser soporte para cumplir mis sueños y darme aliento cuando sentía darme por vencida.

AGRADECIMIENTOS

Agradezco a la Universidad Nacional de San Marcos por la formación profesional brindada; a la entidad bancaria que me permite aportar desde mi rol con iniciativas constantes, y a todos mis compañeros que fueron parte de este proyecto.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Proceso de Testing basado en ISTQB para optimizar los tiempos de pruebas en aplicaciones bancarias

Autor: Checcello Aguirre, Karent Minoshca
Asesor: Alarcón Loayza, Luis Alberto
Título: Trabajo de Suficiencia Profesional para optar el Título
Profesional de Ingeniero de Sistemas
Fecha: Octubre 2023

RESUMEN

En el presente Trabajo de Suficiencia Profesional se detalla la definición de un proceso de pruebas de software que consta de las siguientes fases: Análisis del requerimiento, Diseño y Redacción, Implementación, Ejecución de los casos de prueba para aplicaciones de canales digitales; el cual se apoyó en ISTQB (International Software Testing Qualifications Board), entidad de certificación global en pruebas de Software.

Para la definición y validación del proceso de pruebas se utilizó Scrum por ser la metodología ágil desplegada en la organización.

Como resultado se obtuvo una mejor cobertura de los entregables de pruebas mediante la aplicación de técnicas de diseño; se dispone un gestor de casos de prueba Xray el cual permitió la reutilización de los mismos optimizando el tiempo en el proceso y en la generación de los entregables de pruebas.

Palabras claves: ISTQB, Pruebas de Software, Entidad bancaria, Scrum , Gestor de casos de prueba

**MAJOR NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF SYSTEMS AND COMPUTER ENGINEERING
PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING**

Process based on ISTQB to optimize testing times in banking applications

Author: Checcello Aguirre, Karent Minoshca
Advisor: Alarcón Loayza, Luis Alberto
Title: Professional Sufficiency Work to opt for the Professional Title
of Systems Engineer
Date: October 2023

ABSTRACT

In this work of Professional Sufficiency is detailed the definition of a software testing process consisting of the following phases: Requirement Analysis, Design and Writing, Implementation, Execution of test cases for digital channels applications, which was supported by ISTQB (International Software Testing Qualifications Board); global certification body in software testing.

Scrum was used for the definition and validation of the testing process because it is the agile methodology deployed in the organization.

As a result, a better coverage of the test deliverables was obtained through the application of design techniques; a Xray test case manager was used, which allowed the reuse of the same, optimizing the time in the testing process and in the generation of test deliverables.

Keywords: Testing, ISTQB, Software Testing, Banking, Scrum, Test Case Manager.

ÍNDICE DEL TRABAJO DE SUFICIENCIA PROFESIONAL

CARÁTULA	i
FICHA CATALOGRÁFICA	ii
DEDICATORIA	iii
AGRADECIMIENTOS	iv
RESUMEN	v
ABSTRACT	vi
INDICE DE FIGURAS	ix
INDICE DE TABLAS	xi
INTRODUCCIÓN	1
CAPÍTULO I - TRAYECTORIA PROFESIONAL	3
CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA ..	7
2.1 Empresa – actividad que realiza	7
2.2 Visión	7
2.3 Misión	7
2.4 Organización de la empresa	8
2.4.1 Organigrama General	8
2.4.2 Organigrama del área de Sistemas	9
2.4 Área, cargo y funciones desempeñadas	12
2.5 Experiencia profesional realizada en la organización	12
CAPÍTULO III - ACTIVIDADES DESARROLLADAS	14
3.1 Situación problemática	14
3.1.1 Definición del problema	17
3.2 Solución	18
3.2.1 Objetivos	20
3.2.2 Etapas y metodología	21
3.2.3 Fundamentos Utilizados	43

3.2.3 Implementación de las áreas, procesos, sistemas y buenas prácticas.....	44
3.3 Evaluación Económica.....	49
3.3.1 Evaluación del costo.....	49
3.3.2 Beneficio para la organización	50
CAPÍTULO IV. REFLEXIÓN CRÍTICA DE LA EXPERIENCIA	52
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES	53
5.1 Conclusiones	53
5.2 Recomendaciones	54
5.3 Fuentes de información.....	55
5.4 GLOSARIO	57
ANEXOS	58
ANEXO 1: ALTAS DE ISSUES EN XRAY	58

ÍNDICE DE FIGURAS

Figura 1 Organigrama general de la empresa	8
Figura 2 Organigrama de la gerencia de Sistemas y sus distintas subgerencias	10
Figura 3 Estructura de la subgerencia de Empresas y sus servicios asociados	10
Figura 4 Organización de roles por niveles de las subgerencia de Empresa	10
Figura 5 Organización antigua del área de Ingeniería	15
Figura 6 Implementación de herramientas DevOps en el ciclo de desarrollo de Software	16
Figura 7 Fases del proceso de pruebas propuesto	18
Figura 8 Principales prácticas utilizadas en el proceso propuesto.....	19
Figura 9 Esquema de períodos de desarrollo ágil.....	21
Figura 10 Ceremonias y actividades realizadas dentro de un sprint.....	22
Figura 11 Fases del SDLC y su definición	23
Figura 12 Relación de las fases del SDLC y el proceso de pruebas propuesto.....	24
Figura 13 Matriz de pruebas basadas en Riesgos	26
Figura 14 Esquema de dimensiones de la matriz de pruebas basadas en Riesgo y su valoración	26
Figura 15 Pasos para la aplicación de las pruebas basadas en riesgos	27
Figura 16 Estructura del Gherkin	30
Figura 17 Ejemplo de la redacción de un caso de prueba en Gherkin.....	31
Figura 18 Página web donde se puede utilizar el Pairwise	32
Figura 19 Nomenclatura propuesta para los casos de prueba.....	32
Figura 20 Redacción del lenguaje Gherkin y sus palabras reservadas	33
Figura 21 Procedimiento para la aplicación de técnicas.....	33
Figura 22 Xray y la creación de Test.....	36

Figura 23 Xray y el Test Plan con la lista de casos de prueba	37
Figura 24 Xray con el reporte de cobertura para los Test	38
Figura 25 Organización por funcionalidad en el Repositorio de Test.....	38
Figura 26 Nomenclatura para el agrupador de Test	39
Figura 27 Nomenclatura propuesta para los Test Plan.....	39
Figura 28 Nomenclatura propuesta para los Test Execution.....	40
Figura 29 Nomenclatura para los bugs	40
Figura 30 Pasos para aplicar los lineamientos en la fase de ejecución.....	40
Figura 31 Reporte de trazabilidad	41
Figura 32 Reporte de Métricas de Planes de Testing	42
Figura 33 Reporte de Lista de Procesos de Prueba	42
Figura 34 Roodmap de despliegue de las fases del proceso propuesto	44

ÍNDICE DE TABLAS

Tabla 1 Experiencia laboral	3
Tabla 2 Formación Académica	5
Tabla 3 Certificaciones	5
Tabla 4 Idiomas	6
Tabla 5 Cantidad de funcionalidades registradas y catalogadas por nivel de riesgo	45
Tabla 6 Cantidad de casos de prueba generados por trimestres durante el año 2022 y 2023	45
Tabla 7 Cantidad de casos ejecutados por trimestre durante el año 2022 y 2023	45
Tabla 8 Porcentajes de reutilización por trimestres durante el año 2022 y 2023.	46
Tabla 9 Tiempo para el diseño de casos de pruebas	47
Tabla 10 Tiempos incurridos en el diseño de casos por trimestre en 2022	47
Tabla 11 Tiempos incurridos en el diseño de casos por trimestre en 2023	48
Tabla 12 Comparación de tiempos incurridos sin proceso implementado vs tiempos incurridos con el proceso implementado	49
Tabla 13 Cantidad de horas trabajadas y costos del perfil QA por trimestre	50

INTRODUCCIÓN

Las pruebas de software es un proceso que proporciona información objetiva; mediante la verificación y evaluación de la calidad de un determinado producto. Para una organización es de vital importancia tener un adecuado proceso de pruebas que ayude a incrementar la calidad, apoye a la mejora continua de sus procesos, impulse la comunicación en sus equipos y soporte la ejecución automática de ciertas tareas; ello permitirá liberar de manera más rápida productos al usuario final con la menor cantidad de fallos durante su despliegue y uso.

El riesgo de no contar con un proceso de pruebas adecuado en una organización puede traer como consecuencias: pérdida de confianza en el producto, cuando no hace lo que se espera; pérdida de clientes, con una mala experiencia del usuario; impacto monetario con pérdida de dinero; cuando la correcta operatividad del producto está bajo el seguimiento de una entidad reguladora.

Por ejemplo **Hamilton (2023)** nos relata que Starbucks tuvo que cerrar el 60% de sus tiendas por un fallo en su sistema de puntos de venta y tuvo que regalarle el café a sus clientes porque no se podían procesar sus pagos, Nissan tuvo que retirar 1 millón de carros del mercado por un fallo en las bolsas de aire del cual hubieron dos accidentes automovilísticos, en 1994 Airbus de China Airlines se estrelló a causa de un error en su sistema donde perdieron la vida 264 personas, en 1996 un defecto informático hizo que se abonaran 920 millones de dólares a 823 clientes de un importante banco estadounidense.

En un mundo digitalizado el efecto de un fallo puede aún ser más dramático **Siroky (2017)** nos cuenta que en Julio del 2016 un fallo en el sistema de acciones hizo que los precios de las acciones de conocidas empresas como Amazon, Apple, Alphabet, eBay y Microsoft aparecieran de forma inexacta; el sistema indicaba que algunos precios habían caído y en algunos casos que se habían duplicado, mucha gente comentó lo sucedido por redes sociales creyendo la falsa información, dejando el nombre de las empresas en la polémica por algo que en realidad no había sucedido.

El área de Ingeniería de la entidad bancaria evaluada como parte de su mejora continua implementó un nuevo framework para el desarrollo de software; sin embargo, se observó que para el proceso de pruebas de software no se habían definido prácticas, lineamientos y estándares que ayudarán a cumplir estos objetivos.

En los equipos Scrum las pruebas se realizaban con leve conocimiento del negocio, se tenía poco tiempo para realizar un adecuado análisis y definir una correcta estrategia de pruebas, para el diseño no se contaba con un estándar tecnificado que permitiera sustentar la correcta definición de los casos de prueba; para la redacción no se contaban con lineamientos claros a nivel de nomenclatura y contenido, esto ocasionaba que fueran reescritos varias veces; adicionalmente estos se perdían en el tiempo ya que no se tenía un mecanismo de almacenamiento ni de gestión; esto originó que los usuarios tengan una gran cantidad de incidencias recurrentes que no eran detectadas de manera temprana, lo cual generaba que la entidad bancaria tenga que invertir tiempo y dinero en su resolución.

Ante esta problemática, se vio necesario diseñar e implementar un nuevo proceso de pruebas basado en estándares y con lineamientos claros que permitan tener una mejor organización y trazabilidad en el proceso de pruebas de los entregables con el fin de mejorar la calidad, optimizar tiempos y minimizar los riesgos de incidencias ocurridas.

El presente informe se encuentra estructurado de la siguiente manera:

En el CAPÍTULO I se detalla cronológicamente la trayectoria profesional de la autora considerando sus cargos, funciones y actividades realizadas en diversas empresas, así como los grados y certificaciones obtenidas.

En el CAPÍTULO II se describe información de la empresa donde se desarrolló el trabajo en mención, resaltando su estructura organizacional, misión y visión.

En el CAPÍTULO III se detalla la propuesta del proceso de pruebas de Software basado en ISTQB con el fin de mejorar la cobertura de pruebas y optimizar los tiempos entre otros beneficios cumpliendo los objetivos y alcances planteados.

En el CAPÍTULO IV se realiza un análisis crítico de la propuesta con relación al proceso propuesto comparándolo con el proceso vigente, tomando como caso un proyecto piloto, considerando las lecciones aprendidas.

Finalmente, en el CAPÍTULO V se detallan las conclusiones del proceso propuesto y recomendaciones para los siguientes pasos dentro de la entidad bancaria.

CAPÍTULO I - TRAYECTORIA PROFESIONAL

La autora es bachiller en Ingeniería de Sistemas, especializada en Pruebas de Software con 6 años de experiencia dentro del sector bancario; cuenta con conocimientos en metodologías ágiles, en estrategias y tipos de pruebas tanto como manuales y automatizadas, DevOps y testing continuo.

Se destaca por tener excelente capacidad de organización, liderazgo, ser autodidacta, proactiva siempre con una buena disposición para aprender de manera continua.

Adicionalmente ha llevado cursos de certificación con carácter de valor internacional en los distintos ámbitos que se ha desempeñado.

A continuación, se detalla en la **Tabla 1** la experiencia laboral de la autora en orden descendente:

Tabla 1
Experiencia laboral

Institución y Cargo	Funciones	Año
Entidad Bancaria Quality Enabler (Habilitadora de Calidad)	• Apoyar en la definición y despliegue de un nuevo proceso de pruebas con sus lineamientos e indicadores • Gestionar con las fábricas de pruebas el soporte necesario para los distintos roles testers de los equipos. • Liderar las prácticas de Testing a los perfiles tester de todos los equipos asignados. • Apoyar en la gestión de los pases a producción velando por la calidad de los entregables. • Guiar y fomentar la automatización en los distintos equipos asignados. • Aplicar de Testing Continuo en la automatización de aplicaciones.	Mayo 2021 – Actualidad

Institución y Cargo	Funciones	Año
Entidad Bancaria Quality Assurance Specialist (Especialista Aseguradora de Calidad)	• Construcción de scripts automatizados en aplicaciones de entorno Web con herramientas internas de la entidad bancaria. • Apoyar en la gestión de los pases a producción velando por la calidad de los entregables tanto en proyectos y Reliability. • Análisis, generación y ejecución de casos de prueba para aplicativos en entorno mainframe, Distribuido y Servicios. • Apoyar en la generación de datos mediante scripts automatizados para los casos de prueba. • Creación de scripts para generación de datos de prueba de los principales datos utilizados en entornos previos con la herramienta UFT.	Abril 2020 - Mayo 2021

Institución y Cargo	Funciones	Año
Entidad Bancaria Quality Assurance Training (Aseguradora de Calidad en formación)	• Análisis, generación y automatización de pruebas en entornos web distribuidos con herramientas Open Source WebdriverIO, usando estándares globales. • Construcción de scripts para Servicios Rest con UFT asociadas con Jenkins teniendo la ejecución desde un entorno remoto. • Análisis generación y ejecución de casos de prueba para aplicativos en entorno mainframe, Distribuido y Servicios. • Apoyo en la creación y despliegue de una herramienta web para la búsqueda de datos para pruebas en entornos previos. • Conocimiento en aplicaciones financieras como aplicaciones de inversión, aplicaciones web y móviles de personas naturales y de empresas. • Creación de scripts para generación de datos de prueba de los principales datos utilizados en entornos previos con la herramienta UFT.	Febrero 2019 -Abril 2020

Fuente: Elaboración propia.

En la **tabla 2** se observa la formación profesional de la autora.

Tabla 2

Formación Académica

Formación Académica	Institución	Grado	Año
Ingeniería de Sistemas e Informática	Universidad Nacional Mayor de San Marcos	Bachiller	2013-2018

Fuente: Elaboración propia.

En la **tabla 3** se tienen las certificaciones obtenidas por la autora.

Tabla 3

Certificaciones

Formación	Centro de Certificación	Año
Scrum Master Certified ID:793149	SCRUMstudy	2020
Scrum Fundamentals Certified ID:594645	SCRUMstudy	2017
Scrum Developer Certified ID:744245	SCRUMstudy	2019
SeU™ Certified Selenium Engineer with Java (CSE-Java) ID:Brightest2021012588	Brightest	2021
Holistic Testing ID:000837	Agile Testing Fellowship	2023
ISTQB® Certified Tester - Foundation Level ID:22-CTFL-204598-81	ISTQB	2022
DevOps Essentials Professional Certificate - DEPC ID:FLCWDWBJDQ-SDSSLJGQ-WHFWFQKZKB	Certiprof	2020

Fuente: Elaboración propia.

En la **tabla 4** se listan los idiomas que ha estudiado la autora.

Tabla 4

Idiomas

Idioma	Institución	Nivel	Año
Inglés	ICPNA	Intermedio	2019

Fuente: Elaboración propia.

CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA

2.1 Empresa – actividad que realiza

La empresa es una entidad bancaria con una larga trayectoria (más de 50 años) en el país; siendo parte de un destacado grupo financiero con notable presencia nacional e internacional.

Tiene un gran número de clientes, además brinda una variedad de servicios para personas naturales; pequeñas, medianas y grandes empresas a través de sus canales digitales y oficinas; como préstamos, transferencias, pagos de servicios, retiros, entre otros, todos ellos están orientados a la necesidad específica de cada cliente.

La empresa está en constante mejora continua para innovar sus servicios, ha buscado acelerar su transformación digital, lo cual le permite mejorar la experiencia de sus usuarios, incrementar su fidelización y ubicarse como referente para otras entidades financieras.

2.2 Visión

Ser una de las empresas líderes y referentes en el sector bancario con una conciencia sostenible en el ámbito social y ambiental; digitalizando la mayor cantidad de productos y servicios que permita captar la mayor cantidad de clientes.

2.3 Misión

Apoyar al desarrollo del país generando ocasiones mediante las cuales sus clientes puedan alcanzar sus objetivos a todo nivel, tanto social y personal; brindándoles el apoyo financiero necesario, que les permita crecer de manera conjunta.

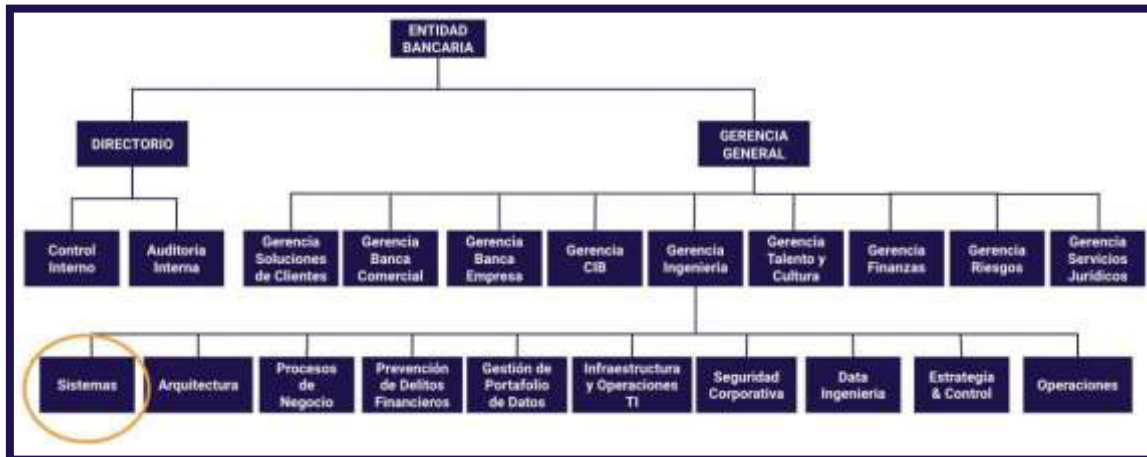
2.4 Organización de la empresa

2.4.1 Organigrama General

En esta sección detallaremos la organización de la empresa con sus principales gerencias:

Figura 1

Organigrama general de la empresa



Nota: El área de Sistemas será detallado a nivel organizacional en la sección 2.4.2.

Fuente: Elaboración propia

A continuación, se detallan las actividades a modo resumen de las distintas áreas de la empresa a nivel gerencial:

A) DIRECTORIO: Es responsable de guiar la estrategia de la organización, así como de supervisar las principales inversiones y gastos que se realizan, aprobar las políticas y reglamentos y velar por la prevención del uso fraudulento de los activos de la entidad.

- **Auditoría Interna:** Es responsable de llevar a cabo una evaluación independiente y objetiva de los riesgos relevantes a los procesos de negocio, de controles para mitigarlos y aspectos legales relacionados.
- **Control Interno:** Diseña y despliega las políticas relacionadas con riesgos no financieros y a una óptima gestión del modelo de control interno de la entidad financiera.

B) GERENCIA GENERAL: Tiene la responsabilidad de dirigir la administración y funcionamiento de la organización la que cuenta con las siguientes áreas:

- **Gerencia Soluciones de Clientes:** Es responsable de crear oportunidades donde los clientes lo requieran; creando soluciones innovadoras, productos que mejoren su experiencia.
- **Gerencia Banca Comercial:** Es responsable de crear la estrategia que asegure la máxima productividad en el negocio.
- **Gerencia Banca Empresas y Corporativa:** Crea la estrategia que proteja la mejor decisión de Negocio, así como el balance de sus resultados.
- **Gerencia Talento y Cultura:** Enfocado en los empleados de la empresa, encargado de buscar los espacios necesarios; apoyando las nuevas dinámicas en los distintos equipos de trabajo.
- **Gerencia Finanzas:** Encargado de optimizar los recursos económicos-financieros; para ello hace uso de herramientas que examinan información gerencial.
- **Gerencia Riesgos:** Garantiza la liquidez de las operaciones a través de un modelo de gestión único, que evalúe y minimice el impacto en los resultados de negocio.
- **Gerencia Servicios Jurídicos:** Asesora a la empresa en temas legales y crea estrategias orientadas a la prevención con el objetivo de cumplir la normativa vigente.
- **Gerencia Ingeniería:** Acompaña a las distintas áreas con el soporte tecnológico necesario, y crea procesos que ayuden a mejorar el rendimiento en las actividades importantes de la organización.

2.4.2 Organigrama del área de Sistemas

Dentro de la gerencia de Sistemas se tienen distintas subgerencias que se organizan de la siguiente manera:

Figura 2

Organigrama de la gerencia de Sistemas y sus distintas subgerencias

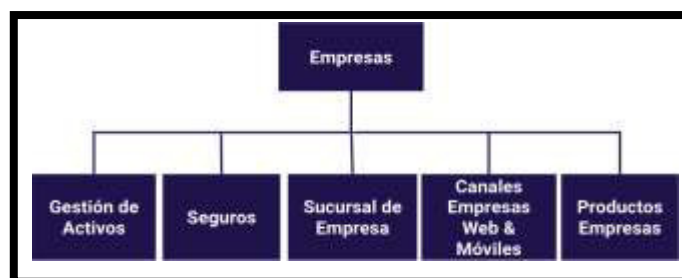


Fuente: Elaboración propia

Dentro de cada subgerencia se tienen distintos equipos agrupados en aplicaciones relacionadas entre sí llamados servicios; en la **figura 3** se grafican los 5 servicios de la subgerencia Empresas.

Figura 3

Estructura de la subgerencia de Empresas y sus servicios asociados

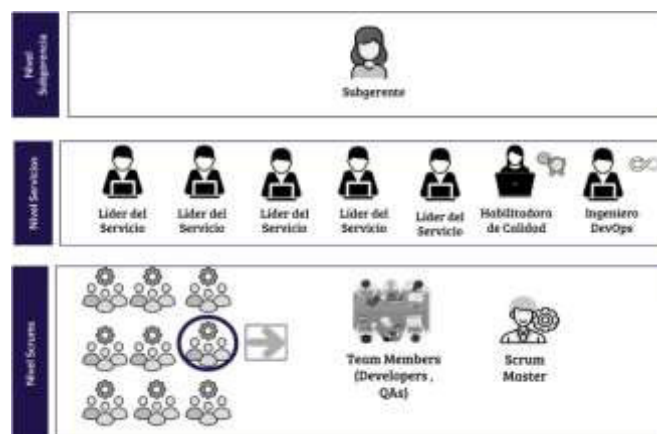


Fuente: Elaboración propia

Dentro de la subgerencia Empresas se tiene 3 niveles que se detallan en la **figura 4**.

Figura 4

Organización de roles por niveles de la subgerencia de Empresas



Fuente: Elaboración propia

A) Nivel Subgerencia: En este nivel corresponde al subgerente:

- **Subgerente:** Quien es el encargado de la visión estratégica del equipo y tiene a su cargo aplicaciones que son gestionadas por los líderes de Servicio.

B) Nivel Servicios: En este nivel está conformado con los siguientes roles:

- **Líderes del Servicio:** Quienes tienen a su cargo distintas aplicaciones y equipos Scrums, los cuales se organizan y distribuyen para el desarrollo técnico de las soluciones priorizadas por el área de Negocio.
- **Ingeniero DevOps:** Quien se encarga en definir, desplegar prácticas y herramientas que ayuden a la mejora en el ciclo de desarrollo del software.
- **Habilitadora de Calidad:** Quien analiza, propone y despliega prácticas que ayuden en la mejora en las pruebas de Software a los distintos equipos Scrums.

C) Nivel Scrums: Este nivel está conformado por:

- **Scrum Master:** Quien lidera a los miembros del equipo (team members) en el desarrollo técnico de los requerimientos encargados por el Product Owner y también en todas las ceremonias ágiles.
- **Team Members:** Quienes desarrollan la solución; en este equipo podemos contar con los desarrolladores; quienes construyen la solución y también QA's; quienes realizan las pruebas de los entregables de los equipos scrums.

Es importante señalar que dentro de las subgerencias aproximadamente el 80 % de los desarrolladores son personal interno de la organización; en promedio el 95% de los QA's son personal externo que trabaja bajo la modalidad de bodyshopping. Como podemos notar en la organización no existe un área de pruebas independiente (como era antes), sino más bien que el personal de pruebas forma parte de cada equipo Scrum.

2.4 Área, cargo y funciones desempeñadas

Dentro de la subgerencia de Empresas la autora desempeñó el cargo de Habilitadora de Calidad teniendo las siguientes funciones:

- **Definición y establecimiento de un nuevo proceso de Pruebas de Software;** dentro del cual se establecen nuevos objetivos y KPIs de la organización.
- **Establecimiento de los lineamientos de prueba en los equipos Scrums;** coordinando con las empresas externas que ofrecen servicio de Pruebas de Software a la entidad bancaria se aseguró que los roles involucrados en el proceso de pruebas:
 - 1) Utilicen herramientas de priorización de funcionalidades para definir un plan de pruebas en base al impacto de las funcionalidades.
 - 2) Apliquen técnicas de diseño de casos de prueba.
 - 3) Redacten los casos con un lenguaje único y entendible a todos los miembros del equipo.
 - 4) Conozcan el uso de una herramienta para la gestión de casos de prueba.
- **Configuración de herramientas de integración continua** como Jenkins, Jira y Bitbucket para la implementación de Testing Continuo y pruebas funcionales automatizadas con herramientas internas de la entidad bancaria.
- **Validación de documentación de pruebas realizadas dentro de los equipos Scrum;** correspondiente a funcionalidades nuevas, incrementos funcionales y mantenimiento, asimismo la confirmación al equipo de gestión de Cambios que se encarga de los despliegues a producción.

2.5 Experiencia profesional realizada en la organización

Entre las actividades desempeñadas en la empresa la autora formó parte de diferentes equipos y proyectos los cuales se detallan de la siguiente manera:

a) Dentro de la Subgerencia de Empresas los logros obtenidos son:

- **Definición y despliegue de buenas prácticas para pruebas manuales y automatizadas en los equipos Scrum:** Se realizó un análisis AS IS al proceso de pruebas de Software, teniendo en cuenta los puntos de mejora se propuso un nuevo proceso para la estandarización y organización en las pruebas de los distintos equipos Scrum.

Se definieron una serie de lineamientos, políticas a fin de optimizar y lograr una mejor trazabilidad entre los casos de prueba, los requerimientos funcionales y los defectos en los distintos equipos Scrum.

- **Implementación de Testing Continuo en aplicaciones de alto valor para el negocio:** Se analizaron los principales flujos de dichas aplicaciones y se realizó el diseño, ejecución y automatización de los principales casos de pruebas.

Dentro del equipo de Calidad los logros obtenidos son:

- **Definición de la arquitectura de un sistema web para la búsqueda de Data de Pruebas:** Se hizo el diseño de una web en PHP, implementando las interfaces en HTML y conectando la base de datos PostgreSQL para la búsqueda de datos de pruebas; el cual nos permite consultar datos de clientes y sus productos asociados.
- **Creación de Scripts automatizados:** Se construyeron scripts automatizados en entornos Mainframe IBM 3270, Servicios Rest, para los flujos más críticos en la generación de datos de prueba como son la creación de clientes y sus productos (cuentas, tarjetas, etc).
- **Análisis y definición de la estrategia de pruebas en distintos equipos Scrums:** Se apoyó en la definición de la estrategia de pruebas funcionales, de regresión y automatizadas en los distintos equipos scrum asignados.
- **Generación de datos de pruebas en entornos previos:** Generación de los datos más críticos de clientes y productos para las pruebas de las aplicaciones asignadas.

CAPÍTULO III - ACTIVIDADES DESARROLLADAS

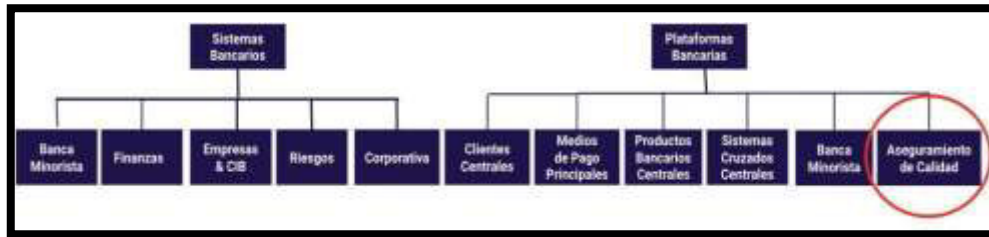
3.1 Situación problemática

Las entidades financieras actualmente cuentan con canales digitales mediante los cuales sus clientes pueden realizar operaciones sin necesidad de ir a sus oficinas. Por ello tienen la necesidad de realizar mejoras incrementando y/o agregando funcionalidades nuevas que les permitan aumentar la captación de usuarios a quienes les puedan ofrecer sus productos y servicios. Siendo necesario que cada entregable que se ponga a disposición del usuario final cuente con la calidad adecuada que asegure su correcto funcionamiento en el menor tiempo posible.

A nivel organizacional el área de Sistemas de la entidad financiera enfrentaba los siguientes principales retos:

- **Evitar incidentes relevantes en las aplicaciones:** Se tiene la necesidad de evitar que los servicios críticos caigan o estén no disponibles debido a un pase a producción inapropiado. La indisponibilidad y errores dentro de una aplicación generan desconfianza en los usuarios, una mala imagen y con ello la pérdida de oportunidades para incrementar las ventas de nuevos productos.
- **Reducir el tiempo de entrega de las soluciones al cliente:** Debido a la competitividad del entorno bancario se ha hecho necesario innovar y ofrecer nuevos productos y servicios en menor tiempo al usuario final.
- **Descentralización del área de Calidad:** Con el fin de buscar la agilidad la organización reestructuró el área de Calidad, la cual se encontraba centralizaba y efectuaba las pruebas funcionales de los distintos equipos Scrum, los analistas de control de calidad que formaban parte del equipo pasaron a formar parte de los distintos equipos de Sistemas Bancarios y Plataformas Bancarias. En la siguiente figura se puede ver en la **figura 5** cómo estaba organizada el área de Sistemas cuando se tenía un equipo de Calidad centralizado.

Figura 5
Organización antigua del área de Ingeniería

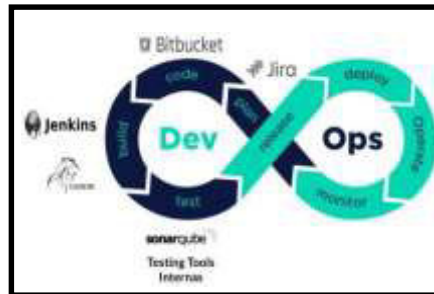


Fuente: Elaboración propia

- **Mejoras realizadas en el desarrollo de Software con la implementación de herramientas DevOps pero no en el proceso de Pruebas:** Con el objetivo de agilizar y mantener la trazabilidad de los desarrollos los Ingenieros DevOps iniciaron el despliegue de lineamientos, herramientas que permitan automatizar ciertas tareas a nivel de desarrollo como **Jira** para la gestión de Requerimientos, **Bitbucket** para el almacenamiento de código de forma segura, **Jenkins** para automatizar el build y deploy de las aplicaciones, **SonarQube** y **herramientas propias del corporativo** para la automatización de pruebas funcionales, **Chimera** para validar que los entregables cumplen criterios de calidad y seguridad a nivel de código. A pesar que el **Jira** contaba con un complemento para gestionar las pruebas llamado **Xray** este no era utilizado; las pruebas funcionales se desarrollaban en documentos independientes. Por ejemplo, el plan de pruebas se hacía en un documento excel nombrado C204 y para las evidencias se registraban en un documento word denominado T907, lo cual no permitía ubicarlos con facilidad y generaba retrabajo tener que diseñarlos nuevamente.

Figura 6

Implementación de herramientas DevOps en el ciclo de desarrollo de Software.



Fuente: Elaboración propia.

A nivel de equipos dentro de los distintos Scrum se tiene como principales retos:

- **Desarrollo ágil con un equipo multidisciplinario:** Dentro de un mismo equipo scrum se tienen diversos roles, como desarrolladores frontend, arquitectos de base de Datos, desarrolladores Backend y analistas de pruebas que trabajan bajo la metodología Scrum por periodos de cada 3 meses llamados Quarters.
- **Aseguramiento de calidad de los entregables de pruebas desarrollados en etapas tempranas (inicio del desarrollo):** El diseño de los casos de prueba se puede ir trabajando con un requerimiento correctamente definido como una feature y/o historia de usuario, mientras el equipo inicia el desarrollo técnico de la solución para que luego puedan ser ejecutadas cuando se reciba la funcionalidad dentro del sprint.

A nivel de las aplicaciones desarrolladas dentro de los principales canales digitales contamos con entornos Web y móviles los cuales enfrentan otros retos a revisar como:

- **Distintos fabricantes en Sistemas operativos móviles:** En las aplicaciones móviles tenemos dos grandes sistemas operativos como **iOS** en el cual Apple es su único propietario y **Android** dentro del cual se manejan distintos fabricantes como Samsung, LG, Motorola, Huawei cada uno de ellos tiene sus propias personalizaciones a nivel de producto lo cual hace que una funcionalidad pueda funcionar correctamente dentro de un Samsung, pero no en un Huawei.

- **Componentes de desarrollo compartidos con otras dependencias en diferentes países y en más de una aplicación:** La entidad financiera al tener presencia en varios países; ha optado por el desarrollo a nivel de código en componentes comunes que puedan ser reutilizados en una misma aplicación, lo cual implica que si hay un cambio de ese componente en un país puede impactar en las aplicaciones de los otros países por ellos se hace necesario hacer validaciones constantes.

3.1.1 Definición del problema

Problema principal

Asegurar un adecuado control de calidad en el desarrollo y despliegue de software en una organización financiera que atraviesa varios retos a nivel de organización del área de Sistemas, equipos Scrums y del proceso de desarrollo de aplicaciones mencionados en la sección 3.1; los cuales se agudizan debido a la falta de un proceso de pruebas que garantice una óptima gestión en el diseño, ejecución y el seguimiento de los casos de prueba como también los defectos identificados; lo cual impiden la trazabilidad entre los requerimientos y las pruebas realizadas por los equipos.

Problemas Secundarios

1. **Falta de priorización en las funcionalidades de las aplicaciones que permitan asegurar su continuidad operativa al momento de definir el plan de pruebas:** Dado que no es posible probar todas las funcionalidades de un aplicativo por temas de capacidad y tiempo.
2. **Falta de un adecuado diseño de casos de prueba:** Necesidad de un mecanismo que nos permita sustentar su diseño. Actualmente en los entregables que forman parte del proceso de pruebas se ven impactados debido a que los casos de prueba se realizaban a juicio experto que era susceptible a un error u omisión lo que conllevaba a incidencias productivas.
3. **Falta de estandarización en la nomenclatura y redacción de los casos de prueba que no permiten su reutilización;** los casos de prueba que escribían los equipos no eran completos y sencillos de entender, ello traía como consecuencia que tuviéramos casos duplicados descritos de diferente manera.

4. **Falta de una herramienta de gestión de pruebas que permita centralizar los casos de prueba a fin de que integren como parte del ciclo de desarrollo de software** ya que actualmente los entregables de prueba se realizaban con documentación independiente de la herramienta de gestión de los requisitos; ello no permitía ubicarlos con facilidad generando retrabajo al describirlos nuevamente.

3.2 Solución

Diseñar un proceso que permita organizar y mejorar la gestión de los casos de prueba, así como optimizar los tiempos incurridos en las pruebas.

Asimismo, utilizar estándares globales como **ISTQB** que aseguren la calidad de los entregables de pruebas antes de su puesta en producción.

Las 4 fases del proceso de pruebas propuesto se pueden apreciar en la **figura 7**.

Figura 7

Fases del proceso de pruebas propuesto



Nota: La figura muestra las fases propuestas en el ciclo de pruebas en las cuales se desarrolló el proyecto. Fuente: Elaboración propia.

A continuación, se explicarán las 4 fases del proceso propuesto:

- 1. Análisis de requerimientos:** En esta fase se recepciona un requerimiento a validar, se determina el alcance de las pruebas, se define la estrategia a seguir y se identifica si la funcionalidad desarrollada es nueva o existente para aplicar el enfoque de pruebas basadas en riesgos que se detalla en la sesión

En esta fase es importante tener en cuenta la criticidad de una funcionalidad para determinar el impacto de la misma en las pruebas.

- 2. Diseño y redacción:** En esta fase se diseñan los casos de prueba; como principal práctica para el diseño se propuso utilizar técnicas que nos ayuden a optimizar y asegurar una óptima cobertura.

Adicionalmente para la redacción de los casos de prueba se propusieron lineamientos a nivel de nomenclatura y de escritura; como por ejemplo la descripción de los casos de prueba se realizarán en Gherkin un lenguaje de fácil comprensión para todos los miembros del equipo.

- 3. Implementación:** En esta etapa se configura el entorno de pruebas, los datos necesarios para los casos de prueba diseñados y también pueden considerarse la construcción de scripts automatizados en herramientas Open Source o en una propuesta por la propia organización.
- 4. Ejecución:** En esta etapa se ejecutan los casos de prueba diseñados; como buena práctica se propuso una herramienta de gestión para el ciclo de vida de las pruebas Xray; por ser complemento de Jira la herramienta de gestión de requerimientos de la organización, que permita dar visibilidad y trazabilidad de los avances en los entregables de prueba.

En la **figura 8** se describen gráficamente las principales prácticas utilizadas en el proceso propuesto.

Figura 8

Principales prácticas utilizadas en el proceso propuesto



Fuente: Elaboración propia

3.2.1 Objetivos

Objetivo General

Diseñar e implementar un proceso integral de pruebas de software en la entidad bancaria que contribuya a mejorar la calidad de los requerimientos del proyecto mediante una adecuada gestión de los casos de prueba y la optimización de los tiempos incurridos usando herramientas.

Objetivos específicos

1. Definir e implementar un mecanismo que nos permita listar y evaluar la criticidad de las funcionalidades de las aplicaciones para priorizarlas en el plan de pruebas.
2. Mejorar el diseño de casos de prueba mediante la aplicación de técnicas para optimizar su cobertura.
3. Establecer estándares en la redacción de los casos de prueba mejorando su calidad para fomentar su reutilización.
4. Definir e implementar una herramienta de gestión de casos de prueba que permita una trazabilidad y el seguimiento al proceso de pruebas.

3.2.1 Alcance

A continuación, se detallan los alcances en los siguientes ámbitos:

Alcance Funcional

El alcance funcional del proyecto incluye:

- La definición e implementación de las pruebas basadas en riesgos para las funcionalidades del aplicativo.
- Aplicación de técnicas para el diseño de casos de pruebas.
- Lineamientos a nivel de nomenclatura y escritura para los casos de prueba.
- Redacción del caso de prueba en un lenguaje comprensible al equipo.
- Trazabilidad en la ejecución de los casos de prueba y sus defectos para las pruebas funcionales de canales digitales dentro de la organización.

Las 4 fases del proceso de pruebas propuesto tendrán impacto dentro del SDLC.

Alcance Organizacional

Comprende la Subgerencia de Empresas de la entidad financiera y estuvo alineado al cumplimiento de los KPIs organizacionales de optimización de tiempos en los despliegues.

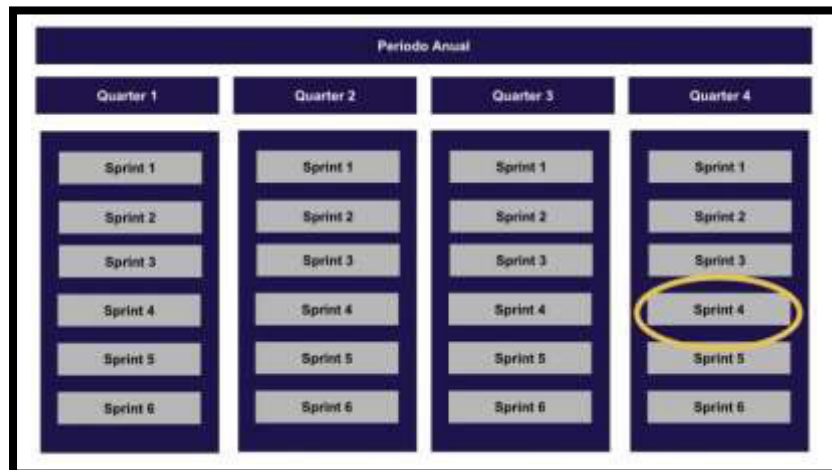
3.2.2 Etapas y metodología

3.2.3.1 Scrum y el ciclo de vida de desarrollo de software (SDLC)

Dentro de la organización los desarrollos se llevan a cabo en periodos trimestrales llamados Quarters; en un año se tienen 4 Trimestres dentro del cual se aplica la metodología Scrum considerando 6 sprints como se muestra en la **figura 9**.

Figura 9

Esquema de períodos de desarrollo ágil

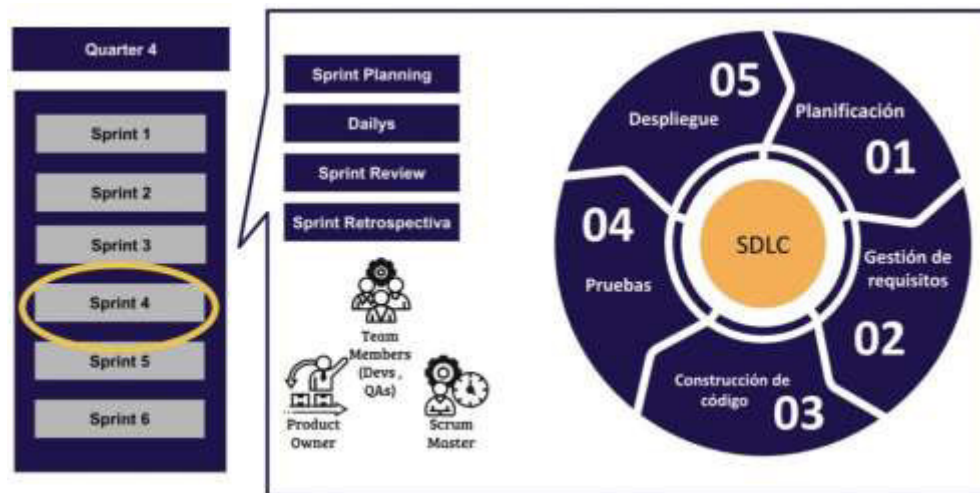


Fuente: Elaboración propia

Dentro de cada Sprint se definen, priorizan los requerimientos para luego ser implementados siguiendo las fases del SDLC; en los cuales se tienen distintas ceremonias, roles como se muestran en la **figura 10**.

Figura 10

Ceremonias y actividades realizadas dentro de un sprint



Fuente: Elaboración propia

Dentro de cada **sprint** se establecen y completan actividades para crear un nuevo requerimiento solicitado por negocio; durante un intervalo de 2 semanas se llevan a cabo **ceremonias** como:

- **Sprint Planning:** Definición y priorización de las actividades a realizar en base a los requerimientos.
- **Dailys:** Seguimiento diario a los avances realizados con el equipo.
- **Sprint Review:** Seguimiento al finalizar el sprint para validar el incremento desarrollado.
- **Sprint Retrospectiva:** Sesión con el equipo donde se relevan puntos de mejora para aplicarlos en el siguiente sprint.

Para el desarrollo de las ceremonias en cada sprint se cuentan con los **siguientes roles**:

- **Scrum Master:** Quien es encargado de liderar las ceremonias y al team member en el desarrollo técnico de la solución.
- **Team Member:** Equipo que implementa la solución.
- **Product Owner:** Visión de la necesidad del cliente y da feedback al equipo acerca de los avances realizados.

Dentro de cada sprint se implementa el **SDLC** en la siguiente **figura 11** se detallan cada una de sus fases.

Figura 11
Fases del SDLC y su definición



Fuente: Elaboración propia

3.2.3.2 Proceso de pruebas propuesto y el SDLC

Según **ISTQB (2021)** es importante tener en cuenta el SDLC para definir una adecuada estrategia de pruebas. Asimismo, cuando se hacen referencia a desarrollos ágiles, las pruebas pueden y deben contemplarse desde el inicio.

Holistic Testing (2022) nos señala que las pruebas pueden iniciarse desde el refinamiento de un requerimiento con una adecuada historia de usuario y los criterios de aceptación claros podemos identificar los riesgos no contemplados con anticipación.

Soler (2022) comenta que bajó del enfoque de pruebas ágiles, cuando las pruebas que suceden en conjunto con el desarrollo permiten a los QAs de los equipos descubrir errores con mayor frecuencia y en cada fase del proceso del SDLC.

Es por ello que el proceso propuesto considera las pruebas tempranas desde la definición de requerimientos hasta su despliegue a producción.

En la **figura 12** se ven las fases del SDLC y la relación con las 4 fases del proceso propuesto. La capa externa es el proceso de pruebas propuesto y la capa interna es el SDLC.

Figura 12

Relación de las fases del SDLC y el proceso de pruebas propuesto



Fuente: Elaboración propia

Mientras se define el requerimiento en la primera fase (**Planificación**) y prioriza en la segunda (**Gestión de requisitos**), el QA apoya en el refinamiento de la Historia de Usuario y sus criterios de aceptación en la fase de **Análisis del requerimiento** aparte de tener en cuenta la criticidad de la funcionalidad para definir la estrategia de pruebas.

Cuando el equipo inicia la tercera fase **Construcción de código** el QA en base al requerimiento refinado previamente va diseñando los casos de prueba en la fase de **Diseño y Redacción**. Adicionalmente el QA define el plan de pruebas con los datos de prueba necesarios, además de configurar el entorno de pruebas ya sean para pruebas manuales y automatizadas durante la fase de **implementación**.

En la fase de **pruebas** el equipo tiene que validar su desarrollo; como ya se tienen definidos los casos de prueba en fases previas el QA puede realizar la verificación de dichos casos de forma manual o automatizada en la etapa de **Ejecución** para que luego puedan ser promovidos al cliente en la etapa de **Despliegue**.

En la siguiente sección se detallarán cada una de las fases del modelo propuesto teniendo en cuenta las herramientas, lineamientos, aplicación y consideraciones.

3.2.3.2.1 Análisis del requerimiento

ISTQB (2021) indica que en esta fase se define el alcance de las pruebas el cual se apoya en fuentes de conocimiento e información como documentación de requerimientos, features e historias de usuario. Es importante analizar y priorizar los criterios de aceptación para poder relacionarlos con los casos de prueba necesarios para su validación. Recomiendan seguir una estrategia de pruebas basadas en riesgos la cual el QA pueda identificar con los PO la criticidad de cada funcionalidad.

IBM (2021) señala que las pruebas basadas en riesgos ayudan al equipo a priorizar bajo ciertos criterios, funcionalidades según su importancia y la probabilidad del impacto en su fallo.

Mena (2022) indica que las pruebas basadas en riesgos previenen el riesgo del fallo de una funcionalidad y su impacto en el negocio. También nos da algunos beneficios como: mejorar la experiencia del cliente, facilita la definición del plan de pruebas mediante la priorización, ayuda a la optimización del tiempo del equipo ya que prioriza sus esfuerzos en pruebas que son de alto valor, aclara en la cobertura y el alcance de prueba para los equipos interesados.

Las pruebas basadas en riesgos son importantes por las siguientes razones:

- Identifica eficientemente la probabilidad de errores y se define en base al requerimiento (historia de usuario/feature), cual debe priorizarse al momento de definir la estrategia de pruebas.
- Permite tener varias evaluaciones para cada criterio de valoración, ya que incentiva distintos puntos de vista y puntuaciones, lo cual da un panorama de cada factor de riesgo y evita una opinión sesgada.
- Optimiza tiempos y da mayor relevancia a los puntos más importantes durante la etapa de pruebas.
- Permite la identificación de riesgos de las funcionalidades de una aplicación y su uso en el análisis, diseño y redacción, implementación y ejecución de pruebas.

En el siguiente apartado se explicarán a mayor detalle el uso de las pruebas basadas en riesgos.

01. Herramienta

Matriz Risk Based Testing es una herramienta que permite listar las funcionalidades y valorar la criticidad de las mismas bajo los atributos como Impacto y Probabilidad.

Figura 13

Matriz de pruebas basadas en Riesgos

Pruebas basadas en Riesgos				
Aplicativo	Funcionalidad	Impacto	Probabilidad	Nivel de Riesgo
App SW	Login	Bajo	Alto	RIESGO MODERADO
App SW	Funcionalidad 2	Medio	Alto	RIESGO ALTO
App SW	Funcionalidad 3	Alto	Alto	RIESGO ALTO
App SW	Funcionalidad 4	Bajo	Bajo	RIESGO BAJO
App SW	Funcionalidad 5	Medio	Medio	RIESGO MODERADO
App SW	Funcionalidad 6	Alto	Medio	RIESGO ALTO

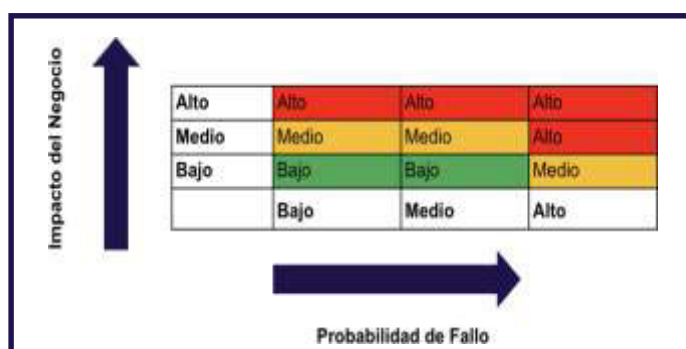
Fuente: Elaboración propia

Según **ISTQB (2021)** nos señala que la determinación del nivel de riesgo suele implicar la evaluación, de la **probabilidad e impacto del riesgo**.

En la **figura 14** se muestra la relación entre **Impacto (Orientado al negocio donde se analiza la funcionalidad y el impacto con respecto al cliente, organización)** y la **Probabilidad (Relacionada a la complejidad de una funcionalidad al momento de desarrollarla)** dependiendo de la valoración asignada podrá determinar un factor de Riesgo.

Figura 14

Esquema de dimensiones de la matriz de pruebas basadas en Riesgo y su valoración.



Fuente: Elaboración propia.

Nos permite catalogar las funcionalidades de una aplicación de la

siguiente manera:

- **Riesgo Alto:** Riesgo crítico, se requieren acciones preventivas sobre la funcionalidad.
- **Riesgo Moderado:** Riesgo considerable, se pueden introducir acciones preventivas para reducir el nivel de riesgo sobre la funcionalidad.
- **Riesgo Bajo:** Riesgo marginal debe controlarse, pero no requiere acciones preventivas sobre la funcionalidad.

02. *Lineamientos*

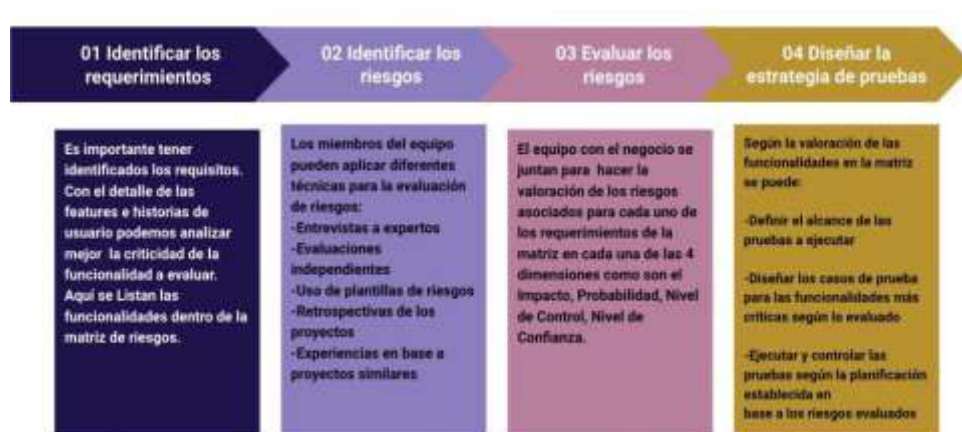
- Uso de la Matriz de Riesgos para determinar la criticidad de las funcionalidades que van siendo implementadas dentro de una aplicación.
- Considerar la funcionalidad a implementar sobre la cual se desarrollarán los casos de prueba al momento de registrar un nuevo requerimiento.

03. *Aplicación*

Para la aplicación de la matriz de pruebas basadas en riesgos es importante tener en cuenta los siguientes pasos mostrados en la **figura 15**.

Figura 15

Pasos para la aplicación de las pruebas basadas en riesgos



Fuente: Elaboración propia

04. Consideraciones

Al momento de realizar el uso de las pruebas basadas en riesgos es importante tener en cuenta lo siguiente:

- **Involucrar a los miembros del equipo interesado en el análisis:**
Trabajar con equipos tanto como técnicos y de negocio es esencial para poder identificar todos los riesgos dentro de una aplicación.
- **Hablar de riesgos en un lenguaje de fácil comprensión:** La comunicación es esencial para el éxito de un requerimiento por ello cuando se habla de riesgos es necesario que se explique de una manera sencilla de entender.
- **Definir un enfoque de prueba teniendo en cuenta los resultados obtenidos:** Riesgos críticos deben ser enfocados de manera diferente, con mayor cobertura, criterios más estrictos, etc. Este enfoque debe ser definido de manera previa, de ese modo se puede dar un uso más efectivo de los recursos de testing.

3.2.3.2.2 Diseño y Redacción

Uno de los insumos importantes en el proceso de pruebas son los casos de prueba; **IBM (2022)** señala que un caso de prueba nos ayuda a definir lo que vamos a validar con el objetivo de que nuestro sistema funcione correctamente.

ISTQB (2021) señala que en esta etapa se **diseñan los casos de prueba** necesarios en base a las técnicas de diseño para cubrir el requerimiento.

Manyela & Andrews (2021) indican que el uso de técnicas de prueba es importante para prevenir riesgos y detectar fallos en el sistema.

Juarez (2020) nos comenta que existen distintos tipos de técnicas que nos permiten seleccionar la menor cantidad de casos con mayor probabilidad de encontrar errores en el sistema; dado que no podemos realizar pruebas exhaustivas teniendo tiempos acotados.

Veenu Singh et al (2023) indica que es importante priorizar e identificar las variables del requerimiento que influyen en el proceso de pruebas.

Dentro de las entidades financieras se cuentan con aplicaciones que poseen muchos datos de entrada que necesitan ser completados para hacer una operación de manera exitosa, por ejemplo, una transferencia o un pago de servicios dentro de una banca móvil o una banca por internet donde tenemos que seleccionar ya sea la cuenta de cargo, la cuenta de abono, el monto de la operación, etc.

Es por ello que se propone el uso de técnicas de diseño de casos de prueba que ayuden a la cobertura de los entregables de la organización.

La **redacción de casos de prueba** suele ser un punto que pasa desapercibido, pero suele ser súper importante porque:

- **Facilita la colaboración del equipo:** Casos de prueba bien redactados son sencillos de entender para los miembros del equipo de desarrollo e interesados apoyando en su constante comunicación.
- **Ayuda en la verificación de requerimientos:** Casos de prueba bien escritos ayudan a validar adecuadamente los requisitos.
- **Facilita su reutilización:** Casos de prueba correctamente diseñados son fácilmente replicables para validar el correcto funcionamiento del sistema lo cual ayuda a identificar y corregir errores de manera efectiva.
- **Impulsa la automatización de pruebas:** Casos de prueba correctamente coberturados son necesarios para la automatización de pruebas ya que nos permite ahorrar tiempo y recursos.
- **Sirve como una documentación viva:** Ayuda a los miembros del equipo a comprender la funcionalidad sobre todo cuando hacen mantenimiento y actualizaciones al sistema.

Según **ISTQB (2021)** los casos de prueba deben ser comprensibles para todas las partes interesadas del equipo de desarrollo e interesados; además deben tener objetivos claros, con criterios que indiquen si son casos positivos o casos fallidos para que puedan ser replicables, verificables y traceables.

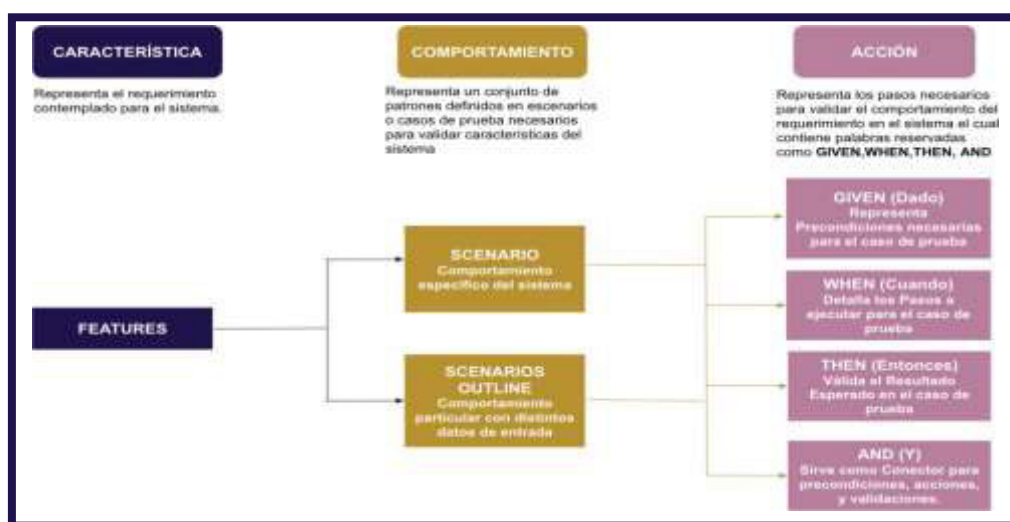
Scaled Agile Framework (2023) dentro de un artículo nos relata acerca de **Behavior Driven Development (BDD)**, desarrollo orientado por el comportamiento, como práctica de equipos ágiles que ayuda a incorporar la calidad desde la definición anticipada de pruebas o como parte de la definición

del comportamiento del sistema, crea un entendimiento compartido entre los equipos ágiles y el negocio. Como objetivo tiene ayudar a guiar el desarrollo, disminuir el retrabajo con la mejor comprensión del equipo. Las pruebas realizadas con esta práctica son escenarios orientados al comportamiento de un requerimiento desde el punto de vista del usuario; cuando se automatizan estos tipos de prueba garantizan que el sistema cumple continuamente el comportamiento especificado incluso cuando existan incrementos funcionales.

BrowserStack (2023) comenta que BDD implica el uso de un lenguaje compartido que mejora la comunicación entre los miembros del equipo técnicos y no técnicos donde las pruebas se centran más en el usuario y en el comportamiento del sistema. Gherkin usado en BDD, es propuesto para la definición de casos de prueba.

Gherkin es un lenguaje de dominio específico (DSL), el cual se caracteriza por su simplicidad y legibilidad. Lo especificado en Gherkin es sencillo de entender por cualquier miembro del equipo tanto para técnicos y no técnicos; el cual es utilizado para definir escenarios de prueba, casos de prueba y comportamientos esperados de una aplicación. En la **figura 16** se muestra la estructura del Gherkin.

Figura 16
Estructura del Gherkin



Fuente: Elaboración propia

En la **figura 17** se muestra un ejemplo de redacción de un caso en Gherkin.

Figura 17

Ejemplo de la redacción de un caso de prueba en Gherkin



Fuente: Elaboración propia

01. Herramienta

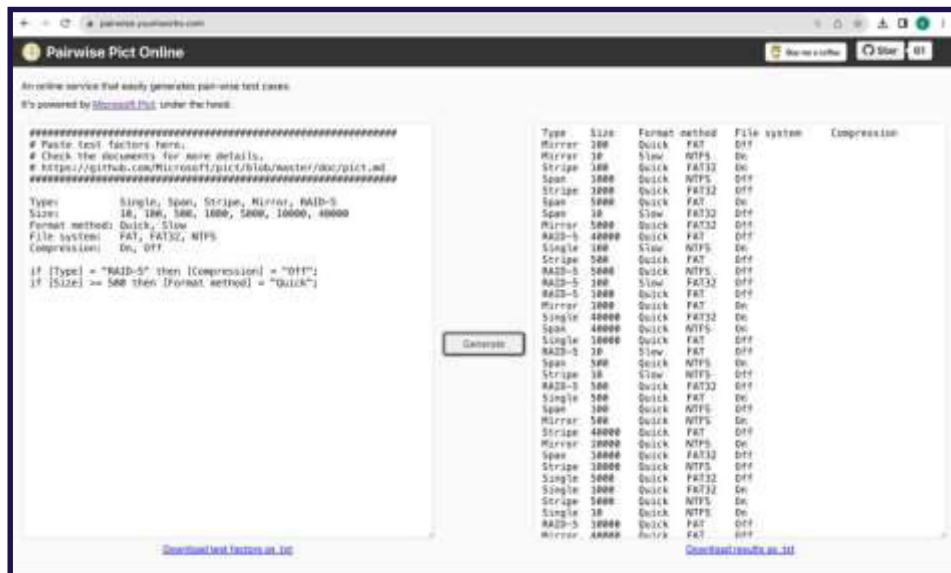
Entre las distintas técnicas de casos de prueba que tenemos se propuso el Pairwise. Según **Kulcsár et al (2020)** indican que la técnica de comparación por pares facilita la identificación por riesgos que de otro modo serían difíciles de evaluar.

Fadhil et al (2022) indica que el uso de técnicas combinatorias puede optimizar el diseño de los casos de prueba hasta un factor del 70 % al 90 %.

ISTQB (2021) detalla diversas técnicas para el diseño de casos de prueba de caja negra como partición de equivalencia, valores límites, tablas de decisión entre las cuales resalta la técnica de Pairwise: esta es recomendada cuando tenemos varios parámetros de entrada, la técnica nos garantiza que todos los parámetros sean validados y coberturados. También nos indica que en www.pairwise.org existen distintas herramientas disponibles que hacen uso del pairwise y que pueden ayudar en esta actividad al QA del equipo. En la **figura 18** mostraremos una web de libre acceso donde se puede aplicar la técnica de Pairwise.

Figura 18

Página web donde se puede utilizar el Pairwise



Fuente: Elaboración propia

02. Lineamientos

- Uso de técnicas para el diseño de los casos de prueba en los entregables de pruebas.
- Generación de una combinatoria de pruebas basada en los parámetros y alcance del requerimiento a validar.
- Propuesta para nombrar el caso de prueba se mostrará en la **figura 19**.

Figura 19

Nomenclatura propuesta para los casos de prueba



Fuente: Elaboración propia

- Propuesta para redactar el caso de prueba en Gherkin se muestra en la **figura 20**.

Redacción del lenguaje Gherkin y sus palabras reservadas

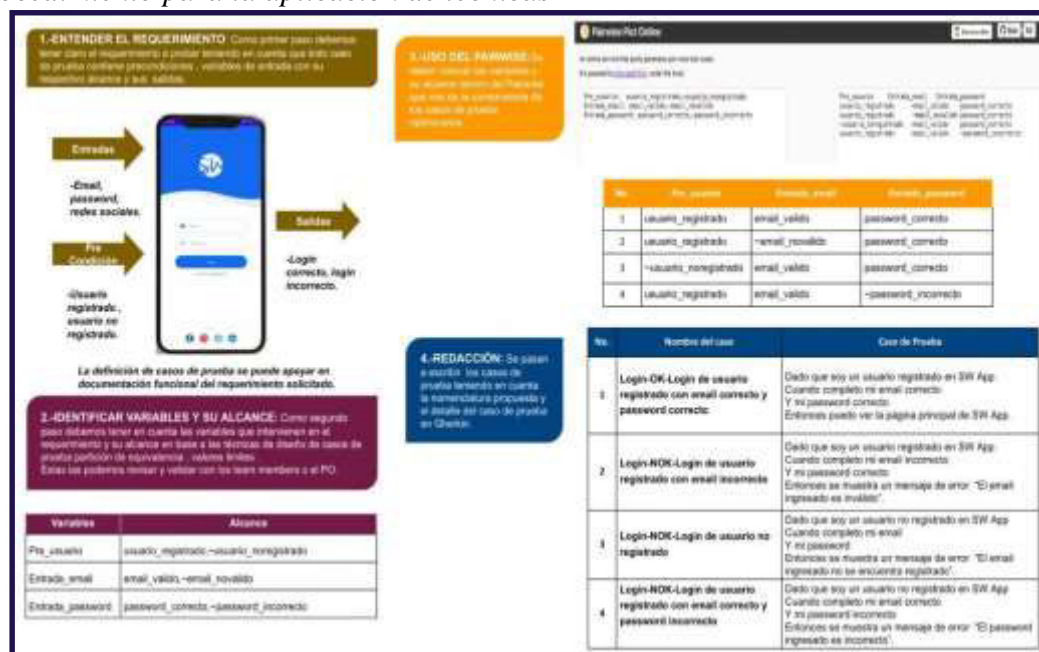


Fuente: Elaboración propia

03. *Aplicación*

En la **figura 21** se muestran las fases de la aplicación de técnicas con sus 4 fases propuestas y su explicación: Entender el requerimiento, Identificar variables y su alcance, Uso de Pairwise y la Redacción.

Procedimiento para la aplicación de técnicas



Fuente: Elaboración propia

04. Consideraciones

Al momento de diseñar y redactar un caso de prueba es importante tener en cuenta las variables que intervienen dentro del requerimiento a desarrollar, debemos considerar las que intervengan de manera directa en la funcionalidad a validar.

Los QAs deben contar con conocimiento previo de técnicas de diseño de casos de prueba y redacción.

3.2.3.2.3 Implementación

ISTQB (2021) indica que en esta etapa se realizan actividades como:

- Definición del plan de pruebas y priorización de los casos de prueba de acuerdo al alcance del requerimiento.
- Creación de scripts de pruebas automatizadas.
- Búsqueda y creación de los datos necesarios para la ejecución de los casos de prueba.
- Preparar el entorno donde se ejecutarán los casos de prueba.
- De utilizarse una estrategia de pruebas basadas en riesgo, el nivel de riesgo será la consideración primordial para la ejecución de los casos de prueba.

01. Herramienta

En la organización se vio la necesidad de poder centralizar los casos de prueba por requerimientos teniendo una herramienta que nos permita gestionarlos, asimismo tener registrados los planes de prueba generados por los equipos, ello debido al área que Auditoría siempre hace revisiones periódicas de los entregables de calidad para ello se propuso Xray como complemento de Jira, esta herramienta será detallada en la fase de ejecución.

3.2.3.2.4 Ejecución

ISTQB (2021) señala que en esta etapa se ejecutan los casos de pruebas dentro del cual se realizan las siguientes actividades:

- Ejecución de pruebas manuales tanto como pruebas exploratorias

- Ejecución de pruebas automatizadas
- Comparación de resultados reales con los esperados
- Reporte de defectos en base a los fallos observados.
- Se crea una instancia de ejecución para la trazabilidad del plan de pruebas y las evidencias de las pruebas realizadas.

En esta etapa es importante contar con un gestor de casos de prueba porque permite:

- **Estructuración y agrupación:** Permite organizar y estructurar los casos de prueba de manera eficiente. Ello ayuda a ubicar fácilmente los requerimientos con las funcionalidades probadas y ver sus resultados probados.
- **Reutilización de casos de prueba:** La reutilización ahorra tiempo y esfuerzo. Permite almacenar los casos de prueba fácilmente y utilizarlos para diferentes releases que genere un aplicativo.
- **Trazabilidad**
Proporcionan trazabilidad entre requisitos, casos de prueba y resultados. Esto es esencial para demostrar que todas las funcionalidades y requerimientos del software han sido validados de manera adecuada.
- **Seguimiento de resultados**
Permiten realizar un seguimiento eficiente de los resultados de la ejecución de pruebas. Esto facilita la identificación y resolución rápida de defectos, así como la generación de informes detallados sobre el estado de la calidad del software.
- **Colaboración y comunicación**
Facilita la colaboración entre los miembros del equipo de pruebas y otros interesados, como desarrolladores, PO. Todos los involucrados pueden acceder a la misma información sobre casos de prueba, resultados y defectos.
- **Automatización de pruebas**
Algunos gestores de casos de prueba están integrados con herramientas de automatización, lo que facilita la transición de casos de prueba manuales a casos de prueba automatizados. Esto mejora la eficiencia y la cobertura de las pruebas.
- **Mejora Continua**
Facilitan la identificación de áreas de mejora en los procesos de prueba. El análisis de datos históricos y la generación de informes permiten tomar decisiones informadas para mejorar la eficacia y eficiencia de las actividades de prueba.

01. Herramienta

Como herramienta para la ejecución de pruebas manuales se propuso el uso del Xray complemento de Jira que se adapta perfectamente a la que utiliza la organización.

Según **Atlassian (2023)** indica que Xray fue creado para la gestión de las pruebas, en la cual todo el equipo puede planificar, ejecutar, realizar el seguimiento de los entregables de prueba de sus requerimientos.

En la cual podemos:

- **Manejar los casos de prueba como Issues de Jira**

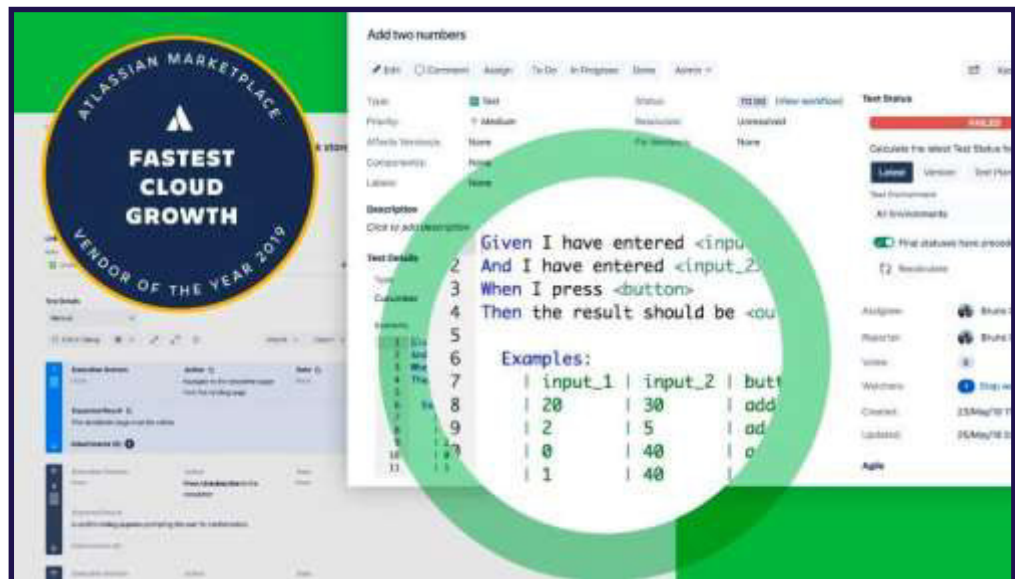
Nos permite gestionar las pruebas manuales y automatizadas como issues de Jira, personalizar pantallas, campos y flujos de trabajo como se observa en la **figura 22**.

Nos permite además especificar los casos de prueba en cucumber e integrarlos con marcos de automatización de pruebas.

Crear casos de prueba en carpetas y agrupadores de pruebas llamados test set.

Figura 22

Xray y la creación de Test



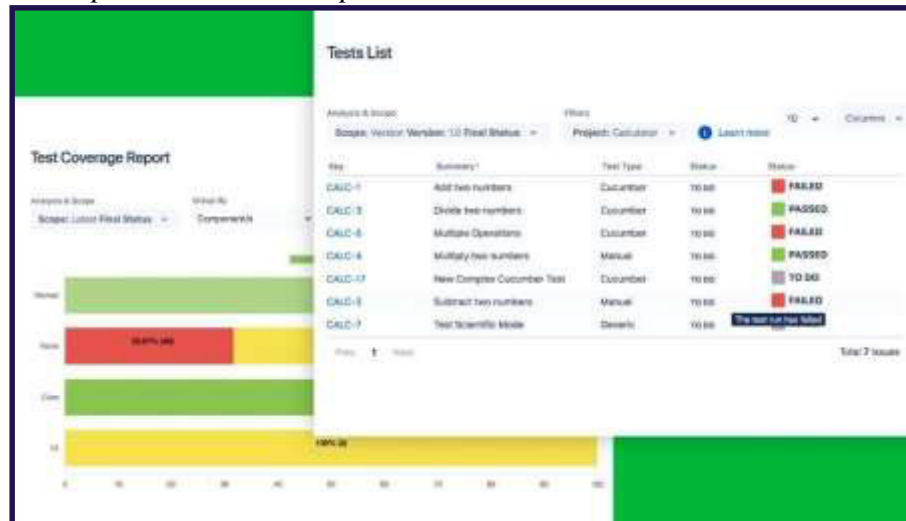
Fuente: Xray Test Management ,2023 (
<https://marketplace.atlassian.com/apps/1211769/xray-test-management-for-jira?tab=overview&hosting=cloud>)

- **Planificar, ejecutar e Integrar**

Dentro de Xray contamos con 4 entidades principales:

Figura 24

Xray con el reporte de cobertura para los Test



Fuente: Xray Test Management ,2023 (<https://marketplace.atlassian.com/apps/1211769/xray-test-management-for-jira?tab=overview&hosting=cloud>)

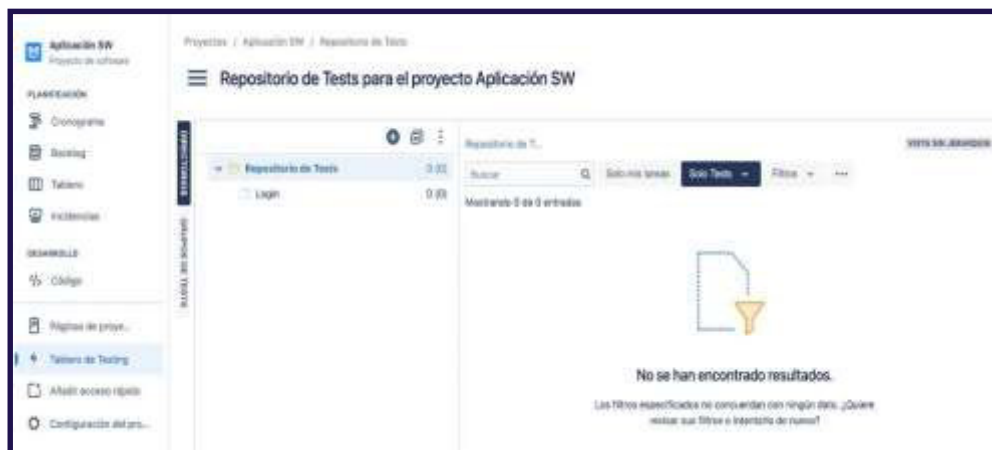
02. Lineamientos

En esta fase debemos tener en cuenta lo siguiente:

- Tener una estructura a nivel de Repositorio de Test para mantener un orden a nivel de funcionalidades de una aplicación para ubicar de manera más fácil los casos de prueba como se observa en la **figura 25**.

Figura 25

Organización por funcionalidad en el Repositorio de Test



Fuente: Elaboración propia.

- Utilizar los agrupadores de Test para agrupar los casos de prueba relacionados a una misma funcionalidad ayuda a mejorar su gestión; para

nombrarlos podemos considerar el nombre de la funcionalidad registrada en el repositorio de Tests como se indica en la **figura 26**.

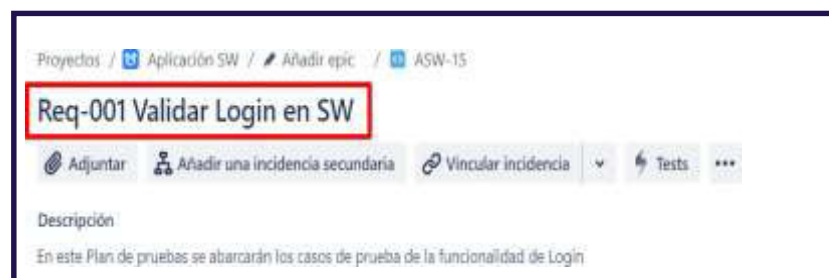
Figura 26
Nomenclatura para el agrupador de Test



Fuente: Elaboración propia.

- Al crear un Test Plan asociar un identificador del requerimiento y una descripción breve del mismo en el nombre como se visualiza en la **figura 27**.

Figura 27
Nomenclatura propuesta para los Test Plan

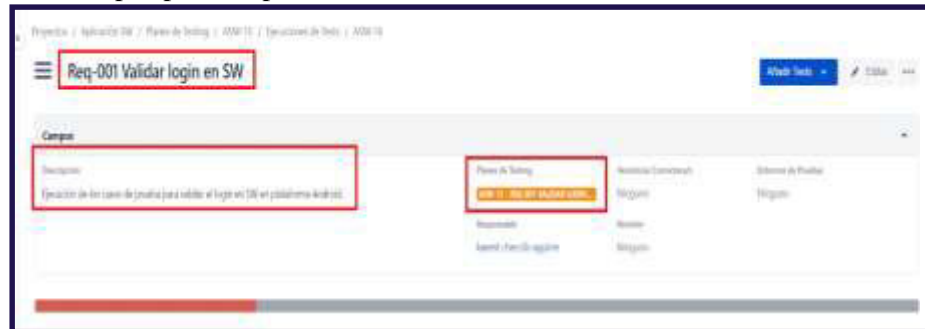


Fuente: Elaboración propia.

- Los Test Execution deben estar asociados a un Test Plan asimismo deben tener asociado un identificador del requerimiento y una descripción del mismo en el nombre como se presenta en la **figura 28**.

Figura 28

Nomenclatura propuesta para los Test Execution

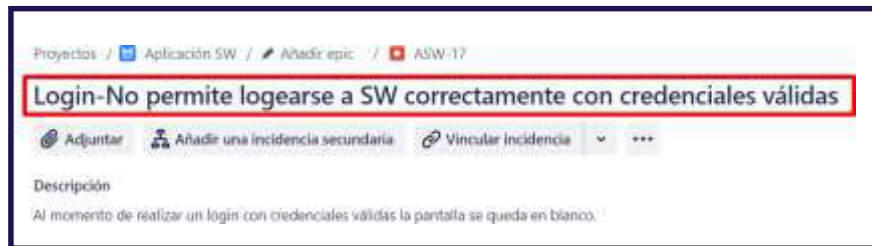


Fuente: Elaboración propia.

- Los defectos o bugs deben contener en el nombre la funcionalidad con un resumen breve y claro como se ve en la **figura 29** asimismo debemos detallarlo en el campo descripción y adjuntarle una imagen de la observación encontrada.

Figura 29

Nomenclatura para los bugs.



Fuente: Elaboración propia.

03. Aplicación

Para aplicar los lineamientos comentados debemos seguir los pasos comentados en la **figura 30**.

Figura 30

Pasos para aplicar los lineamientos en la fase de ejecución

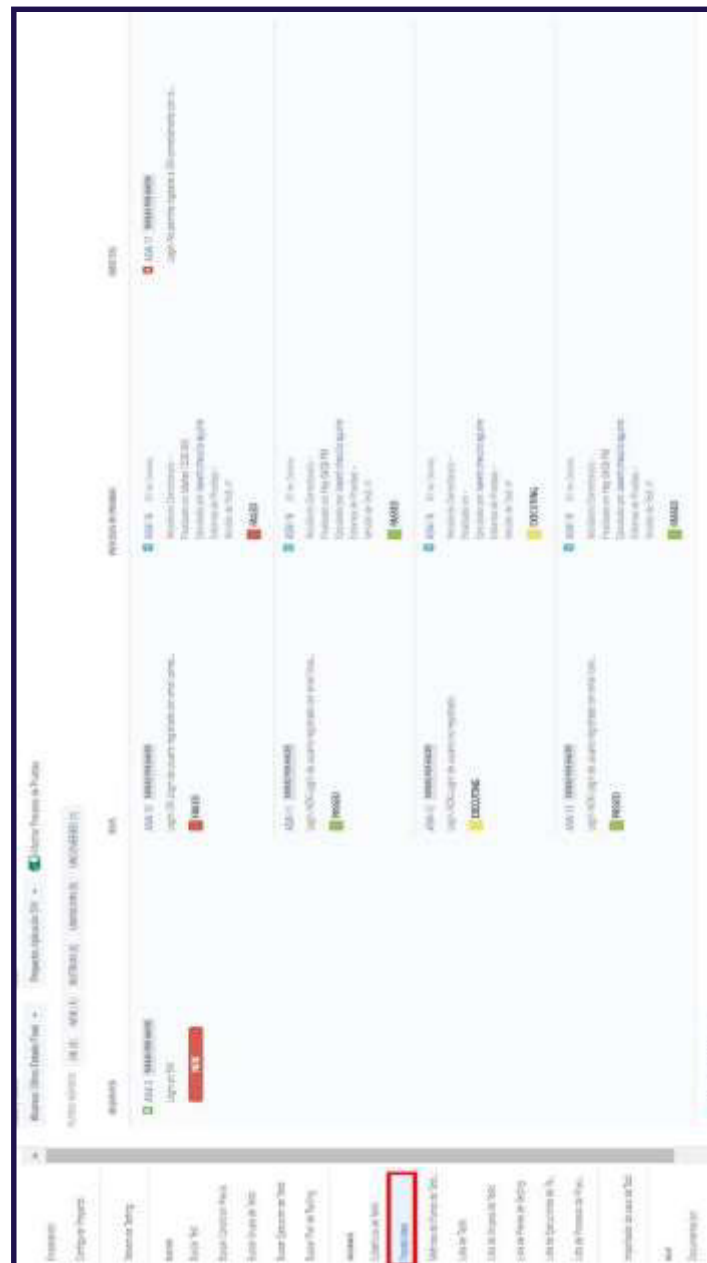


Fuente: Elaboración propia.

Adicionalmente luego de realizar la ejecución de los casos de prueba podemos generar reportes en Xray que nos permitan ver los avances por requerimiento con su trazabilidad y cobertura como lo veremos a continuación:

Reporte de Trazabilidad: En este reporte se muestra la trazabilidad de Requerimiento, casos de prueba, Test Execution con sus defectos asociados como se ve en la **figura 31**.

Figura 31
Reporte de trazabilidad



Fuente: Elaboración propia.

mientras más claros se encuentren permitirán que el equipo pueda utilizarlos y evite retrabajo en crearlos nuevamente.

3.2.3 Fundamentos Utilizados

Para las fases del proceso implementado se utilizaron los siguientes fundamentos:

- **SCRUM** siendo una metodología ágil para el desarrollo de software en la cual se adaptó el proceso propuesto detallado en la sección de la metodología utilizada.
- **ISTQB (2021) International Software Testing Qualifications Board** organización de certificación internacional de calidad del software en la que su material sirvió de apoyo para las fases del proceso propuesto y las actividades implementadas en cada una de ellas.
- **Agile Testing** siendo una práctica de pruebas de software que sigue los principios ágiles para el desarrollo de software; aquí no se ve al testing como una fase independiente sino como parte integral del desarrollo de software donde aplica prácticas como Integración continua, Desarrollo guiado por pruebas (Test Driven Development – TDD), Desarrollo guiado por comportamiento (Behaviour Driven Development – BDD) entre otras.

Agile Testing cuenta con los siguientes principios:

El Testing no es una fase: El testing continuo es la única forma de garantizar avance continuo, por esto, el testing se realiza constantemente junto con el desarrollo de software y demás actividades.

Colaboración constante: Los equipos de prueba colaboran estrechamente con los desarrolladores, los propietarios de productos y otros miembros del equipo para comprender los requisitos y las expectativas del cliente.

Pruebas continuas: En lugar de realizar pruebas solo al final del ciclo de desarrollo, las pruebas se realizan de manera continua a lo largo del proceso de desarrollo. Esto asegura una retroalimentación temprana y permite una corrección rápida de errores.

Enfoque en calidad: Aunque la entrega rápida es crucial, la calidad del software sigue siendo una prioridad. Los equipos de prueba aseguran que se mantengan altos estándares de calidad a lo largo de todo el proceso de desarrollo.

- **Holistic testing** es un enfoque que va más allá de las pruebas unitarias y de componentes individuales, y se centra en evaluar todo el sistema como una entidad completa.

Soler (2022) nos comenta que, bajo el enfoque holístico aplicado a las pruebas de software, todos los miembros de un equipo deben considerar la importancia del testing desde el principio del ciclo de desarrollo de software, incluyendo cómo se debe implementar el código para proporcionar información sobre cómo se está comportando realmente en producción el sistema.

3.2.3 Implementación de las áreas, procesos, sistemas y buenas prácticas

Este proceso se implementó dentro de la subgerencia del área de Empresas para una banca móvil el cual siguió la siguiente planificación para la implantación de cada una de las fases como se muestran en la **figura 34**.

Figura 34

Roadmap de despliegue de las fases del proceso propuesto



Fuente: Elaboración propia.

Durante los 2 últimos trimestres del año 2022 se fomentó mucho con el equipo la adopción de las prácticas y lineamientos planteados, los cuales hasta la época actual ya forma parte de sus actividades habituales dentro de su proceso de pruebas.

Dentro de lo logrado por el equipo en cada una de las fases hasta el 2 trimestre del año 2023 tenemos:

Análisis del requerimiento: Se logró definir y catalogar 61 funcionalidades del aplicativo con las pruebas basadas en riesgos las cuales fueron catalogadas como se muestra en la **tabla 5**.

Tabla 5

Cantidad de funcionalidades registradas y catalogadas por nivel de riesgo

Riesgo Extremo	Riesgo Controlado	Riesgo Bajo
11	14	36

Fuente: Elaboración propia.

Diseño y redacción: El equipo logró diseñar y redactar 835 casos de prueba con técnicas de diseño de casos de prueba y redactarlos en Gherkin

Tabla 6

Cantidad de casos de prueba generados por trimestres durante el año 2022 y 2023.

3 Trimestre 2022	4 Trimestre 2022	1 Trimestre 2023	Mitad 2 Trimestre 2023
117	348	332	38

Fuente: Elaboración propia.

Ejecución: El equipo logró ejecutar en Xray 1637 casos de prueba como se muestra en el cuadro siguiente:

Tabla 7

Cantidad de casos ejecutados por trimestre durante el año 2022 y 2023.

3 Trimestre 2022	4 Trimestre 2022	1 Trimestre 2023	Mitad 2 Trimestre 2023
202	788	589	58

Fuente: Elaboración propia.

Lo cual nos permitió la trazabilidad de 16 requerimientos de la aplicación con los entregables de prueba como son el Test Plan y el Test Execution que reemplazaron a la documentación C204 y T907 que se tenía como entregables de prueba en el proceso anterior.

Haciendo un análisis entre el total de casos en el plan de pruebas, casos de prueba nuevos y reutilizados se obtuvo el siguiente cuadro que se muestra en la **tabla 8**.

Tabla 8

Porcentajes de reutilización por trimestres durante el año 2022 y 2023.

	3 Trimestre 2022	4 Trimestre 2022	1 Trimestre 2023	Mitad 2 Trimestre 2023
Total de casos en el plan de Pruebas	202	788	589	58
Total de Casos de prueba Nuevos	117	348	332	38
Casos de prueba Reutilizados	85	440	257	20
Porcentaje de Reutilización	42.07 %	55.83 %	43.63 %	34.48 %

Fuente: Elaboración propia.

Como puede observarse el tener una correcta organización y los casos claros estos pueden ser reutilizados por los equipos lo cual puede ayudar en la optimización de tiempos como se detalla en la siguiente sección.

Dentro de los equipos el QA maneja los siguientes tiempos aproximados para el diseño de casos de prueba como se ve en la **tabla 9**.

Tabla 9*Tiempo para el diseño de casos de pruebas*

Casos de prueba	Descripción	Tiempo Mins
Caso Nuevo	Diseño de un caso no existente dentro del gestor de casos de prueba y necesita crearse desde el inicio pasando por las fases de análisis de requerimientos, diseño y redacción para luego ser agregado al plan de pruebas en la etapa de implementación.	25 mins
Caso Reutilizado	Caso de prueba que se encuentra en la biblioteca y necesita ser ubicado para ser agregado en el plan de pruebas durante la etapa de implementación.	10 mins

Fuente: Elaboración propia.

Si se hace un ejercicio entre los tiempos con los casos de prueba nuevos y requeridos en los planes de prueba por cada trimestre podemos encontrar la optimización de tiempos aproximados.

En la **tabla 10** se mostrarán los tiempos aproximados incurridos para el total de casos del plan de pruebas, casos nuevos y reutilizados para el 2022.

Tabla 10*Tiempos incurridos en el diseño de casos por trimestre en 2022*

	3 Trimestre 2022	Tiempos (Diseño e Implementación)	4 Trimestre 2022	Tiempos (Diseño e Implementación)
Total de casos en el plan de Pruebas	202	84.16 h	788	328.33 h
Total de Casos de prueba Nuevos	117	48.75 h	348	145 h
Casos de prueba Reutilizados	85	14.16 h	440	73.33 h

Fuente: Elaboración propia

En la **tabla 11** se mostrarán los tiempos aproximados incurridos para el total de casos del plan de pruebas, casos nuevos y reutilizados para el 2023.

Tabla 11

Tiempos incurridos en el diseño de casos por trimestre en 2023

	1 Trimestre 2023	Tiempos (Diseño e Implementación)	Mitad de 2 Trimestre 2023	Tiempos (Diseño e Implementación)
Total de casos en el plan de Pruebas	589	245.41 h	58	24.16 h
Total de Casos de prueba Nuevos	332	138.33 h	38	15.83 h
Casos de prueba Reutilizados	257	60 h	20	3.33 h

Fuente: Elaboración propia.

Como podemos observar en el cuadro de la **tabla 10** en el 3 trimestre 2022 se utilizaron en total 202 casos en los planes de prueba que demoraron aproximadamente en su diseño e implementación **84.16 h** que sería el tiempo que se hubiera necesitado para crear todos los casos de prueba sin aplicar el proceso de pruebas propuesto.

Al realizar el análisis a los casos de prueba diseñados por el equipo se observó que habían 117 casos que habían sido creados como nuevos dado que no existían en el gestor de casos de prueba se necesitó para ello aproximadamente 48.75 h para el análisis del requerimiento, diseño y redacción e incluirlos en el plan de pruebas; sin embargo hubieron 85 casos que formaban parte ya de la herramienta para los cuales se requirió 14.16 h para analizar el requerimiento, ubicarlos y que formen parte del plan de pruebas; si sumamos el tiempo de los casos nuevos y reutilizados nos da un tiempo aproximado de 62.91 h aplicando el proceso propuesto.

Haciendo el mismo análisis por cada uno de los periodos podemos tener el siguiente cuadro del tiempo aproximado sin el proceso implementado y con el proceso implementado para cada los periodos como se ve en la **tabla 12**.

Tabla 12:

Comparación de tiempos incurridos sin proceso implementado vs tiempos incurridos con el proceso implementado.

	3 Trimestre 2022	4 Trimestre 2022	1 Trimestre 2023	Mitad de 2 Trimestre 2023
Sin proceso Implementado	84.16 h	328.33 h	245.41 h	24.16 h
Con proceso Implementado	62.91 h	218.33 h	198.33 h	19.16 h
Tiempo Optimizado	21.25 h	110 h	47.08 h	5 h

Fuente: Elaboración propia.

Como se puede observar tenemos una optimización significativa del tiempo con la aplicación del proceso propuesto en cada uno de los trimestres.

3.3 Evaluación Económica

3.3.1 Evaluación del costo

Haciendo una evaluación de los costos del proceso se tiene lo siguiente:

Costos directos

Como se comentó inicialmente el QA de los equipos scrum está bajo la modalidad de bodyshopping; para la implementación del proceso es necesario contemplar el costo de la fábrica que terceriza el servicio por el perfil solicitado ya que debe tener todos los conocimientos contemplados

A continuación, en la **tabla 13** se mostrará el costo total por perfil QA en cada uno de los trimestres.

Tabla 13:*Cantidad de horas trabajadas y costos del perfil QA por trimestre.*

# Trimestre	Cantidad de FTEs	Perfil	Horas trabajadas	Costo Total
Trimestre 3-2022	2.5	QA Semi Senior	1160	S/. 87742.4
Trimestre 4 - 2022	2	QA Semi Senior	928	S/. 70 193.9
Trimestre 1-2023	2	QA Semi Senior	928	S./ 70 193.9
Mitad Trimestre 2-2023	0.5	QA Semi Senior	232	S./ 17 548.4
Total				S./ 245 678.7

Fuente: Elaboración propia.

El monto total (S/. 245 678.72) es la suma de los costos por trimestre, estos costos se estimaron en base a la cantidad de miembros de perfil QA Semi Senior del equipo en los diferentes trimestres, el tiempo de asignación al requerimiento, ya que en algunos casos contaron con una asignación parcial del 50%.

La entidad financiera adquirió el Jira y el Xray antes de la implementación del TSP, como parte de su estrategia de mejorar el proceso de pruebas, buscó contar con un repositorio que centralice los casos de prueba en todas sus aplicaciones, por lo cual no se consideran los costos de Jira dentro del presente proyecto.

3.3.2 Beneficio para la organización

Dentro de los beneficios podemos encontrar:

- Un proceso de pruebas escalable que permite tener visibilidad de las funcionalidades de una aplicación y su criticidad, donde hacemos uso de las pruebas basadas en riesgos que ayudan en la priorización de los casos de prueba en los entregables de prueba.
- Mejora en la cobertura del diseño de casos de prueba con la aplicación de técnicas.

- Reutilización de los casos de prueba como se puede visualizar en la **tabla 8** que llega aproximadamente hasta un 55.83 %.
- Optimización de tiempos en la fase de diseño e implementación de los casos de prueba como se puede ver en la **tabla 12**.
- Trazabilidad entre requerimientos, casos de prueba y entregables de pruebas mediante el uso de una herramienta de gestión de casos de prueba Xray como se puede ver en los reportes que nos brinda la herramienta.

CAPÍTULO IV. REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

La autora del presente trabajo apoyó en la definición e implementación del proceso comentado que en este documento se expone en conjunto con el equipo de trabajo del cual formó parte (4 Ingenieros DevOps y 2 QEs).

En primer lugar, es importante resaltar la apertura de la subgerencia de empresas y a los líderes de servicios, por su atención en el presente problema e impulsar accionables para su pronta mejoría permitiendo la implementación de este nuevo proceso en sus equipos.

En segundo lugar, es importante comentar que hubo factores que influyeron en la implementación del proceso como fueron la rotación de los QAs dentro de los equipos Scrum de una aplicación a otra y el porcentaje de participación variable asignado durante los diferentes trimestres.

En tercer lugar, fue necesario asegurar los nuevos skills para los QAs con los coordinadores de las fábricas terceras dado que dentro de los equipos habían QAs que conocen del negocio a nivel funcional de una aplicación, pero para ellos eran nuevos los lineamientos del proceso.

En tercer cuarto indicar que, para algunos requerimientos, la definición para el desarrollo no suele ser clara y concisa, lo cual conlleva a generar consultas en sesiones de entendimiento con los expertos de negocio para la definición de los casos de prueba.

Finalmente, se dieron cambios en el proceso de despliegue para pasar al entorno productivo se reemplazó el uso de la documentación habitual de pruebas C204 y T907 por el uso de las entidades de una herramienta de gestión de casos de prueba, pese a que los cambios sumaron gran valor a la organización, se tuvieron que realizar capacitaciones para un mejor entendimiento a los QAs y equipos involucrados, hubo un tiempo de aprendizaje y adaptación a la nueva forma de trabajo.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

El análisis realizado, el diseño y la implementación del proceso de pruebas dan paso a las siguientes conclusiones:

- La visión holística de la participación temprana del QA dentro del equipo de desarrollo ayuda en la mejor comprensión del requisito, además de apoyarse con la priorización de las pruebas basadas en riesgos permite delimitar mejor el plan de pruebas y validación del requerimiento.
- La aplicación de técnicas de diseño como el Pairwise ayuda en la mejora de la cobertura y en la optimización de los casos de prueba.
- Una adecuada nomenclatura y redacción de los casos fomenta su reutilización y sirve de documentación viva para todos los interesados y miembros del equipo lo que permite que trabajen de manera colaborativa.
- El uso de una herramienta de gestión de pruebas permite tener la trazabilidad entre requerimiento, casos de prueba y defectos los cuales pueden ser revisados mediante los reportes en Xray.
- La reutilización de casos de prueba ayuda en la optimización de tiempos al momento de definir el alcance y el plan de pruebas de nuevos requerimientos.

5.2 Recomendaciones

Dentro de las recomendaciones del presente trabajo se tienen:

- Sensibilizar a todo el equipo Scrum sobre la importancia del control de calidad, con el fin de obtener la colaboración y retroalimentación de todos.
- En la etapa de refinamiento de requerimiento es importante la participación de todos los miembros del equipo tanto developers como QAs dado que en agilidad la responsabilidad es del equipo y no de un rol, mientras más claro se pueda tener el alcance de un requerimiento mejor se pueden definir los casos de prueba involucrados.
- Se hace mención de la importancia de tener en el equipo un rol que conozca de estas prácticas propuestas en el proceso de pruebas para que puedan ayudar en acelerar su implementación.
- Se recomienda hacer un seguimiento de la adopción de las prácticas y lineamientos implementados para validar su correcta implementación.

5.3 Fuentes de información

Advanced Topic - Behavior-Driven Development. (2023).

Scaled Agile Framework. Recuperado de
<https://scaledagileframework.com/behavior-driven-development/>

Certified Tester Advanced Level Syllabus (2021).

ISTQB. Recuperado de https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTAL-TA_Syllabus_v3.1.2.pdf

Certified Tester Advanced Level Test Analyst (CTAL-TA). (2021).

ISTQB. Recuperado de <https://www.istqb.org/certifications/test-analyst>

Hamilton, T. (2023). What is Software Testing? Definition.

Guru99. Recuperado de <https://www.guru99.com/software-testing-introduction-importance.html>

IBM. (2021). Lección 6: Creación de una evaluación de riesgo.

IBM. Recuperado de <https://www.ibm.com/docs/es/engineering-lifecycle-management-suite/lifecycle-management/6.0.2?topic=plan-lesson-create-risk-assessment>

IBM. (2022). Visión general de casos de prueba y suites de pruebas.

IBM. Recuperado de <https://www.ibm.com/docs/es/engineering-lifecycle-management-suite/lifecycle-management/6.0.3?topic=testing-test-case-test-suite-overview>

Juarez, E. (2020). Técnicas de diseño de casos de prueba.

Platzi. Recuperado de <https://platzi.com/tutoriales/1421-pruebas-software/9569-tecnicas-de-diseno-de-casos-de-prueba/>

Kulcsár, E., Csiszér, T. y Abonyi, J. (2020). Pairwise comparison based failure mode and effects analysis (FMEA). MethodsX. Recuperado de

<https://www.sciencedirect.com/science/article/pii/S2215016120302272>.

Mena, A. (2022). Testing basado en Riesgos. Recuperado de

<https://www.linkedin.com/pulse/testing-basado-en-riesgos-alexandra-mena/?originalSubdomain=es>.

- Mohammed, F., Mohammed Najm, A. y Mohammed, Y. (2022). Combinatorial Testing Approaches: A Systematic Review. *Journal of Computers, Communications, Control & Systems Engineering*, 22(4).61-79
- Siroky, D. (2017). The glitch economy: Counting the cost of software failures. *Cloud Computing News*. Recuperado de <https://www.cloudcomputing-news.net/news/2017/oct/30/glitch-economy-counting-cost-software-failures/>.
- Soler, J. (2022). Holistic Testing: ¿Cómo ayuda este enfoque a lograr un alto nivel de Calidad? . *Abstracta*. Recuperado de <https://cl.abstracta.us/blog/holistic-testing-enfoque-lograr-alto-nivel-calidad-software/>
- Veenu Singh, Kumar, V., y V.B. Singh. A hybrid novel fuzzy AHP-TOPSIS technique for selecting parameter-influencing testing in software development. *Decision Analytics Journal*, 6(2023), 15.

5.4 GLOSARIO

- **COBERTURA DE PRUEBAS**

Medida de cuánto se comprueba un sistema en base a las pruebas realizadas bajo un determinado criterio.

- **TÉCNICAS DE DISEÑO DE CASOS DE PRUEBA**

Mecanismo mediante el cual los QA's identifican los casos de prueba con alta probabilidad de encontrar fallos y que permitan incrementar la cobertura de las pruebas.

- **PAIRWISE**

Es una técnica eficaz de generación de casos de prueba la cual plantea que los defectos son generados como máximo por dos factores. Esta técnica ayuda a la reducción de casos de prueba, en lugar de probar todas las combinaciones de cada factor probamos de manera efectiva una combinación de dos factores.

- **SDLC (CICLO DE VIDA DE DESARROLLO DE SOFTWARE)**

Es un proceso utilizado por los equipos de desarrollo para crear software de alta calidad el cual contiene actividades relacionadas también al mantenimiento de software.

- **ISSUES DE JIRA**

También llamados incidencias representan el trabajo que deben desarrollar los equipos Scrums para construir un software; dentro de Jira tenemos algunos ejemplos como tareas, historias de usuario, features, bugs.

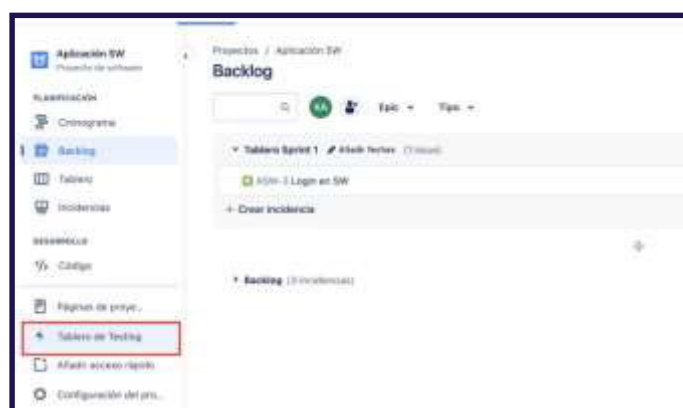
ANEXOS

ANEXO 1: ALTAS DE ISSUES EN XRAY

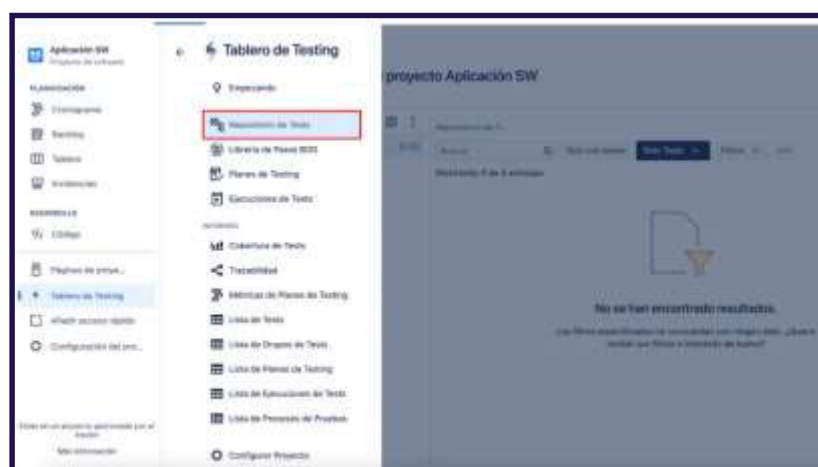
En el presente anexo se presentará una guía de referencia para la creación y asociación de las entidades de Xray asociados en Jira.

Organización del Repositorio de Test

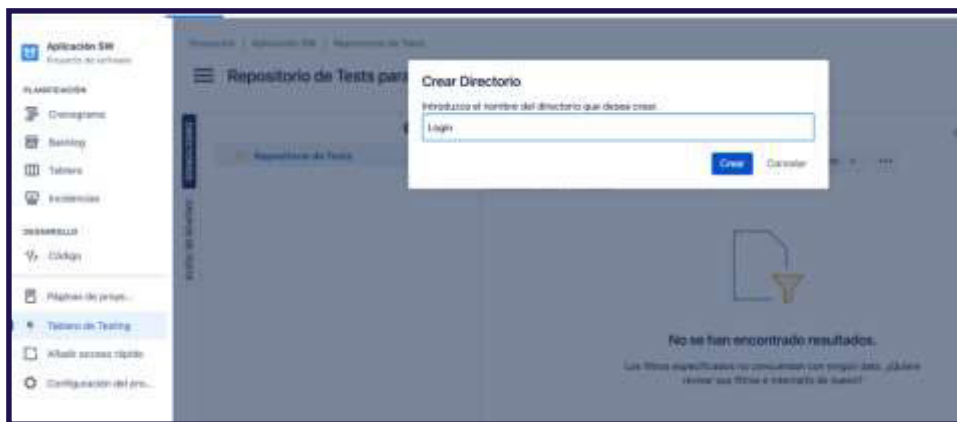
Dentro de cada proyecto Jira tenemos una sección Tablero Jira



Paso 1: Ubicarnos en la sección Repositorio de Tests



Paso 2: Crear un directorio por cada funcionalidad del aplicativo.

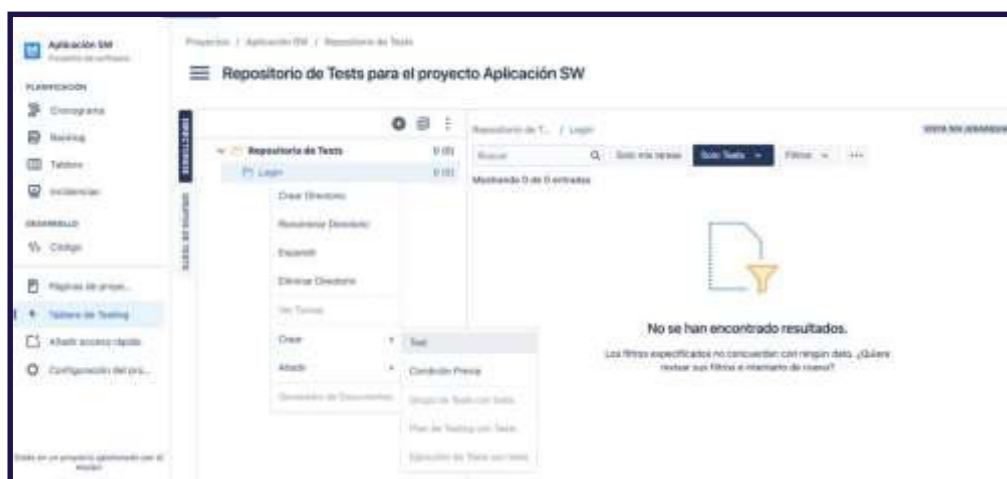


Se podrá visualizar la creación del directorio de la funcionalidad.



Alta de Test (Caso de Prueba)

Paso 1: Ubicarnos dentro del directorio de la funcionalidad direccionarnos en la sección Crear Test.



Paso 2: Completar los datos de nombre y descripción para crear el Test.

Crear incidencia

Importar incidencias

Configurar campos ▾

Proyecto*

Aplicación SW (ASW)

▾

Tipo de Incidencia*

Test

▾

?

Algunos tipos de incidencia no están disponibles debido a alguna incompatibilidad en la configuración de campos o a asociaciones con flujos de trabajo.

Resumen*

Login-OK>Login de usuario registrado con email correcto y password correcto

Descripción

Estilo ▾

B

I

U

A ▾

^A ▾

↻ ▾

≡

≡

@ ▾

+ ▾

≡

El caso de prueba valida el acceso correcto de un usuario registrado en la aplicación SW.

☐ Crear otra

Crear

Cancelar

Al momento de crearlo se mostrará la siguiente pantalla del caso de prueba.

Añadir epic / ASW-10

🔒

👁 1

🔗

🔄

⋮

✕

Login-OK>Login de usuario registrado con email correcto y password correcto

🔗

👤

🔗

⚡

⋮

Descripción

El caso de prueba valida el acceso correcto de un usuario registrado en la aplicación SW.

Detalles del Test

Detalles del Test

Condiciones Previas

Grupos de Tests

Planes de Testing

Proceso de Pruebas

Tipo de Test

Manual

▾

Repositorio de T...

Login

Conjunto de datos

Responsible

Sin asignar

Asígname a mí

Etiquetas

Ninguna

Story point estimate

Ninguno

Sprint

Ninguno

Informador

karén checcilo aguine

Estado del Test

Abrir Estado del Test

Creado hace 17 segundos

Actualizado hace 17 segundos

Configurar

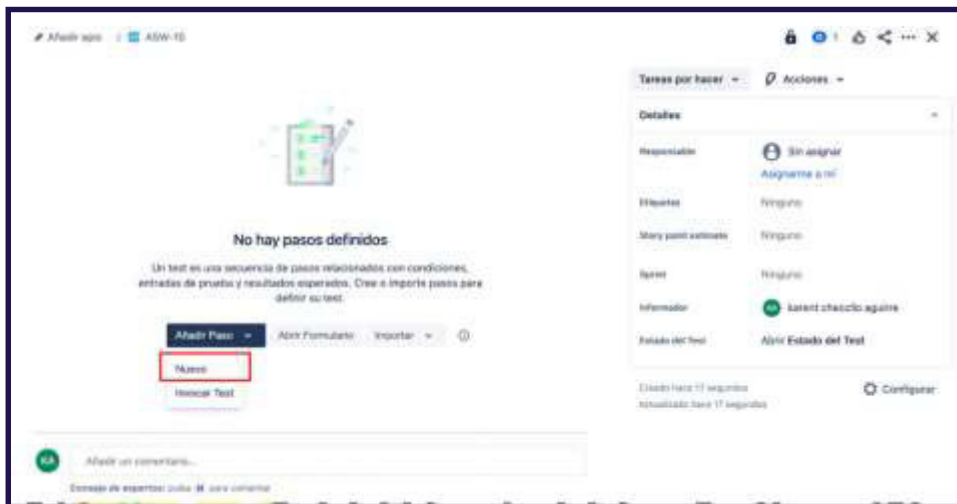
KA

Añadir un comentario...

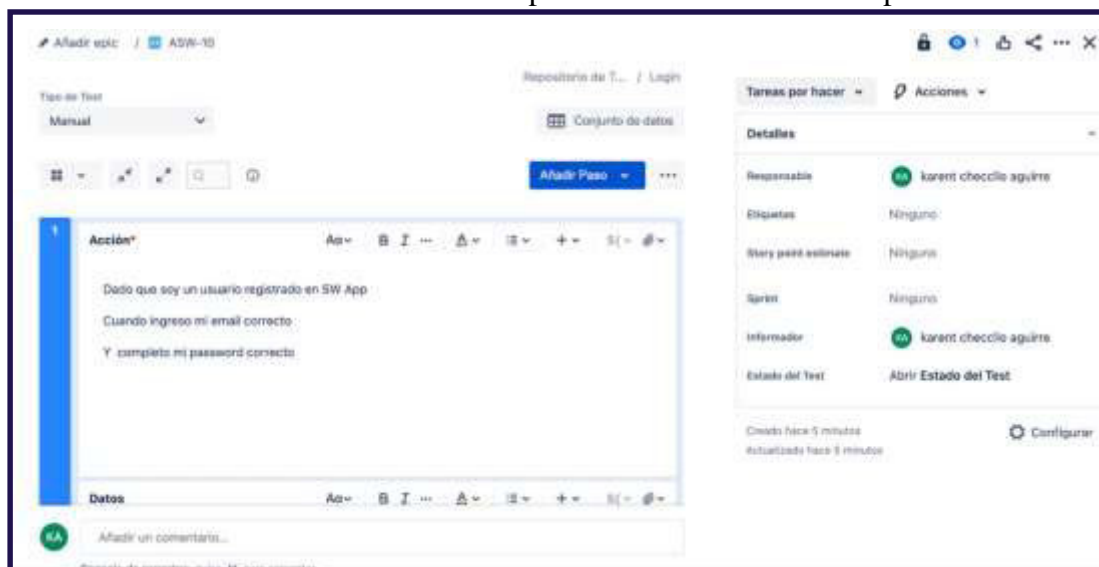
Consejo de expertos: sube M para comentar

60

Paso 3: Ir a la sección Añadir un nuevo paso para completar el contenido del Test.

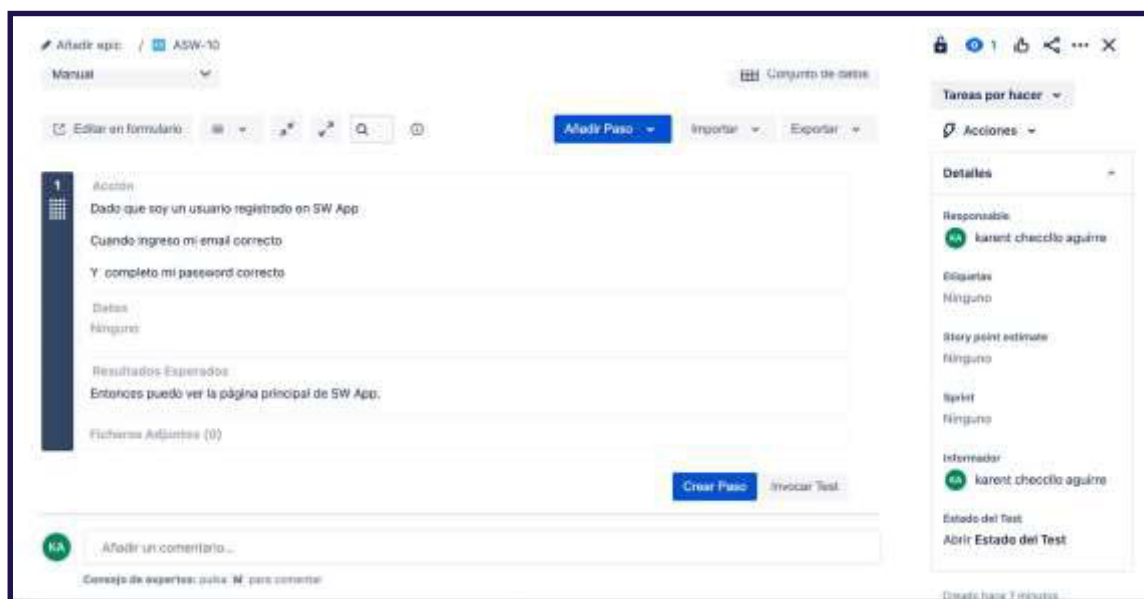


Paso 4: Dentro de la sección acción completar el detalle del caso de prueba.





Paso 5: Dentro de la sección resultados esperados detallar el resultado esperado del caso de prueba.

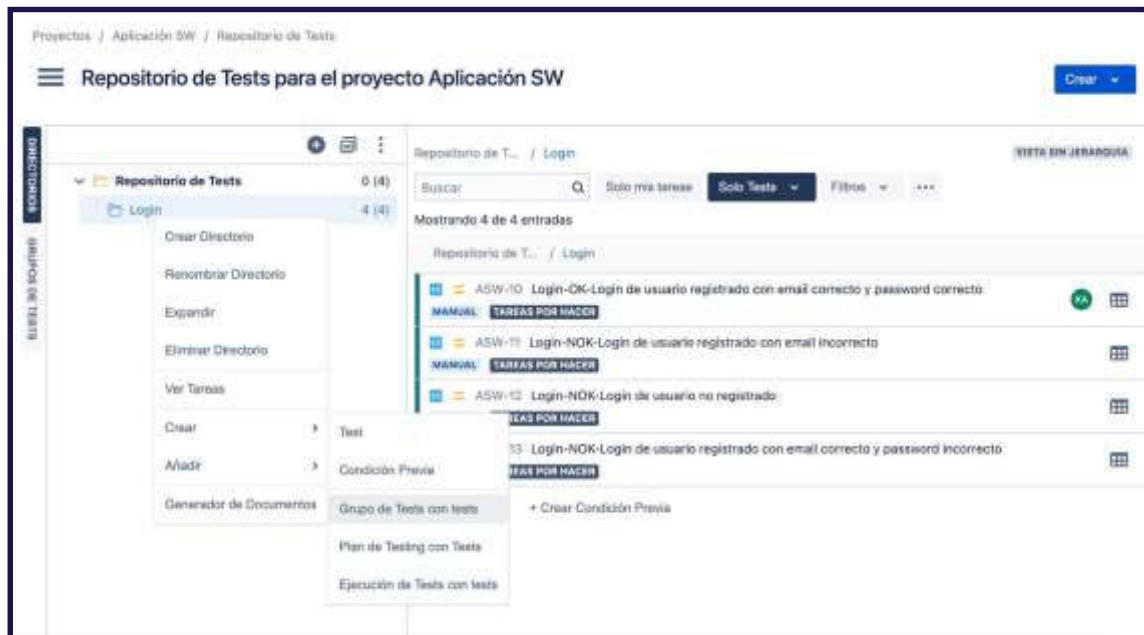


Se podrá visualizar el caso de prueba creado desde la vista del repositorio de Tests.

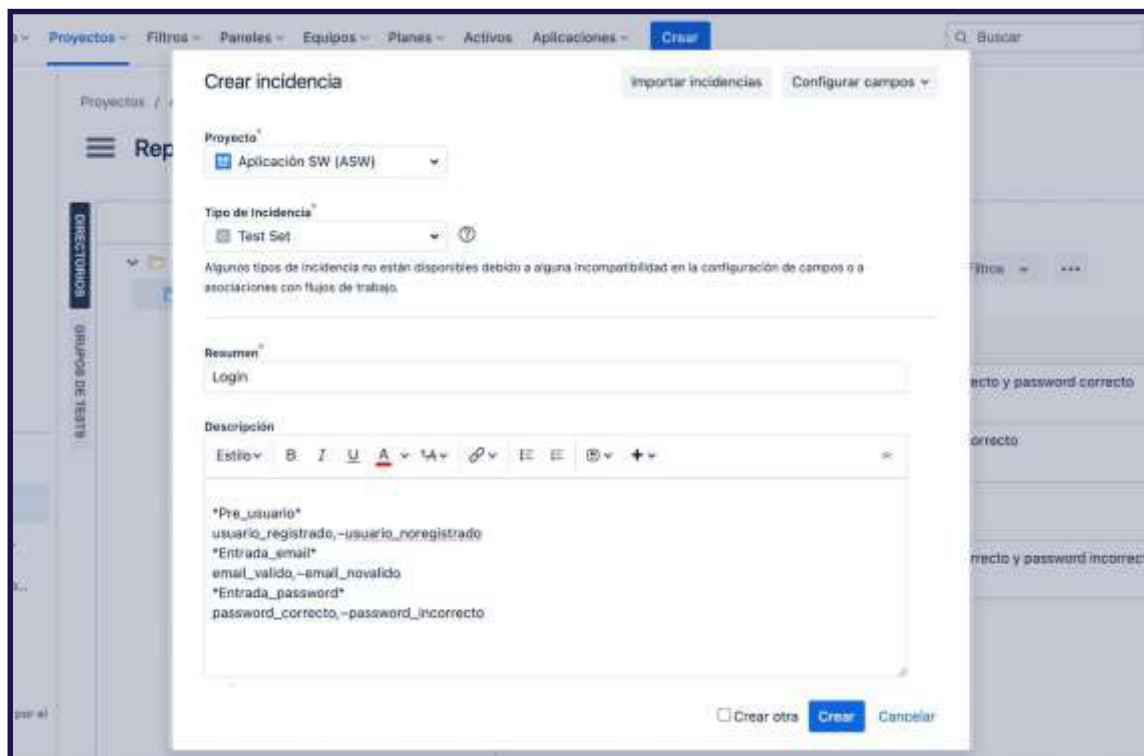


Alta de Test Set (Agrupadores de casos de prueba)

Paso 1: Desde el repositorio de Tests ir a la opción Crear Grupo de Tests con Test.



Paso 2: Completar los datos como Resumen y la descripción.



Luego podremos ver el Test Set creado con los casos de prueba asociados.

Proyectos / Aplicación SW / Añadir epic / ASW-14

Login

Adjuntar

Añadir una incidencia secundaria

Vincular incidencia

Tests

...

Descripción

Pre_usuario

usuario_registrado,~usuario_noregistrado

Entrada_email

email_valido,~email_novalido

Entrada_password

password_correcto,~password_incorrecto

Tests

Añadir Tests

Ver en un tablero

Filtros

10

Columnas

Listado de casos de prueba asociados.

Proyectos / Aplicación SW / Añadir epic / ASW-14

password_correcto,~password_incorrecto

Tests

Añadir Tests

Ver en un tablero

Filtros

10

Columnas

	Rango	Clave	Resumen	Estado	Acciones
☐	1	ASW-10	Login-OK-Login de usuario registrado con email corr...	TAREAS POR HACER	...
☐	2	ASW-11	Login-NOK-Login de usuario registrado con email in...	TAREAS POR HACER	...
☐	3	ASW-12	Login-NOK-Login de usuario no registrado	TAREAS POR HACER	...
☐	4	ASW-13	Login-NOK-Login de usuario registrado con email co...	TAREAS POR HACER	...

Anterior

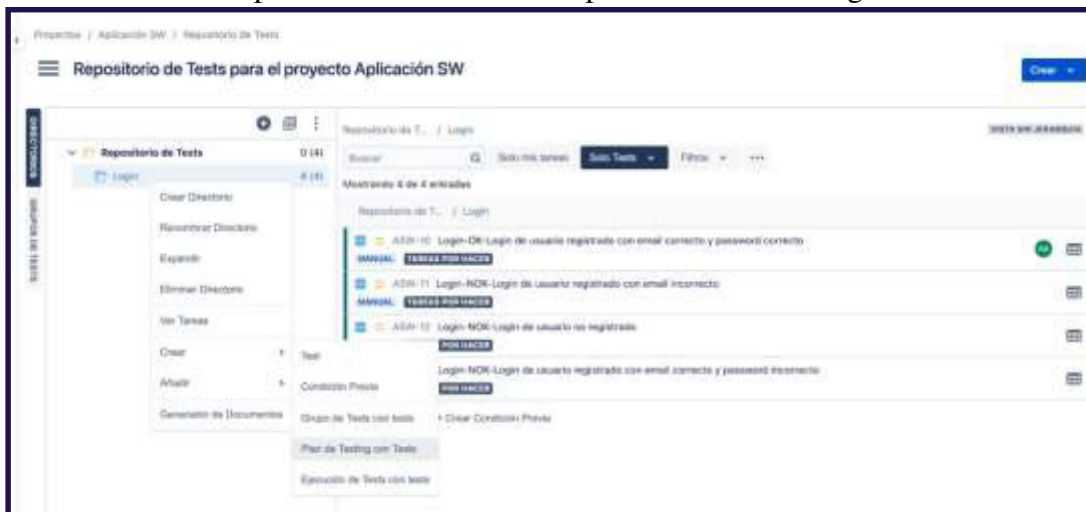
1

Siguiente

Total 4 tareas

Alta de Test Plan (Plan de pruebas)

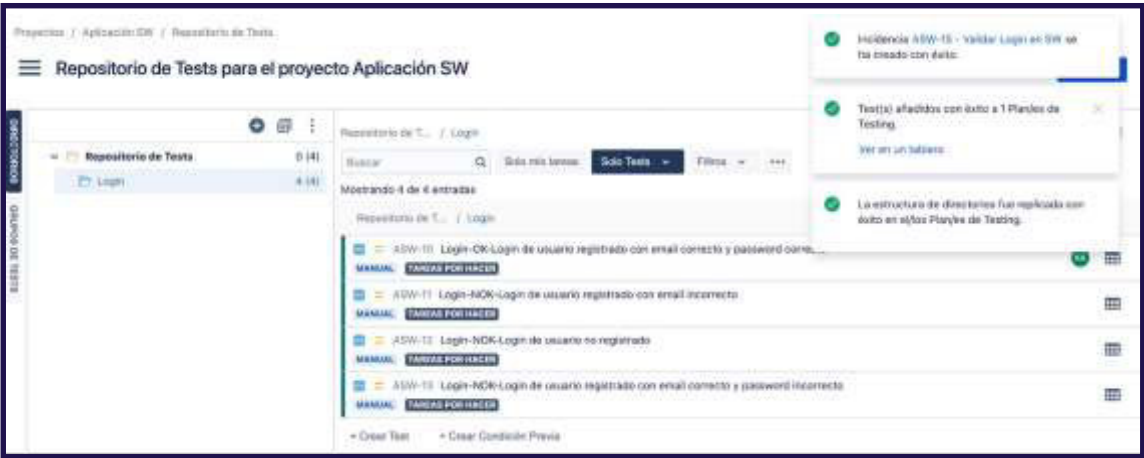
Paso 1: Desde el repositorio de Tests ir a la opción Plan de Testing con Tests.



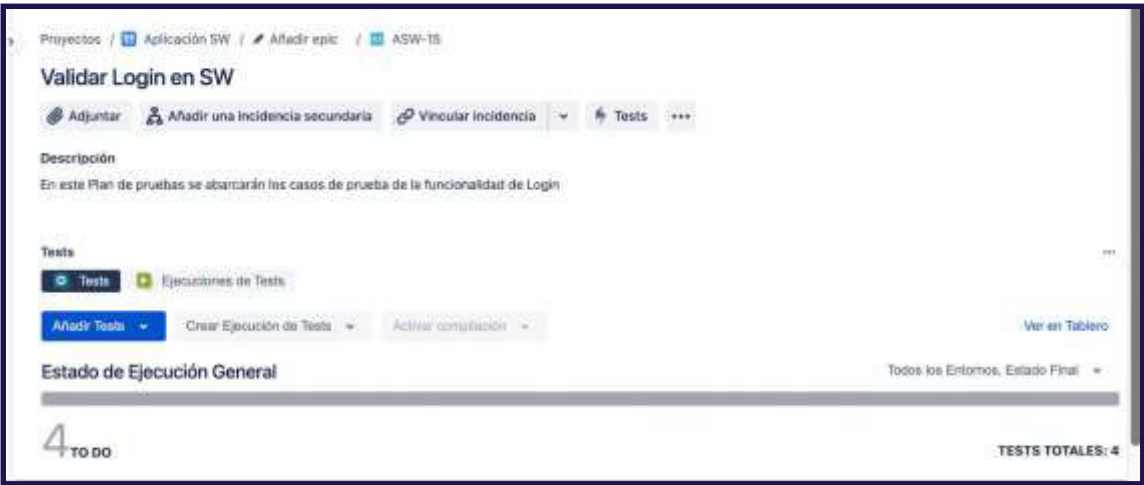
Paso 2: Completar los datos como Resumen y Descripción para el alta de Test Plan.

The screenshot shows the 'Crear incidencia' form. The 'Proyecto' field is set to 'Aplicación SW (ASW)'. The 'Tipo de Incidencia' field is set to 'Test Plan'. Below these fields, there is a message: 'Algunos tipos de incidencia no están disponibles debido a alguna incompatibilidad en la configuración de campos o a asociaciones con flujos de trabajo.' The 'Resumen' field contains the text 'Validar Login en SW'. The 'Descripción' field contains the text 'En este Plan de pruebas se abarcarán los casos de prueba de la funcionalidad de Login'. At the bottom right, there are buttons for 'Crear otra', 'Crear', and 'Cancelar'.

Se visualizará una notificación de la creación del Plan de Pruebas.



Se visualizará el Test Plan creado con los casos asociados desde el repositorio de Test.



Alta de Test Execution (Instancia de Ejecución de pruebas)

Paso 1: Desde el Test Plan seleccionamos la opción Crear ejecución de Tests.



Paso 2: Completar los datos como Sumario y descripción.



Se visualizarán los casos de prueba relacionados a la instancia de ejecución del Test Plan.

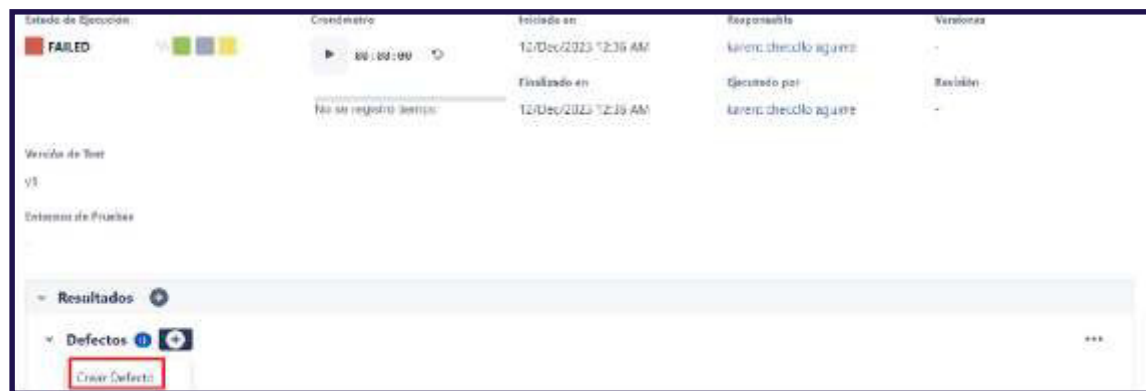


Listado de los casos de pruebas asociados al test execution.

4 TO DO							TESTS TOTALES: 4	
Solo mis Proceso de Pruebas Filtros							10 Columnas	
Rango*	Clave	Resumen	Tipo de Test	Conjunto de datos	#Defectos	Estado	Acciones	
☐ 1	ASW-10	Login-OK-Login de usuario registrado con email corr...	Manual		0	TO DO		...
☐ 2	ASW-11	Login-NOK-Login de usuario registrado con email in...	Manual		0	TO DO		...
☐ 3	ASW-12	Login-NOK-Login de usuario no registrado	Manual		0	TO DO		...
☐ 4	ASW-13	Login-NOK-Login de usuario registrado con email co...	Manual		0	TO DO		...
Anterior 1 Siguiente							Total 4 tareas	

Alta de Bug (Defecto)

Desde la instancia de ejecución de un caso en estado Failed nos ubicamos desde la sección Resultados Defectos y Crear Defectos.



Paso 1: Completar los datos como Resumen y descripción.

Crear incidencia

Importar incidencias Configurar campos

Proyecto*

Aplicación SW (ASW)

Tipo de Incidencia*

Error

Algunos tipos de incidencia no están disponibles debido a alguna incompatibilidad en la configuración de campos o a asociaciones con flujos de trabajo.

Resumen*

Login-No permite logearse a SW correctamente con credenciales válidas

Descripción

Estilo B I U A A Link List Bulleted List Image +

Al momento de realizar un login con credenciales válidas la pantalla se queda en blanco.

Crear otra Crear Cancelar

Luego podremos visualizar el bug defecto creado asociado a la instancia de ejecución del caso de Prueba.

Login-OK-Login de usuario registrado con email correcto y password correcto

Ejecutar con la aplicación Exploradora Copiar los datos Importar los resultados de la ejecución

Defectos

Problemas: 17 0 puntos 0 al principio Test 0 puntos 0 al final Test

Problemas: 17 0 puntos 0 al final Test

Estado de Ejecución	Defecto	Estado de	Responsable	Versiones	Estado de Test
FAILED	17	10/11/2015 12:36 AM	Ignacio Chacón Aguirre	1.0	0.0
		10/11/2015 12:36 AM	Ignacio Chacón Aguirre	1.0	0.0

Resultados 0/0 (0/0)

Defectos 17

17 Login no permite logearse a SW correctamente con credenciales válidas

10/11/2015 12:36 AM