



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

**REDISEÑO DE PROCESO DEL ÁREA DE PRODUCTO DE LA
EMPRESA BUK MEDIANTE LA IMPLEMENTACIÓN DE
ASEGURAMIENTO DE CALIDAD PARA LA DISMINUCIÓN
DE DEFECTOS**

PROYECTO DE GRADO PARA OPTAR AL GRADO DE MAGÍSTER EN
INGENIERÍA DE NEGOCIOS CON TECNOLOGÍAS DE INFORMACIÓN

MÓNICA ALEJANDRA MONCADA PADILLA

PROFESOR GUÍA:
VICTOR HERNANDEZ MARTÍNEZ

MIEMBROS DE LA COMISION:
FELIPE VERA CID
ROCÍO RUÍZ MORENO

SANTIAGO DE CHILE
2023

RESUMEN EJECUTIVO

El presente trabajo se efectúa en la Startup llamada Buk dedicada al desarrollo de software. Tiene como su producto digital una plataforma orientada al rubro de Recursos Humanos que busca centralizar todos los procesos que conforman un área de RRHH. La ideación de la plataforma surge primordialmente a que las empresas realizan diversos procesos de RRHH de manera manual, de modo que la plataforma resulta ser una solución que les permite optimizar, centralizar y automatizar cada uno de sus procesos.

Este proyecto inicia precisamente a la alta cantidad de defectos que se presentan en la plataforma. Estos defectos reportados por los clientes al área de servicio de atención adjudica como resultado clientes molestos que amenazan con retirar sus suscripciones, además de una mala reputación que puede llegar a afectar la adquisición de nuevos clientes. Ambos escenarios afectan directamente a los ingresos de la empresa en donde se estima que anualmente pierde un aproximado de USD 1,585,911.11 en sus ingresos por el retiro de suscripciones. Por esta razón es importante llegar a una solución que permita al cliente poder hacer uso de la plataforma de una manera placentera. En atención a lo cual este proyecto tiene como objetivo disminuir en un 50% los defectos que reportan los clientes.

Se tuvo que realizar un levantamiento de los procesos actuales que intervienen en el desarrollo de la plataforma, se procedió a construir la arquitectura de Macroprocesos para luego proceder en la diagramación BPMN que permita conocer la situación actual. Al realizar el levantamiento se observó que no existe ningún proceso que monitoree la calidad del software que se va desarrollando por lo que para el rediseño se centró en el área de producto considerando se agregue procesos enfocados en el aseguramiento de calidad, mediante el uso del proceso que ofrece el marco COBIT 4.1.

El proyecto abarcara el rediseño del proceso en una de las doce células que conforman el área de producto, en el cual se procede a incorporar aseguramiento de calidad sumando al perfil QA como responsable. El rediseño fue apoyado tecnológicamente mediante herramientas que permitieran poder llevar a cabo los scripts que contienen la lógica de negocio, se hizo uso de la herramienta Katalon Studio que integrándola con la herramienta Katalon TestOps permite poder realizar un mayor análisis a través del Dashboard que brinda. El prototipo se llevó a cabo en 3 meses lo que permitió poder obtener resultados satisfactorios de la implementación logrando una disminución del 40% de defectos reportados por los clientes, resultando no solo viable sino exitoso ya que es sumamente probable en los próximos meses se pueda superar el objetivo planteado de disminuir a un 50% los defectos. Se finaliza con un plan de gestión de cambio que contiene las actividades y el Roadmap del proyecto. Además de una evaluación económica que permita conocer la viabilidad del mismo.

AGRADECIMIENTOS

En primer lugar a Dios, que su gracia permitió la finalización de este proyecto de tesis del Magister.

A mis padres, que sus palabras de aliento y apoyo incondicional fueron imprescindibles para poder cursar este Magíster. Mis logros les pertenecen y mi amor eterno.

Agradecer a mi hermana, quien tiene siempre las palabras correctas en los momentos difíciles. Mi pequeña hermana tienes mi corazón.

A mi amiga Estefany, quien en esos momentos que dudaba de mi misma, ella creía en mí, recordándome que yo era capaz de lograr todo lo que alguna vez inició como un sueño. Tiene todo mi cariño y gratitud incondicional.

A la empresa Buk que me ha permitido poder desarrollarme profesionalmente y brindándome la libertad de poder implementar cada uno de mis conocimientos.

Al profesor Victor, por enseñarme que siempre hay mejores maneras de hacer las cosas, que la búsqueda de la excelencia debe ser un propósito en lo que hagamos.

Y para finalizar al MBE, quien me brindó los conocimientos que me permitirán ser una mejor profesional.

TABLA DE CONTENIDO

CAPÍTULO 1: INTRODUCCIÓN Y CONTEXTO	1
1.1 Historia de la empresa Buk	1
1.2 Descripción General de la Empresa	2
1.3 Acerca del problema y su justificación	4
1.4 Objetivos y Resultados Esperados del Proyecto	6
1.4.1 Objetivo General	6
1.4.2 Objetivos Específicos	6
1.5 Resultados Esperados	6
1.6 Alcance	8
1.6.1 El proyecto abarcará:	8
1.6.2 El proyecto No abarcará:	8
1.7 Riesgos Potenciales	8
CAPÍTULO 2: MARCO TEÓRICO	11
2.1 Metodología de Ingeniería de Negocios	11
2.2 Patrones de Arquitectura y Procesos de Negocio	14
2.2.1 Procesos de la Macroproceso Cadena de Valor (Macro1)	18
2.2.2 Macro 1.3.3 Subproceso Producción	20
2.3 Calidad	21
2.3.1 Definición de Calidad	21
2.3.2 Calidad de software	21
2.3.3 El Costo de la Calidad de software	21
2.3.4 Control de Calidad	23
2.3.5 Aseguramiento de Calidad	23
2.3.5.1 Antecedentes del Aseguramiento de Calidad	24
2.3.5.1.1 Revisiones	25
2.3.5.2 Aseguramiento de Calidad en el Diseño de Software	26
2.3.6 Área de Aseguramiento de Calidad	27
2.3.7 Estrategias de Pruebas de Software	28
2.3.8 Perfiles de Aseguradores de Calidad	28
2.3.8.1 Pruebas	29
2.3.8.2 Tipos de Pruebas	29
2.3.8.2.1 Pruebas Funcionales	29
2.3.8.3 Casos de Pruebas	30
2.3.8.4 Ciclo de Pruebas	30

2.3.9	Prácticas de calidad de Software	31
2.3.10	Metodología Shift Left	31
2.4	Metodología SCRUM	32
2.4.1	Roles en SCRUM	32
2.4.2	El Sprint	33
2.4.3	Historias de Usuarios	33
2.5	Modelos de Calidad de software	34
2.6	COBIT	35
2.6.1	Origen COBIT	35
2.6.2	COBIT 4.1	36
2.6.3	Proceso Administración de la calidad	38
2.6.3.1	Entradas y Salidas del Proceso Administración de la calidad	38
2.6.3.2	Actividades del Proceso Administración de la calidad	39
CAPÍTULO 3: PLANTEAMIENTO ESTRATÉGICO Y ANÁLISIS DE LA SITUACIÓN ACTUAL		40
3.1	Visión y Misión de Buk	40
3.2	Posicionamiento Estratégico	40
3.3	Objetivos estratégicos	41
3.4	Modelo de Negocios	42
3.4.1	Propuesta de valor	42
3.4.2	Segmento de clientes	42
3.4.3	Relaciones con el cliente	43
3.4.4	Canales	43
3.4.5	Recursos claves	43
3.4.6	Socios claves	44
3.4.7	Fuentes de ingresos	44
3.4.8	Estructura de costes	44
3.5	Diagnóstico de la Situación Actual	46
3.5.1	Problemas Identificados	46
3.5.2	Arquitectura de Procesos AS IS	52
3.5.2.1	Macroprocesos de Buk	52
3.5.2.2	Macro 1.3 Desarrollo y entrega de plataforma Buk	54
3.5.2.3	Macro 1.3.1 Desarrollo de plataforma Buk	57
3.5.3	Modelamiento BPMN Detallado de Procesos AS IS	59
3.5.3.1	Ideación y creación de la plataforma Buk	59
3.5.3.2	Levantamiento de defectos	60
3.5.3.3	Desarrollo de plataforma Buk	61
3.6	Cuantificación del Problema u Oportunidad	62
CAPÍTULO 4: PROPUESTA DE DISEÑO DE PROCESOS		63
4.1	Direcciones de Cambio	63
4.1.1	Estructura empresa y mercado	63

4.1.1.1 Situación Actual	63
4.1.1.2 Propuesta de Diseño	63
4.1.2 Anticipación	64
4.1.2.1 Situación Actual	64
4.1.2.2 Propuesta de Diseño	64
4.1.3 Coordinación	65
4.1.3.1 Situación Actual	65
4.1.3.2 Propuesta de Diseño	65
4.1.4 Prácticas de trabajo	65
4.1.4.1 Situación Actual	65
4.1.4.2 Propuesta de Diseño	66
4.1.5 Integración de procesos conexos	66
4.1.5.1 Situación Actual	66
4.1.5.2 Propuesta de Diseño	66
4.1.6 Mantención consolidada de estado	67
4.1.6.1 Situación Actual	67
4.1.6.2 Propuesta de Diseño	67
4.1.7 Arquitectura de Procesos TO BE	68
4.1.7.1 Diseño en IDEF0	68
4.1.7.2 Macroprocesos de Buk	68
Macro 1.3: Línea de desarrollo y entrega de la plataforma Buk	70
4.1.7.3 Macroproceso 1.3.1.1: Desarrollo de la plataforma Buk	71
4.1.8 Modelamiento Detallado de Procesos TO BE	72
4.1.8.1 Proceso “Ideación y Desarrollo de la Plataforma Buk”	72
4.2 Diseño de Lógica de Negocios	81
4.2.1.1 Identificación y análisis de defectos en la plataforma	81
CAPÍTULO 5: PROPUESTA DE APOYO TECNOLÓGICO	87
5.1 Especificación de requerimientos	87
5.1.1 Requerimientos No Funcionales	87
5.1.2 Requerimientos Funcionales	87
5.2 Arquitectura lógica	88
5.2.1 Capa de datos:	88
5.2.2 Capa lógica de negocio:	89
5.2.3 Estructura de un caso de prueba (Testcase) en Katalon Studio:	90
5.2.4 Capa de Visualización:	91
5.3 Herramientas	93
5.3.1 Katalon Studio	93
5.3.2 Katalon TestOps	94
5.4 Infraestructura Tecnológica	100
5.5 Prototipo Desarrollado	102
5.5.1 Flujo crítico de negocio seleccionado	102
5.5.2 Desarrollo de Script de Prueba Automatizada	102
5.5.3 Dashboard en Katalon TestOps	104

5.5.4	Resultados	106
CAPÍTULO 6: GESTIÓN DEL CAMBIO Y PROPUESTA DE IMPLEMENTACIÓN		
	107	
6.1	Gestión del Cambio	107
6.1.1	Liderazgo y gestión del proyecto de cambio	107
6.1.2	Sentido y estrategia del proceso de cambio	108
6.1.3	Cambio y Conservación:	108
6.1.4	Metaobservación, rediseño y seguimiento del proceso de cambio:	109
6.1.5	Prácticas en el cambio:	110
6.1.6	Gestión de las comunicaciones:	110
6.1.7	Gestión emocional del proyecto:	110
6.1.8	Gestión del aprendizaje y las habilidades:	111
6.1.9	Gestión del poder:	111
6.1.10	Evaluación y cierre:	111
6.2	Pasos de Implementación basados en la Gestión del Cambio	112
CAPÍTULO 7: EVALUACIÓN DEL PROYECTO		113
7.1	Evaluación Económica	113
7.1.1	Definición de Beneficios y Costos	113
7.1.2	Datos y Cuantificación	114
7.1.3	Flujo de Caja	114
7.1.3.1	Tasa de descuento CAPM	114
7.1.3.2	Situación Actual Optimizado	115
7.1.3.3	Situación con Proyecto	115
7.1.4	Análisis de Sensibilidad	117
CAPÍTULO 8: CONCLUSIONES		120
8.1	Conclusiones sobre el trabajo realizado	120
8.2	Trabajo futuro	121
CAPÍTULO 9: BIBLIOGRAFÍA		122
CAPÍTULO 10: ANEXOS		124

CAPÍTULO 1: INTRODUCCIÓN Y CONTEXTO

1.1 Historia de la empresa Buk

Fue en el año 2016, cuando Jaime Arrieta y Santiago Lira coinciden en EEUU mientras cursaban un MBA. Es en esta Universidad cuando Jaime y Santiago realizaban un proyecto de clase en donde se percatan que las empresas estadounidenses tenían ya digitalizados los procesos de RRHH, pero la situación en Chile era muy diferente ya que los procesos de las áreas de RRHH eran manuales lo que hacía fueran desordenados y lentos.

Por este motivo Jaime y Santiago al percatarse de la necesidad de esta área en su país de nacimiento Chile, deciden iniciar una Startup, pero al tener ellos el conocimiento en negocio, aún les faltaba el conocimiento tecnológico por lo que se contactan con los hermanos Felipe y Ricardo Sateler que se encontraban en ese momento en Chile, para poder llevar a cabo la creación de la plataforma.

El nombre de “Buk” nació en una reunión entre ellos cuatro. Cuando Jaime y Santiago regresan a Chile, tenían el objetivo de buscar un nombre que fuera corto, de manera que al ver un libro infantil derivaron de la palabra “Book” a lo que es ahora “Buk”, ya que representaba lo que buscaban que era que el producto fuera un libro abierto de conocimientos.

Tanto Jaime, Felipe, Santiago y Ricardo inician la búsqueda de clientes para finales del año 2017, en donde consiguen que una gran empresa llamada WOM del rubro de telecomunicaciones confíe en ellos, pero para lograrlo debían competir contra grandes empresas del rubro del software de Recursos Humanos, que ya se encontraban muy bien posicionados como lo es Oracle y SAP, pero WOM visualizo en Buk una enorme empatía al ser ambos emprendedores y decidió confiar en ellos, lo que resulto para Buk la obtención del inicio de su primer cliente grande.

Para los años 2019 y 2020, Buk se expande a Colombia y Perú, además adquiere en el año 2020 una compañía chilena llamada “Ctrl” que posee un software de asistencia que se integra con la plataforma de Buk.

En el año 2021 Buk inicia sus operaciones en México y a finales de este mismo año para cubrir las necesidades que su software carecía relacionadas a la capacitación de empleados, Buk adquiere dos empresas “Boost” y “Ninja Excel” en donde la primera es una aplicación de capacitaciones y la segunda es una aplicación que enseña Excel de una manera interactiva, lo que vienen a hacer de Buk el software de recursos humanos más completos dentro del rubro.

Actualmente, Buk cuenta con más de 4,000 clientes de diversas industrias buscando posicionarse como líder en Latinoamérica con su software de Recursos Humanos

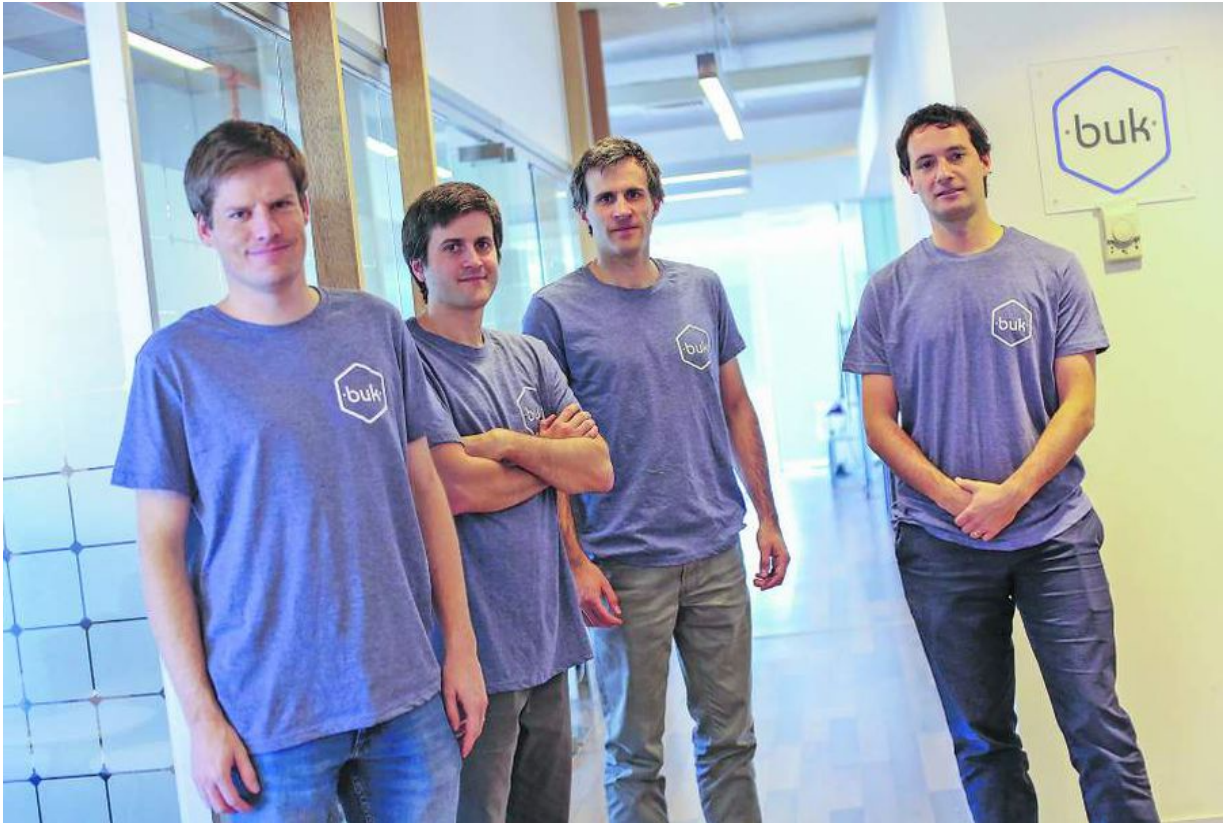


Figura 1: Santiago, Felipe, Ricardo y Jaime, fundadores de Buk.
Fuente: Extraído de La tercera, 2019

1.2 Descripción General de la Empresa

Buk es una Startup chilena que tiene como su producto SAAS, una plataforma que permite gestionar el área de Recursos Humanos, centrándose en el *pago de remuneraciones*, el *desarrollo profesional* y los *finiquitos* teniendo como propósito **“Crear un lugar de trabajo más feliz”**.

La plataforma facilita una mejor forma de llevar todos los procesos relacionados a las personas de la organización, puesto que busca centralizar toda la información en un mismo lugar, en donde cada empleado puede acceder a su portal para visualizar su información personal.

El software de Buk contiene tres pilares fundamentales:

- **Administración:** Incluye la gestión de remuneraciones, liquidaciones, pago PREVIRED, contratos, finiquitos, movimientos de personal, licencias, etc.
- **Talento:** Tiene como objetivo potenciar, identificar, atraer y retener el talento mediante distintas herramientas como evaluaciones de desempeño, desarrollo, capacitaciones etc.
- **Cultura:** Busca fortalecer la cultura en las organizaciones con distintas herramientas como beneficios, encuestas y comunicaciones.

El modelo de negocio de Buk se basa en *suscripciones*, cobrando entre US\$2 y US\$4 por empleado dentro de una empresa, siendo una excelente solución para empresas que tienen planillas variables. Y les permite a las empresas que son pequeñas poder incorporar la plataforma en sus procesos, pagando un bajo costo.

Buk se caracteriza por tener clientes tanto de empresas pequeñas, medianas y grandes de distintos rubros como lo son salud, financiero, educación entre otros. Dentro de sus clientes se pueden encontrar a WOM, SKY, Cornershop, GAP entre muchos más.


<p style="text-align: center;">Buk completo</p> 	TOTAL COLABORADORES	COSTO POR EMPLEADO MENSUAL	COSTO TOTAL MENSUAL
	1 a 40	0.15 UF + IVA	3 a 6 UF + IVA
	41 a 250	0.11 UF + IVA	4,5 a 27,5 UF + IVA
	251 a 500	0.08 UF + IVA	20,1 a 40 UF + IVA
	501 a 1000	0.07 UF + IVA	35,1 a 70 UF + IVA
	1001 a 2500	0.065 UF + IVA	65,1 a 162,5 UF + IVA

Figura 2: Tabla de precios por suscripción a Buk.

Fuente: Buk

Buk actualmente tiene presencia en Chile, Colombia, Perú y México. En Chile ha tenido para el año 2021 una tasa de cierre del 29%, en Colombia del 16% y en Perú del 13%. Iniciando en mayo del 2022 operaciones en su sede en México.



Figura 3: Presencia de Buk en Latinoamérica.
Fuente: Buk

1.3 Acerca del problema y su justificación

Buk se ha caracterizado por ser una empresa que ha experimentado un crecimiento acelerado dentro de las Startups chilenas, en donde cada vez son más los clientes que se van sumando a utilizar la plataforma. Gran parte se debe a que la empresa tiene como centro al cliente, buscando brindarles una plataforma que les de todo el apoyo necesario en las tareas diarias que tiene una empresa respecto a la gestión de remuneraciones, talento y desempeño de sus colaboradores. En aproximaciones con los stakeholder de la empresa y al conocer sobre sus dolores, en las reuniones se menciona constantemente que en lo que respecta a los primeros trimestres del año 2022 se presenta un alza en los defectos que reportan los clientes al usar la plataforma. Estos defectos son canalizados por los agentes del área de servicio de atención al cliente (SAC) mediante ya sea llamadas, correos electrónicos o la misma plataforma. Son los agentes los que tienen como función reportar estos defectos a un desarrollador del área de producto que tiene como rol ser el superhéroe, que nombran como “batman”. El batman cumple con la función de solucionar el defecto reportado por el cliente, este rol resulta ser temporal por un periodo de 2 semanas en el que es

seleccionado, lo que permite que todo el equipo de desarrollo en algún momento reciba este rol.

Se considera con los stakeholder del área de producto que la alza de defecto actualmente está resultando un problema, del cual no se está tomando ninguna medida y que definitivamente está afectando la experiencia del cliente al usar la plataforma, en donde muchos de los clientes no solo amenazan con el retiro de la suscripción sino que un 10% de ellos lo ha llevado a cabo porque consideran que funcionalidades como remuneraciones no pueden fallar debido a que en los cálculos afectan directamente a las nóminas de pago. Las salidas de los clientes descontentos llegan a afectar directamente los ingresos en ventas de la empresa, no solo por la pérdida de dinero en las suscripciones sino también porque es más difícil traer nuevos clientes y si la empresa de Buk quiere sostener que el cliente es su centro debe procurar brindarle la mejor experiencia.

En la figura 4, se presenta la cantidad de defectos críticos (son los defectos que afectan de una manera bloqueante la funcionalidad de la plataforma), obtenidos por la herramienta Freshdesk que utilizan los agentes SAC. Esta herramienta permite extraer la cantidad de defectos reportados en el periodo de julio 2021 a marzo 2022 y así poder visualizar de manera cuantitativa el problema y porque es necesario abordarlo antes que se llegue a un porcentaje demasiado alto de clientes que finalizan su suscripción con Buk.

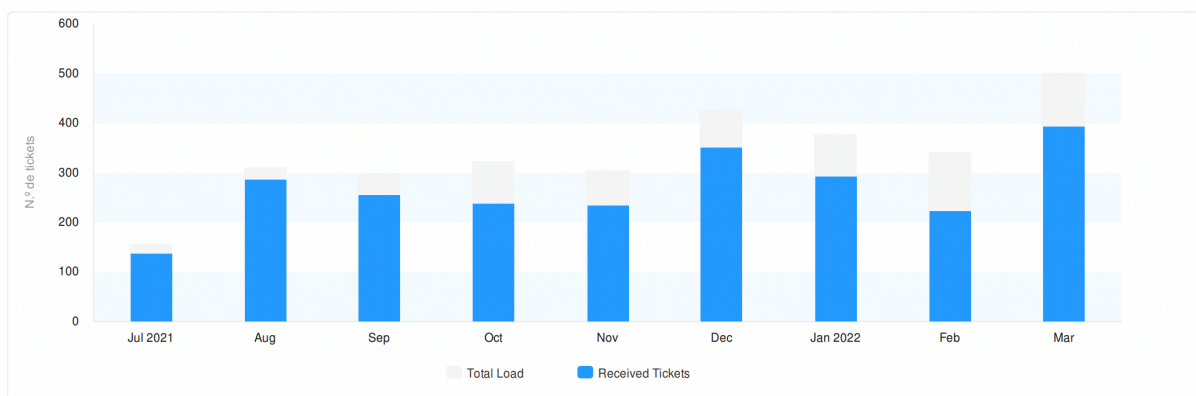


Figura 4: Cantidad de defectos reportados por clientes julio 2021- marzo 2022

Fuente: Obtenido en Freshdesk

1.4 Objetivos y Resultados Esperados del Proyecto

1.4.1 Objetivo General

Rediseñar el proceso del área de producto mediante aseguramiento de calidad de software para disminuir un 50% los defectos en la plataforma de la empresa Buk para el año 2023.

1.4.2 Objetivos Específicos

- Identificar y analizar los procesos actuales del área de producto, en donde se realiza el desarrollo de la plataforma e impactan los defectos reportados por los clientes.
- Rediseñar el proceso del área de producto donde se identifique el problema de la alta cantidad de defectos, implementando un proceso enfocado en aseguramiento de calidad de software.
- Aplicar metodologías y buenas prácticas de calidad de software que apoye al diseño del proceso de aseguramiento de calidad, usando el modelo de calidad de software COBIT 4.1 como complemento a la metodología de rediseño de procesos de Barros.
- Construir un prototipo mediante herramientas tecnológicas que contenga la lógica de negocio. Permitiendo obtener información si realmente el rediseño del proceso del área de producto logra la disminución de los defectos en la plataforma de Buk.
- Incorporar gestión del cambio en la implementación del proyecto ante cualquier resistencia que se presente y permita favorecer la conclusión exitosa proyecto.
- Realizar evaluación del proyecto, basándose en un análisis económico que evalúe la rentabilidad y permita determinar la viabilidad del mismo.

1.5 Resultados Esperados

Es importante destacar que en el rediseño del proceso se pretende tener los siguientes resultados:

- Análisis de los procesos actuales que conforman el área de producto y el área de servicio al cliente que permitan encontrar las hipótesis del problema presentado.

- Rediseño del proceso de desarrollo de producto incorporando metodología de calidad de software llamado COBIT 4.1.
- Mostrar resultados de las ejecuciones realizadas de las pruebas automatizadas mediante el Dashboard que se creara en la herramienta de Katalon TestOps.
- Implementación de prácticas y planes de calidad que permitan monitorear que el desarrollo de la plataforma se entregue al cliente libre de defectos.
- La inserción de una cultura de calidad en el área de producto, conscientizando tanto a los desarrolladores, product owner y bussines analyst la importancia de agregar buenas prácticas de calidad en todo el proceso de desarrollo de la plataforma.
- Presentar la viabilidad económica del proyecto a los stakeholder de la empresa.

A continuación, se presenta una tabla con algunos de los indicadores considerados en los objetivos planteados anteriormente:

Tabla 1: *Tabla de resultados esperados.*
Fuente: *Elaboración propia*

INDICADOR	AS IS	TO BE
Cantidad de defectos reportadas por los clientes	907 (Trimestre 2022)	453 (Trimestre 2023)
Cantidad de errores encontrados en etapa de desarrollo por el profesional con conocimiento en calidad de software	NA	288
Cantidad de dinero perdido por retiro de suscripción	USD 1,585,911.11	USD 15,895.11

El rediseño del proceso incorporando aseguramiento de calidad permitirá disminuir la cantidad de defectos en un 50%. También se espera que los profesionales con conocimientos en calidad de software (QA), sean capaces de encontrar errores en etapa de desarrollo, donde según menciona (Pressman, 2010) puede ser un aproximado de 24 errores los que un QA puede encontrar. Representando en 12 células (las células en Buk son equipos conformados por desarrolladores y personas de negocio como los “Business Analyst y los Product Owner”), un aproximado de 288 errores que se podrán detectar en etapa de desarrollo. Es fundamental destacar que se espera que con el rediseño el porcentaje de salida de los clientes sea del 1%,

impactando positivamente en los ingresos de la empresa, dando como consecuencia que solamente se pierda la cantidad USD \$15,895.11 en comparación con la pérdida actual que es aproximadamente de USD \$1,585,911.11.

1.6 Alcance

1.6.1 El proyecto abarcará:

- El rediseño del proceso se incluirá en toda el área de desarrollo de producto.
- La incorporación de una nueva área de aseguramiento de calidad (QA), que tendrá un profesional encargado de velar por la calidad del producto.
- La definición de los indicadores que permitan medir la relación de calidad con los defectos reportados por SAC.
- La implementación de la lógica de negocio que apoyará al aseguramiento de calidad de la plataforma.

1.6.2 El proyecto No abarcará:

- Debido a que el tiempo es un limitante no se realizará la implementación del proceso en las doce células que conforman el equipo de desarrollo del área de producto. Solamente se implementará en una célula que será la que realiza funcionalidades más críticas de la plataforma que impactan directamente en el negocio. Que para este caso la célula con la que se trabajara el proyecto será la célula de Remuneraciones.
- Para fines del proyecto solamente se abarcará en Buk Chile, los demás países no serán considerados, ya que actualmente no se encuentran datos que permitan conocer el estado de defectos levantados por clientes de esos países.

1.7 Riesgos Potenciales

A continuación, se describen los riesgos que serán presentados por tipos, los cuales serán evaluados con tres ponderadores (*severidad*, *ocurrencia* y *detección*). En donde se parte que la *severidad* es la criticidad que puede afectar a que el proyecto no pueda llevarse a cabo, *ocurrencia* se asumirá si se presenta en múltiples ocasiones y *detección* si se asume que es un riesgo que puede ser prevenido y por lo tanto se deberán tomar medidas. Estos indicadores permitirán analizar la criticidad de los riesgos.

Los ponderadores tendrán una escala de 1 a 10, en el que mayor sea su valor dará como resultado riesgos críticos. Para poder realizar estas evaluaciones se hizo mediante una reunión con el CTO y el Head de producto de Buk, en donde en la tabla 2, se puede observar cómo fue ponderado cada riesgo y qué medidas se tomarán para los riesgos más críticos.

Tabla 2: Tabla de riesgos potenciales.
Fuente: Elaboración propia

Tipo de riesgo	Descripción	Ponderadores			
		Severidad	Ocurrencia	Detección	Promedio
Económico	Presupuesto limitado para la compra de herramientas de calidad que en su mayoría son de pago	10	1	1	4
Económico	Presupuesto limitado para la adquisición de profesionales con conocimientos en calidad de software	10	1	1	4
Externos	Encontrar profesionales con conocimientos en calidad de software, debido a la alta demanda y a los muy pocos perfiles especializados en pruebas manuales y automatizadas	10	8	8	8.6
Externos	Predisposición en el equipo de	10	5	5	6.6

	desarrollo al implementar controles de calidad				
Tecnológicos	Problemas de integración de herramientas de calidad con las tecnologías actuales de la empresa Buk	9	6	5	6.6

Como observamos en la tabla 2, se analizan los riesgos que son críticos por lo que se concluyó en conjunto que se deberán tomar las siguientes medidas para poder mitigarlos que se presentan en la tabla 3:

Tabla 3: Tabla de medidas ante riesgos.
Fuente: Elaboración propia

Tipo de riesgo	Descripción	Medidas
Externos	Encontrar personas con conocimientos en calidad de software, debido a la alta demanda y a los muy pocos perfiles especializados en pruebas manuales y automatizadas	<ul style="list-style-type: none"> • Compartir procesos de reclutamiento en diferentes plataformas (LinkedIn, GetonBoard, etc.) • Apoyarse del beneficio de referidos a los colaboradores de la empresa
Externos	Predisposición en el equipo de desarrollo al implementar controles de calidad	<ul style="list-style-type: none"> • Realizar charlas al equipo de desarrollo sobre el área QA, la importancia de ella y sus funciones. • Evangelizar una cultura de calidad es responsabilidad de todo el equipo del área de producto.
Tecnológicos	Problemas de integración de herramientas de calidad con las tecnologías actuales de la empresa Buk	<ul style="list-style-type: none"> • Buscar e incorporar herramientas que sean compatibles con las tecnologías usadas en Buk.

CAPÍTULO 2: MARCO TEÓRICO

2.1 Metodología de Ingeniería de Negocios

La metodología de ingeniería de negocios impulsada por el autor de (Barros, 2015), desarrolla una metodología que permite integrar la gestión de empresas, la arquitectura empresarial, el rediseño de procesos y las tecnologías de información.

La metodología se compone por seis etapas, que inician con definiciones a nivel macro hasta llegar al detalle:

1. **Definición de Posicionamiento Estratégico:** Busca definir la ventaja competitiva en la cual una empresa domina su mercado en el tiempo. La propuesta de Hax plantea 3 tipos:
 - a. **Estrategias del mejor producto:** Busca el liderazgo en el mercado ya sea persiguiendo el costo más bajo, u otras características como que el producto sea estandarizado, que los clientes sean numerosos y genéricos. Dentro de las desventajas que contiene es que los clientes no tienen mucha lealtad y las guerras de precios entre competencia.
 - b. **La estrategia de solución integral para el cliente:** Pretende incorporarse en la cadena de valor del cliente. Dentro de sus características tiene un conocimiento profundo del cliente e innovación en conjunto con el cliente de sus productos.
 - c. **El lock-in sistémico:** Procura crear condiciones que hagan que para un cliente sea casi imposible prescindir de los servicios de una empresa. Dentro de las características es que su cadena de valor se extiende desde los clientes, proveedores y sus complementadores.

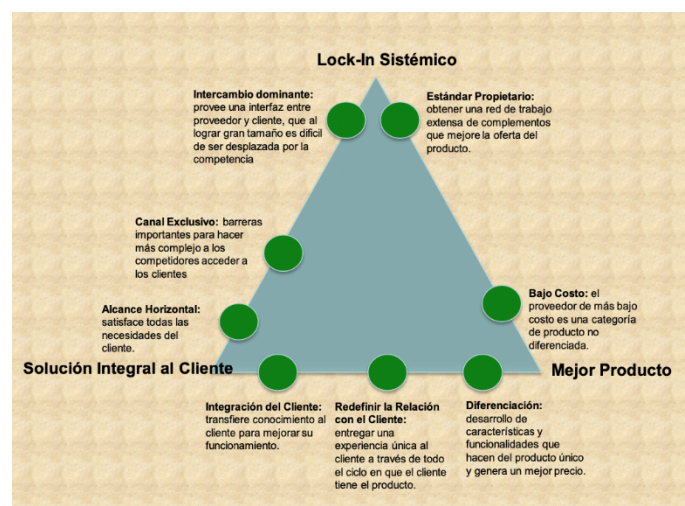


Figura 5: Alternativas de Posicionamiento Estratégico.

Fuente: Barros, 2015

2. **Desarrollo de Modelo de Negocios:** Pretende explicar que es lo que conforma una empresa, quienes son sus clientes, como los generan y que es lo que valoran. Además, un modelo de negocio determina cuales son los procesos que se van a requerir y cuál será su estructura.

Los factores que considera un Modelo de Negocio son:

- a. **Propuesta de Valor Al Cliente:** Cuales son los clientes objetivos y que problema se resolverá o necesidad será suplida.
- b. **Formula de Utilidades:** Lo integra el modelo de ingreso que precios se les dará a los productos o el servicio, cuánto dinero se ganará. La estructura de costos que tendrán y que tasa de uso de recursos será necesario se usen para crear el inventario.
- c. **Recursos claves:** Es lo necesario que permita poder entregar al cliente una interesante propuesta de valor, lo conforman los recursos humanos, los canales, la tecnología, etc.
- d. **Procesos Claves:** Contiene los procesos como la creación del producto, manufactura, mercadeo, entrega. Además de las métricas que giran en torno de la empresa.

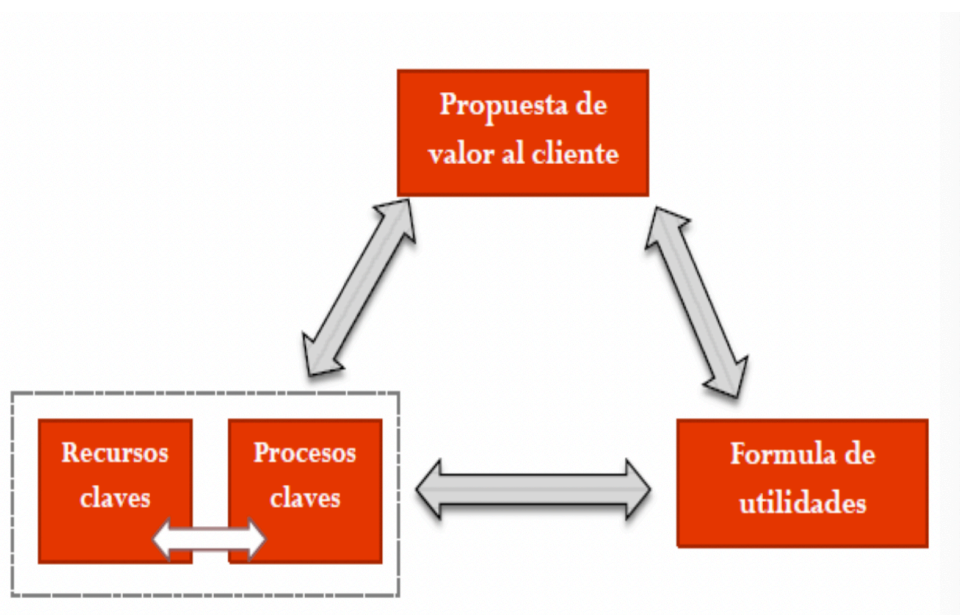


Figura 6: Factores a considerar en un modelo de negocio.

Fuente: Barros, 2015

3. **Diseño de Negocio y Arquitectura Empresarial:** La arquitectura empresarial busca establecer los procesos que son necesarios para implementar las capacidades, el diseño de negocio y la lógica de negocio.
4. **Diseño detallado de procesos:** Para que pueda llevarse a cabo primero es necesario definir el posicionamiento estratégico y el modelo de negocio lo que permita poder crear procesos que vayan alineados a la empresa.
5. **Diseño Apoyo TI:** Busca redefinir los requerimientos de apoyo TI para brindar apoyo a la fase de diseño mediante nuevas arquitecturas tecnológicas y de sistemas.
6. **Construcción, Implementación y Operación:** Lleva a la practicas los diseños de procesos que han sido asociados a TI y en cómo se puede llevar a cabo con las herramientas correctas.

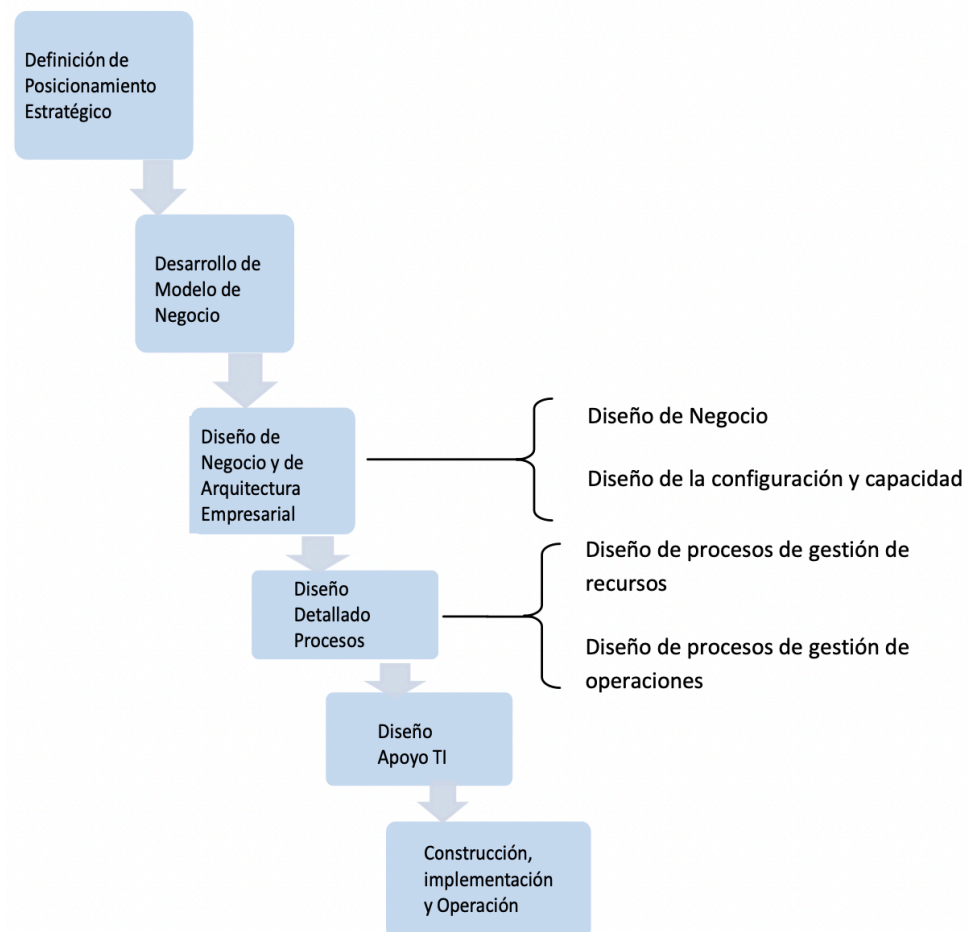


Figura 7: Metodología de Ingeniería de Negocios.
Fuente: Barros, 2015

2.2 Patrones de Arquitectura y Procesos de Negocio

El autor de (Barros, 2015) expone que todas las empresas tienen procesos muy parecidos, por lo que a partir de ello define patrones de arquitectura de procesos para poder realizar a partir de ellos el rediseño de los procesos.

La Arquitectura de Macroprocesos contiene procesos y estos contienen subprocesos y estas actividades. En donde los subprocesos son un conjunto de actividades. Y las Actividades son las encargadas de ejecutar las prácticas de trabajo que contiene la lógica de negocio que contiene el proceso.

Se distinguen cuatro patrones de Macroprocesos como se observa en la figura 8:

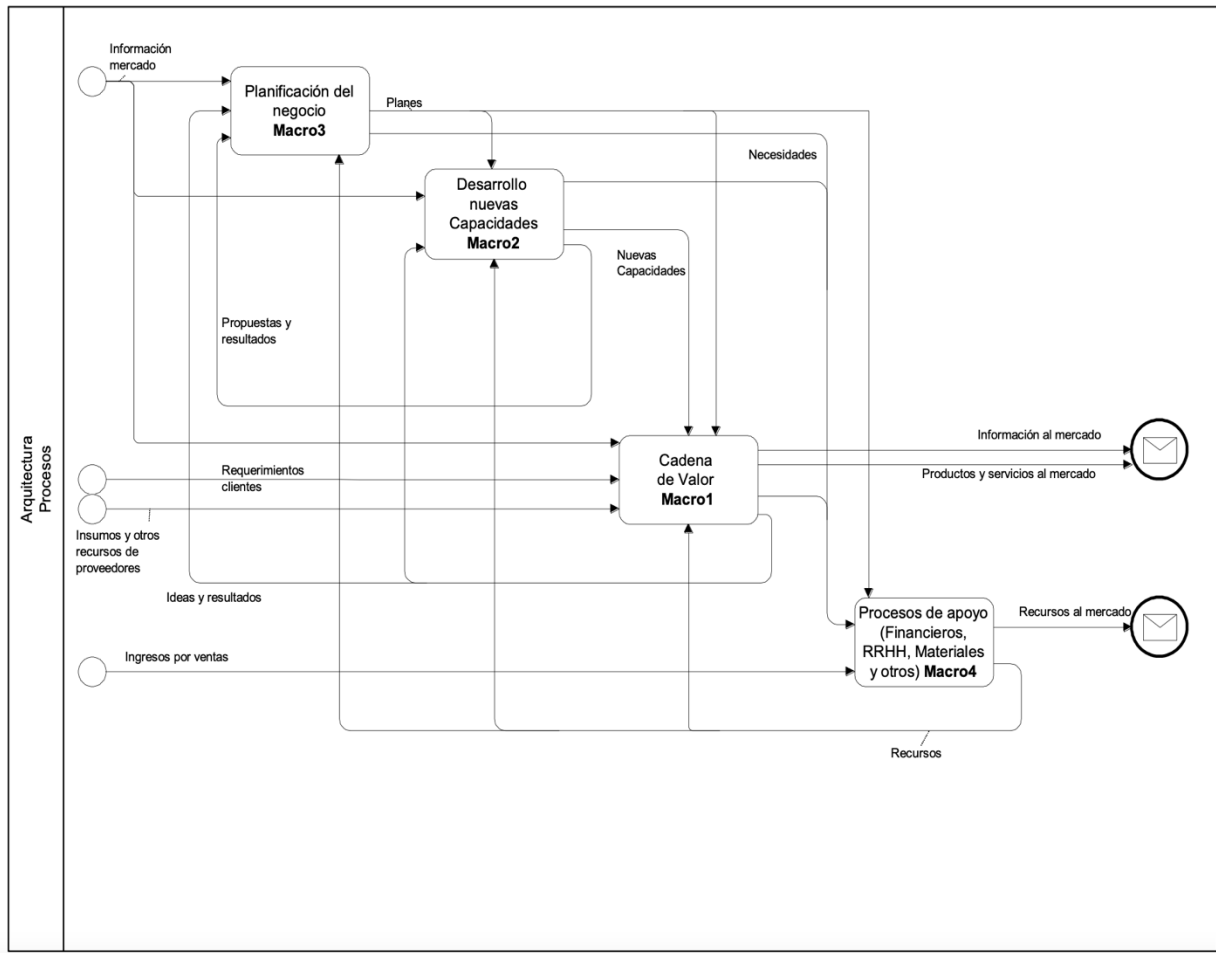


Figura 8: Arquitectura de Macroprocesos.

Fuente: Barros, 2015

1. Macroproceso 1 “Cadena de Valor”: Esta Macro representa los procesos en donde se ejecuta la producción ya sea de productos y/o servicios. Se asemeja a lo que se llama la Cadena de Valor como menciona (Porter, 1996). Se centra en la producción del bien o servicio que se quiere entregar al cliente. Siendo el corazón del negocio.

Esta Macro está compuesta por cuatro procesos:

- **Administración de la relación con el cliente:** Se centra en las actividades que se requieren para mercadear y generar las ventas, el contacto con el servicio al cliente. Busca generar un plan de ventas y brindar un buen servicio al cliente.
- **Administración de la relación con proveedores:** Busca generar todos los requerimientos necesarios para abastecer los inventarios, encontrar los proveedores además de la adecuada planificación y realizar la programación de las entregas.
- **Gestión de producción y entrega:** Se encarga de la planificación, el programa y control que serán requeridos para entregar el producto o servicio. También apoya a realizar el análisis de demanda y la logística de la distribución.
- **Producción y entrega de productos o servicios:** Implementa los planes y programas del proceso de Gestión de producción y entrega, realizando las actividades de distribución del producto o servicio como puede ser manufactura de productos y desarrollo de software, entre otros.

2. Macroproceso 2 “Desarrollo de Nuevas Capacidades”: Esta Macro representan los procesos que desarrollan las nuevas capacidades necesita para poder estar en constante innovación.

Esta Macro está compuesta por cuatro procesos:

- **Evaluación de necesidad de nueva capacidad:** Se encarga de evaluar económicamente el proyecto, mediante estudios de mercado, analizando indicadores que validen la factibilidad.
- **Gestión, diseño y construcción de nueva capacidad:** Responsable de ejecutar las actividades para determinar los recursos que serán necesarios para el proyecto y asignar los recursos.
- **Diseño y construcción de nueva capacidad:** Ejecuta los planes que son originados de los procesos de gestión, se encarga de realizar el prototipo.
- **Mantenimiento de estado:** Recibe los cambios que va produciendo el desarrollo de las capacidades y ofrece la información al proceso de “Diseño y construcción nueva capacidad”.

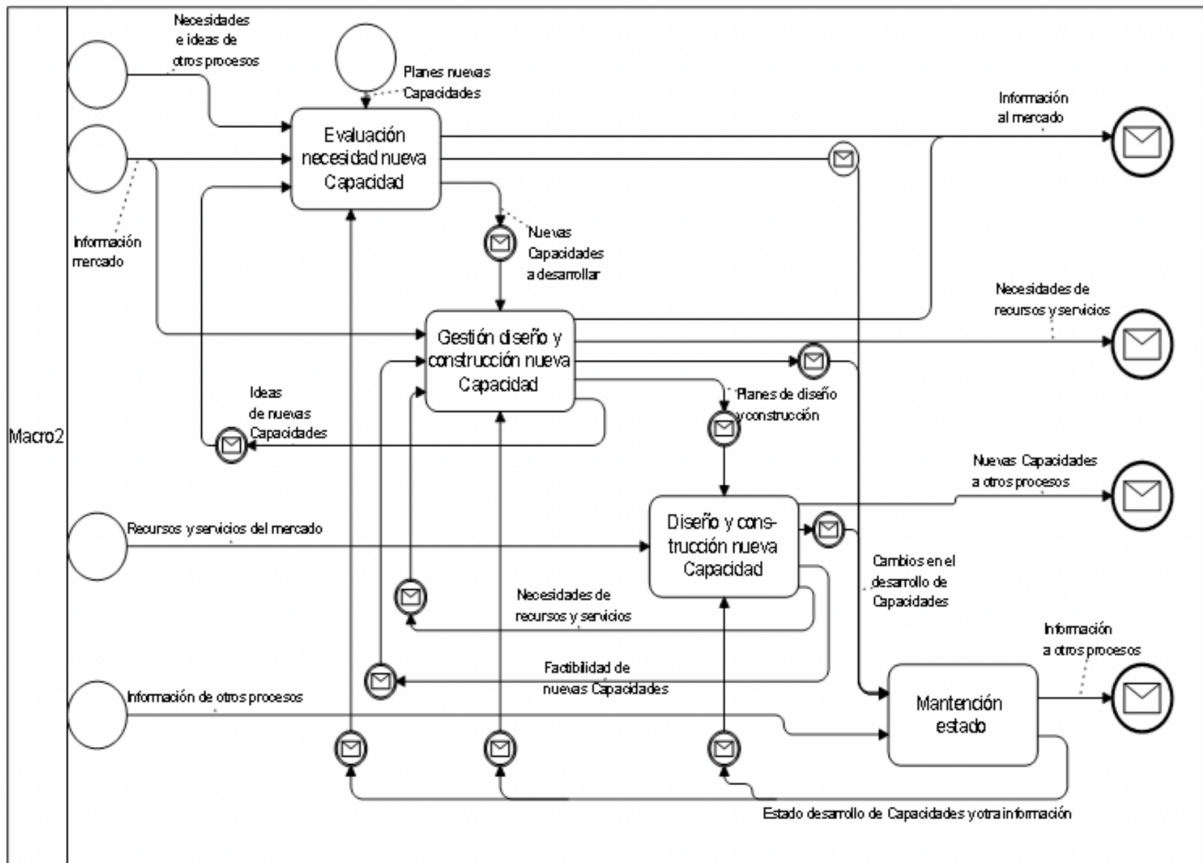


Figura 9: Macroproceso 2 - Desarrollo de Nuevas Capacidades.
Fuente: Barros, 2015

3. Macroproceso 3 “Planificación del Negocio”: Es la Macro que apoya la planificación del negocio, enfocado en las estrategias de la organización que dan como resultados los planes que deben seguir los demás Macros.

Esta Macro está compuesta por cuatro procesos:

- **Definir el concepto del negocio y visión:** Se basa en tendencias económicas, aspectos políticos explotando tendencias de innovación en el negocio. Busca definir la visión, la misión y objetivos.
- **Gestionar el desarrollo de la estrategia:** Produce los planes Desarrollo estrategia, así como asignaciones tanto financieras y de recursos humanos.
- **Desarrollo de la estrategia:** Define la estructura organizacional, necesidades de procesos o rediseños de acuerdo al mercado y rubro de la empresa.
- **Mantenimiento de estado:** Recibe los cambios en la gestión de la estrategia y brinda información sobre el estado al proceso de “Desarrollo de la estrategia”.

4. Macro “Gestión de recursos habilitadores”: Tienen como objetivo ser el apoyo requerido para los procesos, ya sea mediante recursos humanos, financieros, infraestructura, etc.

Esta Macro está compuesta por cuatro procesos:

- **Obtener recurso:** Pretende determinar las necesidades y los responsables de ejecutar las acciones para obtener los recursos. Busca determinar los recursos que tiene la empresa y estimar lo que se necesitara a futuro a partir de los planes de producción de la empresa. Además de cuando se debe contratar gente, adquirir nuevos equipos y provisionar insumos para la oficina.
- **Decidir manejo de recurso:** Se encarga de asignar los recursos que están disponibles de acuerdo a los requerimientos de otros Macroprocesos.
- **Decidir transferencia de recurso:** Determina donde deben enviarse los recursos ya sea equipo, dinero y empleados.
- **Ingreso, manejo y transferencia de recursos:** Trabaja en conjunto con la actividad de manejo de recursos como contratar personal, invertir dinero, pago a proveedores, etc.

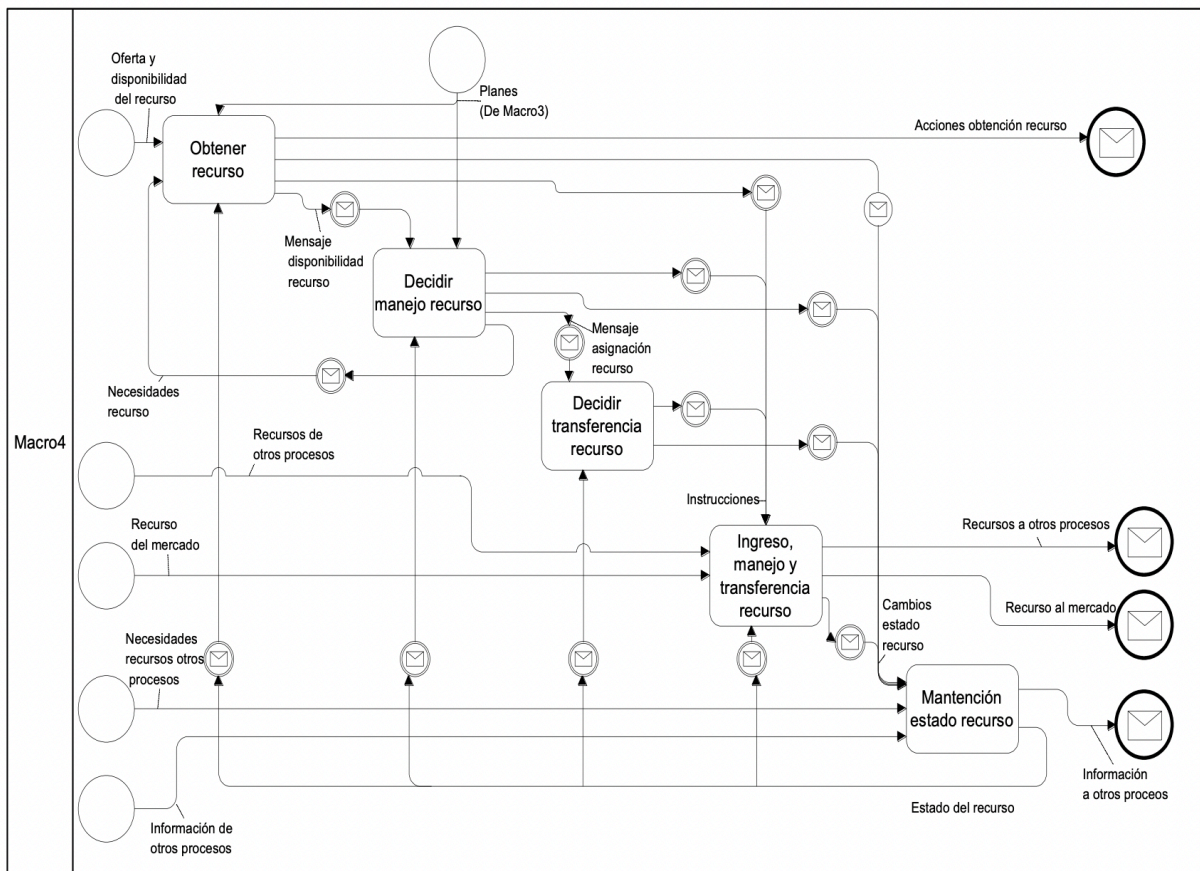


Figura 10: Macroproceso Gestión de recursos habilitadores.

Fuente: Barros, 2015

2.2.1 Procesos de la Macroproceso Cadena de Valor (Macro1)

Como se menciona anteriormente, esta Macro 1 es la encargada de la producción por lo que también es llamada “cadena de valor”. Esta Macro contiene los siguientes procesos:

- **Administración relación con el cliente:** Este proceso tiene como objetivo el enfoque en marketing, las actividades que se centran en ventas y lo referente al servicio al cliente.
- **Administración relación con proveedores:** Se centra en la planificación, inventarios, la planificación de abastecimiento y como debe ser la logística de la distribución.
- **Gestión de producción y entrega:** Se encarga de realizar la planificación, la programación que se debe seguir para la creación y entrega del producto o del servicio, basándose en la demanda, la capacidad.
- **Producción y entrega de productos o servicios:** Se encarga de ejecutar la planificación realizada en el proceso anterior, como por ejemplo lo que es la manufactura de productos, **desarrollos de software**, entre otras.
- **Mantenimiento de Estado:** Recibe toda la información necesaria de acuerdo al estado de producción que se realiza en el proceso de “Producción y entrega de productos o servicios”.

La estructura de Macro 1 se encuentra conformada por el proceso de **Ejecución** “Producción y entrega de productos”; “Administración y relación con proveedores, Administración y relación con el cliente y Gestión y producción y entrega” que son procesos de **Gestión** y “Mantenimiento de estado” que es un proceso homónimo como se observa en la figura 11.

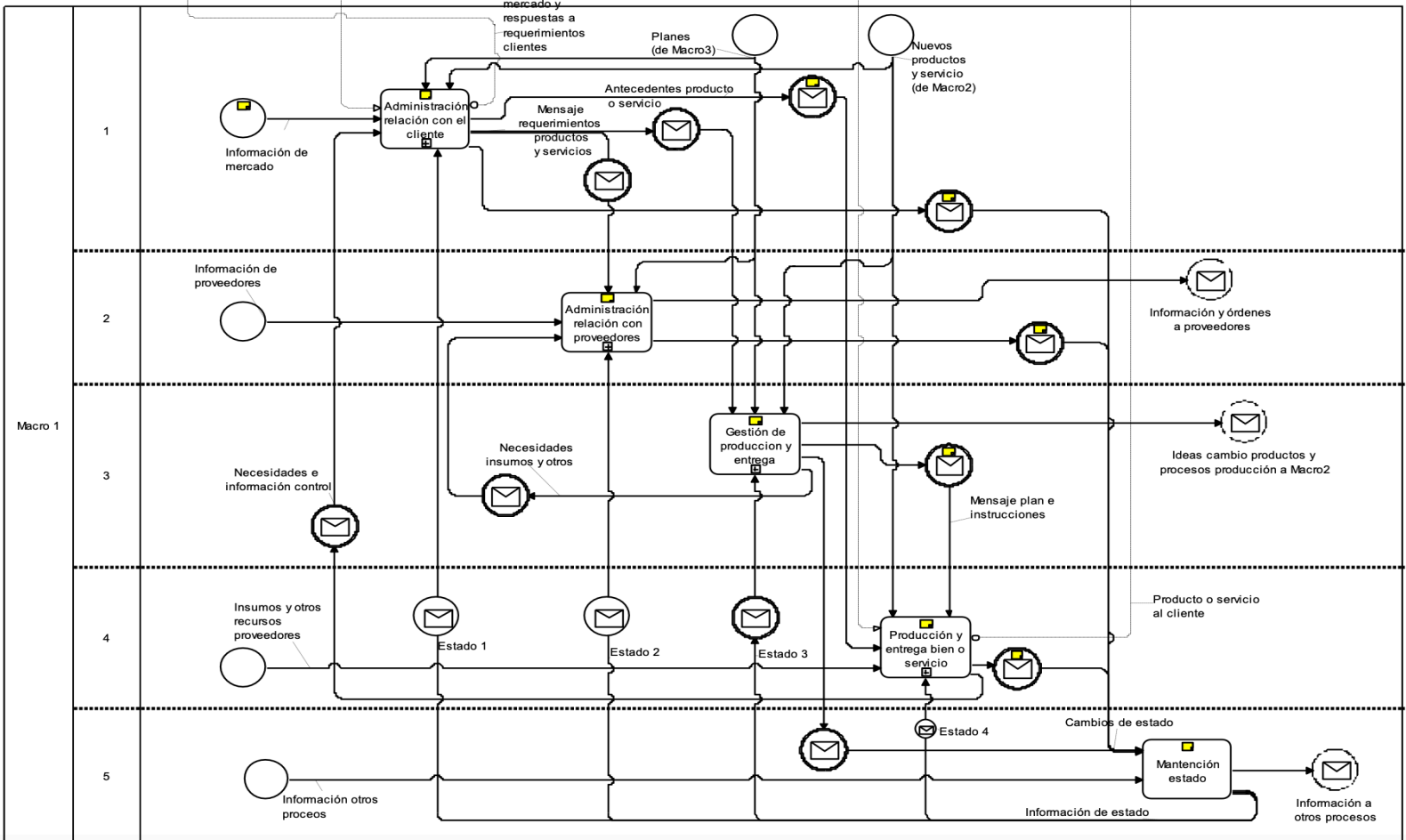


Figura 11: Primer nivel de Macro 1.
Fuente: Barros, 2015

2.2.2 Macro 1.3.3 Subproceso Producción

Pertenece al proceso de “Producción y entrega de bien o servicio”, este subproceso de “Producción” realiza todas las actividades que son necesarias para generar el producto o servicio que será entregado al cliente.

En (Barros, 2015) se menciona que *Producción*, tiene la característica que no puede detallarse finamente ya que depende de lo que se producirá, por lo que no es lo mismo producir un producto alimentario como un producto de software. Sin embargo, se tiene una estructura común que sirve como base y es válida para cualquier producto.

Es en esta macro donde se puede representar lo que es la programación de código de desarrollo de software para la creación de un producto digital. Este proceso es importante porque para poder desarrollar un producto es fundamental medir los resultados en los que se puede retroalimentar por *Mantenimiento de estado* y que pueda ser retroalimentado por *Monitores y control*, siendo el que permite tomar medidas correctivas, sucediendo el *control de calidad* en un proceso de desarrollo de software, en donde se toman los programas y se someten a pruebas para establecer si funcionan de acuerdo a lo previsto, sino tomar acciones correctivas.

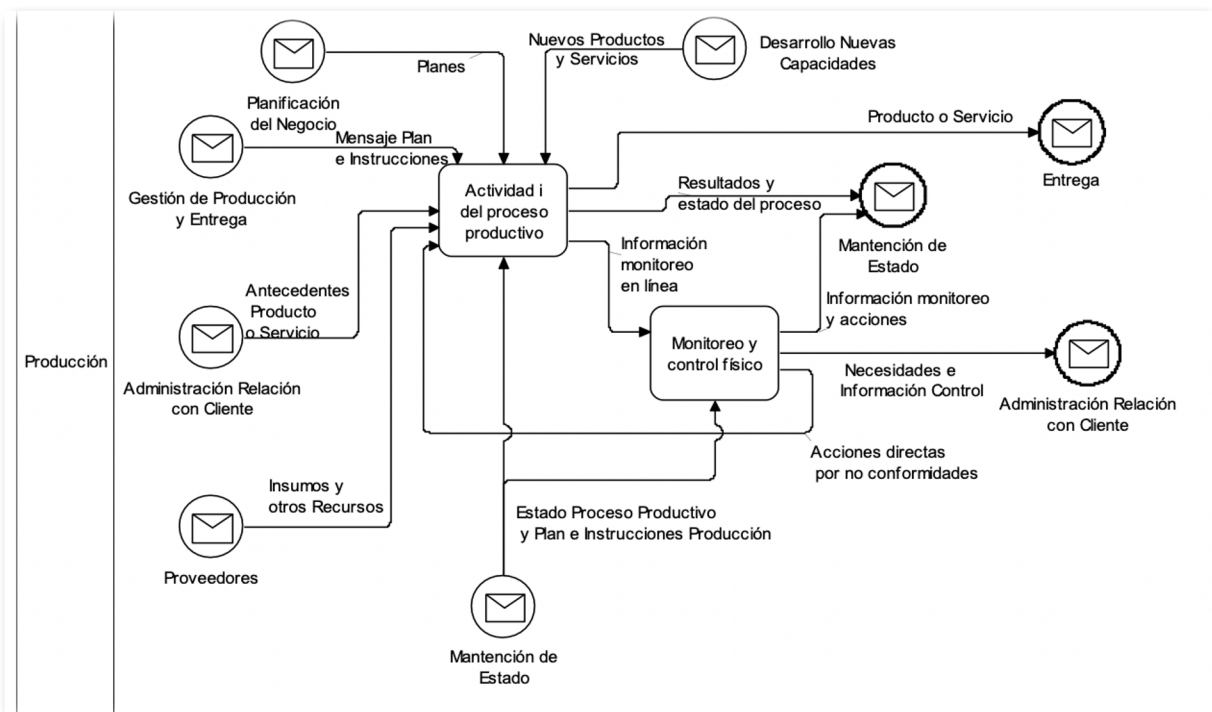


Figura 12: Subproceso de Producción de Macro 1.

Fuente: Barros, 2015

2.3 Calidad

2.3.1 Definición de Calidad

El autor (Juran, 1986) define la calidad como el “rendimiento del producto que da como resultado la satisfacción del cliente”.

2.3.2 Calidad de software

La IEEE Std. 610-1990 menciona que el concepto de calidad se puede definir como “el grado con el que un sistema, componente o proceso logra cumplir con los requerimientos y necesidades del cliente”.

Mientras en (Pressman, 2010) se define la calidad como un proceso eficaz de software aplicado para crear un producto que sea útil que proporciona valor tanto a la persona que lo usa como a quien lo desarrolla.

Según el autor de (Lewis, 2010) la calidad puede ser vista desde dos puntos de vista. El primero se relaciona con cumplir con los requerimientos y el segundo con medir la calidad a través del cliente.

En (Secades, 2007) se menciona que al finalizar el software debe contener las siguientes capacidades:

- **Fiabilidad:** El producto debe operar sin errores.
- **Comprensible:** Facilidad de comprender el producto.
- **Modificable:** Debe ser fácil de realizar cambios.
- **Probable:** Capacidad de construir y ejecutar casos de prueba.
- **Portable:** Producto fácil de manejar.
- **Utilizable:** El producto debe ser sencillo.
- **Rendimiento:** El producto debe ser veloz.

2.3.3 El Costo de la Calidad de software

Se tiene claro que la calidad tiene un costo, pero también la mala calidad lo tiene, no solo afectando al usuario final que utiliza el software sino a la organización del software que lo desarrollo y debe realizar mantenimiento.

Por lo que es importante definir el costo de tener calidad en el software y el costo de software de mala calidad.

El costo de la calidad: Involucra todos los costos que conllevan agregar calidad, y se pueden desglosar de la siguiente manera:

- **Costos de prevención:** Incluye los costos de actividades para realizar la planificación y coordinación de las actividades de aseguramiento de calidad, el costo de planear las pruebas que se deben realizar en el software.
- **Costos de evaluación:** El costo de ejecutar las pruebas y recabar datos para evaluar.
- **El costo de la falla:** Se subdivide en internos y externos:
 - **Costos internos de falla:** El costo de corregir errores.
 - **Costos externos de falla:** Son los defectos encontrados después de que el producto se encuentra en ambiente productivo, el producto que usan los clientes finales. Entre los ejemplos que se puede definir en este tipo de costos son los que se deben solucionar por quejas, ayuda en línea, etc.

Es sabido que un software de mala calidad produce daño en la reputación de una empresa y aunque sea difícil de cuantificar, es real que si se produce software de mala calidad pueden suceder percances muy costosos para la empresa.

En (Pressman, 2010) se menciona que el costo promedio por corregir un defecto durante el desarrollo es de US\$ 1,309.56 por error, pero si el error se descubre en producción el costo promedio es de US\$ 9,564.99 por error. Las cifras que menciona Pressman al ser del año 2010 se actualizaron de acuerdo a la inflación llevándolos a cifras que concuerdan para el año 2022. (INEGI, 2022)

Cigital (Cig, 2007) precisa que una aplicación grande puede llegar a contener 200 errores introducidos durante la etapa de codificación. Por lo que de acuerdo con los datos promedio el costo de encontrar y corregir defectos durante la etapa de desarrollo puede llegar a ser (200 x USD \$ 1,309.56) resultando a un valor aproximado de US\$ 261,911.26.

Sin embargo, de acuerdo a los datos promedio de la industria indican que encontrar y corregir un defecto durante etapas productivas es de USD \$ 9,564.99, suponiendo que en la etapa de desarrollo se encuentran el 25% el costo sería (50 x USD \$ 9,564.99) dando USD \$ 478,249.43. Pero restan 150 errores no corregidos que tendran el costo de USD \$ 18,902.11. Un error encontrado en etapa de producción será de (150 x US\$ 18,902.11) dando como resultado el valor de USD \$ 2,835,316.73. Entonces el costo total de encontrar y corregir los 200 defectos que se encuentran en producción es de USD \$ 3,313,566.16 que resulta sumamente alto comparado con los costos de corregir defectos cuando se implementa aseguramiento de calidad, que es de USD \$ 261,911.26.

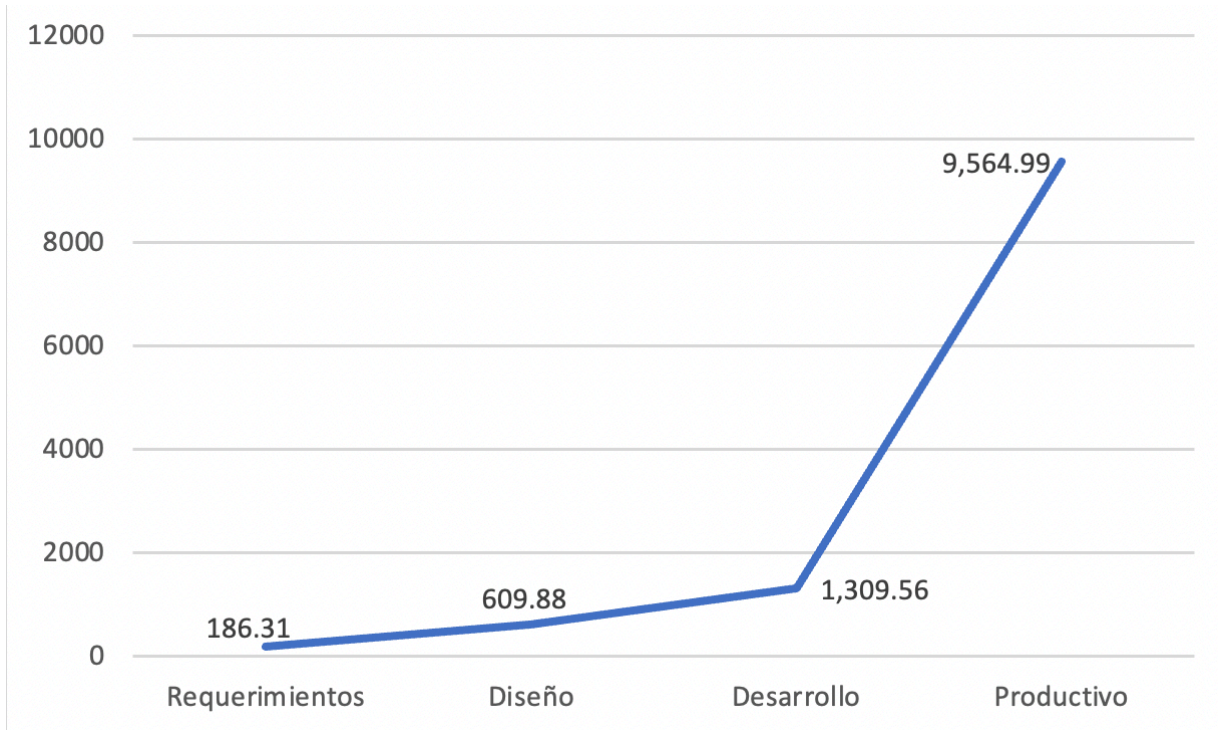


Figura 13: Gráfico costos de defectos.
Fuente: Elaboración propia

Si bien los datos anteriormente son un valor aproximado de los costos que podrían significar encontrar errores en etapa de desarrollo y productivo, se puede observar una significativa diferencia de cuando se encuentran errores en etapa de desarrollo.

Es importante mencionar la diferencia entre **defecto** y **error**. Si bien los dos implican un problema de calidad, el *defecto* es el problema encontrado por el usuario *después* de haberse liberado el software mientras que el *error* es un problema descubierto *antes*, en etapa de desarrollo. (Pressman, 2010).

2.3.4 Control de Calidad

Se puede definir el control de calidad como un conjunto de acciones que ayudan a asegurar que el producto cumpla con las metas de calidad. Se busca inspeccionar teniendo como objetivo descubrir y corregir errores en el software. (Pressman, 2010).

2.3.5 Aseguramiento de Calidad

El Aseguramiento de Calidad es el conjunto de procesos y actividades que proveen de toda información referente a calidad del producto, además establece los métodos y las acciones de control de calidad que tiene como objetivo mantener informado al equipo

de desarrollo referente a la calidad del producto, identificando los errores y realizando análisis de los defectos.

El Aseguramiento de Calidad según menciona el autor de (Pressman, 2010) debe incluir:

- Un proceso de aseguramiento de calidad.
- Control de la calidad, que pueden ser estrategias de pruebas de acuerdo a las necesidades.
- Buenas prácticas de Ingeniería de software, que pueden ser métodos y herramientas.
- Control de los cambios que sufre el software.
- Procedimiento que garantice el cumplimiento de los estándares o requerimientos, cuando sea aplicable.
- Mecanismos que permitan medir y reportar.

2.3.5.1 Antecedentes del Aseguramiento de Calidad

En las décadas de 1950 y 1960, la calidad era responsabilidad únicamente del programador. No es sino hasta 1970 que se introducen en los contratos para desarrollar software militar en EEUU estándares de calidad lo que produjo se introdujeran prácticas de calidad en el software comercial.

El equipo que conforma el Aseguramiento de Calidad, también llamados “Quality Assurance”. Son los responsables de velar que el software cumpla con los factores de calidad, además de ser responsables de planear, ejecutar y hacer reportes acerca de la calidad.

Las tareas que realiza un QA son las siguientes:

- **Preparar el plan de Aseguramiento de Calidad o Plan de Calidad:** En el plan se identifican el tipo de pruebas que se realizarán, los estándares, prácticas y los procedimientos para reportar errores.

El Plan de Aseguramiento de calidad debe contener:

- Propósito y alcance del plan.
- Descripción del producto (modelos, documentos, código fuente).
- Tareas del equipo de QA.
- Herramientas y métodos que apoyen las acciones del equipo QA.
- Roles y responsabilidades en relación a la calidad del producto.

- **Participar en el desarrollo de requerimientos del software:** Revisa en conjunto con otros roles del equipo de producto los requerimientos de clientes que debe tener el producto.
- **Verificar el cumplimiento de los requerimientos:** Debe validar se cumple con lo establecido por negocio.
- **Registrar y reportar todo error y defecto:** Debe documentar los errores encontrados y hacer seguimiento hasta que sean resueltos.

2.3.5.1.1 Revisiones

Son el conjunto de acciones que se requieren como buenas prácticas en la Ingeniería de Software. Cada una de estas acciones requieren de esfuerzo humano que pueden ser las siguientes acciones:

- **Revisiones informales:** Reuniones casuales, que se pueden reunir miembros del equipo para validar código.
- **Revisiones formales:** Realizar instancias donde se pueda validar el software cumpla con los requerimientos.

Lo ideal es que en las reuniones de revisiones se involucren varios miembros del equipo de distintos roles, teniendo como objetivo enfocarse en el producto analizando requerimientos, diseños lo que permita reducir los defectos.

En la figura 14, se puede observar el efecto que contiene realizar revisiones previas enfocadas en la planeación, requerimientos y diseño antes de que se inicie la creación del código.

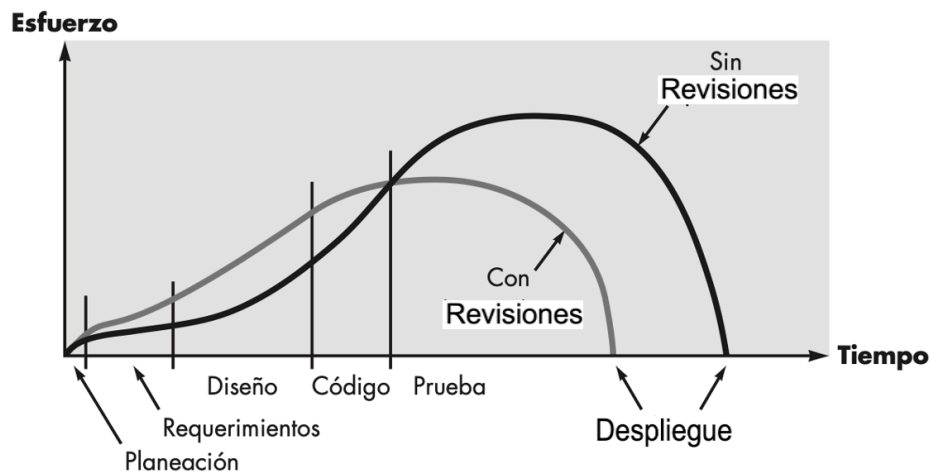


Figura 14: Esfuerzo realizado con o sin revisiones
Fuente: Elaboración propia

2.3.5.2 Aseguramiento de Calidad en el Diseño de Software

El autor de (Pressman, 2010) comenta que se ha demostrado que realizar revisiones en el diseño del software aumenta su calidad, esto sucede porque se crean requerimientos del producto mucho más claros, resultando como consecuencia una eficacia de hasta 75% en disminución de defectos. Al observar en la figura 15 y 16 un ejemplo hipotético, de lo que sucede cuando se realizan pruebas y cuando no.

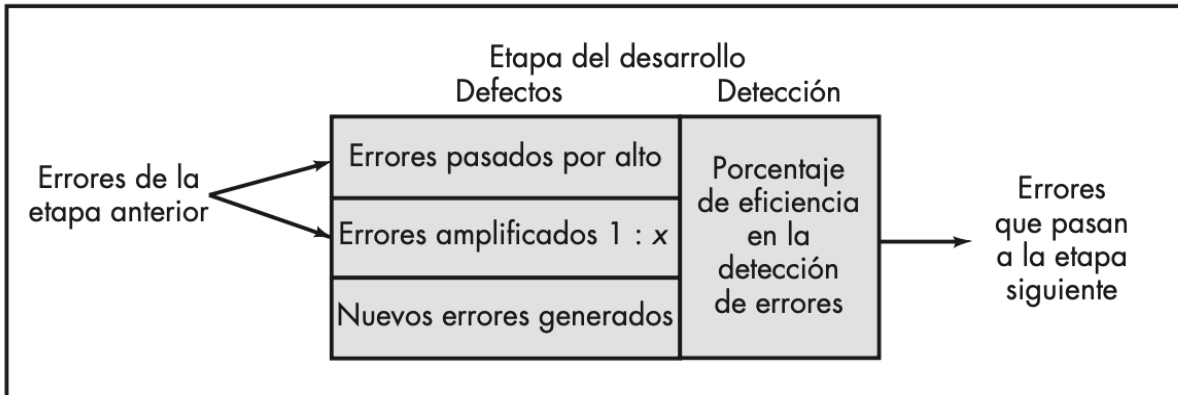


Figura 15: Modelo de amplificación del defecto.
Fuente: Pressman, 2010

En la figura 16, se puede observar que no se efectuaron revisiones en el diseño, resultando como consecuencia que de 10 errores se amplifiquen a 94 errores. Sin embargo, si en el proceso de desarrollo se aplican pruebas al código que validen la calidad y además se verifica que cumplan con los requerimientos de negocio es posible reducir de 94 defectos a 12.

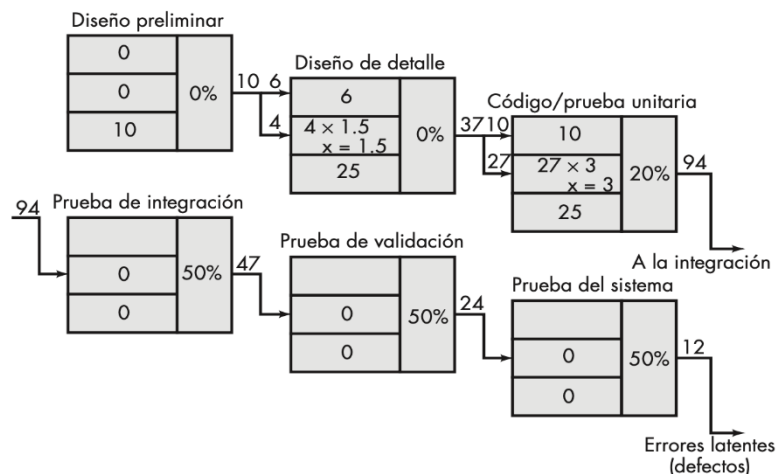


Figura 16: Amplificación del defecto. Sin revisiones
Fuente: Pressman, 2010

Pero en la figura 17, demuestra la importancia del diseño preliminar al momento de desarrollar, ya que, al aplicar correctamente los requerimientos, entendiendo correctamente el objetivo de la funcionalidad facilita el proceso de desarrollo y agregando pruebas de calidad se puede llegar a disminuir significativamente los defectos.

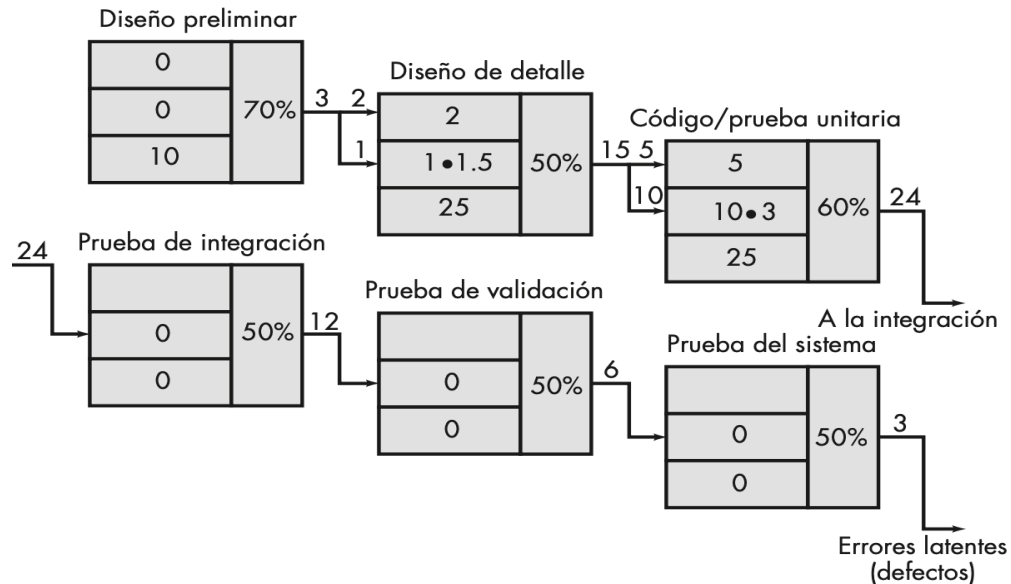


Figura 17: Amplificación del defecto. Con revisiones
Fuente: Pressman, 2010

2.3.6 Área de Aseguramiento de Calidad

Para poder llevar a cabo el Aseguramiento de Calidad es necesario contar con un área que este enfocada en la calidad del software. Para poder incorporar un área de QA en una organización se debe contar con los siguientes principios según la (Qability, 2015):

- **Insertar Aseguramiento de Calidad en el proceso:** Lo primero que se debe realizar es identificar el proceso que necesitará aseguramiento de calidad, por lo que es necesario realizar un análisis en que proceso es vital la presencia del QA.
- **Generar condiciones de pruebas:** Obtener la documentación del producto, ambientes de pruebas, herramientas a utilizar, requerimientos funcionales y no funcionales del producto.
- **Definir proceso de Aseguramiento de Calidad:** Es importante definir las actividades necesarias para poder llevar a cabo las tareas de aseguramiento de

calidad que contendrá este proceso que tendrá como objetivo validar y verificar la calidad del producto.

- **Brindar apoyo al equipo:** En esta etapa es importante generar gestión del cambio, ya que es inevitable no encontrarse con cierta resistencia por parte de los desarrolladores.
- **Formar el equipo de Aseguramiento de Calidad:** Incorporar profesionales que velen por la calidad del producto, generando valor al equipo. Los “QA” son los profesionales encargados de realizar estas funciones, con conocimientos técnicos y de negocio.

2.3.7 Estrategias de Pruebas de Software

Una estrategia de prueba de software provee una guía que define los pasos que deben realizarse como parte de una prueba, cuando deben planearse, que pasos necesita para llevarse a cabo y cuanto esfuerzo de tiempo y recursos se necesitarán.

Tienen como objetivo poder realizar una guía que contempla lo siguiente:

- Cuando se planean y se llevan a cabo las pruebas.
- El esfuerzo que se requerirá.
- El tiempo y recursos que se necesitan para crear y ejecutar las pruebas.

En los que cualquier estrategia de pruebas debe contener:

- La planificación de casos de prueba.
- El diseño de casos de prueba.
- La ejecución de casos de prueba.
- La recolección de los resultados de la ejecución de los casos de prueba.

El autor de (Pressman, 2010) hace referencia que las pruebas tienen dos objetivos principales que son la *Verificación* y *Validación*. En donde la *Verificación* es el conjunto de tareas que garantizan que el software implementa una función correctamente, mientras que la *Validación* se refiere al aseguramiento que el software que se ha construido cumple con los requerimientos de negocio, Por lo que es importante que el “QA” *Valide* y *Verifique* el producto, que se cumplan ambas definiciones.

2.3.8 Perfiles de Aseguradores de Calidad

Para poder llevar a cabo el aseguramiento de calidad es necesario contar con profesionales que tengan los conocimientos en calidad de software llamados Quality

Assurance (QA) o Aseguradores de Calidad, se distinguen dos perfiles que se diferencian de acuerdo a sus conocimientos.

- **QA Funcional o también conocido como QA Manual:** Es el QA que tiene conocimiento en pruebas funcionales, puede analizar requerimientos de negocio y crear a partir de ello casos de prueba que permitan verificar y validar el producto.
- **QA Automatizador:** El profesional Automatizador contiene los conocimientos de un QA Funcional, no obstante lo diferencia sus conocimientos técnicos. Como el manejo de distintos lenguajes de programación y herramientas para automatizar casos de pruebas.

2.3.8.1 Pruebas

Las pruebas son una serie de pasos que permiten verificar funcionalmente un componente o integraciones de componentes que conforman el software. Es el QA quien realiza estas funciones que tiene como objetivo detectar errores y validar los requerimientos del software.

2.3.8.2 Tipos de Pruebas

El autor en (Canelo, 2021) menciona que para poder asegurar la calidad del producto es necesario poder realizar pruebas, de las cuales existen dos categorías de tipos de pruebas que son:

- **Pruebas funcionales:** Verifican que las funcionalidades cumplen con los requisitos de negocio.
- **Pruebas no funcionales:** Son los que consideran el performance del software. Podrían ser pruebas de performance, estrés, seguridad, accesibilidad, etc.

2.3.8.2.1 Pruebas Funcionales

Esta categoría de pruebas tiene un subconjunto de pruebas que son las siguientes:

- **Pruebas Unitarias:** Son las pruebas realizadas por los desarrolladores para probar un fragmento de código.
- **Pruebas de Aceptación (Pruebas Manuales):** Son las pruebas realizadas por los "QA", se enfocan en asegurar que una funcionalidad en desarrollo, cumpla

con los requisitos de negocio. Son las pruebas realizadas por el perfil “QA Manual”.

- **Pruebas de Regresión (Pruebas Automatizadas):** Son las pruebas que son ejecutadas automáticamente mediante alguna herramienta de Automatización como Selenium, Cypress, Katalon Studio, etc. Tienen como objetivo garantizar que el software siga funcionando correctamente al agregar nuevas funcionalidades. Son las pruebas realizadas por el perfil “QA Automatizador”.

2.3.8.3 Casos de Pruebas

En (Prats, 2019) se define un caso de prueba como la condición que se establece para una funcionalidad. Un caso de prueba debe contener el contexto de negocio de la prueba, la acción que se realizara y el resultado esperado que es lo que me permite validar y verificar que cumple con los requisitos de negocio.

Mientras Toledo (Toledo, 2014) menciona que un caso de prueba debe incluir los siguientes valores:

- Paso a paso, el flujo.
- Datos de entrada.
- Estado Inicial.
- Respuesta esperada.
- Estado final esperado.

Lo que define si el resultado es correcto es el *oráculo*. Toledo (Toledo, 2014) define el *oráculo* como la verificación ya sea manual o automático, si el comportamiento del software es el deseado o no. Por lo que se compara el valor esperado con el valor obtenido.

2.3.8.4 Ciclo de Pruebas

La realización de las pruebas, se realizan a través del siguiente ciclo:

- **Creación del plan de casos de prueba:** En esta actividad el QA realiza un análisis de los requerimientos que contienen las Historias de Usuario, lo que permite crear los casos de pruebas necesarios para validar la calidad de la plataforma.
- **Escritura de casos de prueba:** El QA después de realizar el análisis de los requerimientos, inicia la escritura de casos de prueba en un “gestor de casos de prueba” que es donde se registran los casos de pruebas. En algunas empresas que no tienen recursos utilizan archivos Excel.
- **Ejecución de casos de prueba:** Esta actividad se realiza cuando el desarrollador ha terminado la funcionalidad, el QA procede a realizar la validación y verificación de los casos de prueba.

- **Reporte de resultado de casos de pruebas:** El “QA” tiene la responsabilidad de brindar el estado de los casos de pruebas, indicando cuáles de ellos han fallado o funcionan correctamente.

2.3.9 Prácticas de calidad de Software

Para poder obtener un producto con calidad es necesario incluir buenas prácticas de calidad. El autor (Gill, 2022) menciona algunas importantes como:

- **Analizar los requisitos:** Es importante recopilar los requisitos necesarios para realizar el desarrollo incluyendo las características del software, las funcionalidades, la confiabilidad, la usabilidad para validar y verificar si el software cumplirá con los estándares de calidad.
- **Adoptar el principio de pruebas tempranas:** Mediante un desarrollo ágil, es importante incorporar al “QA” en etapas previas del desarrollo, como lo son levantamiento de requerimientos y en refinamientos.
- **Implementar pruebas automatizadas:** Agregar pruebas automáticas que permitan apoyar las pruebas manuales para tener mayor cobertura.
- **Informar errores de manera efectiva:** Al encontrar errores, comunicar al equipo y realizar seguimiento hasta que sea resuelto.
- **Ser un velador constante de la calidad en el producto:** Tratar de realizar distintos tipos de pruebas que ayuden a cubrir el software.

2.3.10 Metodología Shift Left

Es un enfoque que permite mejorar la calidad del producto y reducir los riesgos al descubrir y solucionar defectos en etapas tempranas centrándose en la integración de las actividades de aseguramiento de calidad en etapas previas al desarrollo, a diferencia de enfoques tradicionales donde el “QA” realiza pruebas hasta final del ciclo de desarrollo. Se pretende que el “QA” participe en los requerimientos y diseño del producto. (Palamarchuk, 2021).

Dentro de las actividades que se presentan en una metodología Shift Left son:

- El desarrollador realiza pruebas unitarias y el apoyando para brindar una mejor cobertura.

- Automatizar pruebas en los distintos niveles (pruebas unitarias, de aceptación y/o de regresión).
- Realizar distintos tipos de pruebas no funcionales como performance, seguridad, etc.

2.4 Metodología SCRUM

Es un proceso que contiene un conjunto de buenas prácticas para trabajar en equipo y obtener resultados lo más pronto posible, generando equipos que son altamente productivos, ya que Scrum busca realizar entregas parciales y regulares del software que se está desarrollando. El autor (Lara, 2015) define que Scrum contiene las siguientes fases:

- **El quién y el qué:** Identifica los roles de cada uno de los miembros y la responsabilidad que tienen en el equipo.
- **El dónde y el cuándo:** Que es lo que representa al sprint.
- **El por qué y el cómo:** Son las herramientas que utilizan los miembros de Scrum que lo conforman.

2.4.1 Roles en SCRUM

Los roles que figuran en el equipo son los siguientes:

- **Product Owner/ Business Analyst:** Son los dueños del producto, quienes saben las necesidades del cliente y son los responsables de realizar la planificación además de mantener y priorizar las Historias de Usuario.
- **Scrum Master:** Asegura que el equipo siga la metodología SCRUM, siendo el responsable de remover obstáculos que pueda tener el equipo de los desarrolladores.
- **QA:** Son los encargados de realizar los distintos tipos de pruebas que aseguren la calidad del producto.
- **Desarrolladores:** Son los encargados de convertir los requerimientos de negocio a código de programación que permita crear el software.

2.4.2 El Sprint

Es la característica que hace de Scrum diferente a otras metodologías, es la unidad básica de un equipo Scrum. El Sprint es una iteración que puede ser de 2 o 4 semanas. (Lara, 2015).

2.4.3 Historias de Usuarios

La define (Rehkopf, 2022) como la unidad de trabajo más pequeña en el marco ágil que brinda una explicación general de una función del software escrita desde una perspectiva de usuario final. Fases de Scrum

Para poder implementar la metodología SCRUM, es necesario incorporar las siguientes fases:

- **Planificación del Sprint:** Se realiza una reunión con el equipo donde se deciden que requerimientos o tareas se le asignará a cada uno de los integrantes, asignando el tiempo que se necesitara para poder realizar el desarrollo.
- **Scrum Diario:** Son reuniones diarias en donde sus duraciones no deben tener un máximo de 15 minutos. Cada miembro del equipo deberá responder las siguientes preguntas:
 - ¿Qué hiciste ayer?
 - ¿Qué tienes planeado hacer hoy?
 - ¿Qué obstáculos tienes?
- **Refinamiento:** Los dueños del producto revisan las Historias de Usuarios con el equipo, para estimar el tiempo y esfuerzo.
- **Revisión del Sprint:** También conocido como “Demo”, se presenta el trabajo realizado durante el sprint.
- **Retrospectiva del Sprint:** En esta instancia se centra en buscar puntos de mejora y exaltar lo que se hizo bien durante el sprint.

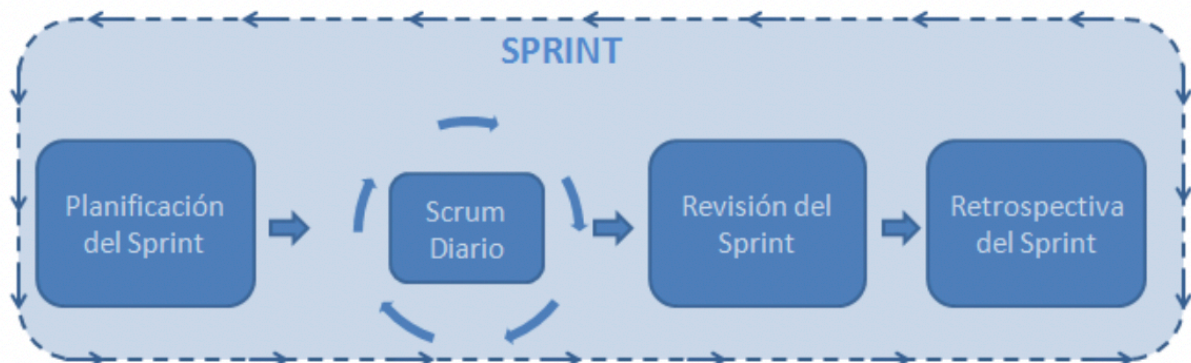


Figura 18: Ciclo del sprint
Fuente: OpenWebinars, Requena

2.5 Modelos de Calidad de software

Los modelos enfocados en la calidad de software se centran en evaluar el proceso del desarrollo del software en cada una de sus etapas. Según (Scalone, 2006) los modelos de calidad integran en mayor parte las mejores prácticas, proponiendo temas de administración en los que cada organización debe enfocarse, además de que integran diferentes prácticas dirigidas a los procesos clave logrando así poder medir los avances en calidad.

El modelo de calidad se debe centrar en poder evaluar el sistema ya sea de forma cuantitativa y cualitativa para así poder medir y proponer mejoras en el proceso en cada una de las etapas de desarrollo del software.

El rol que juegan las métricas como menciona (Mera, 2016) es que permiten estimar el esfuerzo, los tiempos, determinar los incidentes de alto riesgo y poder optimizar así los recursos y el talento del equipo de desarrollo.

Por lo que es importante definir las incidencias como la interrupción no planificada de un servicio o una reducción de la calidad de un servicio de TI (Silva, 2021).

Existen distintos modelos de calidad de procesos entre ellos:

- **ITIL:** Se desarrolló en Reino Unido teniendo como fin fortalecer la gestión gubernamental partiendo de cinco elementos: La perspectiva del negocio, entrega del servicio, soporte del servicio, manejo de infraestructura y manejo de aplicaciones buscando ofrecer un servicio completo. (Callejas, Alarcón & Álvarez, 2017)

- **ISO:** Adaptable para procesos tanto en empresas pequeñas y medianas mediante una estructura de seis niveles: Nivel 0 siendo una organización inmadura, Nivel 1 una organización básica, Nivel 2 una Organización gestionada, Nivel 3 Organización establecida, Nivel 4 Organización predecible y Nivel 5 siendo una organización optimizada. Tiene como objetivo lograr que las organizaciones sean maduras con procesos definidos, en donde se tienen definidas las responsabilidades y se puede llegar a predecir resultados que incrementen la productividad. (Callejas, Alarcón & Álvarez, 2017)
- **COBIT:** Permite que se pueda orientar a negocios y empresas, brindando controles a través de criterios que serán detallados más adelante a profundidad. (Callejas, Alarcón & Álvarez, 2017)
- **Bootstrap:** Es una metodología que permite evaluar la mejora de procesos a partir de seis actividades que son básicas: Examinar la necesidad, iniciar el proceso de mejora, preparación y dirección de la evaluación, análisis de los resultados, implantación y finalización de mejoras. (Callejas, Alarcón & Álvarez, 2017)
- **CMMI:** Este es un modelo que permite la mejora de procesos en las empresas para procesos de desarrollo y el mantenimiento de software. Tiene como propósito verificar que se cumplan con los estándares de calidad a través de la medición de los niveles de madurez. (Lopez, Cabrera & Valencia, 2008).

2.6 COBIT

2.6.1 Origen COBIT

Fue en 1996, la primera edición de COBIT que fue publicada, elaborada por equipos que se encontraban en Europa, Estados Unidos y Australia.

En 1998, se publica su segunda edición que se diferencia de la primera al agregar la “Gestión” y en 2003 se publica por primera vez la versión en línea en el sitio web de ISACA. Es en año 2007 cuando se libera la versión 4.1 de COBIT que contiene un mayor desarrollo del Gobierno TI.

Concepto de COBIT

El marco de referencia de COBIT se conforma por un conjunto de mejores prácticas que permite evaluar la seguridad, eficacia, calidad y eficiencia de los procesos y actividades de TI, así como de las áreas que lo componen para determinar aspectos como los riesgos, gestión efectiva de los recursos, medir el desempeño, cumplimiento de metas y de manera principal medir el nivel de madurez de los procesos de una orga-

nización. Teniendo como finalidad satisfacer las necesidades relacionadas con los requerimientos del negocio y manejo de la inversión en recursos de TI (Institute IT Governance, 2007).

2.6.2 COBIT 4.1

COBIT está compuesto de tres niveles de actividades de TI que permiten administrar sus recursos:

- **Actividades:** Son las acciones que se requieren para obtener un resultado que pueda ser medible.
- **Procesos:** Se le denomina al conjunto de actividades.
- **Dominios:** Se le denomina a la agrupación de los procesos que son responsabilidad de la organización.

Es importante mencionar que las metas que la conforman deben estar alineadas con los objetivos del negocio logrando así satisfacer los requerimientos. COBIT está conformado por 4 dominios que a su vez lo conforman 34 procesos y que a su vez están conformados por 218 actividades (IT Governance Institute, 2007).

COBIT se caracteriza por tener foco en la parte de negocios y proceso, basándose en controles y trabaja con siete criterios de información que son definidos como efectividad, confidencialidad, confiabilidad, efectividad, disponibilidad, cumplimiento e integridad (Guerrero, 2006).

COBIT posee la misión de investigar, desarrollar, publicar y promover un conjunto de objetivos de control generalmente aceptados, autorizados, actualizados por ISACA para ser utilizados en el día a día por la gerencia del negocio, los profesionales de IT y de la seguridad. (IT Governance Institute, 2006).

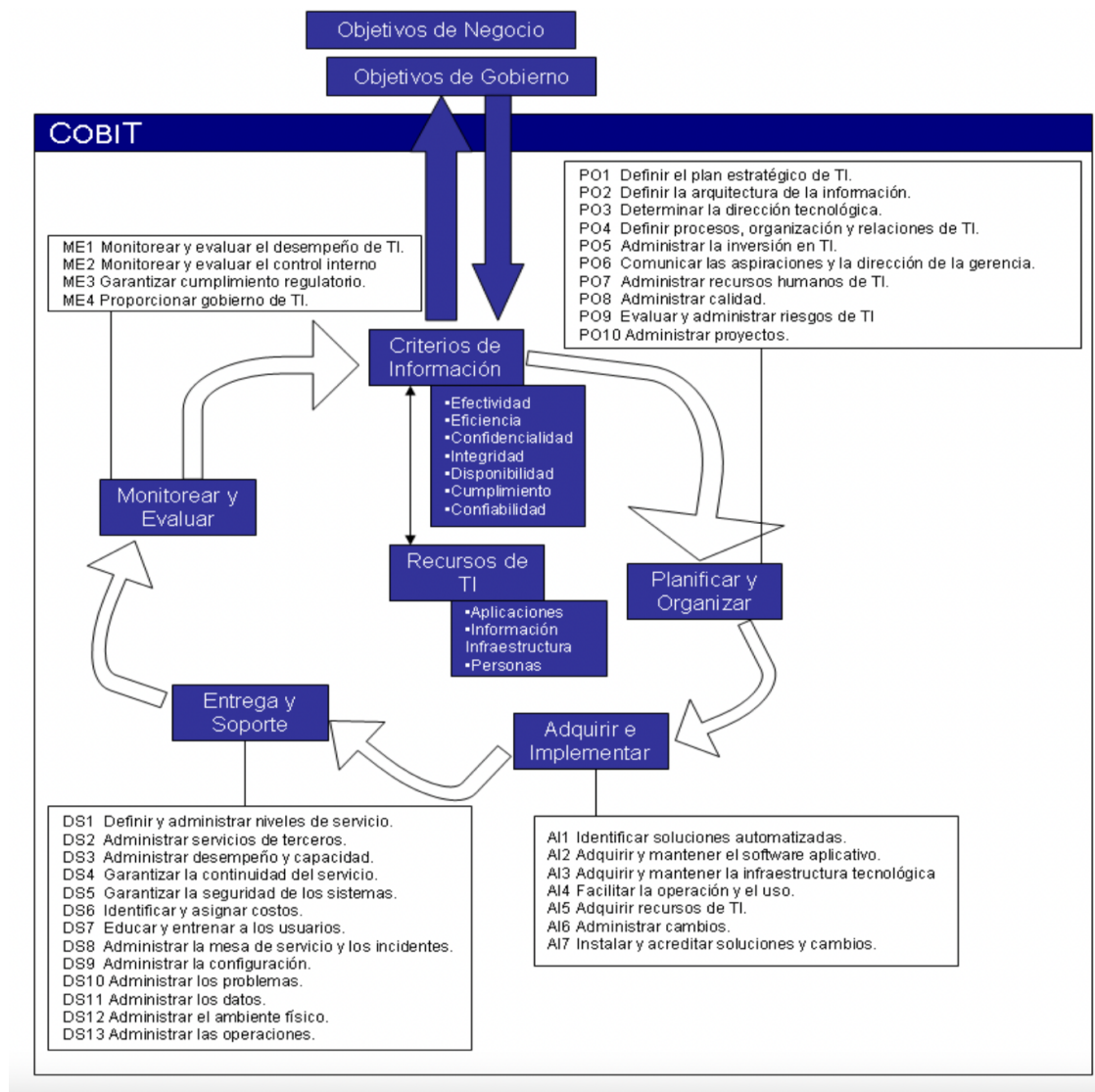


Figura 19: Marco de Trabajo completo COBIT 4.1
Fuente: IT, Governance Institute

2.6.3 Proceso Administración de la calidad

Es el Proceso 08 del dominio “*Planificar y Organizar*”, (IT Governance Institute, 2006) lo describe como el proceso que satisface la mejora continua y medible de la calidad. Se enfoca en definir un sistema de administración de calidad (QMS, por sus siglas en inglés), lográndolo a través de la definición de estándares y prácticas de calidad.

2.6.3.1 Entradas y Salidas del Proceso Administración de la calidad

Este proceso tiene las siguientes entradas:

- **Plan estratégico de TI:** Contiene toda la estrategia y prioridades del negocio, que permita validar todas las oportunidades de mejoras y que planes de acción deben realizarse.
- **Planes detallados de proyectos.** Contiene el plan de aseguramiento de calidad, el plan formal de pruebas, el resultado de las revisiones y el valor de la calidad de los entregables.
- **Planes de acciones correctivas:** Es necesario el proceso de monitoreo que garantice el software cumple con los requisitos y si no es así llevar a cabo las medidas correctivas.

Este proceso tiene las siguientes salidas:

- **Requerimientos de estándares y métricas de calidad:** Presenta resultados de lo que fue la revisión de aseguramiento de calidad del software brindando toda la información necesaria para que se pueda medir la calidad.
- **Medidas para la mejora de calidad:** Permite definir si los procesos brindan los requerimientos de negocio apoyándose de aseguramiento de calidad. Agregando procesos que revisen y garanticen no puedan suceder percances que impacten negativamente al software.

2.6.3.2 Actividades del Proceso Administración de la calidad

El proceso consta de las siguientes actividades:

- **P08.1 Definir un Sistema de Administración de Calidad:**
 - Definir la estructura organizacional para la administración de calidad, cubriendo roles, tareas y responsabilidades.
 - Todas las áreas clave desarrollan sus planes de calidad de acuerdo a los criterios y registran los datos de calidad.

- **P08.2 Establecer y Mantener Estándares y Prácticas de Calidad:**
 - Identificar y mantener estándares, procedimientos y prácticas para los procesos clave de TI para orientar a la organización hacia el cumplimiento del QMS.
 - Usar las buenas prácticas de la industria como referencia al mejorar y adaptar las prácticas de calidad de la organización.

- **P08.3 Crear y Comunicar Estándares de Desarrollo y de Adquisición:**
 - Adoptar y mantener estándares para todo desarrollo y adquisición que siga el ciclo de vida, hasta el último entregable e incluir la aprobación en puntos clave con base en criterios de aceptación acordados.
 - Validación contra requerimientos; planes de pruebas; y pruebas de regresión y de integración.

- **P08.4 Crear y administrar plan de calidad para la mejora continua:**
 - Enfocar la administración de calidad en los clientes, determinando sus requerimientos y alineándose con los estándares y prácticas de TI.

- **P08.5 Medición, Monitoreo, Revisión de las metas y métricas de la calidad:**
 - Definir, planear e implementar mediciones para monitorear el cumplimiento continuo del QMS, así como el valor que el QMS proporciona.

CAPÍTULO 3: PLANTEAMIENTO ESTRATÉGICO Y ANÁLISIS DE LA SITUACIÓN ACTUAL

3.1 Visión y Misión de Buk

3.1.1 La Visión

La motivación e inspiración de Buk a futuro es “Ser la mejor plataforma de gestión de personas de Latinoamérica y hacer a nuestros clientes más felices” (Buk, 2022).

3.1.2 La Misión

Dentro de su misión tiene como propósito “Crear lugares de trabajo más felices” (Buk, 2022).

3.2 Posicionamiento Estratégico

Buk se encuentra en el modelo de Hax dentro del posicionamiento “Diferenciación” como se puede observar en la figura 20, siendo esto conveniente ya que Buk tiene como objetivo brindar un producto que se diferencia del resto del demás software de recursos humanos de la competencia. Buk se mantiene enfocado en la *innovación*, por lo que está en constante observación de las necesidades del rubro de RRHH, incorporando características o funcionalidades que hagan de Buk una plataforma que ofrece nuevas e innovadoras funcionalidades para el área de RRHH de las empresas. Además de hacer integraciones con otros softwares, que permiten que la plataforma de Buk sea valorado por el cliente, como una plataforma en la que puedan realizar todos sus procesos de manera automatizada brindando así a los clientes una excelente solución para acelerar sus procesos organizacionales.

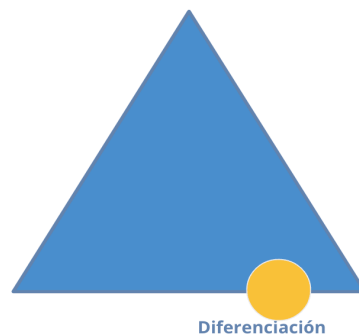


Figura 20: Posicionamiento estratégico de Buk.

Fuente: Elaboración propia

3.3 Objetivos estratégicos

Dentro de la organización de Buk se encuentran los siguientes objetivos estratégicos que componen su balanced scorecard:



Figura 21: Balance Scorecard de Buk.

Fuente: Elaboración propia

- **Financiera:** Tiene como foco consolidar su participación en el mercado latinoamericano.
- **Clientes:** Buk tiene en el centro a sus clientes por lo tanto no pierde su misión de tener clientes más felices.
- **Procesos internos:** La importancia de entregar una plataforma que brinde la mejor experiencia en sus clientes da como resultado estar en constante mejora del ciclo de entrega siendo pertinente la eficiencia en sus procesos internos.
- **Aprendizaje y crecimiento:** Para poder convertirse en la mejor plataforma de Recursos Humanos en Latinoamérica es pertinente contar con la mejor calidad de empleados por lo que es importante la apertura del mejor talento en el mundo y en brindar constante capacitación logrando así tener las personas adecuadas en cada puesto.

3.4 Modelo de Negocios

3.4.1 Propuesta de valor

Buk, plataforma de gestión integral de RRHH, busca entregar todas las herramientas necesarias para RRHH en un mismo lugar, para crear un lugar de trabajo más feliz.

Posee tres pilares fundamentales como se puede ver en la figura 22:



Figura 22: Pilares de Buk.
Fuente: Buk

3.4.2 Segmento de clientes

Actualmente Buk cuenta con una cartera de más de 4,000 clientes de diferentes rubros como lo son el financiero, construcción, salud, retail entre otros como se puede observar en la figura 23.



Figura 23: Segmento de clientes de Buk.
Fuente: Buk

3.4.3 Relaciones con el cliente

Buk busca fidelizarse con sus clientes, apoyando en toda gestión de procesos de RRHH. Buk se centra en el cliente por lo que buscan apoyarlos en todo momento desde que tienen la primera interacción con la plataforma hasta apoyarlos ante cualquier consulta y/o incidencia que les suceda al hacer uso de la plataforma.

3.4.4 Canales

Buk cuenta con un área de marketing que se encarga de dar a conocer el producto mediante publicidad web, televisiva, rótulos en zonas importantes de la ciudad, blog, podcast entre otros que permiten descubrir no solo la plataforma, sino experiencias de clientes además de entender la cultura de la empresa. Sin embargo, no se descarta el valor que se genera en la adquisición de clientes por el boca a boca, donde son los mismos clientes recomiendan a otros potenciales la plataforma, por lo cual es importante brindar la mejor experiencia al usar el software.



Figura 24: Rótulos de Buk.
Fuente: Buk

3.4.5 Recursos claves

- **Empleados administrativos:** Son los encargados de velar por que la gestión de la empresa funcione correctamente, responsabilizándose de lo que son los pagos a los colaboradores, gestiones administrativas, etc.

- **Empleados técnicos:** Son de vital importancia ya que son los encargados de velar por crear y entregar el desarrollo de la plataforma. Está conformado por el área de producto en donde laboran los perfiles como Ingenieros comerciales, informáticos y de telecomunicaciones.
- **Infraestructura y Seguridad de la plataforma:** Al contar con datos altamente sensibles de las empresas como información personal, cuentas de bancos y la cantidad de los ingresos de los colaboradores es necesario brindar la mejor seguridad por lo que Buk tiene el área de SecOps que tiene como objetivo el velar por la seguridad tanto interna como externa, previniendo cualquier vulnerabilidad dentro de la organización. Además de obtener las certificaciones en seguridad correspondiente como la ISO 27001 entre otras, que brinden la mayor confianza a los clientes al usar la plataforma.

3.4.6 Socios claves

Para poder fortalecer la plataforma y brindar una respuesta a todas las necesidades que los clientes puedan tener, Buk adquirió 4 empresas que tienen apps que han sido integradas a la plataforma. Entre ellas están el software de *Asistencia* que permite tener un control de la asistencia de los colaboradores de las empresas, administrando el tiempo en los turnos, permisos, vacaciones, horas extras etc. La app de *Capacitaciones* que brinda acceso a un catálogo de cursos de capacitación laboral. La plataforma de *Ninja* que ayuda a la formación online de Excel y la app de *Adelantos* que facilita a los colaboradores de las empresas poder obtener adelantos de sus sueldos de manera rápida y sencilla. Logrando así que la Plataforma de Buk al integrarse con estos softwares sea la más completa en Recursos Humanos en Latinoamérica.

3.4.7 Fuentes de ingresos

Actualmente Buk tiene un modelo de ingresos, en donde los clientes pagan de acuerdo a un modelo de suscripciones en donde pagan de acuerdo a la cantidad de colaboradores que tienen en sus empresas.

3.4.8 Estructura de costes

- **Sueldos de personal:** La empresa posee varias áreas en donde cada una cuenta con un encargado del área y demás colaboradores a los que se les debe remunerar su trabajo.
- **Gastos operacionales:** Incluye los gastos del edificio, así como de gastos básicos (agua, luz, internet, etc.), gastos en software y hardware.
- **Gastos publicidad:** Incluye los gastos que se pagan por publicidad televisiva, rótulos, en redes sociales, etc.

Modelo Canvas de Buk



Figura 25: Modelo de Canvas de Buk.
Fuente: Elaboración propia

3.5 Diagnóstico de la Situación Actual

3.5.1 Problemas Identificados

Para poder comprender la situación actual de la plataforma fue necesario reunirse con las personas que son los responsables en desarrollarla, también con el equipo de servicio de atención al cliente quienes son los que perciben los dolores de estos al hacer uso de la plataforma y por el equipo de negocio quienes son los que se responsabilizan de crear y adelantarse a las necesidades de los clientes. Toda información que se levantara sería de gran importancia para poder realizar un análisis profundo del estado de la plataforma. Para poder llevarlo a cabo se realizó en distintas sesiones de las cuales se pudo obtener información y generar los problemas que cada equipo tenía en su radar.

- **Sesión con el equipo de Tech Leads:** El perfil del Tech Lead es el encargado de velar porque se cumpla con el desarrollo que se planifica en un sprint, se enfoca en ser el apoyo de los desarrolladores ante cualquier duda técnica. Buk se encuentra dividido por células en donde cada una está encargada del desarrollo de algún modulo o funcionalidad de la plataforma. Para esta instancia se contó con la presencia de todos los Tech Leads que conforman el área de producto. La reunión fue enfocada en los dolores que desde su punto de vista se percibían hacía la plataforma de los cuales se mencionaron entre ellos la deficiencia en los controles de calidad al realizar el desarrollo, quienes son los mismos desarrolladores los que hacen sus pruebas en su código y en la mayoría de los casos no alcanzan en tiempo de agregar pruebas unitarias que permitan validar la correcta funcionalidad de lo desarrollado y al existir en Buk despliegues concurrentes, realizándose 2 despliegues semanales sin un exhaustivo aseguramiento de calidad, permite la fuga de defectos en la plataforma.



Figura 26: Problemas levantados por Tech Leads.
Fuente: Elaboración propia

- **Sesión con el equipo de Servicio de Atención al Cliente (SAC):** Es el equipo de SAC el encargado de ser quienes tienen ese primer contacto con el cliente desde consultas de cómo usar la plataforma hasta recibir defectos que son expuestos ante los clientes. Por consiguiente reciben de buena fuente los dolores de la plataforma, lo que permitió poder levantar los siguientes puntos a considerar ya que a medida que la plataforma va creciendo en cantidad de colaboradores que tienen las empresas está más expuesta a que sean encontrados mayor cantidad de defectos, presentándose en los últimos meses del año 2021, una alza de defectos reportados por clientes lo que da como resultado a clientes molestos ya que la solución de los defectos al ser muchos no permite al súper héroe designado poder solucionarlo de manera inmediata.



*Figura 27: Problemas levantados por SAC.
Fuente: Elaboración propia*

- **Sesión con el equipo de Negocio:** Conformado por los Product Owner y los Business Analyst quienes son los dueños del producto, se encargan de analizar que nuevas funcionalidades deben crearse para la plataforma y permitir adelantarse a la necesidad del cliente. Su punto de vista era primordial ya que poseen vasto conocimiento de la plataforma, de lo cual permitió poder obtener los siguientes dolores, comunicando que es el equipo de negocio quien en algunas ocasiones es el encargado de verificar los nuevos desarrollos sin embargo mencionan que por cuestión de tiempo no logran probar de manera exhaustiva y que en muchas ocasiones solamente confían en que el desarrollador lo hizo correctamente. En la instancia levantan que se han percatado que las redacciones de las Historias de Usuario (HU) no están descritas con toda la información oportuna, no contemplan esos casos bordes

en los que podría fallar la plataforma, mencionan que también al tener falta de información tanto de negocio como técnica da apertura a que las estimaciones sean incorrectas y por lo tanto no se pueda finalizar con todas las HU planificadas para el sprint atrasando así la entrega de nuevas funcionalidades a los clientes y demoras en las soluciones de los defectos reportados por SAC.

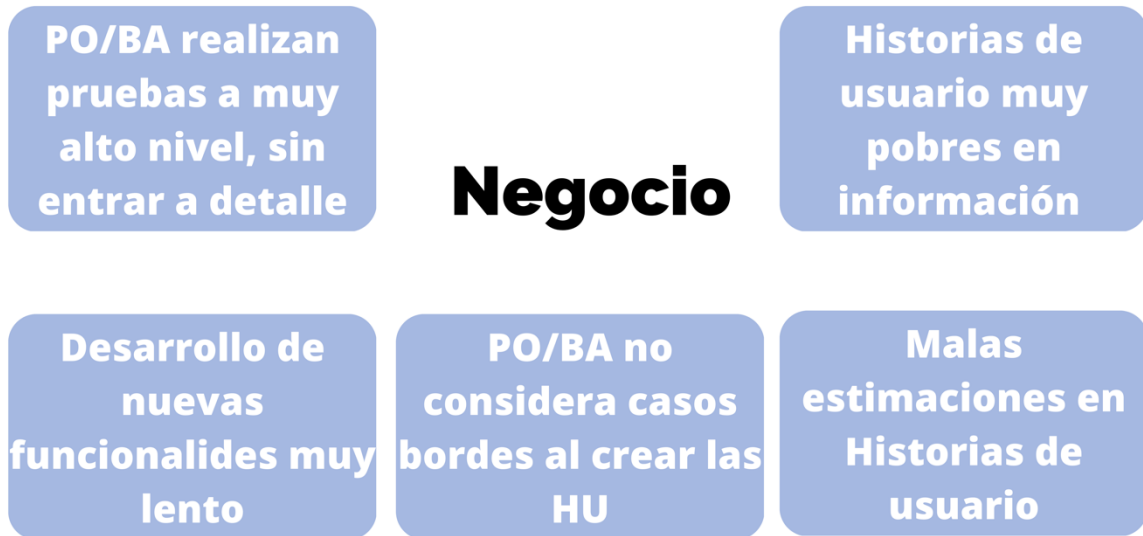


Figura 28: Problemas levantados por Negocio.

Fuente: Elaboración propia

Al realizar el levantamiento de problemas en las instancias con los equipos, se inició agrupando lo recabado para encontrar problemas en común y como se enlazan entre ellos. Se agrupó en problemas que afectarán la usabilidad de la plataforma, problemas orientados a negocio, problemas por lentitud en desarrollo y problemas de comunicación entre áreas.

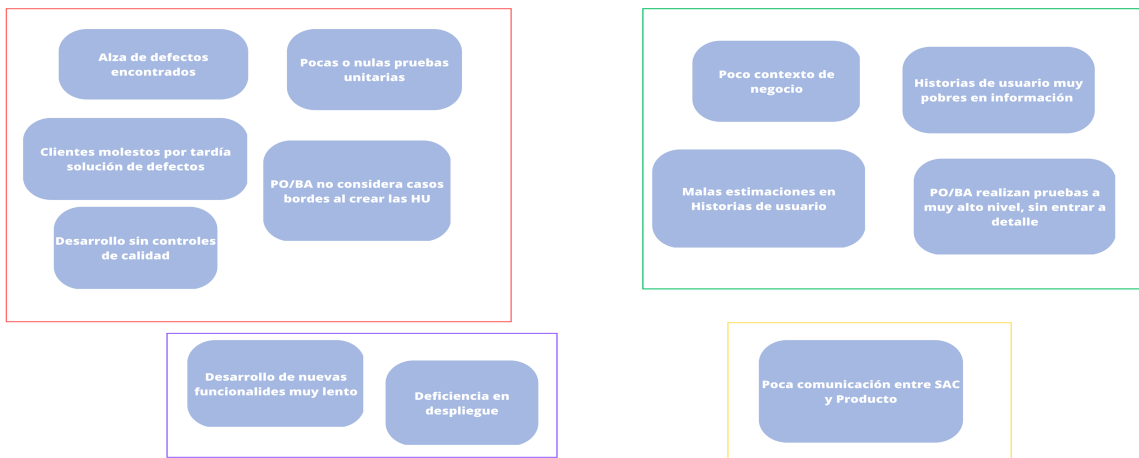


Figura 29: Agrupación de problemas.

Fuente: Elaboración propia

Al finalizar las agrupaciones, se llevó a cabo una reunión con el CTO y Head de Producto presentando los problemas levantados por los equipos. Fue importante que en la conversación se discutiera el problema que para los stakeholder fuera relevante llegando a la conclusión que al ser Buk una empresa que tiene como centro al cliente, es importante escuchar su dolor. De lo levantado por los equipos menciona el equipo de SAC que el cliente reporta sintiendo molestia los defectos y su tardía solución. Debido a lo cual era oportuno tomar el problema del alza de defectos para ahora partir con la pregunta, *¿Cómo disminuimos el alza de defectos reportados por los clientes?*

Ahora, partiendo del problema seleccionado se llevó a cabo el ejercicio del “Árbol How” con algunos representantes de los equipos del área de producto. Esto permitió poder encontrar oportunidades para solucionar el problema y lograr desarrollar las siguientes hipótesis que se observan en la figura 30.

Las hipótesis fueron ponderadas bajo dos criterios como se observa en la tabla 4, el impacto que tiene en negocio, así como el esfuerzo de llevarlo a cabo. Se pondero con un rango de 1 a 10, participando representantes de los equipos, los números que se muestran en la tabla 3 son el promedio del número asignado entre todos los participantes en cada hipótesis. Para evaluar estas dos variables que se midieron se sacó el promedio de las 4 hipótesis, resultando ser potenciales a desarrollar debido a que también puede trabajarse una solución conjunta tal que si se trabaja la H1 y la H2 impactan la H3 con el apoyo de la H4 como muestra la figura 31. Logrando así solventar el problema presentado por los clientes y que para la empresa Buk es importante considerar ya que se juega la baja calidad del producto, mala reputación que se pudiese dar por malas experiencias de los clientes y poca innovación, lo cual sería un gran impedimento en cumplir con su misión que es ser la plataforma de Recursos Humanos más importante en Latinoamérica.

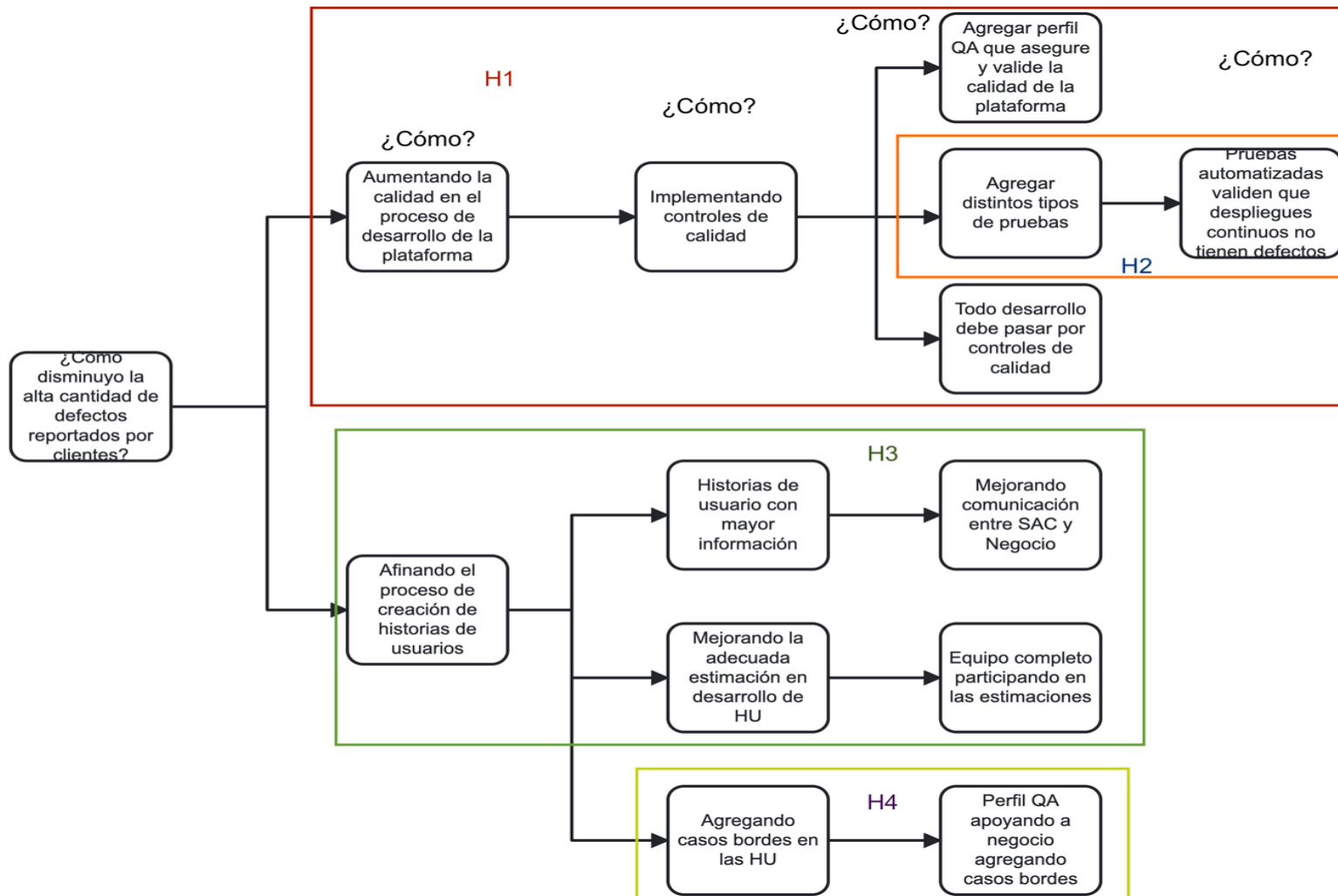


Figura 30: Árbol How de problema en Buk
Fuente: Elaboración propia

Tabla 4: Tabla de hipótesis de cómo solucionar el problema.
Fuente: Elaboración propia

Hipótesis	Impacto en el negocio	Es-fuerzo	Pro-me-dio
H1: Si se rediseña el proceso de creación del producto aplicando controles de calidad disminuirán los defectos.	10	9	9.5
H2: Si se realizan distintos tipos de pruebas al desarrollo se encontrarán defectos en etapa temprana de desarrollo.	10	9	9.5
H3: Si se realiza una mejor escritura en las historias de usuario se podrá prevenir errores en etapa de desarrollo y mala estimación de las historias de usuario.	10	8	9
H4: Añadiendo un nuevo perfil QA en las células de trabajo permitirá aplicar controles de calidad y apoyar a negocio en la escritura de casos bordes en las historias de usuario.	10	8	9

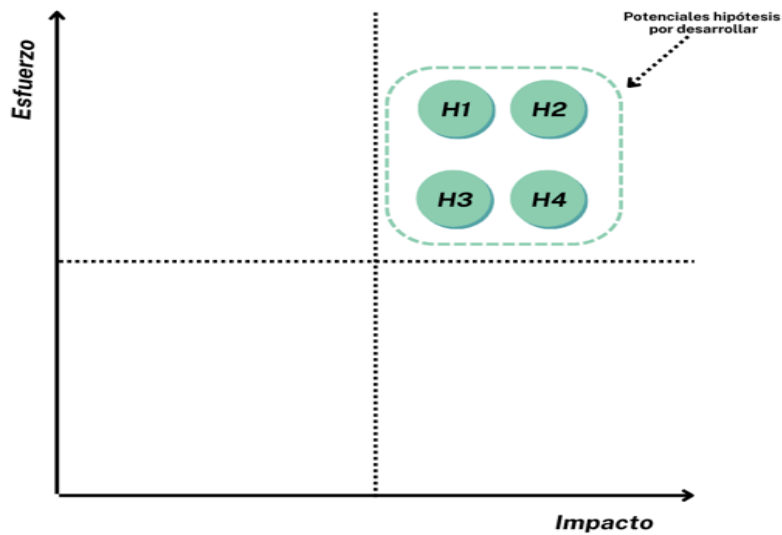


Figura 31: Gráfico Esfuerzo vs Impacto.

3.5.2 Arquitectura de Procesos AS IS

3.5.2.1 Macroprocesos de Buk

Los patrones de arquitectura y de procesos de negocio de Barros, pueden ser adaptados a cualquier dominio con el fin de que puedan ser implementados en cualquier organización, que para este proyecto será para la empresa Buk.

- **Línea de Desarrollo de Buk** Es el Macroproceso que representa la cadena integral de valor de una empresa, de donde provienen los procesos encargados de gestionar los requerimientos de los clientes, de planificar y ejecutar la producción además de realizar la entrega del producto o servicio. Para Buk esta Macro 1 tiene la misión de concretar el producto al que llamaremos *plataforma*. Es en esta Macro donde se acogen los requerimientos de los clientes que pueden corresponder desde nuevas funcionalidades hasta mejoras de las funcionalidades existentes de la plataforma, también se atienden requerimientos de Instituciones como PREVIRED, AFP entre otras para que se cumplan con los estándares y la información del mercado respecto a nuevas tecnologías que podría ser implementadas en la plataforma. Como control atiende los planes referentes a la organización brindada por la Macro 3 que buscan orientarse a las necesidades de la empresa de acuerdo a lo planificado y el desarrollo de las nuevas capacidades de la Macro 2, que se orienta de la innovación. Esta macro es donde se establece los planes del desarrollo de la plataforma, mediante metodología Scrum que tiene como objetivo realizar entregas continuas en periodos también llamados “sprint” de dos semanas, entregando por lo tanto de manera constante nuevas funcionalidades y/o la solución de defectos que habían reportado los clientes.

Resulta inevitable no relacionar el problema concerniente de la alta cantidad de defectos en la plataforma con esta Macro, puesto que es la que lleva a cabo el proceso de ideación y desarrollo de la plataforma. Así que será concerniente realizar un desglose de esta Macro para poder conocer y realizar el análisis como Buk lleva a cabo el desarrollo de la plataforma y como se pretenderá solucionar el problema mediante el rediseño del proceso.

- **Desarrollo de nuevos tipos de soluciones Buk:** Este Macroproceso se encarga de desarrollar las nuevas capacidades que la empresa requiere para ser competitiva, como nuevas apps a integrar en la plataforma, nuevas funcionalidades a desarrollar, en este Macro se lleva a cabo la innovación. Buk se encarga de realizar estudios referentes al mercado en conocer la competencia, las necesidades de los clientes que permitan hacer de la plataforma Buk el software más completo en Recursos Humanos.
- **Planificación de Negocio Buk** Enfocado en la planificación del negocio para definir el futuro de la empresa a manera de estrategias que son llevado a cabo

mediante planes. Buk plantea sus estrategias en esta Macro, definiendo sus OKR que son presentados trimestralmente a toda la organización para conocer el avance de la empresa tanto interno como externamente.

- **Gestión de recursos habilitadores:** Son los procesos que funcionan como apoyo como recursos financieros, humanos, infraestructura y materiales. Buk en esta Macro brinda todos los recursos necesarios a la Macro 1 para que pueda desarrollar la plataforma.

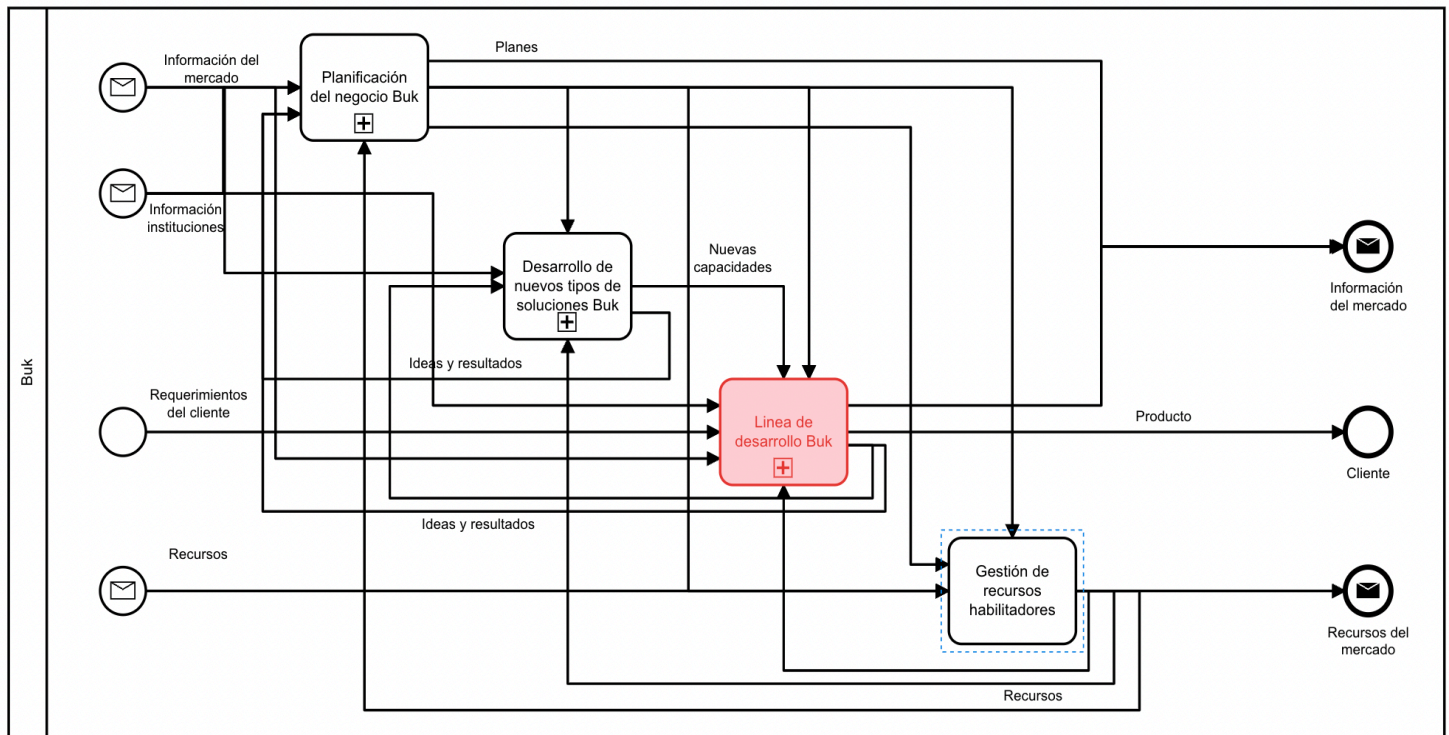


Figura 32: Macroprocesos Buk, Situación actual.

Fuente: Elaboración propia

3.5.2.2 Macro 1.3 Desarrollo y entrega de plataforma Buk

Su foco principal es llevar a cabo el desarrollo y entrega de la plataforma. Y para poder efectuarlo necesita recibir los requerimientos que han sido anteriormente gestionados en el subproceso de “Administración relación con el cliente” que contienen los requerimientos de clientes, instituciones y mercado permitiendo poder llevar a término las necesidades levantadas. Además, acá se acogen las incidencias que son reportadas a SAC, estos son los defectos que contiene la plataforma y deben ser solucionados por los desarrolladores que conforman este subproceso. Dentro de recursos proveedores, son los que se encargan de ofrecer el servicio en la nube a los clientes al ser una plataforma SaaS.

Como control recibe las planificaciones efectuadas por la organización del Macro 3, que tiene la finalidad de alinear los objetivos estratégicos de la empresa con las nuevas funcionalidades desarrolladas y las nuevas capacidades que son las innovaciones a incorporar en la plataforma para que siga la empresa posicionándose en “Diferenciación” como actualmente lo es, debido a las distintas funcionalidades que tiene a diferencia de la competencia como las integraciones con otros softwares. También recibe mediante el subproceso “Gestión, desarrollo y entrega de la plataforma Buk” los planes de desarrollo que se ejecutarán en los periodos llamados “Sprint”. Además, recibe los recursos que son necesarios para poder realizar el desarrollo, desde la gestión de personas como gestión económica. La mantención de estado le brinda el análisis de los requerimientos y si estos están siendo cumplidos asimismo informar a otros subprocesos del estado del desarrollo de la plataforma.

De salida este subproceso realiza la entrega de la plataforma con sus nuevas funcionalidades que fueron levantados como requerimientos, además de los desarrollos que son las soluciones de los defectos levantados por los clientes.

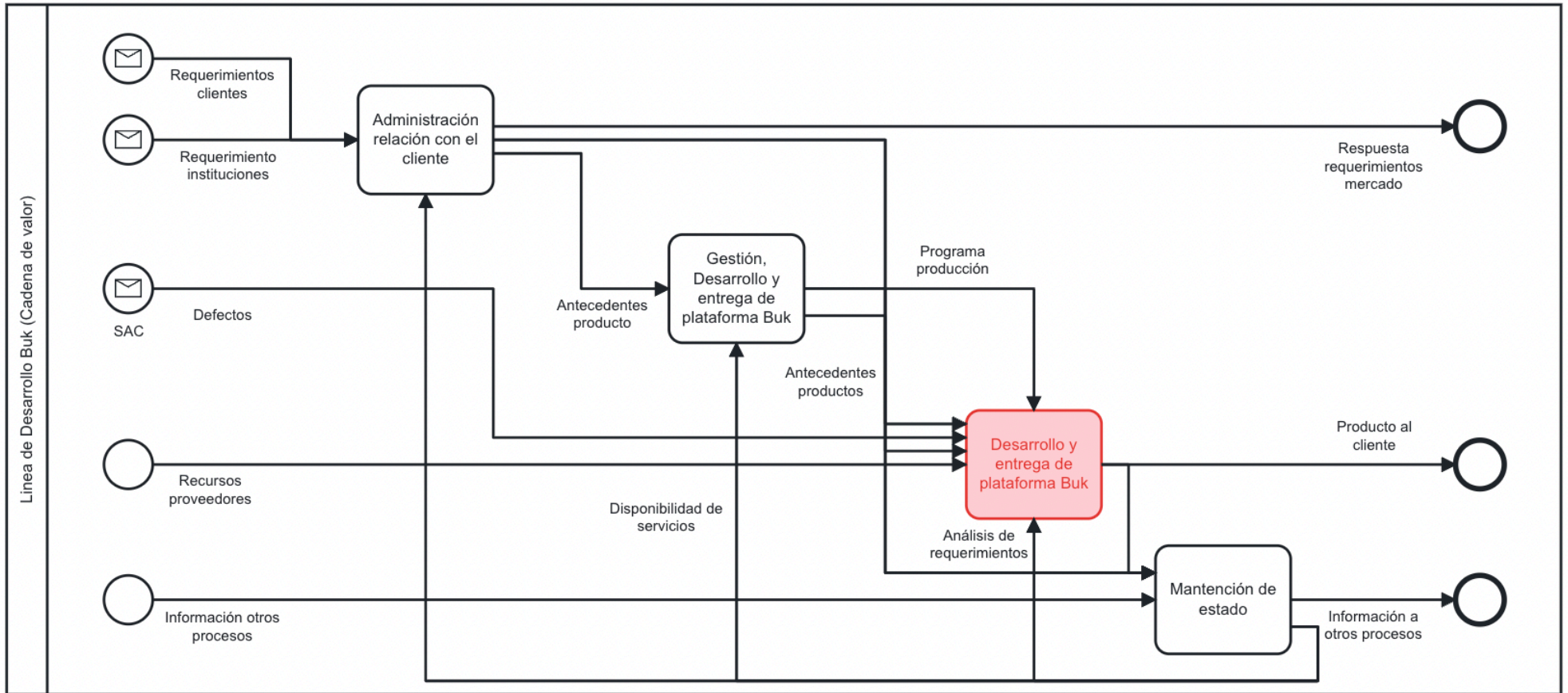


Figura 33: Macro 1 Línea de Desarrollo Buk.
Fuente: Elaboración propia

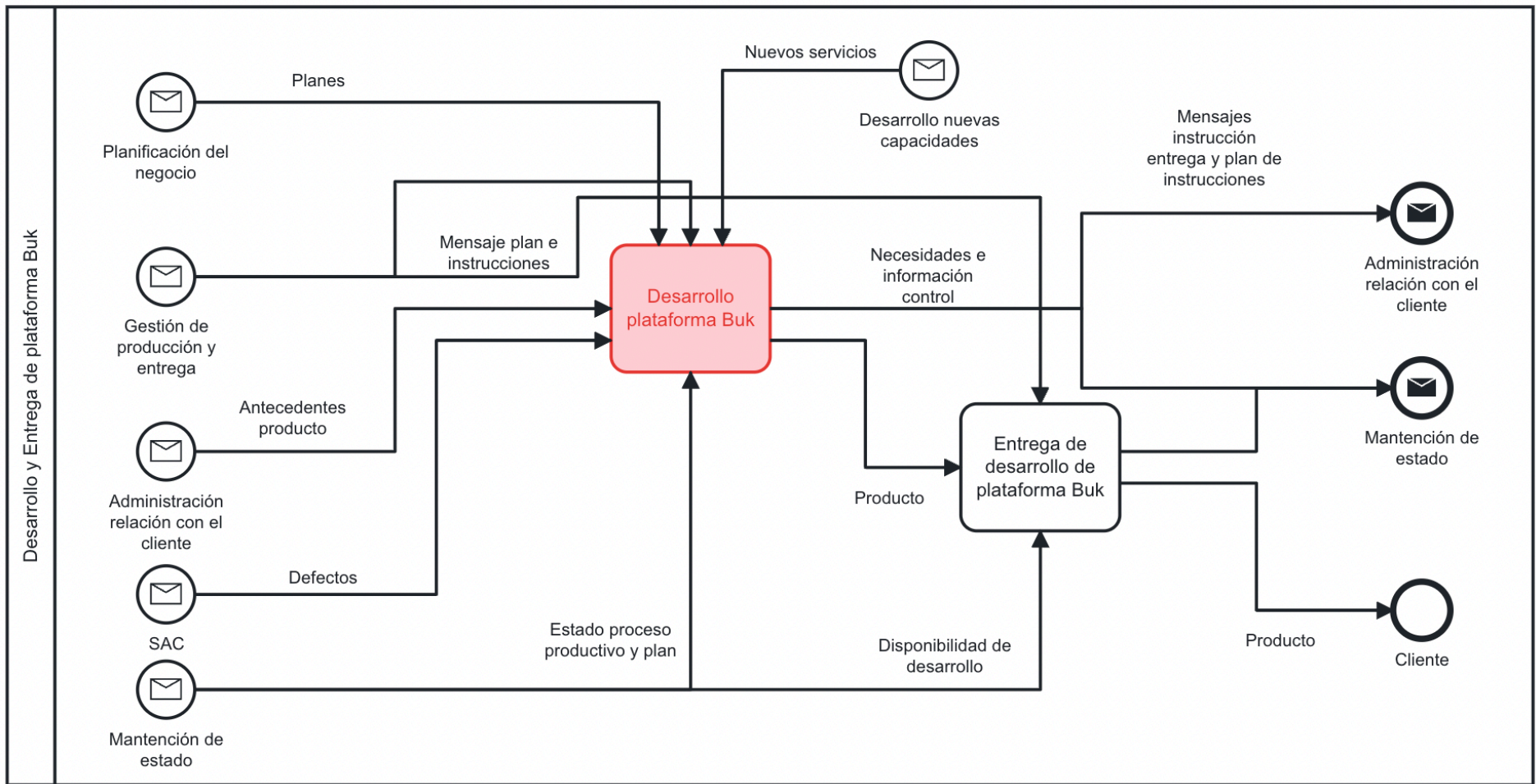


Figura 34: Macro 1.3 Desarrollo y entrega de plataforma Buk. Situación actual.
Fuente: Elaboración propia

3.5.2.3 Macro 1.3.1 Desarrollo de plataforma Buk

Este proceso se distingue debido a que acá es donde se realiza la creación de la plataforma o también podemos decir que es donde se programa el código necesario para la creación de la plataforma.

Para poder efectuar el desarrollo es necesario contar con distintas entradas para que sea llevado a cabo, como lo son los requerimientos de clientes que son proporcionados por “La administración relación con el cliente”, también la planificación del desarrollo brindado por la “La gestión de producción y entrega” y las incidencias levantadas por SAC que son los defectos que contiene la plataforma toda esta información permite que el equipo del área de Producto conformado por desarrolladores, Product Owner y Bussines Analyst lleven a cabo la ideación y creación de la plataforma.

Como control recibe los planes elaborados en la Macro 3 y los planes del proceso “Gestión, producción y entrega” que permiten que el desarrollo sea congruente con las estrategias levantadas por la organización.

La salida de este subproceso “Desarrollo plataforma Buk” contiene las nuevas funcionalidades creadas por los desarrolladores y las soluciones de las funcionalidades que presentaron defectos cuando los clientes hacían uso de la plataforma y por lo tanto reportaron a SAC. Asimismo, se observa que no cuenta con un subproceso de “Monitoreo”, que permita poder efectuar un control tanto de las buenas prácticas, cómo de calidad del desarrollo que va realizando el equipo, lo que posibilita que puedan realizarse fuga de defectos en los desarrollos.

Este proceso es importante sea alimentado por “Mantenimiento de estado” lo que le permite intercambiar información con los procesos que se relaciona y así alimentarse de esta información para poder llevar a cabo sus actividades.

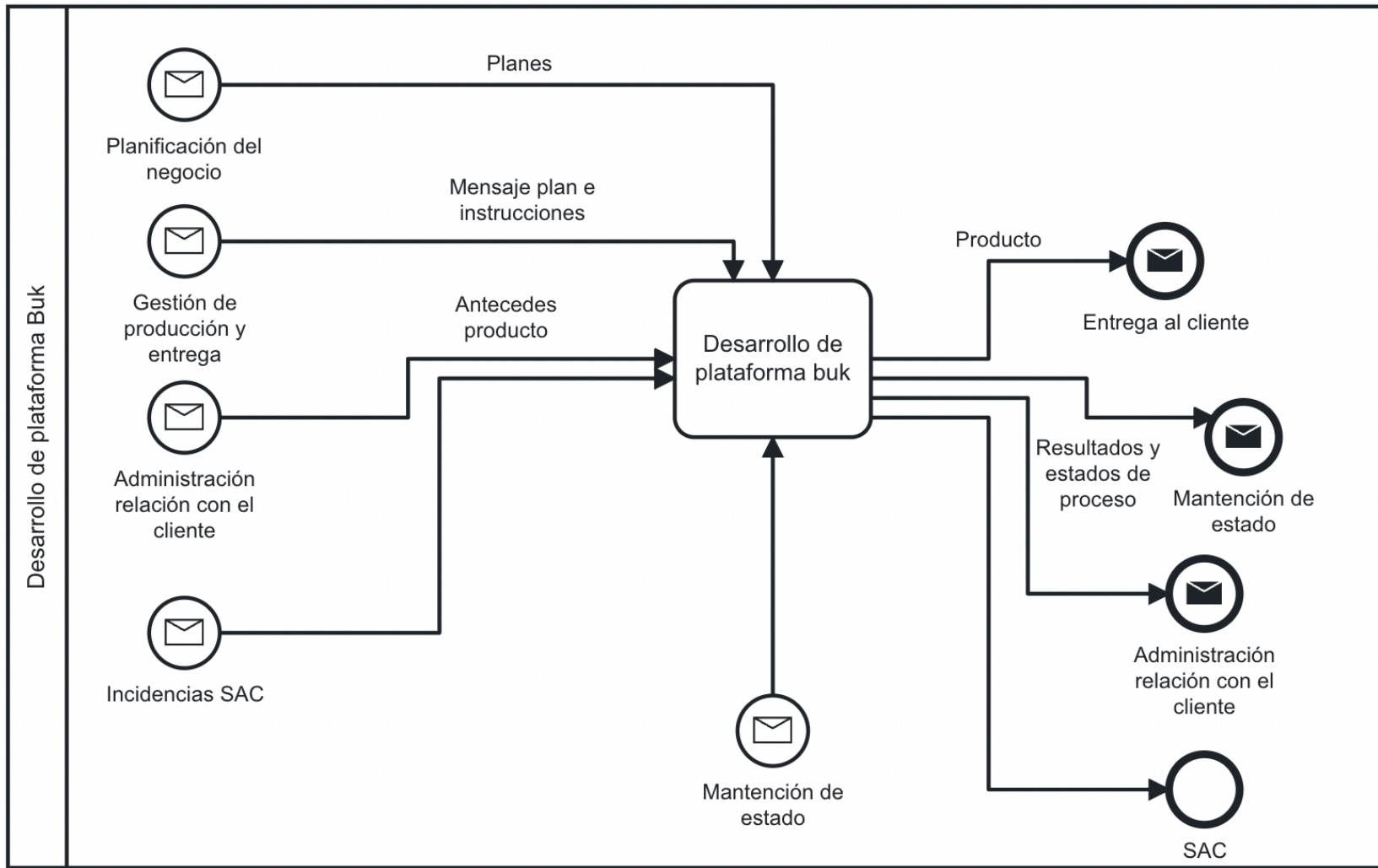


Figura 35: Macro 1.3.1 Desarrollo de plataforma Buk. Situación actual.

Fuente: Elaboración propia

3.5.3 Modelamiento BPMN Detallado de Procesos AS IS

Para poder comprender cómo se realiza la ideación y creación del desarrollo de la plataforma de Buk en el área de producto y de cómo funciona el levantamiento de las incidencias reportadas por SAC se realizará un análisis detallado de los procesos que intervienen.

3.5.3.1 Ideación y creación de la plataforma Buk

Es en este proceso donde se centra en la ideación y creación del desarrollo de la plataforma y está conformado por las siguientes actividades y dos subprocesos, en donde en uno se realiza el desarrollo de la plataforma y en el otro se realiza el levantamiento de las incidencias que son los defectos reportados por los clientes.

Se continua a detallar cada una de las actividades y subprocesos que conforman este proceso, y las dos áreas que lo conforman el área de producto y el área de atención al cliente (SAC).

- **Análisis de Negocio:** Esta actividad es realizada por las personas de negocio representados por los Product Owner (PO) y los Business Analyst (BA), ellos son los encargados de reunir y seleccionar los requerimientos o mejoras que son pedidos por el cliente, la ideación de nuevas funcionalidades y los requerimientos de Instituciones externas a Buk como PREVIRED, AFP, etc., que serán utilizadas para desarrollos futuros.
- **Refinamiento:** Esta actividad es realizada por el PO/BA y tiene como objetivo tomar los requerimientos seleccionados para la creación de nuevos desarrollos o mejoras y transformarlas en Historias de Usuario (HU) para que puedan ser desarrolladas por el equipo de desarrollo.
- **Estimación:** Esta actividad es realizada por el equipo que está conformado por el PO/BA y los desarrolladores, tiene como objetivo estimar el tiempo que tomará la creación de la funcionalidad o mejora del desarrollo de la HU.
- **Subproceso “Desarrollo”:** Es en este subproceso en donde se realiza la escritura de código, donde los desarrolladores toman las HU y parten a crear las funcionalidades además también reciben las incidencias de los defectos levantadas por SAC.
- **Subproceso “Levantamiento de incidencias”:** Es en este proceso en donde se reciben las incidencias que son los defectos levantados por los clientes y son los agentes de SAC los encargados de comunicarlo al área de desarrollo.

- **Comunicación al cliente:** Esta actividad tiene como objetivo que el PO/BA comunique a los clientes por distintos medios las nuevas funcionalidades que fueron desarrolladas por el equipo de desarrollo.

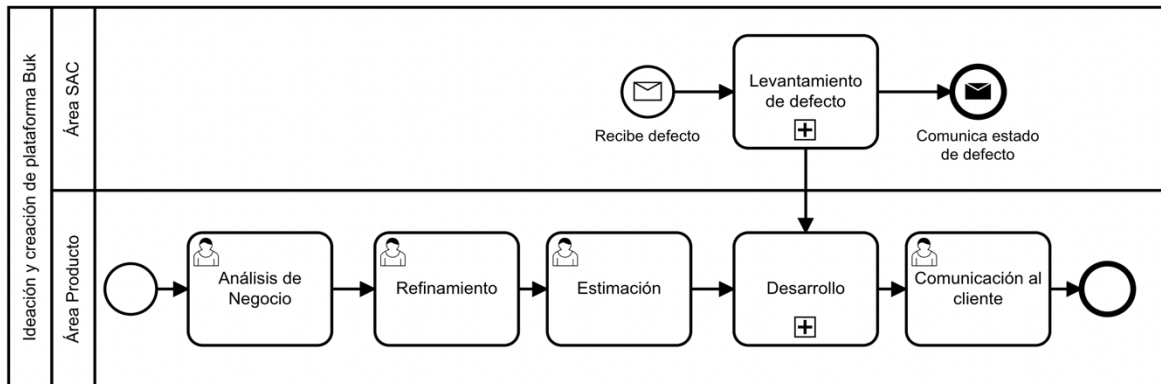


Figura 36: Diagrama BPMN Ideación y creación de plataforma Buk. Situación actual.

Fuente: Elaboración propia

3.5.3.2 Levantamiento de defectos

Es en este subproceso donde ocurre el levantamiento de las incidencias que son los defectos de la plataforma. Inicia cuando el agente de SAC recibe la incidencia que puede ser mediante correo, la misma plataforma Buk, teléfono, etc. Al recibir la incidencia, crea un ticket en la plataforma “Freshdesk” donde agrega todos los detalles referente al defecto explicando lo que está ocurriendo, en qué funcionalidad de la plataforma está afectando, adjuntando imágenes o videos y los paso a paso para poder replicar el defecto. Al realizar el análisis y verificar si es crítico, procede a informarlo al desarrollador que se encuentra como “Batman” que es el asignado para solo resolver los defectos criticos durante el sprint, este rol de superhéroe es asignado a un desarrollador de manera temporal por un sprint, por lo que en cada sprint hay un desarrollador de turno, cada equipo cuenta con un superhéroe al cual tiene distintos nombres asignados de acuerdo al país, por ejemplo en Chile es nombrado “Batman” y en México llamado “Chapulín”.

Cuando el Batman recibe el ticket procede a solucionarlo, si tiene alguna duda se lo comenta al agente SAC para que agregue más información, una vez solucionado se le informa al agente SAC para que se lo comunique al cliente.

Si el agente SAC al refinar el ticket y realizar el análisis y se percata es de prioridad baja entonces el ticket se guarda en backlog y se procederá a solucionarlo cuando el desarrollador en algún momento se encuentre libre.

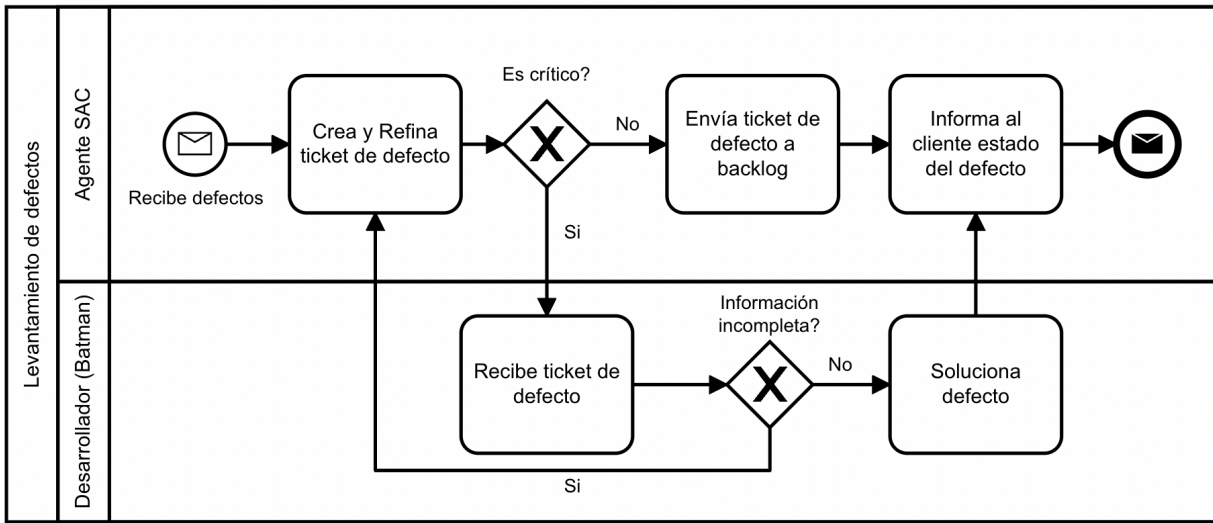


Figura 37: Diagrama BPMN Levantamiento de incidencias. Situación actual.

Fuente: Elaboración propia

3.5.3.3 Desarrollo de plataforma Buk

Es en este subproceso donde se realiza la creación del código, el desarrollo de la plataforma. Este proceso inicia cuando el perfil Business Analyst (BA) ha realizado los subprocesos de “Refinamiento” y “Estimación” lo que le permite tener claridad de lo planificado para los Sprint. Los equipos de producto trabajan con metodología Scrum, lo que permite el BA seleccione las HU que serán trabajadas por el equipo de desarrolladores por un periodo de dos semanas llamados también “sprint”. Cuando han sido seleccionadas se procede a asignar cada HU a un desarrollador que será el responsable de trabajarla durante el sprint, es el BA quien se encarga de revisar cumple con los requerimientos que se escribieron en la HU y si esta correcto desplegar el desarrollo en el ambiente productivo para poder comunicar al cliente las nuevas funcionalidades disponible en la plataforma. Si el BA encuentra que el desarrollo no cumple con algún requerimiento procede a informarlo al desarrollador responsable hasta que todos los requerimientos estén completos.

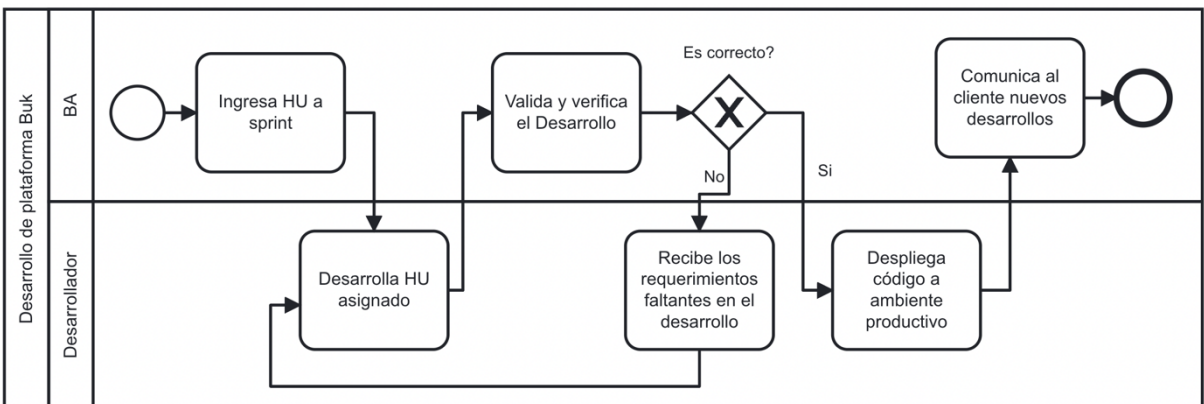


Figura 38: Diagrama BPMN Desarrollo de plataforma Buk. Situación actual.

3.6 Cuantificación del Problema u Oportunidad

En la figura 4, se da a conocer el total de defectos, que contiene la plataforma reportadas por los clientes siendo segmentadas de manera mensual en un periodo de julio 2021 a marzo 2022, donde se puede percatar que la cantidad de defectos ha tenido una mayor alza desde diciembre del 2021, representando en el primer trimestre del año 2022 fue de 907 incidencias, lo que manifiesta que cada uno de estos defectos provocaron molestias en clientes ya que impedían realizar de manera eficiente su trabajo y por lo tanto debieron acudir a SAC a reportar y buscar una pronta solución. Que los clientes divulguen su mala experiencia al usar la plataforma permite que se pueda presentar mala reputación para la empresa impactando negativamente en la adquisición de nuevos clientes y además reflejarse en retiros de suscripciones, representando en ambos escenarios pérdida de ingresos para la empresa.

Para poder evaluar el impacto económico del problema se procedió a realizar un cálculo basado en el actual modelo de negocios que utiliza Buk, a fin de realizar una estimación anual de las ganancias de Buk, basándose en la cantidad de suscriptores en Chile (este proyecto solo se trabaja con este país), por la cantidad de pago por suscriptor. Con el propósito de tener el dato correcto referente al porcentaje de salida de cliente, el área de fidelización de Buk que son los encargados de llevar el control de los clientes, proporciono que el porcentaje de salida de los clientes anuales en Buk es del 10%, con este dato se logra obtener un aproximado de lo que la empresa percibe como pérdidas por retiro de suscripciones de USD 1,585,911.11.

También es importante destacar que los altos defectos además de impactar económicamente en pérdidas de dinero, también afectan en lo que es el desarrollo de nuevas funcionalidades. Se logra concluir de acuerdo a dato proporcionado por el área de producto, que se cumple solamente con un estimado del 6% de las funcionalidades que son planificadas afectando el desarrollo de nuevas funcionalidades que limita a que en la plataforma se desarrollen innovaciones y por consecuencia afecte a la misión de la empresa que es ser la mejor plataforma de RRHH en Latinoamérica.

CAPÍTULO 4: PROPUESTA DE DISEÑO DE PROCESOS

En este capítulo se pretende explicar lo que será una propuesta de rediseño basada en la combinación de la metodología de rediseño de procesos propuesta por (Barros, 2000) y el framework COBIT 4.1 (ITGI, 2007).

Se partirá con las direcciones de cambio que nos permitirá conocer lo actualmente existente y lo que ofrece el rediseño. Posteriormente se presenta una arquitectura de procesos y lo que será el diseño TO-BE. Además de cómo el apoyo tecnológico aportara el incorporamiento de la lógica de negocio para lograr el rediseño.

Con el rediseño se tiene como objetivo la mejora de la calidad del producto, procurando la disminución de defectos, que permita brindar a los clientes una mejor experiencia en las funcionalidades de la plataforma.

4.1 Direcciones de Cambio

Se pretende conocer con las Direcciones de Cambio la diferencia entre la situación actual y lo que ofrece el rediseño centrado en innovación respecto al proceso.

4.1.1 Estructura empresa y mercado

Esta dirección está centrada en la organización para conocer su posición estratégica y el poder de decisión que tienen las áreas que conforman la organización.

4.1.1.1 Situación Actual

Actualmente la empresa se encuentra ubicada en la posición estratégica basada en *mejor producto* siendo más específico en lo que es *diferenciación* debido a que en comparación con lo que es la competencia brinda una plataforma que contiene distintas funcionalidades a las que ofrece el mercado.

La empresa Buk, se caracteriza por ser descentralizada cada una de sus áreas toman sus decisiones de manera independiente además de estar enfocada en ser una empresa horizontal lo que permite una toma de decisiones más rápida y eficiente.

4.1.1.2 Propuesta de Diseño

Con el rediseño la estructura actual de la empresa no se ve afectada, ya que lo que pretende el rediseño es seguirse posicionando como *mejor producto* al enfocarse en la calidad permitiendo que el producto al ser entregado al cliente sea cada vez de mayor valor.

Tabla 5: Variable de diseño: Estructura empresa y mercado.

Fuente: Elaboración propia

a.	Estructura empresa - mercado	AS-IS	TO-BE
a1.	Servicio integral al cliente	No	No
a2.	Lock-in sistemático	No	No
a3.	Integración con proveedores	No	No
a4.	Estructura interna centralizada o descentralizada	Descentralizada	Descentralizada
a5.	Toma de decisiones centralizada o descentralizada	Descentralizada	Descentralizada

4.1.2 Anticipación

Esta dirección está centrada en la prevención por lo que para este proyecto es de las más importantes a considerar ya que permite enfocarse en lo que será las necesidades del cliente, pero más que todo en poder anticiparse a que sea el perfil QA quien encuentra los defectos antes que el cliente.

4.1.2.1 Situación Actual

La empresa Buk actualmente toma en cuenta los requerimientos del cliente para sus próximos desarrollos, pero al no existir planes de calidad que velen por un producto sin defectos y al tener una plataforma con una alta cantidad de defectos los cuales deben ser solucionados de inmediato por los desarrolladores, resultando como consecuencia retraso en el cumplimiento de requerimientos pedidos por clientes.

4.1.2.2 Propuesta de Diseño

Con el rediseño se pretende realizar una correcta planificación de gestión de la calidad enfocada en mejorar el producto lo que permitirá enfocarse en la prevención de defectos y dando como consecuencia se pueda disminuir las incidencias reportadas por el cliente así poder enfocar al desarrollador para que pueda cumplir con los requerimientos que son pedidos por los clientes.

Tabla 6: Variable de diseño: Anticipación.

Fuente: Elaboración propia

b.	Mantenimiento consolidada de estado	AS-IS	TO-BE
b1.	Planificación de calidad	No	Si

b2.	Percepción de los requerimientos de los clientes	Si	Si
b3.	Modelo de prevención de defectos	No	Si

4.1.3 Coordinación

Esta dirección está centrada en la coordinación entre las funcionalidades de la empresa y la de los clientes.

4.1.3.1 Situación Actual

Actualmente en la empresa Buk, no se encuentran establecidas reglas enfocadas en ser un apoyo en una correcta gestión de calidad por lo tanto no se brinda colaboración entre las áreas de las empresas.

4.1.3.2 Propuesta de Diseño

Con el rediseño se pretende establecer un plan de gestión de calidad que contendrá las reglas tanto de negocio como de calidad que permitan brindar un producto que cumpla con los mejores estándares y requerimientos de negocio. Por lo que será necesaria la colaboración de áreas como SAC que conocen los dolores del cliente y sus necesidades.

*Tabla 7: Variable de diseño: Coordinación.
Fuente: Elaboración propia*

c.	Mantenimiento consolidada de estado	AS-IS	TO-BE
c1.	Reglas	No	Si
c2.	Colaboración	No	Si

4.1.4 Prácticas de trabajo

Esta dirección es la encargada de la formalización de lo que es la implementación de reglas y procedimientos.

4.1.4.1 Situación Actual

Actualmente en la empresa Buk, no se encuentran establecidos procedimientos y reglas que permitan poder enfocarse en la calidad del producto, por lo tanto, no se encuentra la lógica de negocio. Tampoco se tiene un control de medición.

4.1.4.2 Propuesta de Diseño

Con el rediseño se incorpora la lógica de negocio que contiene los requerimientos de negocio que permitirá prevenir defectos en donde la herramienta será la encargada de encontrar defectos antes que el producto llegue al cliente. Al incorporar el marco COBIT 4.1 permitirá el uso de controles que nos brindaran tener indicadores que nos dará información referente a la calidad del producto y su estado.

Tabla 8: Variable de diseño: Prácticas de trabajo. **Fuente:** Elaboración propia

d.	Mantención consolidada de estado	AS-IS	TO-BE
d1.	Lógica de negocio automatizada	No	Si
d2.	Lógica y procedimientos de medición de desempeño y control	No	Si

4.1.5 Integración de procesos conexos

Esta dirección está centrada en la interacción de los procesos dentro de los Macroprocesos o entre distintos Macroprocesos.

4.1.5.1 Situación Actual

Actualmente en la empresa Buk, el proceso del desarrollo de producto se encuentra en el macro 1 aislado sin interacción con algún proceso de monitoreo que se encargue de medir la calidad en el proceso de desarrollo de software, que permita someterlo a pruebas para establecer su funcionamiento de acuerdo a lo previsto y si no es así tomar acciones correctivas.

4.1.5.2 Propuesta de Diseño

Con el rediseño se pretende que el proceso del desarrollo de producto se encuentre un subproceso de "Monitoreo de Calidad" que permita poder incorporar pruebas siguiendo procedimientos, reglas enfocados en la gestión de calidad mediante el Framework COBIT 4.1 y se pueda asegurar el correcto funcionamiento de la plataforma.

Tabla 9: Variable de diseño: Integración de procesos conexos. **Fuente:** Elaboración propia

e.	Mantención consolidada de estado	AS-IS	TO-BE
----	----------------------------------	-------	-------

e1.	Proceso aislado	Si	No
e2.	Todos o la mayor parte de los procesos de un Macroproceso	No	Si
e3.	Dos o más macros que interactúan	No	No

4.1.6 Mantención consolidada de estado

Esta dirección está centrada en la integración, en que las actividades tengan conocimiento de lo que sucede en las demás actividades del proceso.

4.1.6.1 Situación Actual

Actualmente la empresa hace uso de la herramienta “Freshdesk” que es la que centraliza y le permite obtener toda la información referente a las consultas, reportes de incidencias y nuevos requerimientos de los clientes. Por lo que se mantiene una constante interacción y comunicación entre el área de SAC y Producto. No se mantiene integración con sistemas de otras empresas u organizaciones.

4.1.6.2 Propuesta de Diseño

Con el rediseño se pretende seguir haciendo uso de la herramienta “Freshdesk” para poder centralizar toda la información referente a clientes. Tampoco se necesitará la integración con sistemas externos a la empresa.

Tabla 10: Variable de diseño: Mantención consolidada de estados.

Fuente: Elaboración propia

f.	Mantención consolidada de estado	AS-IS	TO-BE
f1.	Datos propios	Si	Si
f2.	Integración de datos con otros sistemas de la empresa	Si	Si
f3.	Integración de datos con sistemas de otras empresas	No	No

4.1.7 Arquitectura de Procesos TO BE

4.1.7.1 Diseño en IDEF0

A continuación, se presenta el rediseño enfocado en gestión de calidad haciendo uso de la metodología de rediseño de procesos propuesta por (Barros, 2000), en donde estos modelos Macro detallan las actividades y flujos de los procesos que contiene una organización.

Barros presenta su arquitectura genérica de Macroprocesos dentro de los cuales en algunos casos pueden ser aplicables en situaciones más específicas como sus patrones para Hospitales, Bancos, etc. Sin embargo Barros con su arquitectura genérica nos permite diseñar a partir de estas instancias por lo que para poder realizar este rediseño las Macro de Barros no abarcaban el suficiente alcance para lograr el objetivo del proyecto que es el aumentar la calidad del producto, por lo cual se necesitó la integración de un marco enfocado en la calidad, por lo que se procedió a trabajar con el marco COBIT 4.1 ya que nos proporciona un proceso enfocado en la "Administración de la calidad". Al integrar la metodología de Barros y el marco COBIT 4.1 se obtuvo un modelamiento detallado enfocado en la gestión de calidad del desarrollo del producto Buk.

4.1.7.2 Macroprocesos de Buk

Se presenta los Macroprocesos que conforman la empresa Buk y en como con el rediseño impactara directamente a la Macro 1. Es acá donde se ha centrado debido a que es la encargada de recibir los requerimientos de los clientes, el desarrollo de la plataforma basándose en estos y además con el rediseño agregar un proceso de monitoreo de calidad que permita realizar las pruebas necesarias para poder velar por la correcta funcionalidad de la plataforma y poder entregar un producto que contenga prácticas de calidad.

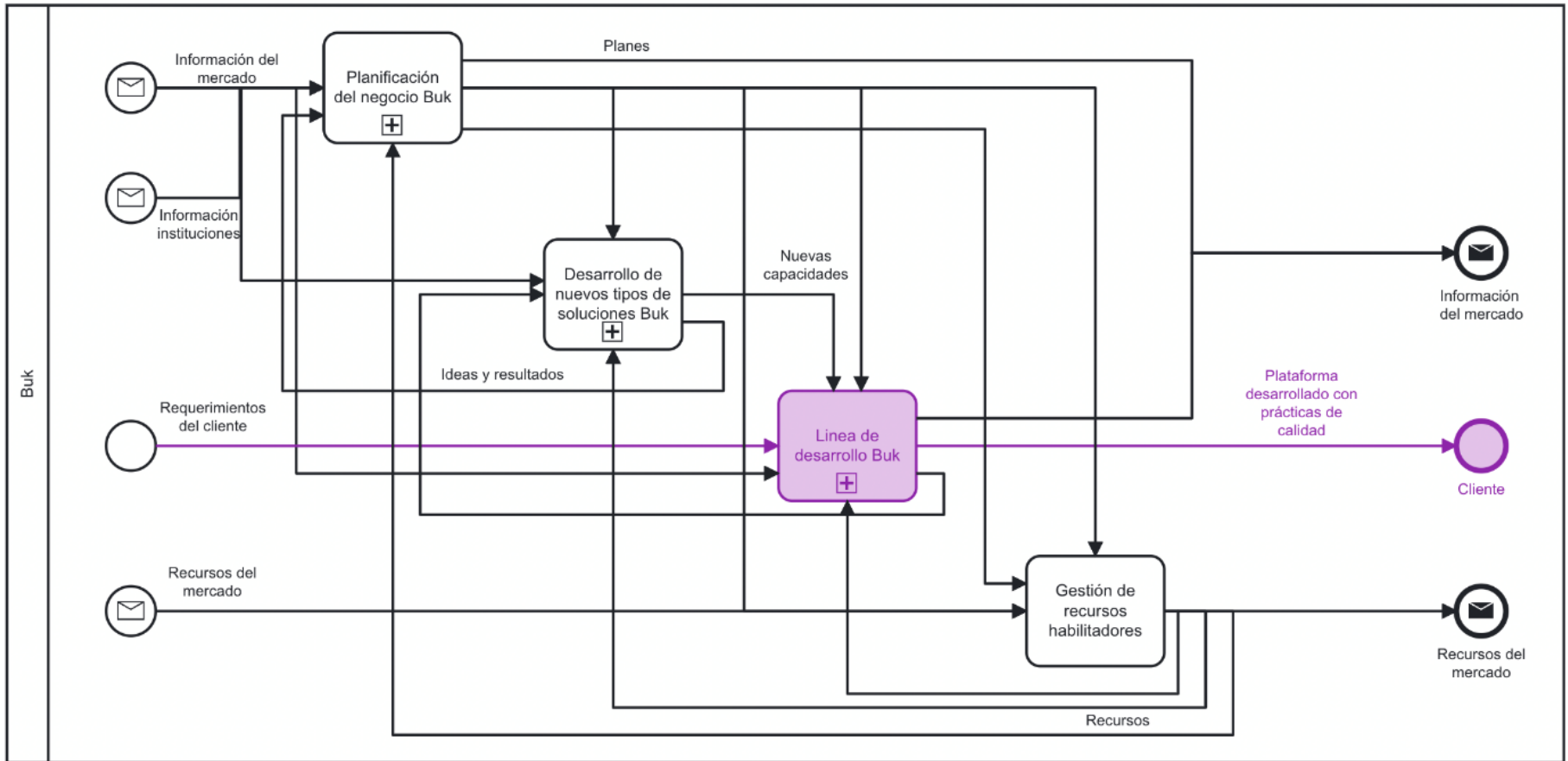


Figura 38: Macroprocesos de Buk.
Fuente: Elaboración propi

Macro 1.3: Línea de desarrollo y entrega de la plataforma Buk

Este rediseño se centra en el proceso de “Desarrollo y entrega de la plataforma Buk” en donde recibe como control los requerimientos obtenidos por los clientes de acuerdo a estos se procederá al desarrollo de la plataforma y además se realizará la gestión de calidad basada en COBIT 4.1. Como salida se tendrá un producto que contiene prácticas de calidad que habrán sido establecidos en el plan de calidad que será desarrollado por el QA, para poder entregar el producto a los clientes. La mantención de estado recibirá información referente a los resultados de las pruebas de calidad.

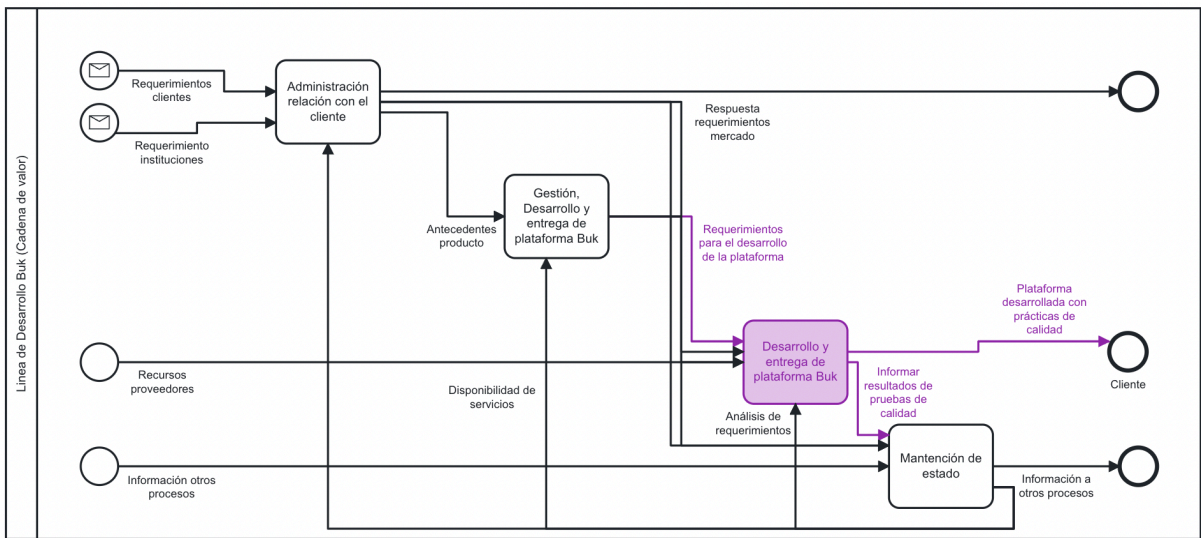


Figura 39: Macro 1.3 Línea de Desarrollo y Entrega de plataforma Buk.

Fuente: Elaboración propia

4.1.7.3 Macroproceso 1.3.1.1: Desarrollo de la plataforma Buk

Este rediseño se enfoca en el proceso “Desarrollo de la plataforma Buk”, tiene como objetivo principal la creación del código realizado por los desarrolladores que vienen a ser las funcionalidades que se agregarán a la plataforma, en este proceso también se realizan las ejecuciones de las pruebas de calidad basándose en los planes de acuerdo a actividades que determina el marco COBIT 4.1.

El proceso de “Desarrollo de la plataforma Buk” recibirá como entrada las prácticas y planes de calidad, requerimientos de clientes, los defectos reportados por SAC. Como control recibe la planificación de negocio que le permita estar sincronizado con las estrategias de la empresa. De salida brindara los resultados de las pruebas, entregará los planes de prueba a negocio y velara por entregar una plataforma con prácticas de calidad que esté libre de defectos al ser entregada al cliente realizando distinto tipos de pruebas tanto manuales como automatizadas.

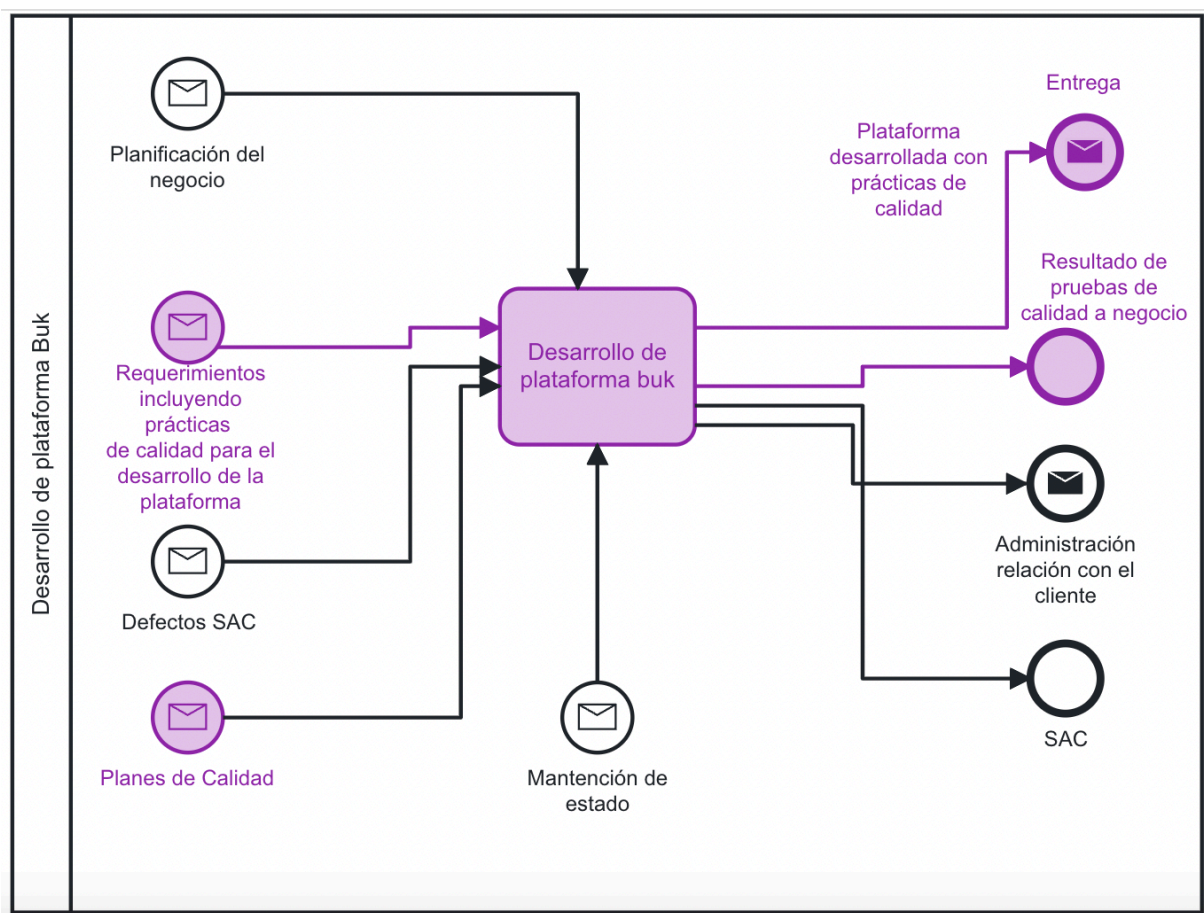


Figura 40: Macro 1.3.1.1 Desarrollo de plataforma Buk.
Fuente: Elaboración propia

4.1.8 Modelamiento Detallado de Procesos TO BE

Para poder llevar a cabo el rediseño del proceso del área de producto, se procedió a incorporar el proceso centrado en *Administración de Calidad* de COBIT 4.1.

COBIT 4.1 cuenta con el dominio llamado “Planificar y Organizar” el cual consta de 10 procesos de los cuales el proceso llamado “Administración y planificación de la calidad” que renombraremos como “Planificación de Aseguramiento de Calidad” y “Ejecución de planes de calidad”, para poder tener un proceso que fuese eficiente se dio la necesidad de dividir en dos subprocesos. Teniendo uno como objetivo la creación de los planes de calidad de acuerdo a los requerimientos de negocio y el otro subproceso enfocado en las ejecuciones de las pruebas de acuerdo a su tipo y teniendo como responsabilidad brindar el resultado de las ejecuciones a negocio.

4.1.8.1 Proceso “Ideación y Desarrollo de la Plataforma Buk”

Es el proceso donde ocurre la creación de la plataforma, desde su ideación hasta su entrega. Es el proceso donde se identificó el problema, por ende se partió de este para el rediseño. Se incorporaron los subprocesos “Construcción de requerimientos del producto”, “Incorporación de calidad en los requerimientos del producto” y el rediseño del subproceso “Desarrollo de la plataforma” de los cuales posteriormente se explicarán a detalle, explicando como el QA apoya a estos subprocesos. Teniendo como finalidad que en este proceso este centrado en la creación de un producto que contenga los mejores controles de calidad y entregarle al cliente un producto sin defectos.

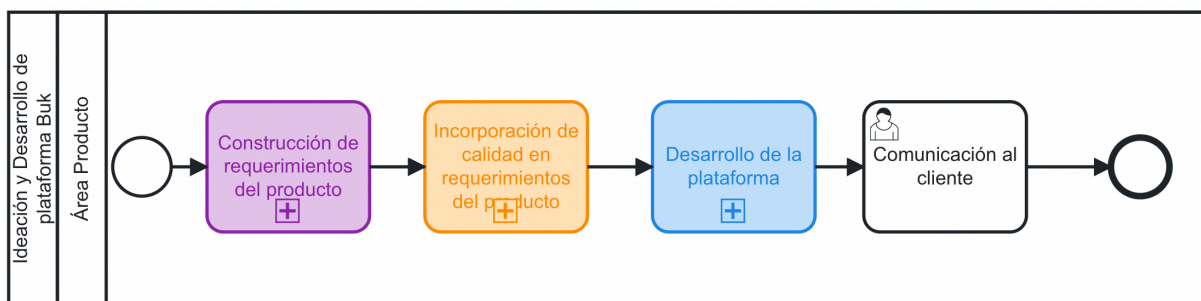


Figura 41: Diagrama BPMN para el proceso “Ideación y Desarrollo de la plataforma Buk”.

Fuente: Elaboración propia

4.1.8.1.1 Subproceso “Construcción de requerimientos del producto”

En este subproceso se realiza la interacción del Product Owner (encargado del producto) con los requerimientos que son pedidos por los clientes que pueden ser desde nuevas funcionalidades o mejoras de la plataforma y de las instituciones externas como puede ser PREVIRED, AFP quienes brindan requerimientos las cuales debe cumplir la plataforma para un adecuado uso en pago de planillas, seguros de salud, etc. También SAC cumple con la función de comunicar los defectos que son reportados por los clientes para que el Product Owner les de prioridad por ende el Product Owner procederá a recibir y reunir los requerimientos del producto, de lo cual procede a realizar un filtro de acuerdo a su prioridad para poder asignarlos a los Sprint en que se desarrollaran para poder ser entregados al cliente.

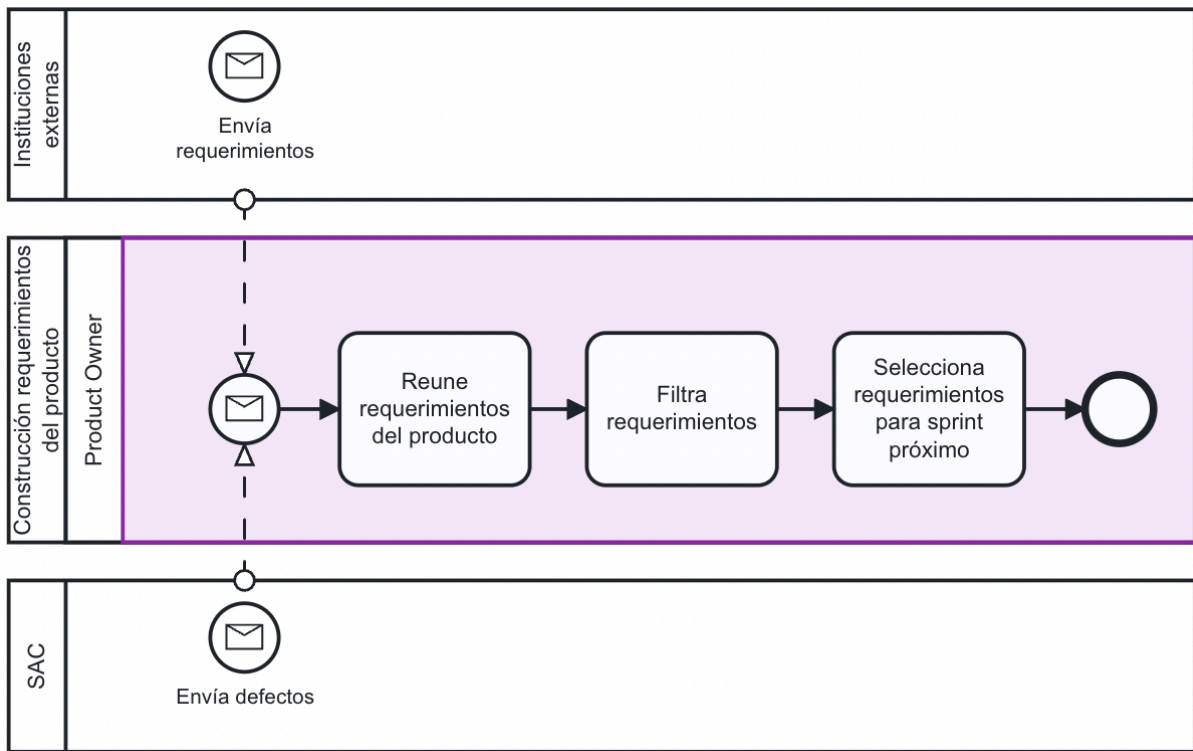


Figura 42: Diagrama BPMN para el subproceso “Construcción y requerimientos del producto”.
Fuente: Elaboración propia

4.1.8.1.2 Subproceso “Incorporación de calidad en los requerimientos del producto”

En este subproceso interactúa el Analista de Negocio (BA) y el “QA”. Cuando se ha seleccionado las Historias de Usuario (HU) por el Product Owner que ingresarán al sprint se incorpora una instancia llamada “Sesión Post-Refinamiento”, en donde se hace uso de este subproceso en el cual al tener las HU seleccionadas, el BA procede a agregar criterios de aceptación y al finalizar el “QA” agregara casos bordes que son estos escenarios muy raros que podría un cliente realizar pero en los que podría fallar la plataforma por lo que con esta instancia se pretende evitar cualquier situación que arroje un defecto en la plataforma. En este subproceso se contempla lo definido en la “Metodología Shift Left”, que busca incorporar al “QA” en etapas tempranas del desarrollo que son los requerimientos y diseño del producto, por lo que al incorporarlo se busca sea apoyo para la disminución de defectos, como menciona la literatura el incorporamiento del “QA” en estas etapas puede disminuir hasta un 75% los errores en etapa de desarrollo.

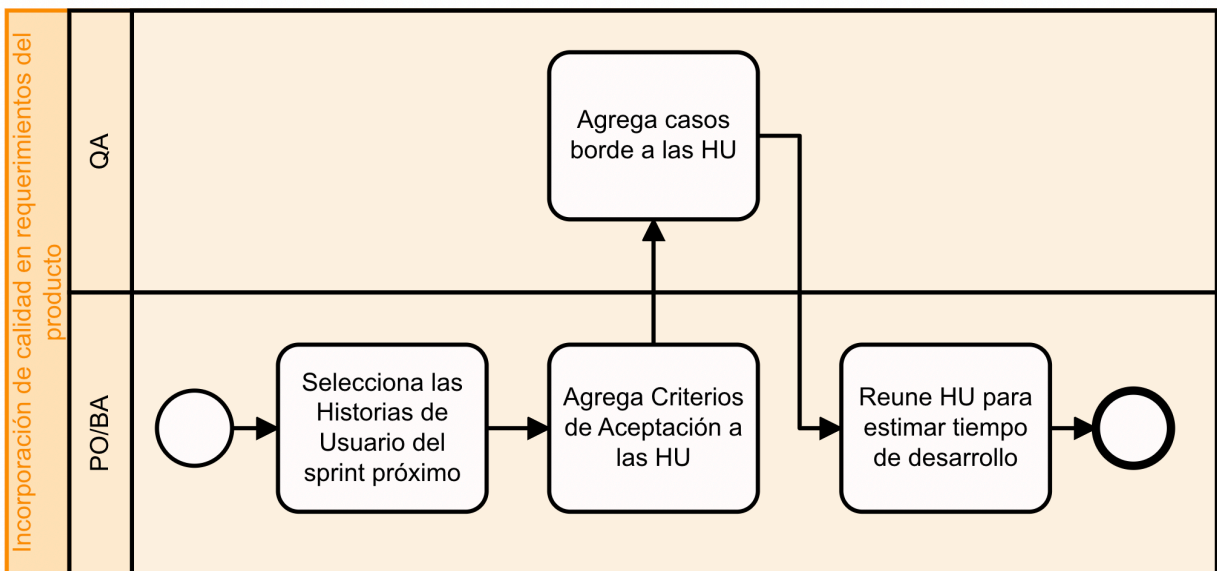


Figura 43: Diagrama BPMN para el subproceso “Incorporación de calidad en requerimientos del producto”.

Fuente: Elaboración propia

4.1.8.1.3 Subproceso “Desarrollo de la plataforma Buk”

Es en este subproceso es donde se ha realizado la mayor parte del rediseño del proceso “Desarrollo de Plataforma Buk” y contiene la lógica de negocio.

- Inicia con el subproceso “**Refinamiento de Historia de Usuario con funcionalidades a desarrollar**”, enfocado en evaluar si las funcionalidades candidatas a desarrollar son viables técnicamente. Parte con el PO/BA seleccionado las HU que se agregarán en los próximos sprints, teniendo claro esto agrega toda información oportuna referente a la funcionalidad y se lo comunica al Tech Lead, quien validará si la funcionalidad es viable técnicamente. Al realizar el análisis informa al PO/BA cuales de estas son técnicamente posibles de desarrollarse de modo que el PO/BA puede ingresarlas en el sprint como se observa en la figura 45.
- Se prosigue con el subproceso “**Planificación de Aseguramiento de Calidad**”, centrado en la creación de los planes de calidad que se deberán seguir por el equipo QA, el cual comunicara a negocio para conocer el alcance que tendrán las pruebas. Al ver la figura 46, notamos inicia con el equipo de QA recibiendo las funcionalidades que se desarrollaran, lo que le permitira crear los planes de calidad para definir que practicas y metodologias debe realizar todo el equipo agregando los tipos de pruebas que serán necesarios para incrementar la calidad de la plataforma y una vez finalizado comunicar a negocio el plan de calidad. Las actividades de este subproceso se basaron del proceso de calidad de COBIT 4.1.
- Será el PO/BA representando a negocio quien dará inicio al sprint agregando las Historias de Usuario que contienen las funcionalidades que seránb creadas por el equipo de desarrollo.
- Al equipo de desarrollo se le asignara la HU de la cual será responsable de crear el código necesario en el subproceso “**Desarrollo de funcionalidades de la plataforma**” , al finalizar el código será revisado por el Tech Lead quien validará cumpla los requisitos técnicos. Si cumple técnicamente se desplegara el código a ambiente de pruebas para que pueda ser validado y verificado por QA en cambio sino cumple el desarrollador recibira las correcciones.
- Es en el subproceso “**Ejecución de Planes de Pruebas**” de la figura 48, donde se lleva a cabo la lógica de negocio. Inicia con el QA Manual recibiendo las HU que ya se encuentran desarrolladas, verifica de acuerdo al plan de calidad el tipo de prueba que será necesaria. Si la prueba es manual será el QA Manual, el encargado de crear los casos de prueba de la funcionalidad y ejecutarlos lo que le permita conocer el estado de calidad de la funcionalidad, si el QA Manual

encuentra errores deberá informar al desarrollador para que sea corregido. En cambio sino encuentra errores comunica funciona correctamente. En el caso de que una prueba sea del tipo automatizada, es el QA Automatizador el responsable de crear los escenarios de pruebas y convertirlos en scripts que contiene la lógica de negocio. Estos scripts incluyen escenarios de pruebas que validan y verifican que las funcionalidades desarrolladas en el sprint no agreguen errores a las funcionalidades existentes, permitiendo monitorear y realizar revisiones periodicas. Y si es el caso en que se encuentre un error alertar oportunamente para que pueda ser corregido por el equipo de desarrollo y este defecto no sea encontrado por el cliente. Es el QA Manual quien se encarga de enviar a negocio el estado de resultado de las pruebas manuales y automatizadas además de las métricas obtenidas como la cantidad de errores encontrados en etapa de desarrollo y si los scripts previnieron defectos.

- Al recibir negocio el resultado de las pruebas y conocer que el estado de calidad de la plataforma es el correcto entonces procede a aprobar el despliegue en el ambiente productivo y a comunicar a los clientes las nuevas funcionalidades añadidas a la plataforma para que puedan hacer uso de ellas.

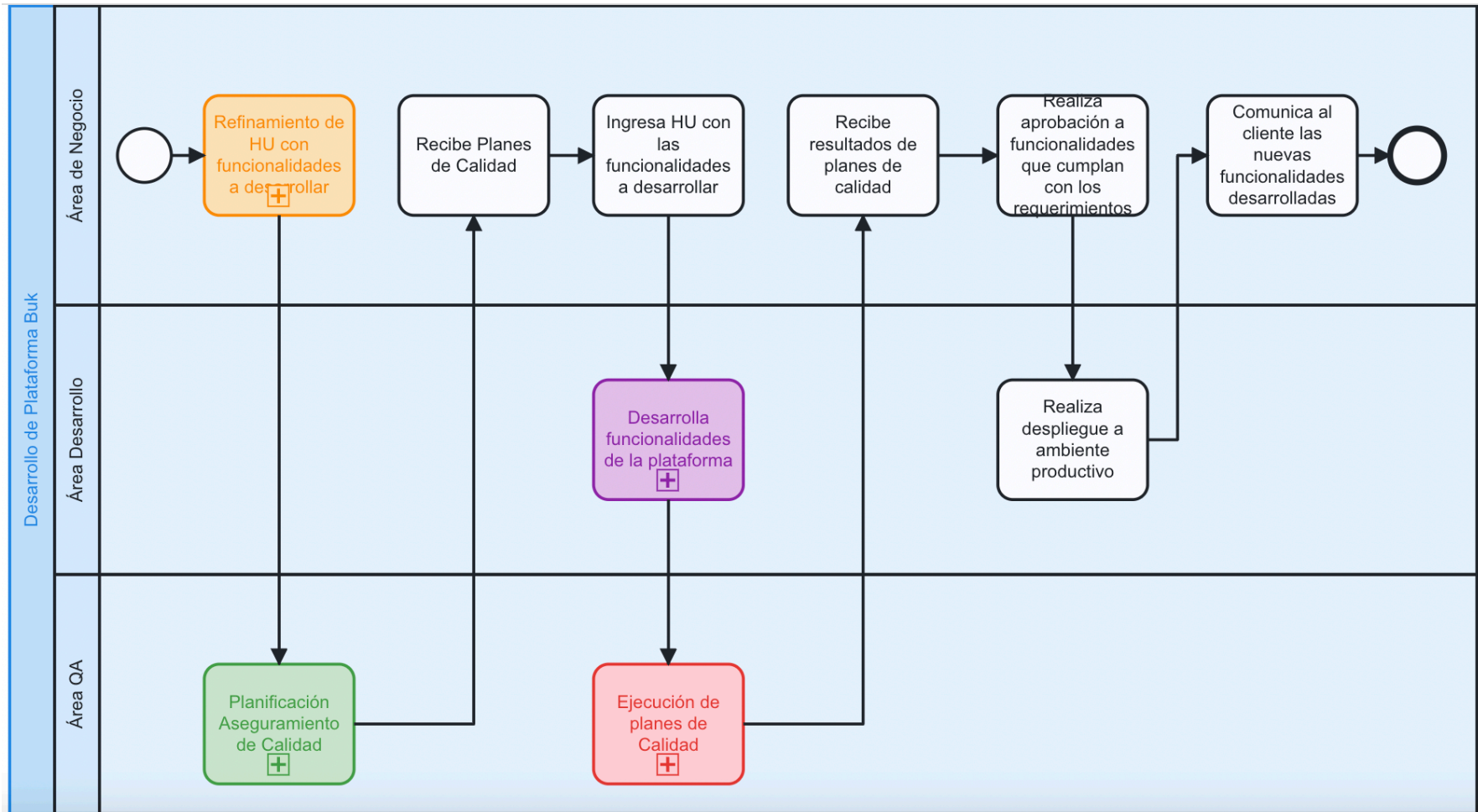


Figura 44: Diagrama BPMN para el proceso “Desarrollo de plataforma Buk”.

Fuente: Elaboración propia

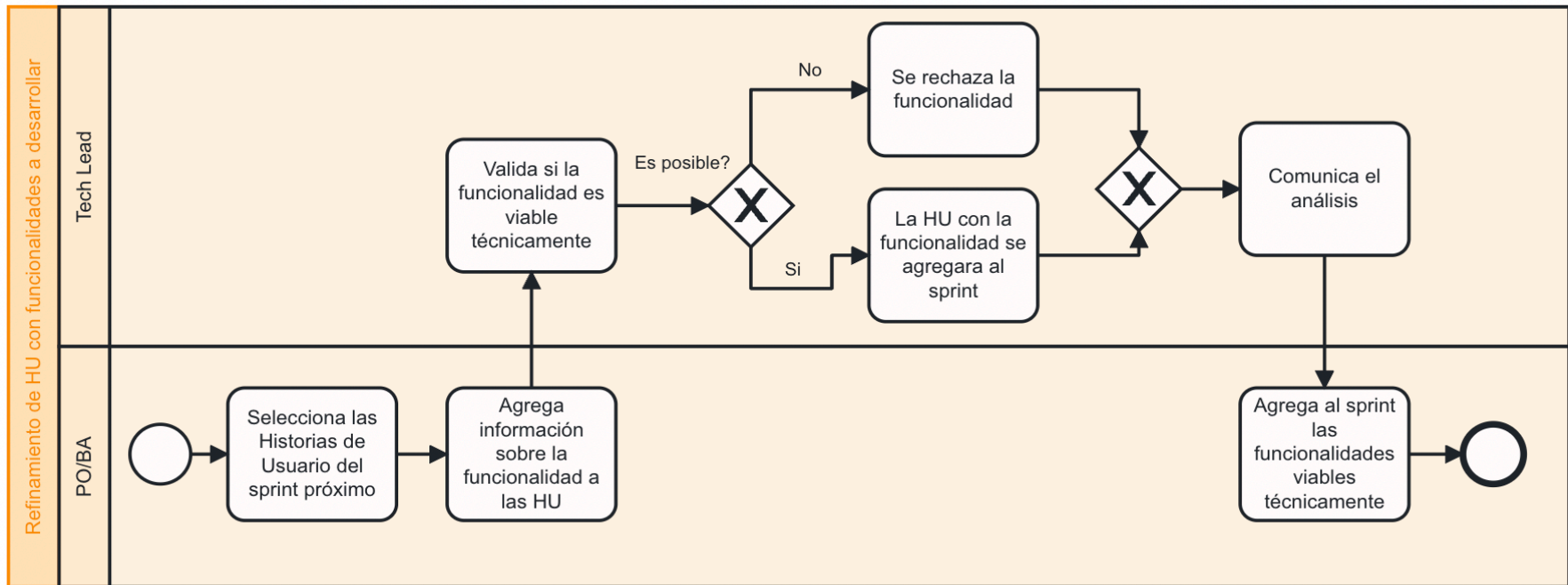


Figura 45: Diagrama BPMN para el subproceso "Refinamiento de HU con funcionalidades a desarrollar". Fuente: Elaboración propia

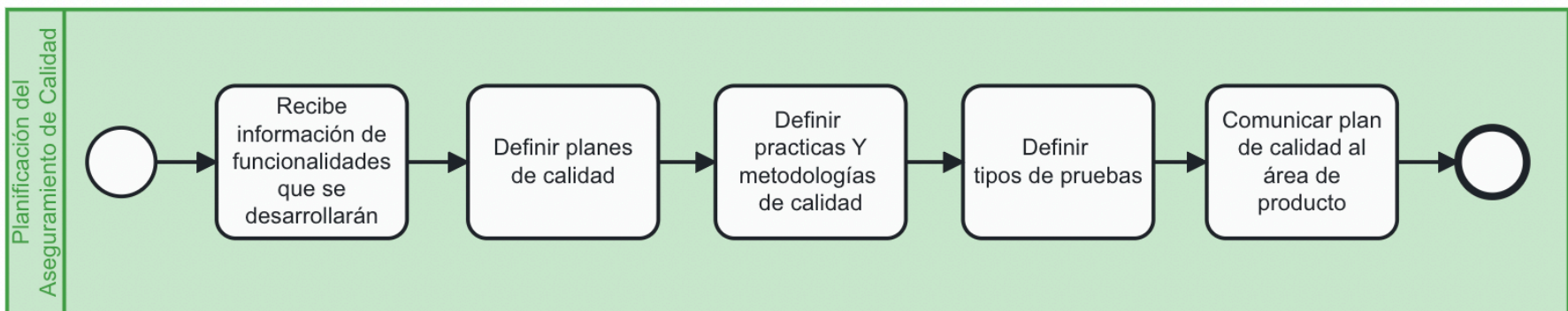


Figura 46: Diagrama BPMN para el subproceso "Planificación del Aseguramiento de Calidad". Fuente: Elaboración propia

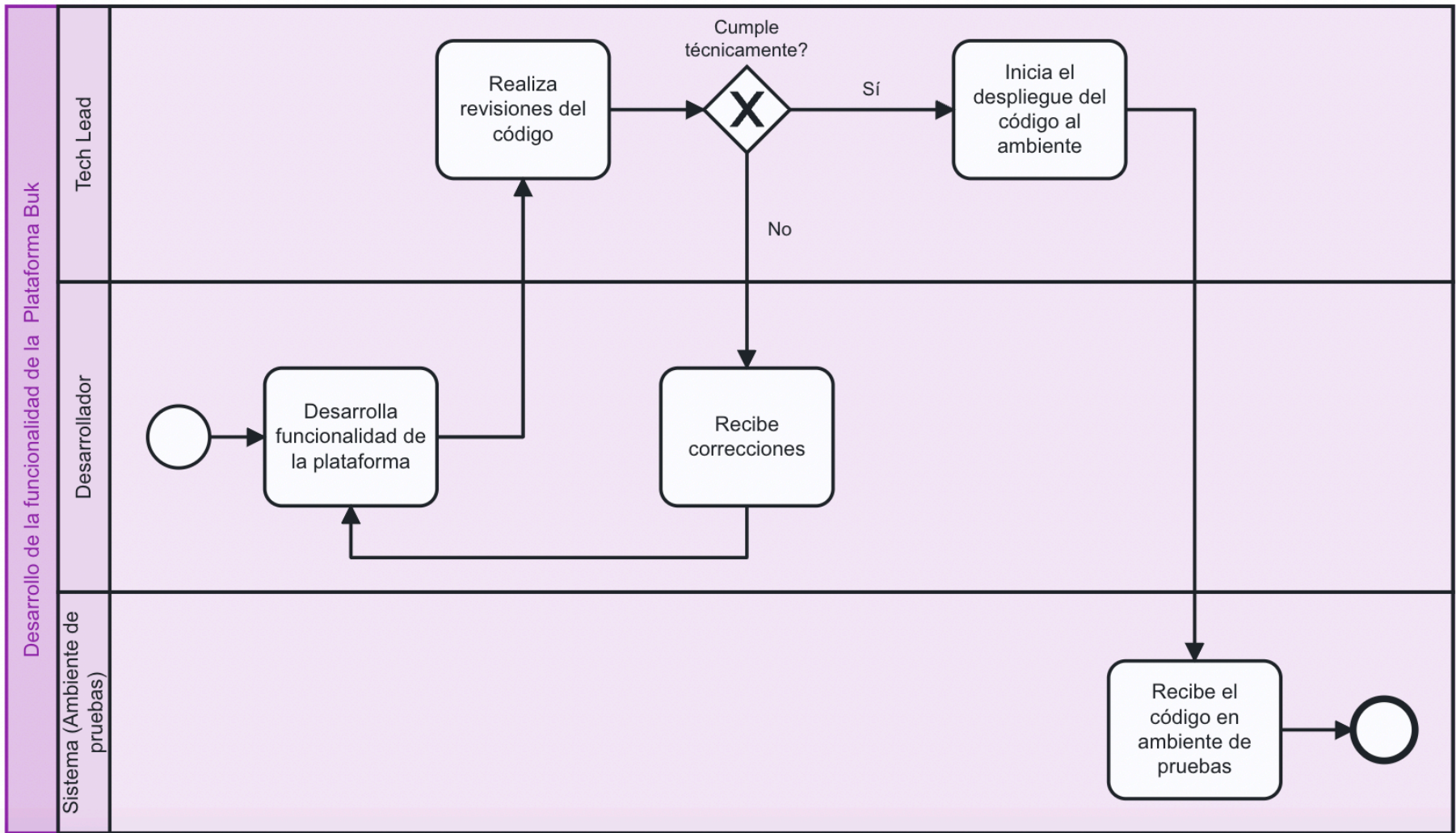


Figura 47: Diagrama BPMN para el subproceso “Desarrollo de la funcionalidad de la plataforma”.
Fuente: Elaboración propia

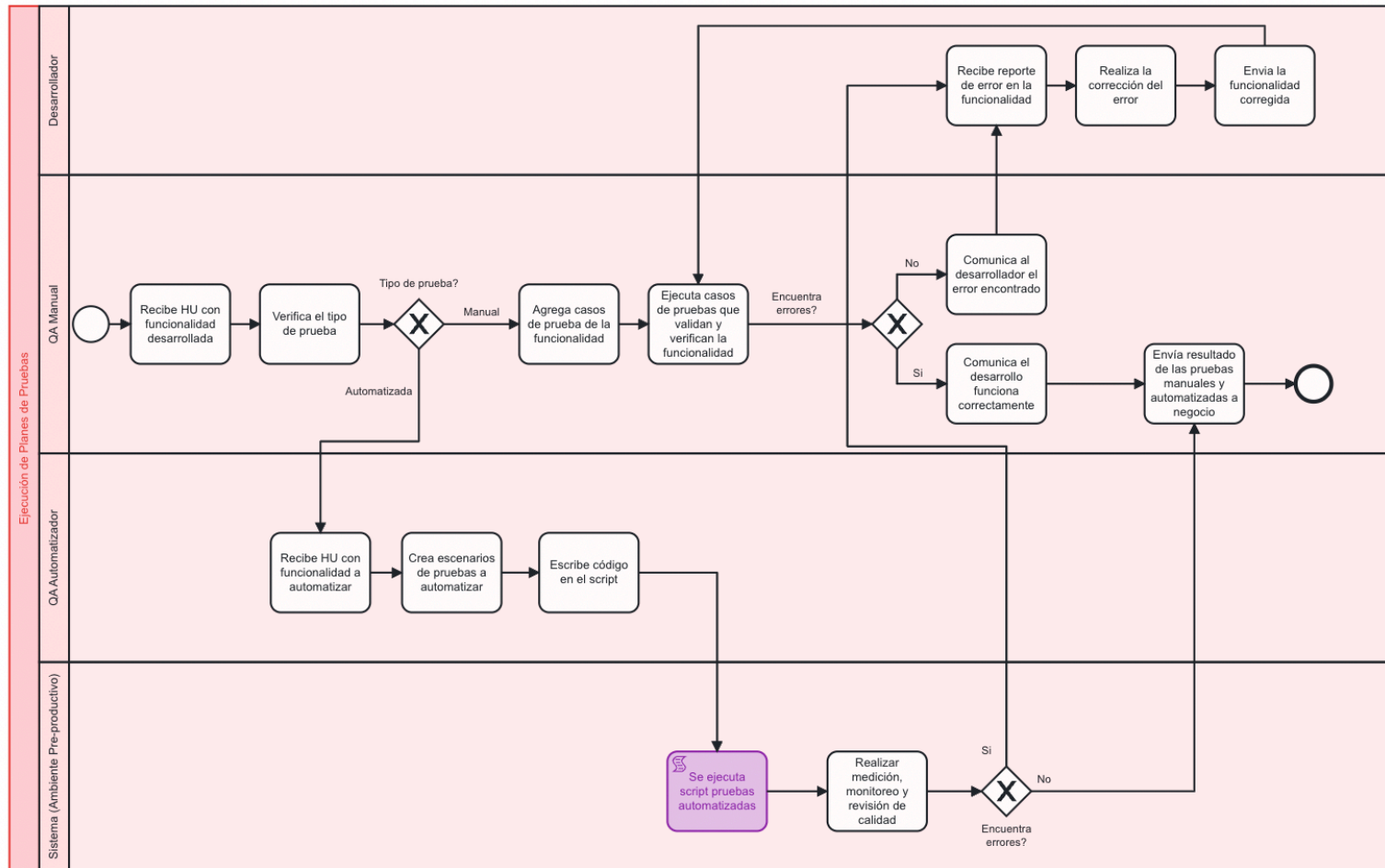


Figura 48: Diagrama BPMN para el proceso “Ejecución de planes de pruebas”.

Fuente: Elaboración propia

4.2 Diseño de Lógica de Negocios

A continuación, se pretende presentar la lógica de negocio para el rediseño del proceso presentado anteriormente.

4.2.1.1 Identificación y análisis de defectos en la plataforma

Esta lógica de negocio tiene como objetivo ser apoyo al área de “QA” proporcionándole información necesaria que le permita poder tomar mejores decisiones respecto a la planificación de aseguramiento de calidad y tener un mayor conocimiento del estado de calidad del producto.

Por lo tanto, para poder llevar a cabo esta lógica de negocio se propone la creación de scripts de pruebas automatizadas, que serán creados en la herramienta llamada “Katalon Studio”. Esta herramienta permite poder crear pruebas automatizadas mediante scripts que suplantan las acciones que los clientes realizan con su mouse y teclado en la interfaz de usuario de la plataforma, por lo tanto, es muy factible para encontrar defectos, antes de que sea publicado en ambiente productivo. Estas pruebas automatizadas tienen como salida el resultado del estado de los casos de prueba, que brindan información si los casos de pruebas son aprobados o fallidos.

Para poder generar valor de los resultados que brindan estas pruebas se realiza una integración con la herramienta “Katalon TestOps”, en donde acá se recopila y centraliza toda la información de los resultados de las pruebas y permite mediante un Dashboard poder interpretar los resultados brindando información como la siguiente, que permita al área QA una mejor toma de decisiones:

- Estadísticas de pruebas ya sea de forma diaria/ semanal/ mensual de la cantidad de pruebas con su estado aprobado y fallido, lo que permitirá conocer:
 - ¿Se realizan pruebas con frecuencia todos los días/semana/mes?
 - ¿Cuál es el estado de las ejecuciones de las pruebas si son ejecuciones con defectos o libre de defectos?
 - ¿Cuánto tiempo lleva ejecutar las pruebas?
- Comprobar la duración total de las ejecuciones y compararla con un historial de ejecuciones de días/meses anteriores. Esto permite conocer día en que las pruebas se ejecutan más lentamente y días en que se ejecutan más rápidamente.

- Supervisar la productividad del progreso en el desarrollo de los scripts, permite conocer el tiempo que conlleva escribir scripts.
- Conocer el número total de los defectos en cada versión de la plataforma.
- Verificar si el equipo ha aumentado o disminuido las actividades de prueba cada mes, para visualizar el rendimiento del equipo.
- Conocer la calidad de los casos de pruebas, que se interpreta que un caso que falle contantemente en las ejecuciones significa que la calidad de los casos de prueba es mala.
- Conocer el porcentaje de ejecuciones de los casos de prueba en distintas plataformas, así como en distintos navegadores y en distintos SO.
- Observar la cobertura de los requisitos de los casos de pruebas en sus ejecuciones.

Para generar un mayor valor al script se escribirá en lenguaje Gherkin, que es un lenguaje que utiliza una sintaxis básica para la descripción de pruebas para que puedan ser comprendidas tanto por personas con conocimientos técnicos, así como los PO/BA que son personas de negocio.

Como se mencionó anteriormente se propone la creación de scripts que se diseñaran de acuerdo a los requerimientos del negocio. Los requerimientos estarán basados en las funcionalidades que contiene la plataforma. A continuación, se describen los pasos. Para poder ejemplificar se utilizará el requerimiento de *creación de un colaborador por importadores en la plataforma*, se elige este flujo debido a que es fundamental ya que son los colaboradores a los que se les registra su información por lo tanto no debe contener errores ya que sino la plataforma no cumpliría con su objetivo.

- El primer paso, consistirá en reunirse con la persona encargada del producto (PO/BA) para definir cuáles son los flujos críticos de la plataforma, para este ejemplo definimos el flujo de creación de un colaborador por importadores en la plataforma, ya que permite agregar más de 1 colaborador a la vez.
- Para poder crear el script es necesario contar con los datos para poder ejecutar la prueba. Para este escenario se hará uso de un archivo XLS, permitiendo agregar más un colaborador a la vez. Este archivo tiene campos como el nombre de empleado, tipo de pago, forma de pago entre otros como se puede observar en la figura 48.
- Definido el flujo y con la data necesaria se procede a la creación de script en la herramienta Katalon Studio debido a que se realizara mediante lenguaje Gherkin entonces la herramienta genera una carpeta llamada "include" que

contiene dos carpetas llamadas “feature” y scripts”. La carpeta feature contiene la descripción a un alto nivel de las funciones de la plataforma en donde se agruparán los escenarios que contienen las reglas de negocio. Se seguirá el patrón GIVEN, WHEN, THEN para escribir los escenarios que contiene la funcionalidad. En donde un escenario representa una posible acción que el cliente pueda realizar al usar la plataforma, por lo que en estas pruebas se pretende enfocarse en los escenarios en los que la plataforma podría desencadenar un defecto.

- En la carpeta scripts, tendrá el código de programación para que la herramienta pueda ejecutar las acciones.

	A	B	C	D	E	F	G	H	I	J
1	Libro: Creación de Colaborador									
2										
3										
4										
5										
6	2022-12-01	262			Juanito	Perez			Cheque	
7	2022-12-01	88	8987867665	Corriente	Jaime	Arriagada		Mensual	Transferencia Bancaria	
8	2022-12-01	312			Lina	Varoli			Servipag	
9	2022-12-01	313			Leah	Varina			Servipag	
10	2022-12-01	348			Mike	Tyson			Cheque	
11	2022-12-01	323			Patricio	Campos		Mensual	Servipag	
12	2022-12-01	132	12018844	Vista	Viviana Carmen	Bustamante			Transferencia Bancaria	
13	2022-12-01	101			Ignacio	Diaz			No Generar Pago	
14	2022-12-01	364	12018844	Vista	Viviana Carmen	Bustamante			Transferencia Bancaria	
15	2022-12-01	80			Sofia	Crespo			Cheque	
16	2022-12-01	14	18325464	Vista	Keyla	Correa			Transferencia Bancaria	
17	2022-12-01	18	15020063961	Corriente	Javiera María	Salazar			Transferencia Bancaria	
18	2022-12-01	372	123456	Vista	Trinidad	Ruiz-Tagle			Transferencia Bancaria	
19	2022-12-01	368			Mike	Tyson			Cheque	
20	2022-12-01	84			Paz	Bascuñan			Vale Vista	
21	2022-12-01	120	2450003	Corriente	Antonia Emilia	Camus			Transferencia Bancaria	
22	2022-12-01	258	64788367	Corriente	Alvaro	Carrasco			Transferencia Bancaria	
23	2022-12-01	105	12356789	Corriente	María Jose	Castro			Transferencia Bancaria	
24	2022-12-01	28	10740069610	Corriente	Carola	Cea			Transferencia Bancaria	
25	2022-12-01	116			Nico	Celedon			No Generar Pago	
26	2022-12-01	115			Nicolas	Celedon			No Generar Pago	
27	2022-12-01	24	101019207	Vista	Carlos	Cerda			Transferencia Bancaria	
28	2022-12-01	37	73067200	Corriente	Cristian	Cerda			Transferencia Bancaria	
29	2022-12-01	353			Nathalie	Cole			Servipag	
30	2022-12-01	350			Vivian	Cole			Servipag	
31	2022-12-01	309			Karen	Conley			Cheque	
32	2022-12-01	231	19748301	Vista	Camila Francisca	Contreras			Transferencia Bancaria	
33	2022-12-01	113			Valentina	Contreras			No Generar Pago	
34	2022-12-01	233			Benjamin	Cornejo			Cheque	
35	2022-12-01	57	7754914	Corriente	Camila Andrea	Correa			Transferencia Bancaria	

Figura 48: Datos para crear script

Fuente: Elaboración propia

Luego, de tener la estructura de nuestras carpetas se procede a escribir los distintos escenarios basados en el flujo crítico que se está trabajando, en este caso sería la creación de un colaborador en la plataforma. Supongamos los distintos escenarios:

- Creación correcta de un colaborador: En este escenario se pretende crear el flujo correcto de lo que sería la creación de un colaborador.
- Creación de un colaborador existente: En este escenario se debe validar que si el colaborador existente, no realice la creación del colaborador.

- Validar campos obligatorios en la creación del colaborador: En este escenario se debe validar que los campos obligatorios propuestos en los CA se cumplan, de lo contrario alertar que se encontró un defecto. Los campos que se deben validar serán los siguientes: Nombre, Apellido, Rut, Teléfono, País de nacimiento, Sexo, Forma de Pago y Fecha de ingreso.
- Validar tipo de dato de los campos, por ejemplo, en teléfono no permita números, en sexo no permita caracteres.
- Validar permita subir foto del colaborador y si no lo hace generar una alerta de defecto.
- Validar que en los campos de selección solo permita la selección de un solo valor.
- Validar se asigne la previsión correctamente.

Se puede observar en la figura 49, el ejemplo de cómo se escribe el caso de prueba en la feature siguiendo el lenguaje Gherkin.

```

1  @Importador
2
3  Feature: Crear colaborador mediante importador, asignar prevision y empleo
4
5  @EmpleadoImportador @smoke_test
6  Scenario Outline: Colaborador importadores
7  Given Queremos crear colaborador, asignar prevision y empleo mediante importadores '<BuscarEmpleadoImportador>'
8  When Seleccionamos empleado en importadores "Colaborador.xls"
9  When Seleccionamos Trabajo en importadores '<BuscarTrabajoImportadores>' seleccionamos archivo "Trabajos.xls"
10 Then Se valida que se haya creado con exito la ficha del trabajador '<NombreTrabajadores>'
11
12 Examples:
13 | BuscarColaboradorImportador | BuscarTrabajoImportador | NombreTrabajador |
14 | Colaborador                 | trabajos                 | Importador         |
15

```

Figura 49: Feature de Katalon Studio
Fuente: Katalon Studio

Y en la figura 49, se ve la estructura de cómo se realiza la escritura de código del script, acá es donde se hace la validación del resultado esperado con el resultado obtenido y comprueba si el caso de prueba ha fallado o ha sido correcto.

```
*CreacionEmpleadoPrevisionTrabajo.feature  CreacionEmpleadoImportadores.groovy  empleados.feature
39 import com.kms.katalon.core.webui.exception.WebElementNotFoundException
40 import co.buk.businesslogic.BLEmpleadoImportador
41
42 import cucumber.api.java.en.And
43 import cucumber.api.java.en.Given
44 import cucumber.api.java.en.Then
45 import cucumber.api.java.en.When
46
47
48
49 class CreacionEmpleadoImportadores {
50     @Given("Queremos crear empleados, asignar prevision y empleo mediante importadores {string}")
51     public void Queremos_crear_empleados_asignar_prevision_y_empleo_mediante_importadores(String BuscarEmpleadoImportador) {
52         String url = GlobalVariable.Url ;
53         WebUI.navigateToUrl( GlobalVariable.Url + 'data_load');
54         WebUI.takeScreenshot();
55         BLEmpleadoImportador.NavegarMenuImportadores(BuscarEmpleadoImportador);
56     }
57
58     @When("Seleccionamos empleado en importadores {string}")
59     public void Seleccionamos_empleado_en_importadores(String nombreArchivoEntrada) {
60         WebUI.takeScreenshot();
61         BLEmpleadoImportador.CrearEmpleadoNuevo(nombreArchivoEntrada);
62     }
63
64     @When("Seleccionamos Trabajo en importadores {string} seleccionamos archivo {string}")
65     public void Seleccionamos_Trabajo_en_importadores(String BuscarTrabajoImportador, String nombreArchivoEntrada) {
66         WebUI.takeScreenshot();
67         BLEmpleadoImportador.CrearTrabajoNuevo(BuscarTrabajoImportador, nombreArchivoEntrada);
68     }
69
70     @Then("Se valida que se haya creado con exito la ficha del trabajador {string}")
71     public void Se_valida_que_se_haya_creado_con_exito_la_ficha_del_trabajador(String NombreTrabajador) {
72
73         String url = GlobalVariable.Url ;
74         WebUI.navigateToUrl( GlobalVariable.Url + 'employees');
75         WebUI.takeScreenshot();
76         BLEmpleadoImportador.EliminarEmpleado(NombreTrabajador);
77     }
78 }
```

Figura 50: Script Katalon Studio, Fuente: Katalon Studio

Ahora, al ejecutar el script que contiene estos casos de prueba, si la herramienta encontrase un defecto, se generará un reporte que contendrá que funcionalidades fallaron y adjuntará imágenes o grabación que permitirá brindar información precisa para poder solucionar el defecto, el cual el QA será el encargado de brindar esta información al desarrollador.

Al realizar la integración con “Katalon TestOps”, toma el resultado de la ejecución de los casos de prueba y mediante un Dashboard centraliza y muestra resultados para poder realizar un análisis de acuerdo a los resultados obtenidos. Y proveen apoyo al área QA.

Caso Exitoso

The screenshot displays the Katalon TestOps interface for a successful test run. The top navigation bar includes 'smoke-test_integración', 'Dashboard', 'Planning', 'Tests', 'Executions', 'Reports', 'Visual Testing', and 'Configurations'. The main header shows 'Test Result' for 'Empleado importadores' with a 'Passed' status. Below this, there are three screenshots of the test execution, each with a file name and size: '1678980029715.png (105.76 KB)', '1678980074635.png (308.99 KB)', and '1678980134677.png (258.3 KB)'. A 'Failed Assertions' section is empty, showing 'There is no data to display.' The 'Logs' section shows a list of steps, all marked as successful (green checkmarks): 'Start Test Case : SCENARIO [CO-C2011] Empleado importadores', '[TEST_CASE][PASSED] - Start Test Case : SCENARIO [CO-C2011] Empleado importadores: SCENARIO [CO-C2011] Empleado importadores', 'Queremos crear empleados, asignar prevision y empleo mediante importadores 'Empleados'', 'Seleccionamos empleado en importadores "Empleados.xls"', 'Seleccionamos Trabajo en importadores "trabajos" seleccionamos archivo "Trabajos.xls"', and 'Se valida que se haya creado con éxito la ficha del trabajador "Importador'.

Caso Fallido

The screenshot displays the Katalon TestOps interface for a failed test run. The top navigation bar is the same as in the previous screenshot. The main header shows 'Test Result' for 'Empleado importadores' with a 'Failed' status. Below this, there are three screenshots of the test execution, each with a file name and size: '1678980029715.png (105.76 KB)', '1678980074635.png (308.99 KB)', and '1678980134677.png (258.3 KB)'. A 'Failed Assertions' section is empty, showing 'There is no data to display.' The 'Logs' section shows a list of steps, with the first step marked as failed (red X): 'Queremos crear empleados, asignar prevision y empleo mediante importadores 'Empleados''. The log details the failure: '[TEST_STEP][FAILED] - Queremos crear empleados, asignar prevision y empleo mediante importadores 'Empleados': Queremos crear empleados, asignar prevision y empleo mediante importadores 'Empleados' FAILED. Reason: com.kms.katalon.core.exception.StepFailedException: Unable to set text 'Empleados' of object '' at com.kms.katalon.core.webui.keyword.internal.WebUIKeywordMain.stepFailed(WebUIKeywordMain.groovy:64) at com.kms.katalon.core.webui.keyword.internal.WebUIKeywordMain.runKeyword(WebUIKeywordMain.groovy:26) at com.kms.katalon.core.webui.keyword.builtIn.SetTextKeyword.setText(SetTextKeyword.groovy:66) at com.kms.katalon.core.webui.keyword.builtIn.SetTextKeyword.execute(SetTextKeyword.groovy:37) at com.kms.katalon.core.keyword.internal.KeywordExecutor.executeKeywordForPlatform(KeywordExecutor.groovy:74) at com.kms.katalon.core.webui.keyword.WebUIBuiltInKeywords.setText(WebUIBuiltInKeywords.groovy:988) at co.buk.businesslogic.BLEmpleadoImportador.NavegarMenuImportadores(BLEmpleadoImportador.java:11) at co.buk.businesslogic.BLEmpleadoImportador.NavegarMenuImportadores.call(Unknown Source) at co.buk.stepdefinition.CreacionEmpleadoImportadores.Queremos_crear_empleados_asignar_prevision_y_empleo_mediante_importadores(CreacionEmpleadoImportadores.groovy:24) at *.Queremos crear empleados, asignar prevision y empleo mediante importadores 'Empleados' (/home/ubuntu/workspace/buk-katalon/Include/features/Colombia/Importadores/CreacionEmpleadoPrevisionTrabajo.feature:24) Caused by: com.kms.katalon.core.webui.exception.WebElementNotFoundException: Web element with id: '' located by '//input[@type='search'] [1]' not found at com.kms.katalon.core.webui.common.WebUICommonHelper.findElement(WebUICommonHelper.java:1378) at com.kms.katalon.core.webui.keyword.internal.WebUIAbstractKeyword.findElement(WebUIAbstractKeyword.groovy:27) at com.kms.katalon.core.webui.keyword.internal.WebUIAbstractKeyword.findElement(WebUIAbstractKeyword.groovy:26)'. The 'Failed Assertions' section is empty, showing 'There is no data to display.'

Figura 51: Caso exitoso y fallido del caso de prueba
Fuente: Katalon TestOps

CAPÍTULO 5: PROPUESTA DE APOYO TECNOLÓGICO

5.1 Especificación de requerimientos

La herramienta que se especifica a continuación cumple la función de apoyo tecnológico a lo que será el rediseño que contiene la lógica de negocio, que tiene como objetivo detectar y localizar defectos en la plataforma de forma preventiva antes que sean encontrados por los clientes. La herramienta seleccionada a utilizar para este proyecto es “Katalon Studio”, debido a que cumple con los requerimientos funcionales y no funcionales que necesita el proyecto para poder llevarse a cabo.

5.1.1 Requerimientos No Funcionales

- La herramienta a utilizar para la automatización de los scripts debe:
 - Ser gratuita.
 - Puedan ser escritos en lenguaje java.
 - Soporte lenguaje Gherkin.
 - Ser compatible con distintos navegadores webs (Chrome, safari, etc.).
 - Ser fácil de usar.
 - Tener amplia documentación.
 - Se pueda realizar integración CI/CD con Jenkins.

5.1.2 Requerimientos Funcionales

- La herramienta debe crear scripts automatizados basándose en los flujos de negocio críticos de la plataforma.
- La herramienta de automatización debe permitir al QA seleccionar uno o más scripts automatizados.
- La herramienta debe permitir poder programar la fecha y hora de ejecución de los scripts.
- La herramienta debe permitir poder agregar y modificar datos de entrada de los scripts automatizados.
- La herramienta debe generar reportes que brinden los resultados de las pruebas automatizadas con su evidencia y que pueden ser imágenes o videos.
- La herramienta deba poder alertar cuando una prueba automatizada del script haya fallado.
- Los scripts puedan ser guardados en un repositorio como Gitlab, Github, etc.

5.2 Arquitectura lógica

A nivel funcional, la herramienta consiste en la ejecución de scripts que contienen pruebas automatizadas que realizan una comparación entre los resultados obtenidos y los esperados de la plataforma, en donde la herramienta al ejecutar el script, levanta un navegador e inicia a simular las acciones que realiza un cliente en la plataforma. Si la herramienta encuentra diferencia en los resultados obtenidos y esperados entonces procede a alertar al QA, brindándole evidencia ya sea en forma de imagen o video, para la pronta identificación y detección del defecto.

Para poder cumplir con lo anterior, se propone una arquitectura de tres capas: una capa de datos, una capa de lógica del negocio y una capa de presentación, como se puede observar en la figura 52.

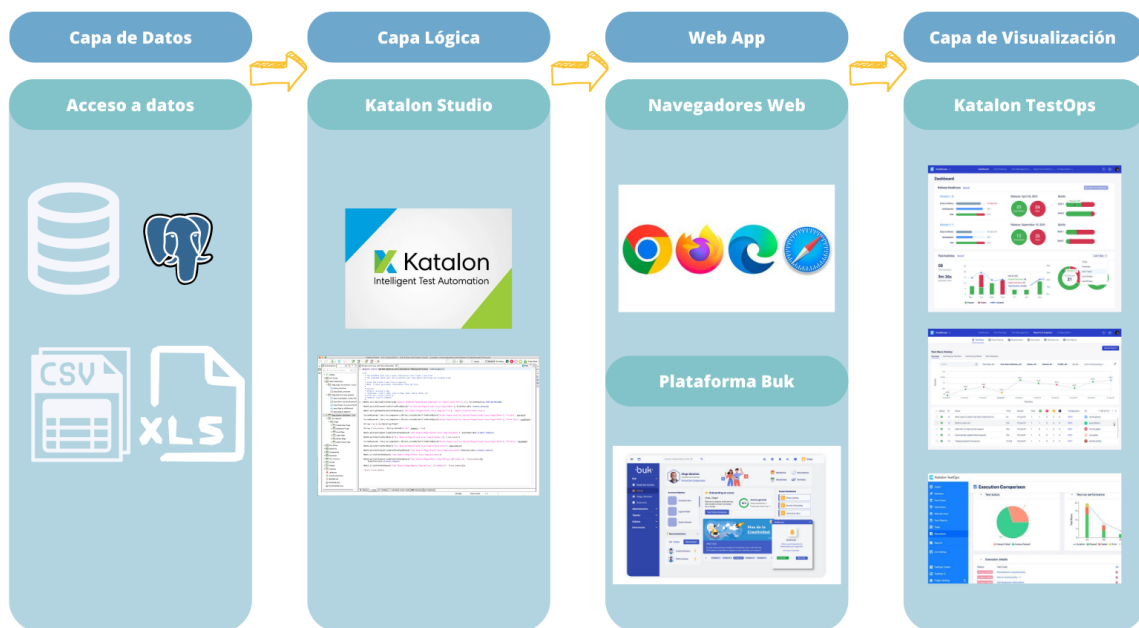


Figura 52: Arquitectura de capas propuesta para el rediseño.

Fuente: Elaboración propia

5.2.1 Capa de datos:

Esta capa contiene todo un conjunto de información perteneciente al contexto de negocio de la empresa. Estos conjuntos de datos se extraen mediante distintas fuentes necesarias como: Base de datos, archivos CSV, archivos Excel, etc.

Para poder llevar a cabo la ejecución de las pruebas automatizados es necesario utilizar datos de prueba. La herramienta Katalon Studio permite poder importar archivos de datos en distintos formatos como CSV, XLS, Bases de datos, entre otros. Para modo de ejemplo se obtendrán datos desde un archivo XLS, lo que permitirá poder ejecutar un escenario de prueba al que se le agregará “Saldo de vacaciones” a un colaborador, realizando los siguiendo los siguientes pasos:

- **Paso 1 y 2:** Se deberá importar el archivo de tipo XLS que contendrá los datos para poder ejecutar el escenario de prueba.
- **Paso 3:** Se visualizan los datos en la herramienta de Katalon Studio.
- **Paso 4:** En el documento feature creado para la elaboración del escenario de prueba, se hace el llamado del archivo importado.

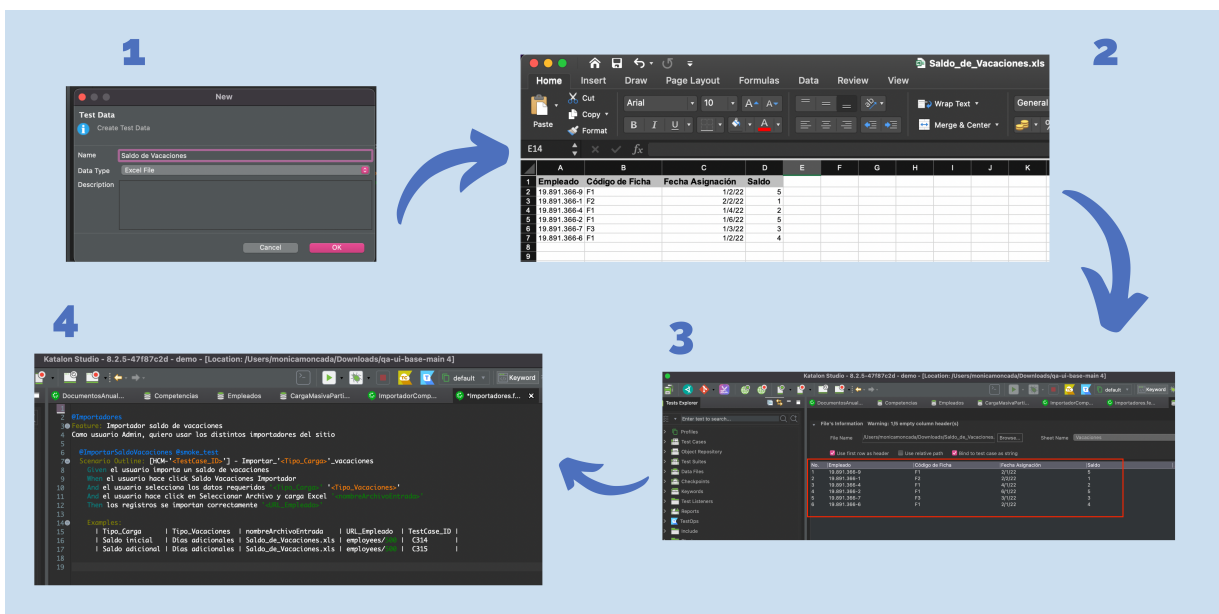


Figura 53: Capa de datos – Importación de datos de un archivo XLS en Katalon Studio.

Fuente: Katalon Studio

El poder obtener data de distintas fuentes facilita el poder crear pruebas con datos muy similares a los de un cliente real al momento de ejecutar el script.

5.2.2 Capa lógica de negocio:

Es esta capa la que permite que las pruebas automatizadas puedan llevarse a cabo. Por lo que elegir la herramienta que cumpliera con los requerimientos era esencial. Katalon Studio permite al QA Automatizador como se observa en la figura 54, la facilidad de:

- Agendar las ejecuciones en la fecha y hora conveniente, además de realizar ejecuciones paralelas para disminuir tiempo de ejecución.
- Escribir los casos de prueba (testcase) en lenguaje gherkin dentro del archivo feature.
- Poder agrupar y seleccionar los testcase en testsuite para poder ejecutarlos de manera organizada y rápida.
- Ejecutar los testsuite en distintos navegadores web como (Chrome, Mozilla, etc).
- La herramienta notifica mediante correo electronico cuando alguna ejecución fallo y los casos de prueba fallidos.
- Obtener informes avanzados que permitan conocer la causa raiz de la falla del caso de prueba.

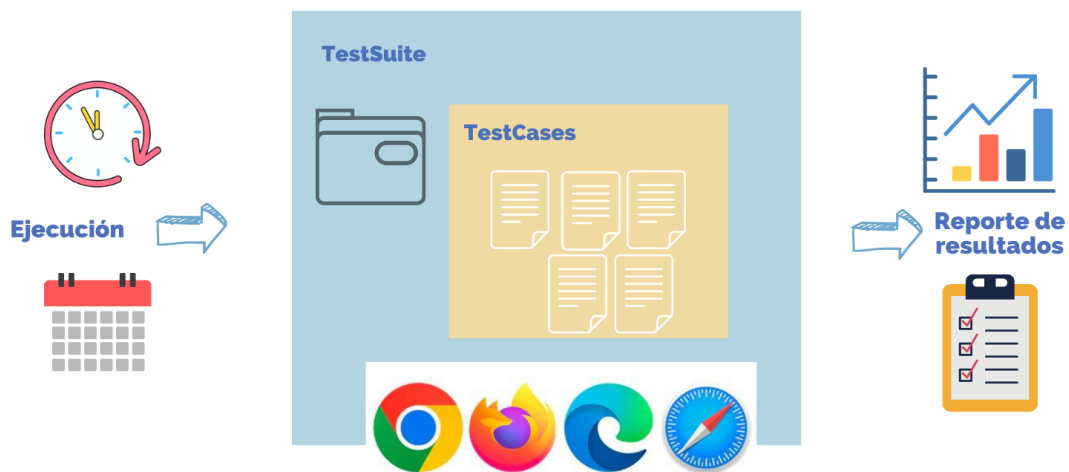


Figura 54: Ejecución de pruebas automatizadas en Katalon Studio.

Fuente: Elaboración propia

5.2.3 Estructura de un caso de prueba (Testcase) en Katalon Studio:

Para poder crear los casos de prueba automatizados, en la herramienta de Katalon se deberán crear dos archivos, el “**Feature**” y el “**Step Definition**”.

En el “**Feature**” se escribirán los casos de prueba en forma de escenarios elaborados en lenguaje Gherkin para que puedan ser comprendidos por negocio como se puede observar en la figura 53.

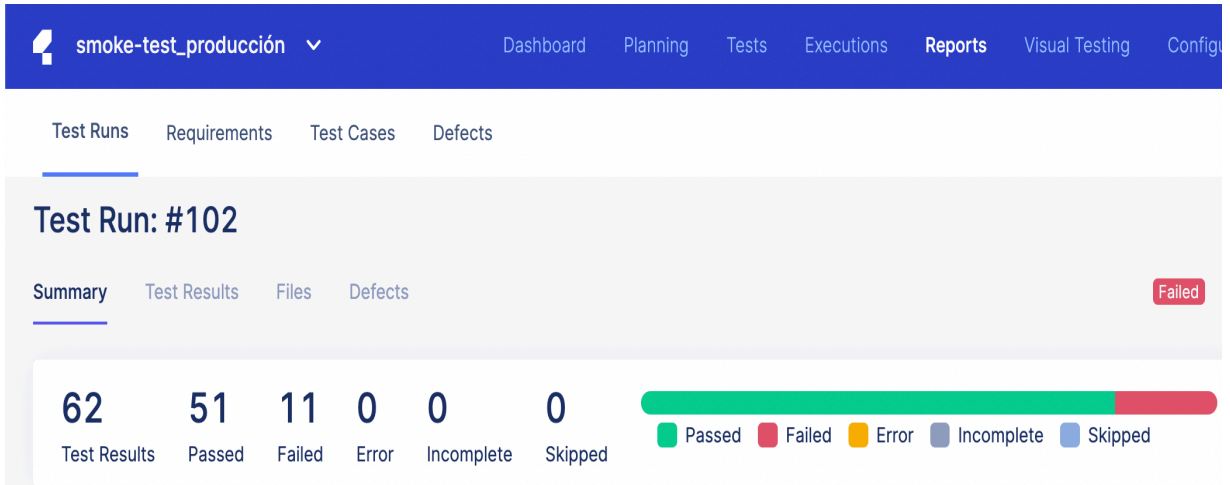


Figura 56: Resultados de ejecuciones de testcase.
Fuente: Katalon TestOps

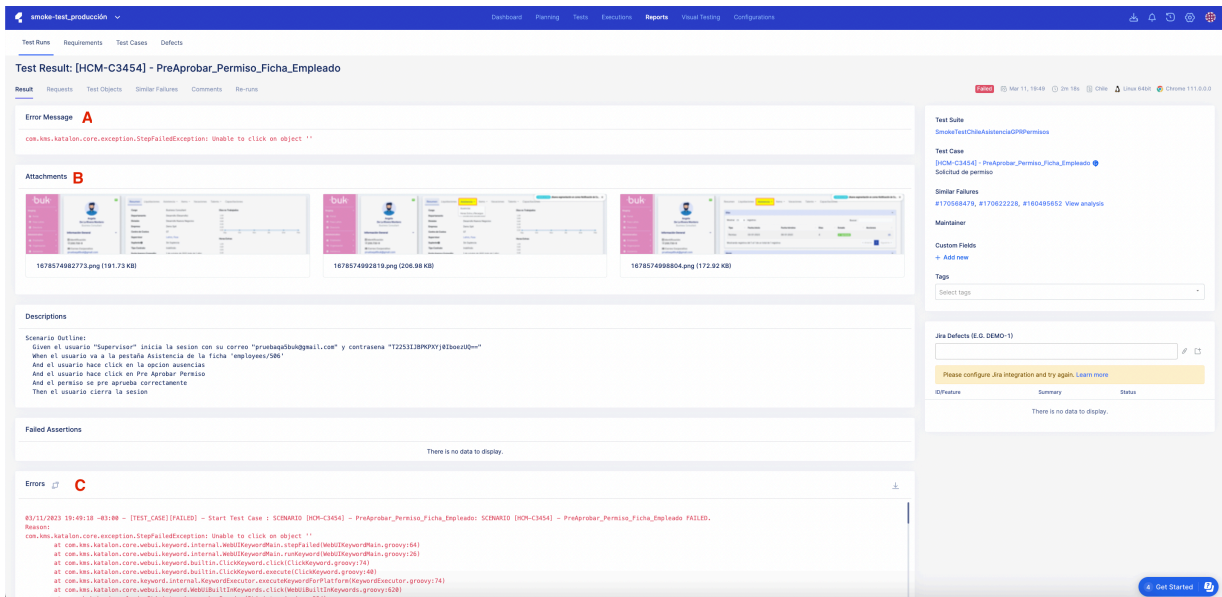


Figura 57: Estructura de un testcase fallido.
Fuente: Katalon TestOps

5.3 Herramientas

5.3.1 Katalon Studio

Es una herramienta de automatización que permite ejecutar scripts tanto para sitios web, móviles, api, etc.

Está basada en Selenium y Appium que son framework de automatización.

La estructura que conforma Katalon permite poder organizar las pruebas en test suites (conjunto de pruebas) o solamente en test cases (casos de pruebas), además de poder ejecutar el script en distintos navegadores web y sistemas operativos.

También, permite poder escribir pruebas en formato BDD, que es lenguaje Gherkin que facilita a las personas no técnicas poder entender las pruebas automatizadas. Además, Katalon nos permite poder hacer las validaciones referentes a los resultados esperados con los resultados obtenidos, mediante los comandos “Verify” que se pueden observar en la figura. También permite poder agregar data desde distintas fuentes como base de datos, csv o archivos Excel.

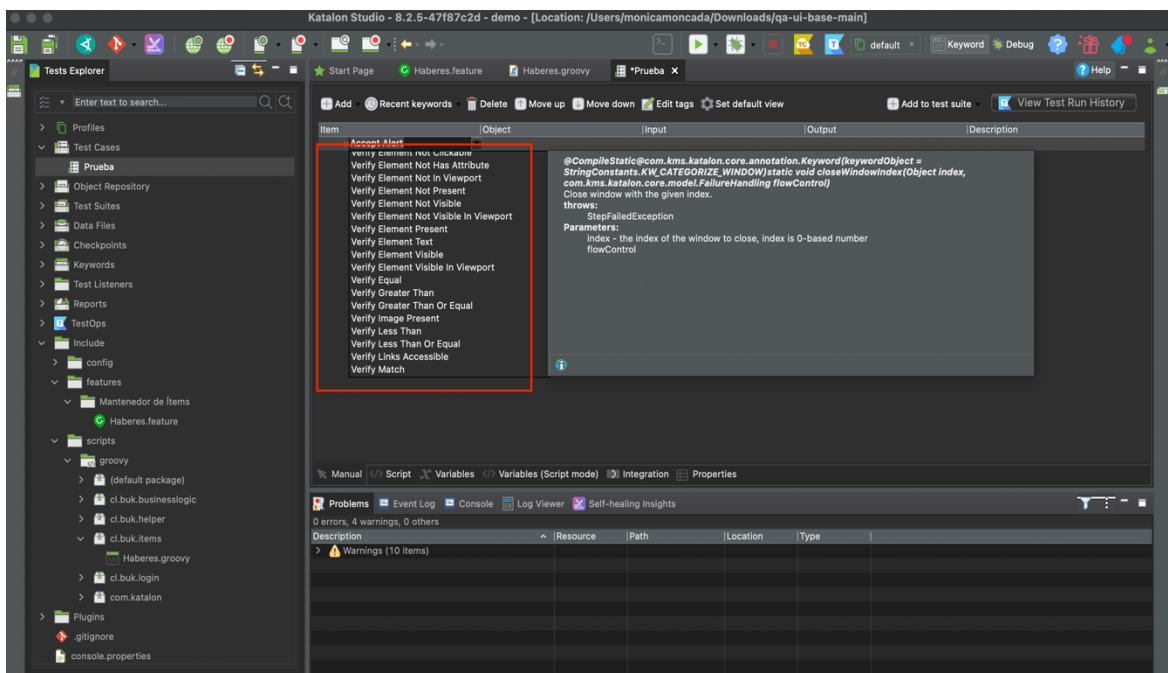


Figura 58: Estructura de Katalon Studio IDE.

Fuente: Katalon Studio IDE

5.3.2 Katalon TestOps

Esta herramienta que se integra a Katalon Studio, permite coordinar varias actividades como planificación, programación y ejecución de los casos de pruebas, recopilar datos de distintas fuentes como Katalon Studio, brindar informes detallados realizando análisis de los datos obtenidos por Katalon Studio.

Brinda un Dashboard del cual podemos obtener la siguiente información:

- **Actividades de pruebas:** Permite conocer si el producto funciona correctamente, brinda datos como el total de ejecuciones de pruebas y el resultado de las ejecuciones de pruebas.

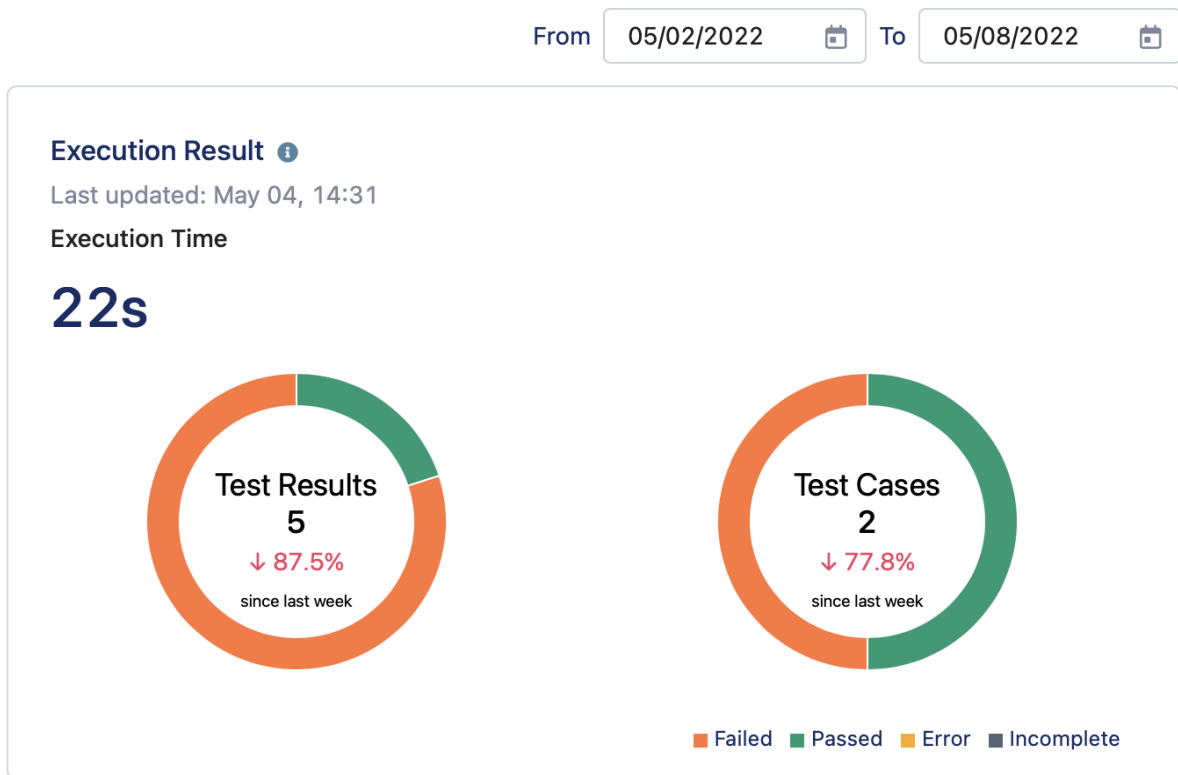


Figura 59: Actividades de pruebas Dashboard.

Fuente: Katalon TestOps

- **Preparación de lanzamiento:** En esta sección brinda información del estado de lanzamientos previos, recordatorios de fecha de entrega, productividad del desarrollo de pruebas y cantidad de defectos encontrados.

Release Readiness ?

Enable Jira Integration



Figura 60: Preparación de lanzamientos Dashboard.

Fuente: Katalon TestOps

- **Productividad:** Permite monitorear la cantidad de casos de prueba que se han creado en el desarrollo del producto, realizar comparaciones entre meses, etc.

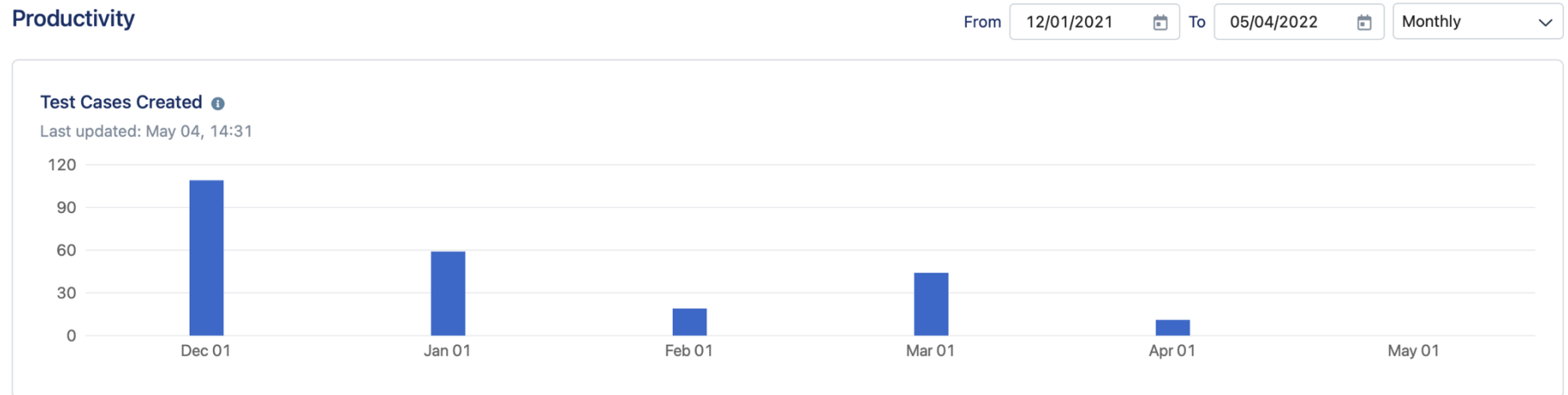


Figura 61: Productividad de pruebas Dashboard.

Fuente: Katalon TestOps

- **Calidad:** Permite monitorear la calidad de los casos de prueba en base al historial de la ejecución de casos de pruebas.

Quality Beta

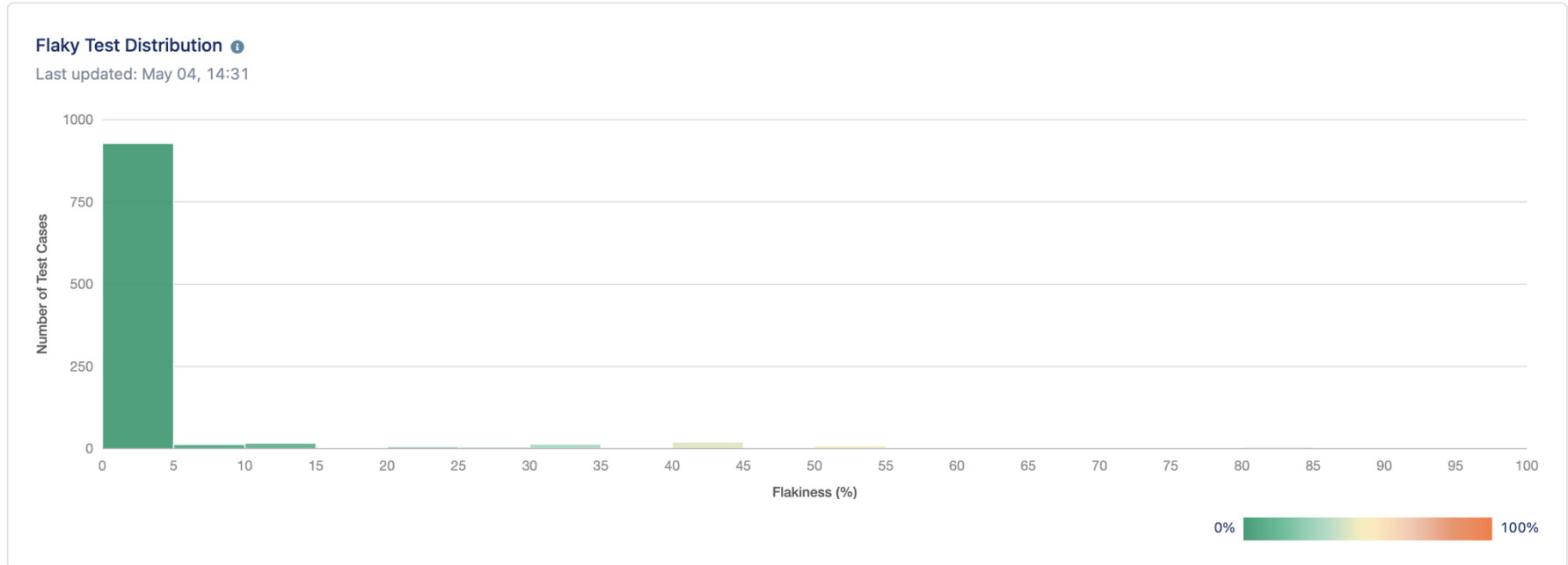


Figura 62: Calidad de pruebas Dashboard.

Fuente: Katalon TestOps

- **Cobertura de plataforma:** Pretende mostrar la cobertura de los casos de prueba de acuerdo a los distintos navegadores web y sistemas operativos.

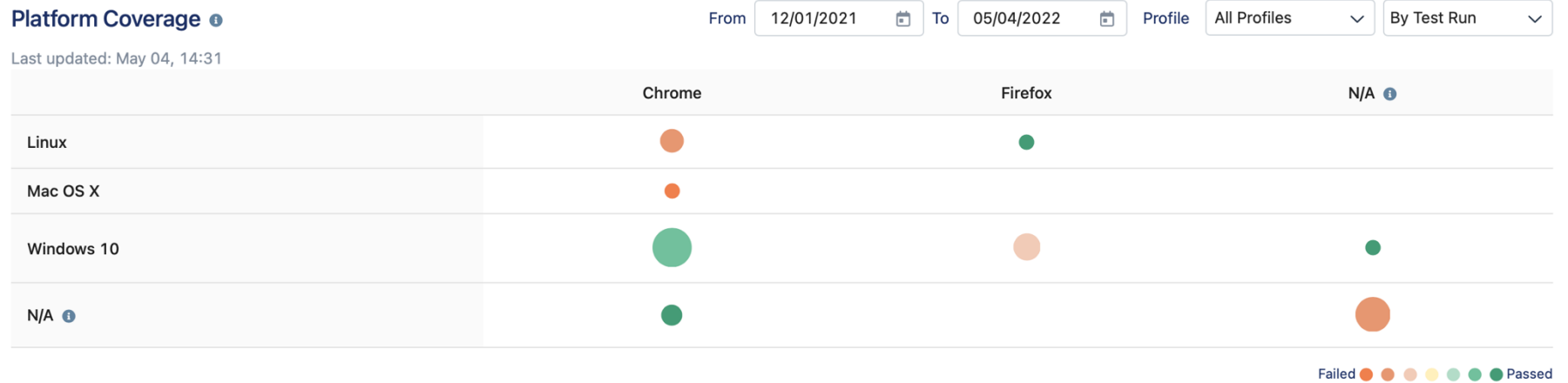


Figura 63: Cobertura de plataforma Dashboard.

Fuente: Katalon TestOps

- **Cobertura de requisitos:** Brinda información que permite conocer la cobertura de requerimientos que conforman el producto.

Requirement Coverage 🔍

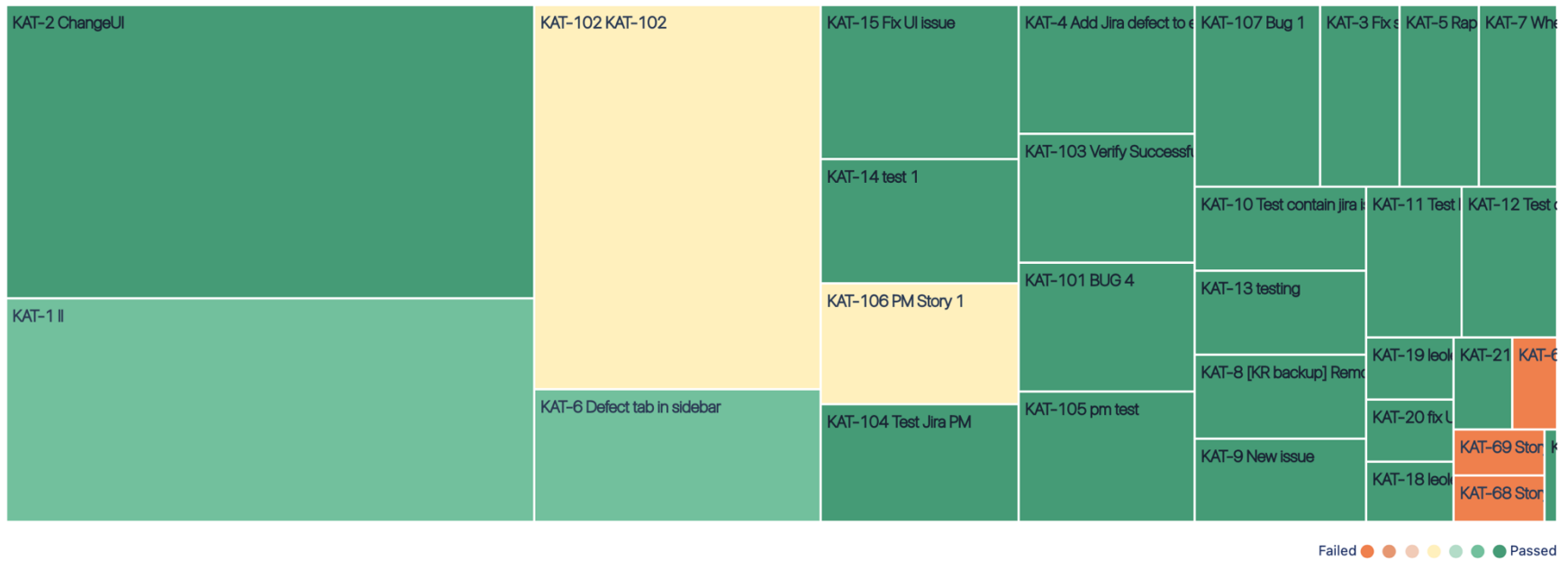


Figura 64: Cobertura de requisitos Dashboard.

Fuente: Katalon TestOps

5.4 Infraestructura Tecnológica

Para poder llevar a cabo el proyecto, se necesitará contar con una infraestructura tecnológica llamada *Integración Continua & Entrega Continua (CI/CD)*.

CI/CD nos permite garantizar que los componentes de software funcionan juntos.

Mediante las siguientes actividades:

- **CI:** es quien detecta cambios en el repositorio de código fuente, realiza análisis de calidad del código fuente, ejecuta pruebas unitarias, genera el pase al ambiente de pruebas (ambiente qa). El repositorio de código fuente será con la herramienta Github.
- **CD:** es el encargado de implementar el software en producción, pero antes realiza la ejecución del script de pruebas automatizadas y si las pruebas detectan un defecto entonces brinda la información al equipo indicándole en que parte de la funcionalidad se produjo el fallo.

Para poder realizar el CI/CD se realizará con la herramienta Jenkins que brinda todas las funcionalidades y la facilidad de poder integrarse con Katalon Studio además de integrar Katalon TestOps que apoya al análisis mediante un Dashboard, sobre los resultados que brinda Katalon Studio.

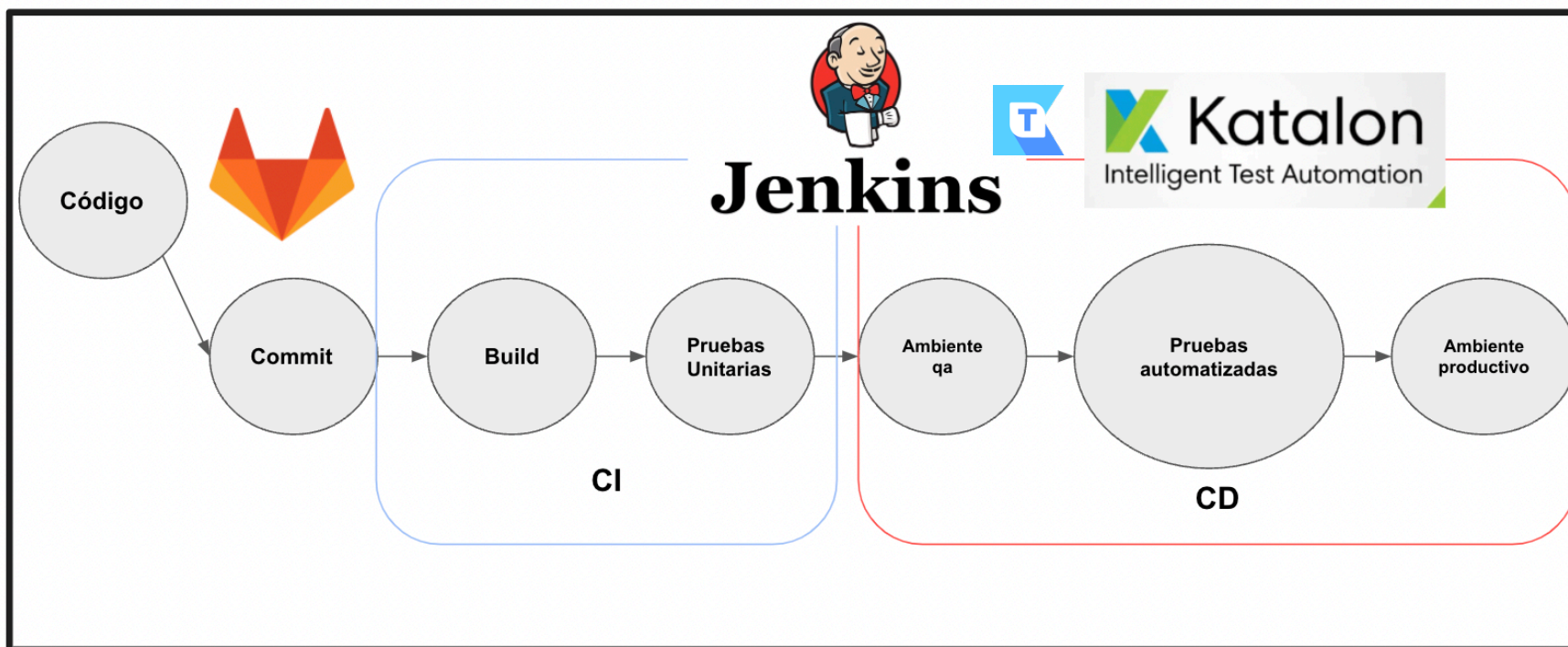


Figura 65: Infraestructura CI/CD propuesta para el rediseño.

Fuente: Elaboración propia

5.5 Prototipo Desarrollado

5.5.1 Flujo critico de negocio seleccionado

Para poder llevar a cabo la creación del prototipo fue necesario reunirse con el Product Owner (PO) de la célula de remuneraciones, siendo esta la célula con la que se iniciara la implementación del proyecto. En la reunión se le explico al PO la importancia de las pruebas automatizadas y que para llevarlas a cabo era necesario realizar el levantamiento de los flujos críticos de negocio, que representan las funcionalidades de la plataforma que exponen la criticidad, por lo tanto, deben no deben presentar ningún error. De la selección de flujos críticos para poder realizar el prototipo se seleccionó la funcionalidad llamada “Procesos”, esta funcionalidad representa una sección de la plataforma que permite segmentar remuneraciones para generar archivos por ejemplo de banco, PREVIRED, liquidaciones, contabilidad entre otros.

5.5.2 Desarrollo de Script de Prueba Automatizada

Al tener el flujo seleccionado que para ejemplificar se trabajara con la funcionalidad de “Procesos”, se procede en Katalon a generar un archivo del tipo feature que permitirá representar el caso de prueba en formato Gherkin que permite sea legible para cualquier persona de negocio.

El caso de prueba tendrá la siguiente estructura:

Scenario Outline: Es palabra clave dentro del lenguaje Gherkin y representa el título del caso de prueba que para este ejemplo será “Ejecutar el proceso de forma correcta. Lo que se espera con este caso de prueba es que se permita probar que distintos procesos se ejecuten correctamente, como lo son PREVIRED, liquidaciones, contabilidad, entre otros.

Luego en las palabras claves Given, When y Then se estructurará de la siguiente manera:

Given el usuario quiere ejecutar los procesos generales del mes de <Mes> en el año <Año>
When ejecuta el proceso <Proceso> de mes < Status_Mes >
And el sistema despliega mensaje informativo ‘alert-succed alert-disimissible’ de ejecución correcta del job
And espera a que el sistema se actualiza
Then el <Proceso> con estado <Status_Mes> termina con resultado <Resultado>”.

Los datos de entrada que tendrá, será mediante las variables
Y se mostrara en Katalon de la siguiente manera: <Proceso>, <Status_Mes>, <Resultado>, <Mes> y <Año>.

En la herramienta Katalon se mostrará cómo se observa en la figura 66.

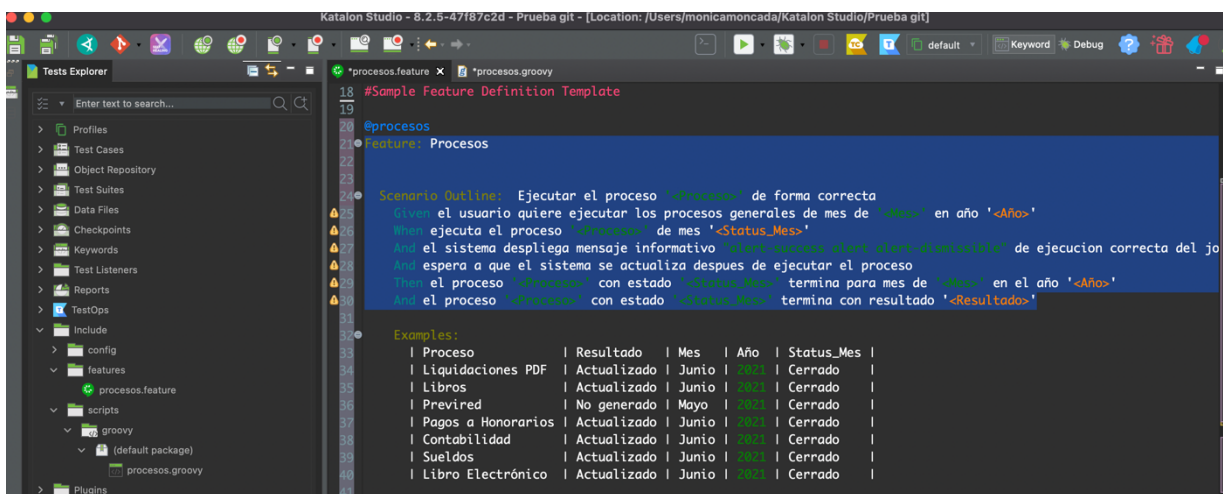


Figura 66: Feature Procesos Katalon Studio.

Fuente: Elaboración propia

Al tener la feature se procede a realizar el script que contendrá la lógica de negocio del proyecto, con ella se pretende que el caso de prueba automatizado valide y verifique que la funcionalidad se encuentra libre de errores por lo tanto funciona correctamente. El script es representado por un conjunto de código de programación que permite se ejecute las acciones de estos pasos a pasos

Se toma cada una de las palabras claves de la feature y se agregan las acciones en forma de código, como se puede observar en la figura 67. El script completo se mostrará en anexos del proyecto.

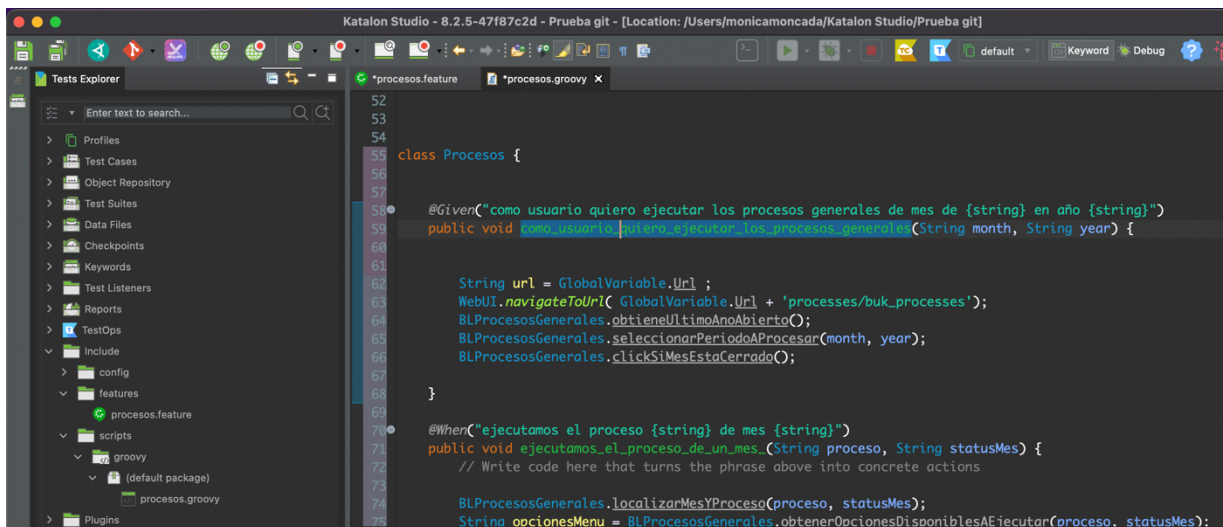


Figura 67: Script Procesos Katalon Studio.

Fuente: Elaboración propia

5.5.3 Dashboard en Katalon TestOps

Al ejecutar el script automatizado en Katalon Studio se obtiene el resultado de la ejecución, pero al integrar con Katalon TestOps obtiene cada uno de los resultados de las ejecuciones y mediante el Dashboard se pueden interpretar los resultados lo que permite conocer el estado de calidad de la plataforma, de las ejecuciones realizadas se han obtenido los siguientes datos:

El historial de pruebas ejecutadas y el estado de las pruebas, lo que permite que el QA pueda validar las pruebas que fallaron y reportarlas a los desarrolladores para que puedan ser solucionadas y el defecto no sea visible para el cliente.

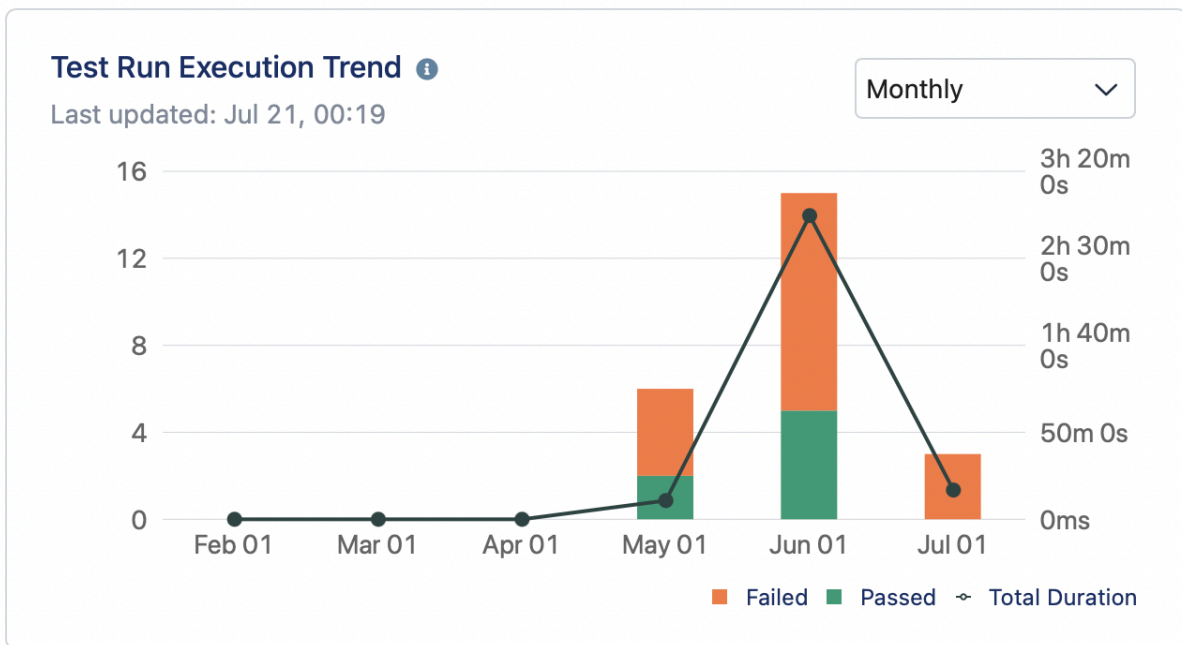


Figura 68: Historial ejecución de pruebas.

Fuente: Katalon TestOps

Conocer el navegador de internet en donde suceden más defectos

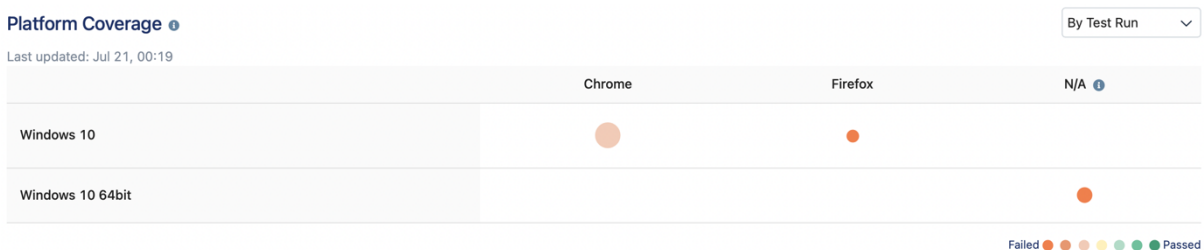


Figura 69: Navegadores con más defectos.

Fuente: Katalon TestOps

Se puede obtener información del caso fallido, brinda información como el navegador en el que fallo, el sistema operativo, el error de mensaje, logs, captura de pantalla en donde sucedió el error.

The screenshot shows the Katalon TestOps interface for a failed test run. The main title is "Test Result: [RMCL-C305] Ejecutar el proceso 'Previred' de forma correcta". The status is "Failed" and it occurred on "Jun 10" at "2m 3s". The error message is: "com.kms.katalon.core.exception.StepFailedException: Actual object 'No generado' and expected object 'Actualizado' are not equal". An attachment shows a screenshot of the application interface with the error highlighted. The right sidebar shows details for the Test Suite "SmokeTestChile - RMCL", Test Case "[RMCL-C305] Ejecutar el proceso 'Previred' de forma correcta", and Similar Failures with IDs #138582723 and #134384390.

Figura 70: Caso de prueba fallido.

Fuente: Katalon TestOps

El objetivo que tiene poder obtener estos resultados es que sean apoyo al QA Lead para poder tomar mejores decisiones tanto en productividad del equipo, como conocer cuáles son las funcionalidades de la plataforma que contiene más defectos, mejorar tiempos de ejecución.

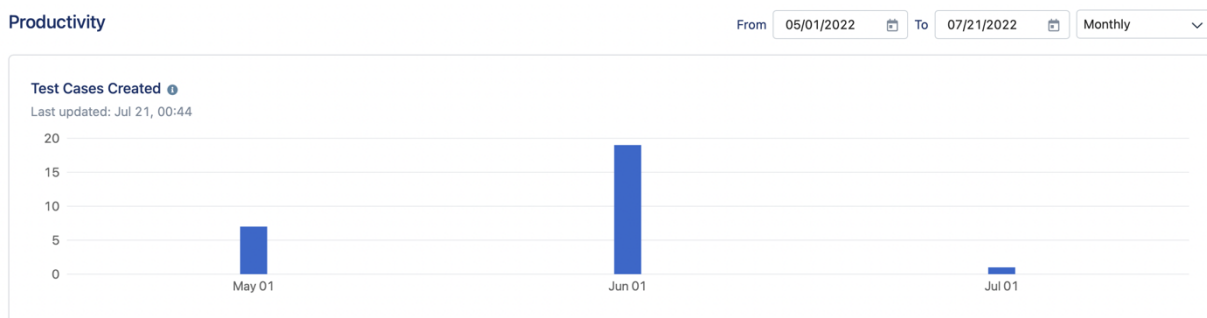
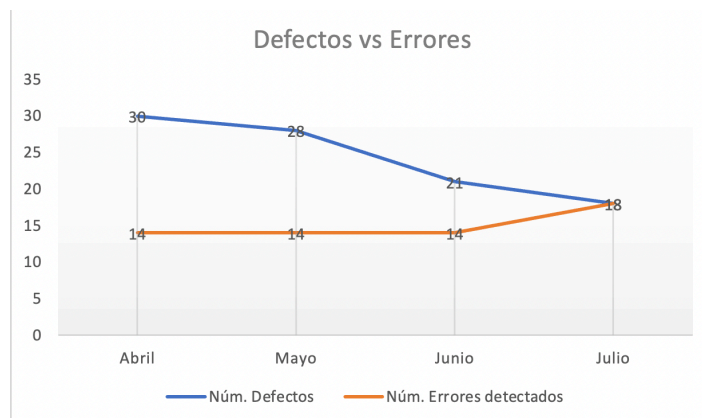


Figura 71: Productividad de Casos de Prueba.

Fuente: Katalon TestOps

5.5.4 Resultados

La implementación del proyecto inicia en el mes de mayo del 2022, en la célula de Remuneraciones como la primera célula en donde se llevará a cabo el rediseño, comenzó incorporando aseguramiento de calidad en donde el QA Manual y QA Automatizador son los encargados por velar por la calidad de la plataforma. El rediseño del proceso contiene la lógica de negocio que permite poder encontrar errores en la plataforma antes que los clientes sean quienes lo reporten como anteriormente sucedía. Al incorporar el prototipo mostrado anteriormente de las pruebas automatizadas en la célula de Remuneraciones ha permitido poder evitar defectos, ya que los scripts realizados que representan casos de prueba al existir un caso de prueba fallido brinda la información de una manera oportuna que permite al QA Automatizador comunicarlo al desarrollador para que esa funcionalidad pueda ser solucionada y así ese defecto no impacte a la funcionalidad de la plataforma, como resultado de la incorporación de procesos que contienen aseguramiento de calidad, perfil encargado de velar por la calidad de la plataforma, una cultura de calidad ha permitido que se refleje una disminución del **40%** en la cantidad de defectos reportados por los clientes en tan solo 3 meses que se inició con la implementación del proyecto. Además de que en solo 4 meses se logró igualar la cantidad de errores encontrados por los QA en etapa de desarrollo como resultado de la implementación de aseguramiento de calidad versus los defectos reportados por los clientes. Teniendo como objetivo en los próximos meses que el número de errores encontrados en desarrollo aumente y el número de defectos encontrados por los clientes disminuya.



Mes	Abril	Mayo	Junio	Julio
Núm. Defectos	30	28	21	18
Núm. Errores detectados	14	14	14	18

Figura 72: Historial defectos reportados por clientes en célula Remuneraciones de abril a julio del 2022.

Fuente: Freshdesk

CAPÍTULO 6: GESTIÓN DEL CAMBIO Y PROPUESTA DE IMPLEMENTACIÓN

6.1 Gestión del Cambio

Para poder llevar a cabo con éxito el proyecto es necesario aplicar “Gestión del Cambio”, buscando con ello hacer que los procesos de cambio puedan fluir de manera adecuada, debido a que cualquier cambio en una empresa suele impactar directamente a las personas por lo que se pretende evitar algún tipo de resistencia que pueda convertirse en obstáculo para la correcta finalización del proyecto.

Existen muchos tipos de modelos de “Gestión del Cambio” en donde la mayoría de ellos se enfocan en paradigmas que son autoritarios, para este proyecto se buscó un modelo que se enfocará en el cuidado de las personas como lo es el modelo CHES centrado en la Conciencia, en ser Holístico, ser Experiencia, Sistemático y Sustentable. (Olguín, 2005)

6.1.1 Liderazgo y gestión del proyecto de cambio

El liderazgo es esencial para poder llevar a cabo el rediseño del proceso, la naturaleza del proyecto toca puntos sensibles en el área de producto. Al existir un área de QA que se enfocará en la calidad de la creación de la plataforma podría causar en los desarrolladores algunas molestias ya que su trabajo será revisado con mayor detalle, por lo que es importante iniciar con una cultura de calidad en el área para poder llevar a cabo. Se necesitará realizar las siguientes actividades:

- **Estructura área QA y asignaciones de roles:** El área QA estará conformado por el **QA Lead** será el profesional que velará por la correcta implementación del rediseño del proceso, las estrategias que se deberán implementar, realizar seguimiento de los hitos y realizar seguimiento de los objetivos de los QA. El **QA Manual** será el encargado de realizar las pruebas manuales, de validar y verificar que las funcionalidades de las Historias de Usuario cumplan con los Criterios de Aceptación. Mientras el **QA Automatizador** será el encargado de realizar los scripts de pruebas automatizadas y validar el resultado además de brindarle la información al QA Lead para poderse apoyar de los resultados para poder tomar mejores decisiones.
- **Declarar los propósitos del proyecto:** Será de mucha importancia que el QA Lead se reúna con el CTO y el Head de Producto esto para explicar los objetivos que tiene el rediseño del proceso del área de producto, cuáles serán los

resultados que se podrán obtener si se tiene una correcta implementación. Y como se verá beneficiada la calidad de la plataforma para los usuarios.

6.1.2 Sentido y estrategia del proceso de cambio

El sentido que se establece en un proyecto puede ser de gran impacto por lo que es importante en el área de producto que se pretenda seducir explicando los beneficios que se tendrán al realizar el rediseño del proceso. Para esto que se necesitarán de las siguientes actividades:

- **Instancia para compartir importancia de una cultura de calidad en el área de producto:** Se pretende reunir a todo el equipo del área de producto. Se puede realizar por roles, área de negocio y área de desarrollo con el fin de que en cada instancia se les explique que una cultura de calidad es responsabilidad de todo el equipo no solo del área QA, en cómo todos pueden apoyar y los beneficios que se obtendrán si se aplican buenas metodologías de aseguramiento de calidad.
- **Iniciar implementación del rediseño de proceso en una célula:** Al ser producto un área conformado por 12 células, se pretende iniciar el rediseño del proceso con una célula. Esto permitirá que las demás células puedan observar la implementación y en cómo busca ser apoyo en la calidad de la plataforma.

6.1.3 Cambio y Conservación:

Buk se caracteriza por tener una cultura de trabajo que permite tener comunicaciones asertivas, apoyo y buena disposición, estos son valores que es necesario conservar para poder realizar una exitosa implementación del rediseño del proceso de desarrollo en el área de producto. Debido a la naturaleza del rol QA es el encargado de revisar el trabajo realizado por los desarrolladores se pueden presentar en algunos casos por parte de los desarrolladores resistencia a los nuevos cambios en el proceso de desarrollo por lo que es importante explicarle al desarrollador que el QA tiene como objetivo mejorar la calidad de su trabajo por ello es importante el QA tenga muy buenas habilidades blandas que le permita poder empatizar y así poder brindar una buena comunicación asertiva. Normalmente se responsabiliza por la calidad solamente al QA sin embargo, es necesario cambiar este pensar conscientizando al equipo que lo conforman tanto los desarrolladores como los product owner, que la calidad es responsabilidad de todos, siendo necesario promover una cultura de calidad que actualmente no existe lo que dara como resultado la facilidad de que el proyecto pueda llevarse a cabo exitosamente.

6.1.4 Metaobservación, rediseño y seguimiento del proceso de cambio:

Para llevar a cabo un proyecto es necesario tener hitos claros que permitan a todo el equipo conocer el estado en el que se encuentra el proyecto y los logros que se han podido obtener, así que será necesario presentar el siguiente Roadmap al equipo de producto que estará compuesto de:

- **Incorporación de proceso de calidad:** Se pretende incorporar al QA a la célula para así poder implementar los procesos de aseguramiento de calidad.
- **Implementar sesiones post refinamiento:** Se busca implementar el proceso de requerimientos de negocios en donde el QA en conjunto con el BA/PO agregan casos bordes a las Historias de Usuario. Esto permitirá que se realice un mejor diseño en las funcionalidades, lo cual disminuye errores en código.
- **Levantamiento de flujos críticos:** En esta instancia se pretende el PO apoye al QA para la creación de los flujos que son críticos para negocio y permitir puedan ser automatizados por el QA Automatizador.
- **Creación de pruebas automatizadas:** Es la instancia en la que se dedica más tiempo debido a lo costoso en recursos que es la creación de los scripts de pruebas automatizadas.

Es importante realizar seguimiento del proyecto cada 3 semanas, que permitirá evaluar si se está cumpliendo con el Roadmap presentado en la figura 73.

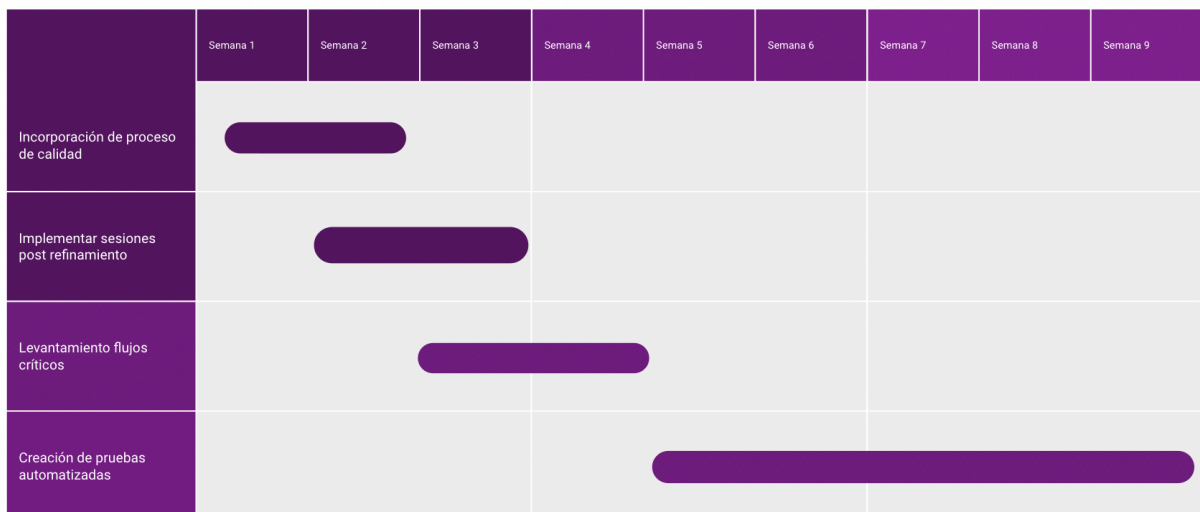


Figura 73: Roadmap proyecto. **Fuente:** Elaboración propia

6.1.5 Prácticas en el cambio:

Buk se caracteriza por ser una empresa centrada en los clientes, por lo que es importante estar atento a los dolores del cliente. El área de SAC es un gran apoyo ya que son los que tienen esa cercanía con el cliente. Para ello será necesaria la siguiente actividad:

- **Reunión área de QA y área SAC:** Esta instancia se llevará a cabo cada 2 semanas. Tiene como objetivo que el área de SAC les brinde información referente a los clientes respecto a cómo son sus dolores, la experiencia de usuario del cliente, mejoras que podría tener la plataforma, los defectos que más reportan los clientes para así el área QA tener un mayor conocimiento de los clientes y en cómo poder mejorar la calidad de la plataforma.

6.1.6 Gestión de las comunicaciones:

Es importante diseñar instancias que permitan comunicar lo relevante al proyecto y sus avances por lo que se realizarán las siguientes actividades:

- **Reunión con Head de producto:** Será una instancia que se realizará una vez a la semana la cual tendrá como propósito informar los avances que se van teniendo en el proyecto tanto exitosos como los obstáculos que se van presentando.
- **Envío informe ejecutivo por sprint:** El QA tendrá la tarea de enviar al finalizar el sprint el informe de su trabajo en la célula, reportando información relevante como las Historias de Usuario que certifico QA, los errores reportados entre otra información a los actores relevantes como lo son el Head de producto, el PO y el Engineer Manager de su célula.

6.1.7 Gestión emocional del proyecto:

Es importante reconocer que los estados de ánimo de un proyecto pueden impactar positivamente o negativamente por lo que será necesario mantener un estado de ánimo positivo, la Lead QA será la embajadora de esta emoción ante todo el equipo velará porque el QA sea bien recibido en la célula y la implementación del proceso de aseguramiento de calidad sea llevado a cabo de la mejor manera posible. Para realizar seguimiento del estado de ánimo del equipo es necesario tener las siguientes actividades:

- **Reunión de uno a uno:** En esta instancia se pretende el QA Lead se reúna semanalmente con los QA para poder hacer seguimiento de objetivos, pero

además poder conocer como ha sido la incorporación del QA cómo se ha sentido al respecto sino ha sentido al equipo de desarrollo indispuerto si el proceso ha fluido de la manera correcta.

- **Reunión QA Lead con el PO:** En esta instancia se busca que el PO brinde comentarios tanto positivos como negativos referentes a lo que ha sido la incorporación del QA en las células, que áreas de mejoras se podrían tener en el proceso y cómo están los estados de ánimo del equipo de desarrolladores.

6.1.8 Gestión del aprendizaje y las habilidades:

Para todo proceso de cambio es necesario ocurra un proceso de aprendizaje siendo en su mayoría un aprendizaje transformacional que incluye aprender a pensar, a sentir de una manera distinta. Siendo necesario para las organizaciones desarrollar nuevas habilidades y en otros casos desaprender otras. Existen dos líneas de aprendizajes que deben estar presentes. La primera se enfoca en las habilidades y conocimientos que se requieren para el proceso de cambio que para este proyecto sería conocimientos en aseguramiento de calidad. Mientras que el segundo se requiere de habilidades necesarias para ejecutar de manera exitosa el proceso de cambio que para este proyecto es importante el QA sea curioso, con ojo al detalle, comunicativo, proactivo y empático al momento de reportar defectos al desarrollador.

6.1.9 Gestión del poder:

Empoderar el área QA será imprescindible para poder llevar a cabo el proyecto, será necesario se le brinden todos los recursos y el apoyo necesario de parte del área de producto para que cada una de las actividades sean llevadas a cabo y lograr el rediseño del proceso sea un éxito, esto debido a que el rediseño presentará cambios en el proceso de desarrollo en donde se deberán implementar prácticas de calidad que el equipo de desarrollo deberá seguir por lo tanto es importante las directrices que brinde el área QA sean respetadas para poder entregar una plataforma de calidad libre de defectos.

6.1.10 Evaluación y cierre:

Si bien la implementación del proyecto iniciará en una célula para luego replicar la implementación en las demás células del área es importante según el modelo CHES que en los equipos no se cultive el miedo a las evaluaciones por lo que será necesario al finalizar la implementación de la primera célula, se realicen evaluaciones cualitativas y cuantitativas con las personas involucradas en el proyecto que permitan conocer la

experiencia en la implementación del rediseño del proceso, su percepción de la integración de prácticas de calidad en el desarrollo y si el nuevo cambio no afecto la cultura de trabajo. También será necesario medir de manera cuantitativa el proyecto, observando las métricas de los defectos reportados por los clientes en donde si se presenta disminución resulta ser un indicador que el proyecto ha resultado exitoso.

6.2 Pasos de Implementación basados en la Gestión del Cambio

Se definen las actividades que serán necesarias para la implementación exitosa del proyecto basándose en la gestión del cambio de la metodología CHES:

1. Realizar reunión con el CTO y el Head de producto para compartir el plan del rediseño del proceso del área de producto y declarar los propósitos del proyecto.
2. Establecer estructura del área QA y la asignación de roles.
3. Iniciar implementación del rediseño de proceso en una célula que permita incorporar los procesos de aseguramiento de calidad.
4. Realizar instancias en el área de producto para compartir la importancia de una cultura de calidad en el área de producto.
5. Implementar las sesiones post-refinamiento en la célula.
6. Realizar el levantamiento de flujos críticos en la célula.
7. Iniciar la creación de scripts de pruebas automatizadas.
8. Realizar instancias una vez al mes entre el área QA y SAC.
9. Realizar instancias una vez al mes entre el QA Lead y el PO de la célula en donde se está implementando aseguramiento de calidad.
10. Realizar instancias una vez a la semana entre el QA Lead y el QA Ssr que se encuentra asegurando la calidad en su célula asignada.
11. El QA asignado en la célula deberá enviar informes al finalizar el sprint del estado de calidad de su célula tanto al QA Lead, Head de producto y el CTO.
12. El QA Lead deberá presentar mensualmente indicadores respecto al área de QA y estado de calidad de la plataforma al Head de producto y el CTO.

CAPÍTULO 7: EVALUACIÓN DEL PROYECTO

7.1 Evaluación Económica

Para poder llevar a cabo el proyecto es necesario realizar un análisis económico que permita conocer si el proyecto es viable, por lo que se pretenderá hacer una comparativa basándose en el flujo de caja de la situación actual y el flujo de caja con proyecto, además de presentar distintos escenarios que permitan realizar un análisis de sensibilidad.

7.1.1 Definición de Beneficios y Costos

Los beneficios del proyecto impactan directamente a los ingresos de la empresa, al lograr disminuir significativamente los defectos en la plataforma se logrará mantener una buena reputación de la empresa lo que dará como consecuencia el ingreso de nuevos potenciales clientes y la retención de los actuales clientes. Dentro de los costos que implicará el proyecto será el pago de salarios de los nuevos roles QA que se necesitarán para el aseguramiento de calidad de la plataforma, no se incluirán los costos de desarrollo e implementación dado que se incluyen en las funciones que realizará el QA Automatizador. Sin embargo, se añade el costo de suscripción de la herramienta que apoyará tecnológicamente al proyecto .

Tabla 11: Tabla de beneficios

Fuente: Elaboración propia

Beneficios
Disminución de defectos en la plataforma Buk
Mantener buena reputación de la empresa Buk
Disminuir porcentaje de salida de los clientes

Tabla 12: Tabla de costo.

Fuente: Elaboración propia

Costo
Salarios roles QA
Suscripción de herramienta Katalon

7.1.2 Datos y Cuantificación

Al reflejar los costos del proyecto en un valor monetario es necesario realizar el cálculo de los sueldos que conllevaría la creación del área QA, para ello será necesario contar con un QA Sr y un QA que tenga conocimientos en pruebas manuales y automatizadas, el salario se obtiene de acuerdo al promedio ofrecido por la empresa Buk además se deberá pagar un costo anual por suscripción de la herramienta que permite automatizar casos de prueba, el costo fue obtenido del sitio web de Katalon. Cabe mencionar que se realizara el costo para el año 2022 calculando un promedio de 6 profesionales QA y para el año 2023 el costo de 12 profesionales QA debido a que el área de producto se conforma por 12 células de las cuales se espera que en cada una se encuentre un QA a cargo de la calidad de las funcionalidades de la plataforma.

Tabla 13: Datos para realizar caja de flujo.

Fuente: Elaboración propia

Costo	Valor (USD)
Salario QA Sr (Manual & Automatizador)	\$3200
Salario QA Ssr (Manual & Automatizador)	\$2200
Suscripción Anual de herramienta Katalon	\$1,499

7.1.3 Flujo de Caja

7.1.3.1 Tasa de descuento CAPM

Para poder obtener la tasa de descuento fue necesario utilizar la metodología CAPM, Utilizando el β calculado por Damodaran del sector de *Software (Systems & Application)*.

Tabla 14: Tasa de descuento. **Fuente:** Tasas obtenidas por sitio web de Aswath Damodaran

Tasa libre de riesgo para Chile	1.51
Libre de riesgo	4.24
β	1.37
Premio riesgo país	1.8
Tasa de Descuento	8.22

7.1.3.2 Situación Actual Optimizado

La situación actual con la que se compara la situación con proyecto, se determina basando en los ingresos actuales obtenidos por la empresa descontando el 10% de salida de los clientes que impactan al ingreso, asumiendo se considera en el año 2 un 12% en porcentaje de salida de clientes y para el año 3 un 15% de porcentaje de salida de clientes en caso que no se tomen medidas para solucionar la calidad del cliente. Lo cual afectara aumentando el porcentaje de salida de cliente además de la adquisición de potenciales nuevos clientes para la empresa. Para este caso se asume no considerar costos fijos, ya que no se cuenta con área QA que se necesite pagar salarios y por lo tanto ni en costos variables que serían los de infraestructura.

Tabla 15: Flujo de caja actual optimizado.
Fuente: Elaboración propia

	Año 0	Año 1	Año 2	Año 3
Ingreso por venta		1427319.99	1256041.591	1067635.353
Costos fijos	0	0	0	0
Costos variables	0	0	0	0
Utilidad antes de impuesto		1,427,320	1,256,042	1,067,635
Impuesto a la renta (27%)		-385376.40	-339131.2296	-288261.5452
Utilidad después de impuesto		1,041,944	916,910	779,374
Flujo de Caja Operacional		1,041,944	916,910	779,374
Flujo de Caja de Capitales	0	0	0	0
Flujo Caja privado Neto	0.00	\$ 1,041,943.59	\$ 916,910.36	\$ 779,373.81

7.1.3.3 Situación con Proyecto

La situación actual por la que se llevó a cabo el proyecto es porque se presenta una perdida en los ingresos de la empresa por un aproximado de USD 1,585,911,11 esto debido al retiro de suscripciones de los clientes anualmente, lo que se pretende que con el proyecto es disminuir a 1% la salida de los clientes y por lo tanto aumente los ingresos por suscripciones de tal forma que los actuales clientes no se retiren y aumenten los ingresos de nuevos clientes.

Con el proyecto se asume lo siguiente:

- La situación con proyecto incluye en los costos fijos los que serán los salarios de los QA que serán necesario para implementar con éxito el proyecto en donde en el primer año se espera la incorporación de 6 QA's y para el año 2 se espera el incorporamiento de 6 QA's más logrando así que en cada una de las 12 células se encuentre un QA velando por la calidad de la plataforma. Además, para el año 3, se aumenta un 10% en los salarios.

- Con el proyecto se espera la disminución de suscripciones por cliente a un 1% por lo que impacta positivamente en el ingreso de ventas de la empresa, se aproxima que en el año 2 se tenga un 10% de crecimiento en ventas y en el año 3 un 15% de crecimiento.
- Las empresas privadas se encuentran sujetas a un IDPC del 27%. (EDIG, 2022)
- Se agrega en costos variables la suscripción por el pago de la herramienta Katalon en donde se calcula que anualmente podría tener un aumento del 1% en su costo.
- No se necesitará de inversión externa por lo tanto no se incurrirá en préstamos.
- Para el proyecto no es necesario obtener activos fijos dando como consecuencia no se presente ni depreciación, ni amortización.
- Los valores estarán en dólar.
- El horizonte de evaluación se realizará en 3 años con una tasa de descuento de 8.22 en el que, al realizar el flujo de caja, se obtiene un VAN de \$ **138,625.35**.

Tabla 16: Flujo de Caja, Situación con Proyecto.

Fuente: Elaboración propia

	Año 0	Año 1	Año 2	Año 3
Ingreso por venta		1570051.989	1648554.588	1813410.047
Costos fijos	0	14,200	28400	31240
Costos variables	0	1,499	1,514	1,529
Depreciación activo fijo Tangible	0	0	0	0
Amortización activo fijo intangible	0	0	0	0
Ganancias o perdidas por ventas de activos	0	0	0	0
Utilidad antes de impuesto		1,554,353	1,618,641	1,780,641
Impuesto a la renta (27%)		-419675.31	-437032.9589	-480773.0828
Utilidad después de impuesto		1,134,678	1,181,608	1,299,868
Depreciación activo fijo Tangible	0	0	0	0
Amortización activo fijo intangible	0	0	0	0
Ganancias o perdidas por ventas de activos	0	0	0	0
Flujo de Caja Operacional		1,134,678	1,181,608	1,299,868
Préstamo	0	0	0	0
Amortización préstamo	0	0	0	0
Flujo de Caja de Capitales	0	0	0	0
Flujo Caja privado Neto	0.00	\$ 1,134,677.68	\$ 1,181,607.63	\$ 1,299,867.96
VAN	\$138,625.35			

El flujo de caja diferencial se aprecia en la siguiente tabla 17, teniendo el Delta VAN positivo de \$13,835.79.

Tabla 17: Flujo de Caja Delta.

Fuente: Elaboración propia

	Año 0	Año 1	Año 2	Año 3
Ingreso por venta		142731.999	392512.9973	745774.6948
Costos fijos	0	14,200	28400	31240
Costos variables	0	1,499	1,514	1,529
Depreciación activo fijo Tangible	0	0	0	0
Amortización activo fijo intangible	0	0	0	0
Ganancias o pérdidas por ventas de activos	0	0	0	0
Utilidad antes de impuesto	0	127,033	362,599	713,006
Impuesto a la renta (27%)	0	-34298.91	-97901.73	-192511.54
Utilidad después de impuesto		92,734	264,697	520,494
Depreciación activo fijo Tangible	0	0	0	0
Amortización activo fijo intangible	0	0	0	0
Ganancias o pérdidas por ventas de activos	0	0	0	0
Flujo de Caja Operacional	0.00	92,734	264,697	520,494
Préstamo	0	0	0	0
Amortización préstamo	0	0	0	0
Flujo de Caja de Capitales	0	0	0	0
Flujo Caja privado Neto	0.00	\$ 92,734.09	\$ 264,697.27	\$ 520,494.16
Δ VAN	\$ 13,835.79			

7.1.4 Análisis de Sensibilidad

Para poder realizar el análisis de sensibilidad fue necesario presentar tres distintos escenarios, al tener claro que el porcentaje de salida de clientes impactan directamente al ingreso de ventas se procede a los siguientes escenarios:

- **Escenario pesimista** en donde se representa un 35% en salida de clientes, dando como resultado un VAN de \$86,246.88 que si bien es un VAN positivo se puede observar que comparado con lo esperado con el proyecto es aproximadamente un 60% menos de ingresos que se podrían obtener si se lleva a cabo el proyecto.

Tabla 18: Flujo de caja escenario pesimista.

Fuente: Elaboración propia

	Año 0	Año 1	Año 2	Año 3
Ingreso por venta		1030842.215	670047.4398	435530.8358
Costos fijos	0	14,200	28400	31240
Costos variables	0	1,499	1,514	1,529
Depreciación activo fijo Tangible	0	0	0	0
Amortización activo fijo intangible	0	0	0	0
Ganancias o pérdidas por ventas de activos	0	0	0	0
Utilidad antes de impuesto		1,015,143	640,133	402,762
Impuesto a la renta (27%)		-274088.67	-172836.0287	-108745.6957
Utilidad después de impuesto		741,055	467,297	294,016
Depreciación activo fijo Tangible	0	0	0	0
Amortización activo fijo intangible	0	0	0	0
Ganancias o pérdidas por ventas de activos	0	0	0	0
Flujo de Caja Operacional		741,055	467,297	294,016
Préstamo	0	0	0	0
Amortización préstamo	0	0	0	0
Flujo de Caja de Capitales	0	0	0	0
Flujo Caja privado Neto	0.00	\$ 741,054.55	\$ 467,297.41	\$ 294,016.14
VAN	\$ 86,246.88			

- **Escenario Intermedio** que es el que se representa actualmente que es una salida del 10% por parte de los clientes que representa un VAN de \$123,586.74 siendo positivo pero que puede llegar a ser un mayor valor.

Tabla 19: Flujo de caja escenario probable.

Fuente: Elaboración propia

	Año 0	Año 1	Año 2	Año 3
Ingreso por venta		1427319.99	1284587.991	1156129.192
Costos fijos	0	14,200	28400	31240
Costos variables	0	1,499	1,514	1,529
Depreciación activo fijo Tangible	0	0	0	0
Amortización activo fijo intangible	0	0	0	0
Ganancias o pérdidas por ventas de activos	0	0	0	0
Utilidad antes de impuesto		1,411,621	1,254,674	1,123,360
Impuesto a la renta (27%)		-381137.67	-338761.9776	-303307.2518
Utilidad después de impuesto		1,030,483	915,912	820,053
Depreciación activo fijo Tangible	0	0	0	0
Amortización activo fijo intangible	0	0	0	0
Ganancias o pérdidas por ventas de activos	0	0	0	0
Flujo de Caja Operacional		1,030,483	915,912	820,053
Préstamo	0	0	0	0
Amortización préstamo	0	0	0	0
Flujo de Caja de Capitales	0	0	0	0
Flujo Caja privado Neto	0.00	\$ 1,030,483.32	\$ 915,912.01	\$ 820,052.94
VAN	\$ 123,586.74			

- **Escenario optimista** es el ideal de lo que se espera con la implementación del proyecto en donde se obtiene como resultado el 1% de salida de clientes, en donde los clientes están con un grado de satisfacción alto y se aumenta el ingreso de venta por nuevos clientes en el año 2 del 10% y en el año 3 del 15%, por lo que se tiene un VAN de \$139,460.54 muy cercano a lo esperado al escenario con proyecto que tiene un VAN de \$138,625.35.

Tabla 20: Flujo de caja escenario optimista.

Fuente: Elaboración propia

	Año 0	Año 1	Año 2	Año 3
Ingreso por venta		1570051.989	1727057.188	1986115.766
Costos fijos	0	14,200	28400	31240
Costos variables	0	1,499	1,514	1,529
Depreciación activo fijo Tangible	0	0	0	0
Amortización activo fijo intangible	0	0	0	0
Ganancias o pérdidas por ventas de activos	0	0	0	0
Utilidad antes de impuesto		1,554,353	1,697,143	1,953,347
Impuesto a la renta (27%)		-419675.31	-458228.6607	-527403.6268
Utilidad después de impuesto		1,134,678	1,238,915	1,425,943
Depreciación activo fijo Tangible	0	0	0	0
Amortización activo fijo intangible	0	0	0	0
Ganancias o pérdidas por ventas de activos	0	0	0	0
Flujo de Caja Operacional		1,134,678	1,238,915	1,425,943
Préstamo	0	0	0	0
Amortización préstamo	0	0	0	0
Flujo de Caja de Capitales	0	0	0	0
Flujo Caja privado Neto	0.00	\$ 1,134,677.68	\$ 1,238,914.53	\$ 1,425,943.14
VAN	\$ 139,460.34			

Se puede observar en la tabla 21, si bien los tres escenarios resultan con un VAN positivo, por consecuencia hace que el proyecto resulte ser viable económicamente de manera que si el rediseño cumple con el objetivo de disminuir la salida de clientes impactara de manera positiva los ingresos en ventas de la empresa.

Tabla 21: Análisis de Sensibilidad de los tres escenarios.

Fuente: Elaboración propia

Análisis de Sensibilidad	% de Salida	VAN
Pesimista	35%	\$ 86,246.88
Intermedio	10%	\$ 123,586.74
Optimista	1%	\$ 139,460.34

CAPÍTULO 8: CONCLUSIONES

8.1 Conclusiones sobre el trabajo realizado

- El trabajo realizado propone un rediseño para el área de producto específicamente en el proceso que tiene como objetivo el desarrollo de la plataforma Buk en donde se plantea la incorporación de subprocesos que monitoreen la calidad de la plataforma mediante aseguramiento de calidad, basándose en prácticas de calidad que ofrece el marco COBIT 4.1. Se pretende como objetivo lograr disminuir a un 50% los defectos que son encontrados por los clientes. El impacto que puede tener una implementación exitosa dará como resultado una buena reputación para la empresa. Además de brindar confianza al equipo de desarrollo que su trabajo será verificado por el QA, quien evaluará si la plataforma incluye los requerimientos de negocio y la calidad adecuada. Logrando impactar positivamente en los ingresos de ventas de la empresa, dado que como consecuencia el rediseño deberá disminuir la cantidad de clientes que se encuentran inconformes con la plataforma por los defectos que contiene y al tener buena reputación atraerá nuevos prospectos de clientes.
- Dentro de los riesgos que contiene el proyecto es encontrar perfiles QA con conocimientos tanto de pruebas manuales y automatizadas por esta razón fue necesario que el perfil no solo se buscara en Chile sino en otros países para tener mayor oportunidad de dar con los perfiles adecuados. Además, se buscó la herramienta de Katalon Studio que puede realizar pruebas en lenguajes como Java que es el más común que manejan los QA, aparte de que la herramienta tiene una curva de aprendizaje corta, por lo tanto, el QA de manera rápida puede iniciar a automatizar. Y que con la integración de Katalon TestOps se podrá obtener indicadores que nos permite poder medir la calidad de la plataforma. De igual manera la suscripción anual no resulta costosa y fue aprobada por el Head del área de producto a realizar el costo.
- Para no tener predisposición por el equipo de desarrollo fue necesario realizar reuniones con las células, explicándoles la importancia de la incorporación de buenas prácticas de calidad en sus desarrollos y en como el QA ayudara al momento de poder validar y verificar que los desarrollos están saliendo con el correcto aseguramiento de calidad y siempre alineados a los requerimientos de negocio permitiendo la entrega de una plataforma libre de defectos. Buscando esencialmente la incorporación de una cultura de calidad en el área de producto promoviendo que la calidad es responsabilidad de todos.
- Dentro de lo que se hizo bien en el proyecto ha sido que se procuró escuchar a las personas importantes como el CTO, el Head de Producto y las células que conforman el área de producto poniendo máxima atención a sus dolores. La importancia de explicarles en como agregar aseguramiento de calidad sería de gran

ayuda para disminuir los defectos que reportan los clientes e imprescindible que sean conscientes que la calidad también es responsabilidad de todo el equipo. De lo que se tuvo que mejorar para poder entender el origen de los defectos fue el refinamiento de las tarjetas que reporta el equipo SAC en donde se les pidió que al momento que el cliente reporte un defecto estas tarjetas contengan toda la información necesaria ya que anteriormente quedaban campos vacíos de la tarjeta por lo tanto se tenía información incompleta o redundante.

- Finalmente, el prototipo fue realizado en la célula de Remuneraciones, en donde el equipo estuvo abierto en todo momento y con muy buena disposición a incorporar estas prácticas de calidad, de lo cual como resultado se obtuvo una disminución de 40% de defectos en el periodo de 3 meses que se trabajó con la implementación con el rediseño del proceso. Por este motivo se puede considerar que el proyecto apunta a ser un éxito ya que como objetivo se tiene llegar a una disminución de 50% que sin duda en los próximos meses se podrá alcanzar a cumplir de manera satisfactoria. Además logrando que el QA en 4 meses encontrase un aproximado de 60 errores con la implementación de aseguramiento de calidad, dando como resultado que el mes de junio del 2022 los errores encontrados por el QA fuera de 18 y los defectos encontrados por el cliente sea la misma cifra, apuntando que para los próximos meses disminuyan los defectos reportados a SAC y aumente el indicador de errores encontrados por el QA en etapa de desarrollo.

8.2 Trabajo futuro

- El trabajo actual se inicia en una de las doce células que pertenecen al área de producto, en donde se han obtenido de manera exitosa muy buenos resultados, por lo que se pretende como trabajo futuro la implementación del rediseño del proceso en las células restantes del área incorporando aseguramiento de calidad en lo que es toda el área de producto. Conformando así un área de QA que pueda ir madurando e implementando una cultura de calidad en toda la empresa.
- Se espera que a medida se agregan funcionalidades a la plataforma los scripts automatizados se incorpore mayor cantidad de casos de pruebas que permita tener una mayor cobertura de las funcionalidades, para así poder tener un total control de aseguramiento de calidad.

CAPÍTULO 9: BIBLIOGRAFÍA

- Buk (2022). Obtenido de Buk:
<https://www.buk.cl/blog/el-propósito-y-los-valores-corporativos-que-guían-a-buk#:~:text=La%20Visi%C3%B3n,a%20nuestros%20clientes%20m%C3%A1s%20felices%22>.
- Callejas. M & Alarcon. A. & Álvarez. A. (2017). Modelos de calidad del software, un estado del arte. En: Entramado. Vol. 13, no. 1, p. 236-250.
- Da Silva, D. (2021). Obtenido de zendesk: <https://www.zendesk.com.mx/blog/gestión-de-incidentes/>
- Deming. Edwards. (1986). Out of the Crisis, Cambridge University Press.
- E. Serna y F. Arango. (2011). “Prueba del software: más que una fase en el ciclo de vida”, Revista de Ingeniería, no. 35, pp. 34-40.
- F. Scalone. (2006). “Estudio comparativo de los modelos y estándares de calidad del software”, tesis de maestría, Facultad Regional Buenos Aires, Universidad Tecnológica Nacional, Buenos Aires, Argentina.
- Guerrero, G. (2006). “Cobit 4.0”, in IT GOVERNANCE INSTITUTE, ISBN 1-933284-37-4 COBIT 4.0
- Hax. C. Arnoldo. (2010). The Delta Model, Reinventando tu estrategia de negocio.
- ISO 9000-3, UNE-EN ISO 9000-3:1998
- IT Governance Institute. (2007). Cobit 4.1. USA: ITGI.
- Juran, J. (1998). Juran’s Quality Handbook, Ed. McGraw Hill.
- Lewis, (1999) “A Comparison of IEEE/EIA 12207, ISO/IEC 12207, J-STD-016, and MIL-STD-498 for Acquirers and Developers.” Abelia Corporation, Fairfax, Virginia, USA
- Lopez Echeverria, A. M., Cabrera, C., & Valencia Ayala, L. E. (2009). Introducción a la calidad del software. Scientia et Technica Año XIV.
- Martínez, E. & García. (2011). Gobierno de TI a través de COBIT 4.1 y cambios esperados en COBIT 5.
- Mera. J. (2016). Análisis del proceso de pruebas de calidad de software, Ingeniería Solidaria, vol. 12, no. 20.
- Mercado. V.H. & Zapata, J., & Ceballos, Y.F. (2015). Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles. Rev.investig.desarro.innov, 6(1), 73-83 .
- Morales, S. F., Medina Martínez, J. C., Cuevas, R. E., & Molina, F. (s.f.). Criterios de calidad en el desarrollo de software de tipo web.
- Barros. (2015). Ingeniería de Negocios: Diseño Integrado de Servicios, sus Procesos y Apoyo TI.
- Pressman. R (2010) Ingeniería de Software, 7ma edición
- Rehkopf. M (2022). Obtenido de Atlassian:
<https://www.atlassian.com/es/agile/project-management/user-stories>

- Scalone. F. (2006) Estudio comparativo de los modelos y estándares de calidad del software. Tesis Ingeniería de Calidad. Buenos Aires: Universidad Tecnológica Nacional Regional de Buenos Aires. 488 p.
- Secades. V. (2007). "Gestión del Conocimiento: Modelos de transferencia del conocimiento y Calidad en los Medios de Comunicación, Organizaciones y empresas" en Curso nivelatorio de Doctorado, Universidad Pontificia de Salamanca, España.
- Sistemas de gestión de calidad – Conceptos y vocabulario (Traducción certificada), ISO 9000:2000, ISO 2000.
- Standard Glossary of Software Engineering Terminology, IEEE STD 610-1990, IEEE 1990.
- (2022). Obtenido de EDIG: <https://edig.cl/2022/10/21/impuesto-de-primera-categoria/>
- (2020). Obtenido de Ambir: <https://www.ambit-bst.com/blog/metodolog%C3%ADa-til-gesti3n-de-incidencias-y-objetivos>
- (2020). Obtenido de Manageengine: <https://www.manageengine.com/latam/service-desk/itil-incident-management/que-es-la-gestion-de-incidentes-itil.html>

ANEXO

Script de casos de pruebas automatizadas:

Feature:

Feature: Procesos

nosotros ejecutaremos los smoke test de todos los procesos

@smoke_test

Scenario Outline: [RMCL-C305] Ejecutar el proceso '<Proceso>' de forma correcta

Given nosotros queremos ejecutar los procesos generales de mes de '<Mes>' en año '<Año>'

When ejecutamos el proceso '<Proceso>' de mes '<Status_Mes>'

And el sistema nos despliega mensaje informativo "alert-success alert alert-dismissible" de ejecucion correcta del job

And espero a que el sistema se actualiza despues de ejecutar el proceso

Then el proceso '<Proceso>' con estado '<Status_Mes>' termina para mes de '<Mes>' en el año '<Año>'

And el proceso '<Proceso>' con estado '<Status_Mes>' termina con resultado '<Resultado>'

Examples:

Proceso	Resultado	Mes	Año	Status_Mes
Liquidaciones PDF	Actualizado	Junio	2021	Cerrado
Libros	Actualizado	Junio	2021	Cerrado
Previred	No generado	Mayo	2021	Cerrado
Pagos a Honorarios	Actualizado	Junio	2021	Cerrado
Contabilidad	Actualizado	Junio	2021	Cerrado
Sueldos	Actualizado	Junio	2021	Cerrado
Libro Electrónico	Actualizado	Junio	2021	Cerrado

@smoke_test

Scenario: [RMCL-C366] Ejecutar Etapa 1 del proceso Generar con boton Procesar

Given nosotros queremos ejecutar los procesos generales de mes de 'Marzo' en año '2022'

When presionamos el boton procesar en pagina

And seleccionamos las etapas a procesar

| Etapa | Accion |

| Etapa 1 | Unselect |

#| Etapa 2 | Select |

| Etapa 3 | No |

#| Etapa 4 | Unselect |

And presionamos boton procesar en formulario

Then Los procesos correspondientes terminan de forma correcta:

| Proceso | Mes | Year | Status_Mes | Status_Job |

| Liquidaciones PDF | Junio | 2021 | Cerrado | Actualizado |

| Sueldos | Junio | 2021 | Cerrado | Actualizado |

@smoke_test

Scenario Outline: [RMCL-C306] Verificar que el Proceso de Otros Pagos pueda ser ejecutado de forma correcta

Given nosotros queremos ejecutar los procesos generales de mes de '<Mes>' en año '<Año>'

When enviamos el proceso "Otros pagos" de mes '<Status_Mes>'

And el sistema nos despliega el pop up de otros pagos con título "Envío de Transfer Bancario"

And seleccionamos la opción '<Opcion>
And seleccionamos el bono '<Bono>' si corresponde
And presionamos el boton "Enviar" en formulario de envío de otros pagos
Then El proceso otros pagos se ha enviado de forma correcta

Examples:

Opcion	Bono	Mes	Año	Status	Mes
Transfer bancario por ítem	Aguinaldo	18	Mayo	2021	Cerrado
Anticipos	N/A	Mayo	2021	Cerrado	

@smoke_test

Scenario: [RMCL-C310] Verificar que el Proceso de Libros PDF

Given nosotros queremos ejecutar los procesos generales de mes de "Agosto" en año "2020"

When ejecutamos el proceso "Libros PDF" que descarga de mes "Cerrado"

And el sistema nos despliega el pop up de descarga de archivo con título "Generando Archivo"

Then El proceso de generación de archivo ha descargado en la ruta "C:/Users/mmoncada/Downloads" el archivo "Libro Remuneraciones PDF.zip" de forma correcta

Script:

```
package core.buk.stepdefinition
import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase
import static com.kms.katalon.core.testdata.TestDataFactory.findTestData
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject
```

```
import com.kms.katalon.core.annotation.Keyword
import com.kms.katalon.core.checkpoint.Checkpoint
import com.kms.katalon.core.checkpoint.CheckpointFactory
import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile
import com.kms.katalon.core.model.FailureHandling
import com.kms.katalon.core.testcase.TestCase
import com.kms.katalon.core.testcase.TestCaseFactory
import com.kms.katalon.core.testdata.TestData
import com.kms.katalon.core.testdata.TestDataFactory
import com.kms.katalon.core.testobject.ObjectRepository
import com.kms.katalon.core.testobject.TestObject
import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
```

```
import org.openqa.selenium.WebElement
import org.openqa.selenium.WebDriver
import org.openqa.selenium.By
```

```
import com.kms.katalon.core.mobile.keyword.internal.MobileDriverFactory
```

```

import com.kms.katalon.core.webui.driver.DriverFactory

import com.kms.katalon.core.testobject.RequestObject
import com.kms.katalon.core.testobject.ResponseObject
import com.kms.katalon.core.testobject.ConditionType
import com.kms.katalon.core.testobject.TestObjectProperty

import com.kms.katalon.core.mobile.helper.MobileElementCommonHelper
import com.kms.katalon.core.util.KeywordUtil

import com.kms.katalon.core.webui.exception.WebElementNotFoundException

import cucumber.api.java.en.And
import cucumber.api.java.en.Given
import cucumber.api.java.en.Then
import cucumber.api.java.en.When

import core.buk.businesslogic.BLProcesosGenerales
import core.buk.helper.HelperTestObject;

import internal.GlobalVariable
import io.cucumber.datatable.DataTable

class SubProcesos {

    @Given("quiero crear un proceso específico para el mes por defecto que se
    utilizara")
    public void
    quiero_crear_un_proceso_especifico_para_el_mes_por_defecto_que_se_utilizara() {

        String url = GlobalVariable.Url
        WebUI.navigateToUrl( GlobalVariable.Url + 'processes/buk_processes')
        BLProcesosGenerales.obtieneUltimoAnoAbierto()
    }

    @When("presiono el boton {string} proceso")
    public void presiono_el_boton(String boton) {
        switch(boton) {
            case "Crear" :

                BLProcesosGenerales.presionaCrearProcesoUltimoMesAbierto();
                break;
        }
    }

    @When("agrego el nombre {string} del proceso personalizado")
    public void agrego_el_nombre_del_proceso_personalizado(String nombre) {

```

```

        BLProcesosGenerales.ingresarNombreProceso(nombre);
    }

    @When("agrego los empleados de la lista:")
    public void agrego_los_empleados_de_la_lista(DataTable dataTable) {

        List<Map<String, String>> data = dataTable.asMaps(String.class,
String.class);
        for (row in data) {

            //Seteamos el elemento correcto - FHNI
            String Identificador = row.Identificador;
            String accion = row.Selecciona;

            BLProcesosGenerales.agregarEmpleadoByIdentificador(Identificador);

        }

        BLProcesosGenerales.agregarParticipantesSeleccionadosEnProcesosPerson
alizados();
    }

    @When("empleados son agregados de forma correcta al proceso
personalizado")
    public void
empleados_son_agregados_de_forma_correcta_al_proceso_personalizado() {
        // Write code here that turns the phrase above into concrete actions

        BLProcesosGenerales.confirmaEmpleadosAgregadosCorrectamenteAProceso
();
    }

    @When("guardo el proceso personalizado")
    public void guardo_el_subproceso() {

        BLProcesosGenerales.creoProcesoPersonalizado();
    }

    @Then("el proceso personalizado es creado de forma correcta con mensaje
{string}")
    public void el_proceso_personalizado_es_creado_de_forma_correcta(String
attributteValue) {

```

```

        assert
WebUI.verifyElementAttributeValue(BLProcesosGenerales.obtenerMensajeDeEvento
DeEjecucion(), "class", attributteValue, 5);
    }

    @Then("selecciono el proceso personalizado {string}")
    public void selecciono_el_proceso_personalizado(String nombre) {

        BLProcesosGenerales.obtieneUltimoAnoAbierto();

        BLProcesosGenerales.seleccionaProcesoPersonalizadoPorNombre(nombre);
    }

    @Then("elimino proceso personalizado de forma correcta con mensaje
{string}")
    public void
elimino_proceso_personalizado_de_forma_correcta_con_mensaje(String mensaje) {

        BLProcesosGenerales.eliminaProcesoPersonalizado();
        assert
WebUI.verifyElementAttributeValue(BLProcesosGenerales.obtenerMensajeDeEvento
DeEjecucion(), "class",mensaje, 5);
    }
}

```