



MAESTRÍA EN GERENCIA DE SISTEMAS Y TECNOLOGÍA EMPRESARIAL

Fortalecimiento de la fábrica de software en una empresa de desarrollo de
soluciones tecnológicas y de comunicaciones

AUTOR

Edison Fernando Chancusig Casa

AÑO

2021



FACULTAD DE POSGRADOS

MAESTRÍA EN GERENCIA DE SISTEMAS Y TECNOLOGÍA EMPRESARIAL

**FORTALECIMIENTO DE LA FÁBRICA DE SOFTWARE EN UNA EMPRESA DE
DESARROLLO DE SOLUCIONES TECNOLÓGICAS Y DE COMUNICACIONES**

AUTOR

Ing. Edison Fernando Chancusig Casa

TUTOR

MsC. Germán Pancho Carrera

AÑO

2021

DECLARACIÓN DE AUTORÍA

"Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes."



Edison Fernando Chancusig Casa

CI. 1714770847

AGRADECIMIENTOS

Quiero agradecer a Dios, por guiar mi camino, por darme la fuerza necesaria para seguir y sobre todo por llenarme de bendiciones cada día.

Mi agradecimiento a la Universidad de las Américas, directivos y profesores quienes con sus conocimientos y actitud contribuyeron en el desarrollo de la presente maestría.

Un profundo agradecimiento a todas las personas que me apoyaron a lo largo de la presente maestría, mi familia y amigos que siempre me brindaron su apoyo y eterna colaboración.

DEDICATORIA

El presente trabajo de investigación lo quiero dedicar a:

Mi amada esposa Jenny y mis queridos hijos Sofía y Mateo quienes con su amor, paciencia y apoyo incondicional durante este periodo me han motivado para seguir, han sido el pilar fundamental para que pueda llegar a conseguir el objetivo que hace tiempo me había trazado.

RESUMEN

El presente trabajo muestra un análisis integral del estado actual de la fábrica de software en una empresa de desarrollo de soluciones tecnológicas. La empresa posee un gran catálogo de aplicaciones, cuyo principal objetivo es brindar a sus clientes servicios de calidad. Para cumplir con sus metas la empresa cuenta con tecnología y personal que le permiten trabajar de acuerdo a las exigencias del mercado.

Actualmente la empresa tiene una serie de inconvenientes relacionados con sus servicios digitales, errores de funcionamiento en ambientes de producción, inexistencia de metodologías ágiles, falta de herramientas que permitan optimizar el tiempo de cada recurso, así como una mala planificación en la ejecución de los proyectos.

El objetivo del presente trabajo es proponer alternativas que contribuyan a solucionar los problemas indicados anteriormente, para cumplir con el objetivo se implementa una arquitectura basada en TOGAF, la cual nos permite tener una visión completa del estado actual de la fábrica de software, donde se utilizará una serie de prácticas sugeridas por marcos de trabajo como Scrum, Devops y referentes en el desarrollo de software como CMMI-DEV.

ABSTRACT

The present research shows a comprehensive analysis of the current state of a software's factory in a company which develops technology solutions. The corporation owns a large catalog of applications; whose main objective is to provide its clients worthy quality services. The company has technology and personnel that allow it to work in accordance to market demands, in order to accomplish the aimed goals.

Currently, the company struggles with several problems related to its digital services, malfunctions in production conditions, non-existent agile methodologies and a lack of tools which permit to optimize the time of each resource. Furthermore, a poor management planning in projects' execution.

The main purpose of this work is to put forward alternatives which can contribute to solve the problems mentioned above, therefore an architecture based on TOGAF is supposed to be implemented to achieve the goal. This allows us to have a whole view of the current state of the software factory, where several suggested practices will be used by work framework like Scrum, Devops and benchmarks in the software development such as CMMI-DEV.

Índice de contenido

1. FASE PRELIMINAR.....	1
1.1. Contexto	1
1.1.1. Análisis Foda	2
1.2. Organización impactada	3
1.2.1. Organigrama empresarial	3
1.2.2. Nivel de impacto	3
1.3. Stakeholders y expectativas de valor	4
1.4. Motivadores	6
1.5. Marcos de referencia complementarios	7
1.6. Equipo de arquitectura	8
1.7. Matriz Raci	9
1.8. Catálogo de principios	10
2. VISIONAMIENTO ARQUITECTÓNICO	15
2.1. Requerimientos de alto nivel.....	15
2.2. Visionamiento y escenarios de la solución.....	16
2.2.1. Referentes en la industria	16
2.2.2. Marcos de referencia	17
2.2.3. Visionamiento	20
2.3. Análisis de brechas	20
2.4. Definición de la arquitectura objetivo	23
2.4.1. Target de la arquitectura de negocio	23

2.4.2.	Target de la arquitectura de datos	23
2.4.3.	Target de la arquitectura de aplicación	24
2.4.4.	Target de la arquitectura de infraestructura base	24
2.5.	Road Map	25
3.	ARQUITECTURA DE NEGOCIO DE LA FÁBRICA DE SOFTWARE	26
3.1.	Modelo de procesos.....	26
3.1.1.	Arquitectura de negocios base	26
3.1.2.	Detalle de la arquitectura objetivo.....	28
3.1.3.	Análisis de brechas.....	32
3.2.	Modelo organizacional y talento humano	38
3.2.1.	Arquitectura base.....	38
3.2.2.	Detalle de la arquitectura objetivo.....	39
3.2.3.	Impacto de la arquitectura objetivo en la situación actual.....	47
3.2.4.	Road Map	47
4.	ARQUITECTURA DE APLICACIONES Y DE DATOS	48
4.1.	Arquitectura de aplicaciones.....	48
4.1.1.	Arquitectura actual de aplicaciones base	48
4.1.2.	Arquitectura de aplicaciones objetivo	53
4.1.3.	Trazabilidad de nuevas aplicaciones	53
4.1.4.	Análisis de brechas e iniciativas	58
4.2.	Arquitectura de datos	62
4.2.1.	Arquitectura actual de datos base	63

4.2.2.	Arquitectura de datos objetivo	64
4.2.3.	Análisis de brechas.....	64
4.2.4.	Iniciativa para cerrar la brecha	64
5.	ARQUITECTURA DE INFRAESTRUCTURA BASE	66
5.1.	Infraestructura de la fábrica de software.....	66
5.1.1.	Infraestructura base.....	66
5.1.2.	Entornos de desarrollo.....	71
5.2.	Arquitectura de infraestructura base objetivo	72
5.2.1.	Seguridad	74
5.2.2.	Disponibilidad	74
5.2.3.	Monitoreo.....	74
5.2.4.	Loggin.....	74
5.2.5.	Integración continua	75
5.3.	Análisis de brechas.....	76
5.4.	Impacto de la arquitectura objetivo en la situación actual	78
6.	OPORTUNIDADES Y SOLUCIONES	79
6.1.	Arquitectura de negocio	80
6.1.1.	Iniciativa: Implementar el marco de trabajo Scrum	80
6.1.2.	Conceptualización de la implementación.....	81
6.1.3.	Iniciativa: Seleccionar el personal adecuado para formar el equipo Scrum.....	82
6.1.4.	Iniciativa: Implementar las áreas de proceso de Ingeniería de CMMI-DEV para el desarrollo de software	86

6.2. Arquitectura de Aplicaciones y Datos	89
6.2.1. Iniciativa: Implementar la herramienta colaborativa Jira	90
6.2.2. Iniciativa: Implementar CircleCI	92
6.2.3. Iniciativa: Implementar SonarQube.....	94
6.2.4. Iniciativa: Implementar Confluence.....	96
6.2.5. Iniciativa: Implementar PowerBI	99
6.2.6. Iniciativa: Implementar Cypress.....	101
6.3. Arquitectura de infraestructura base	103
6.3.1. Iniciativa: Fortalecer la infraestructura en la nube	103
7. PLAN DE MIGRACIÓN	107
7.1. Análisis de impacto	107
7.2. Análisis de esfuerzo.....	108
7.3. Fases.....	110
7.4. Análisis de dependencias.....	112
7.5. Mapa de ruta.....	112
7.5.1. Cronograma de actividades	112
Conclusiones.....	115
Recomendaciones.....	116
Referencias.....	117
ANEXOS	121

Figuras

Figura 1. Análisis FODA	2
Figura 2. Organigrama empresarial y unidades impactadas	3
Figura 3. Scrum.....	7
Figura 4. CMMI-DEV	7
Figura 5. Diagrama conceptual requerimientos de alto nivel.....	16
Figura 6. Componentes del modelo CMMI. (CMMI, 2010)	19
Figura 7. SFIA. (SFIA Foundation, 2021)	19
Figura 8. Análisis de brechas	22
Figura 9. Arquitectura de aplicación	24
Figura 10. Hoja de Ruta	25
Figura 11. Modelo del proceso de desarrollo de software.....	27
Figura 12. Arquitectura objetivo Scrum y Cmmi-Dev.....	32
Figura 13. Estructura organizativa y talento humano	39
Figura 14. Scrum y equipo developers.....	41
Figura 15. Estructura organizativa objetivo	42
Figura 16. Hoja de ruta arquitectura de negocio objetivo	47
Figura 17. Aplicaciones existentes	50
Figura 18. Arquitectura de aplicaciones objetivo	57
Figura 19. Análisis de brechas arquitectura actual vs arquitectura objetivo	62
Figura 20. Fuentes de datos y su relación con la información.....	63
Figura 21. Infraestructura actual.....	70
Figura 22. Entornos de desarrollo. Adaptada de	72
Figura 23. Cuadrante de Gartner, plataforma e infraestructura en la nube.	73
Figura 24. Beneficios de la integración continua. (SmartNodus, 2021)	75
Figura 25. Impacto de la infraestructura objetivo.....	78
Figura 26. Conceptualización de iniciativa implementar Scrum	82
Figura 27. Conceptualización de iniciativa seleccionar equipo Scrum	86
Figura 28. Áreas de proceso de Ingeniería. (CMMI, 2010)	88

Figura 29. Conceptualización de iniciativa implementar áreas de proceso de CMMI-DEV	89
Figura 30. Conceptualización de iniciativa implementar Jira	92
Figura 31. Conceptualización de iniciativa implementar CircleCI	93
Figura 32. Conceptualización de iniciativa implementar SonarQube	95
Figura 33. Conceptualización de iniciativa implementar Confluence.....	98
Figura 34. Conceptualización de iniciativa PowerBI.....	100
Figura 35. Conceptualización de iniciativa implementar Cypress.....	103
Figura 36. Costos estimados del fortalecimiento de la infraestructura. (Calculadora de precios de Google Cloud Platform, 2021).....	105
Figura 37. Conceptualización de iniciativa fortalecer la nube de la fábrica de software.....	106
Figura 38. Priorización de iniciativas	111
Figura 39. Road Map de iniciativas	114

Tablas

Tabla 1 Unidades impactadas	4
Tabla 2 Stakeholders y expectativas de valor	5
Tabla 3 Business Motivation Model.....	6
Tabla 4 Marcos de referencia complementarios.....	7
Tabla 5 Equipo de arquitectura	8
Tabla 6 Raci	9
Tabla 7 Principio de negocio 1	10
Tabla 8 Principio de negocio 2	10
Tabla 9 Principio de aplicación 1	11
Tabla 10 Principio de aplicación 2.....	11
Tabla 11 Principio de datos 1	12
Tabla 12 Principio de datos 2.....	13
Tabla 13 Principio de tecnología 1	13

Tabla 14 Principio de tecnología 2	14
Tabla 15 Niveles de madurez CMMI. Adaptado de (CMMI, 2010)	18
Tabla 16 Análisis de brechas	20
Tabla 17 Procesos de Scrum (SCRUMstudy, 2017)	29
Tabla 18 Áreas de proceso de Ingeniería CMMI-DEV. Adaptado de (CMMI, 2010)	30
Tabla 19 Análisis de brechas procesos actuales vs objetivo.....	33
Tabla 20 Talento humano	38
Tabla 21 Perfil personal para la arquitectura objetivo (SFIA Foundation, 2021) ...	42
Tabla 22 Arquitectura de aplicaciones actual. Catálogo de aplicaciones	48
Tabla 23 Procesos vs aplicaciones	51
Tabla 24. Roles vs aplicaciones	52
Tabla 25 Rúbrica de madurez	58
Tabla 26 Análisis de brechas	59
Tabla 27 Documentación versus responsables.....	65
Tabla 28 Infraestructura actual.....	66
Tabla 29. Rúbrica de madurez	76
Tabla 30. Análisis de brechas e iniciativas.....	77
Tabla 31 Arquitecturas e Iniciativas.....	79
Tabla 32 Responsabilidades del equipo Scrum. (SCRUMstudy, 2017)	83
Tabla 33 Roles y características de Scrum. Adaptado (SCRUMstudy, 2017)	84
Tabla 34 Escala de Impacto.....	107
Tabla 35 Análisis de impacto.....	108
Tabla 36 Escala de esfuerzo	109
Tabla 37 Escala de ponderación, recursos, complejidad, capacidad de TI.	109
Tabla 38 Análisis de esfuerzo	109
Tabla 39 Fases de implementación.....	110
Tabla 40 Cronograma de implementación por fases.....	112

1. FASE PRELIMINAR

1.1. Contexto

Soluciones Tecnológicas y de Comunicaciones que de ahora en adelante será conocida como STYC, es una empresa multinacional presente en diferentes países como Ecuador, Perú, México y Colombia, con más de 14 años de experiencia brindando servicios digitales que se encuentran continuamente innovando.

Desde su fundación en el año 2008 hasta el año 2012, la empresa generó grandes ingresos ofreciendo servicios de entretenimiento por voz, entretenimiento multimedia a diferentes dispositivos móviles. El avance de la tecnología, el apareamiento de smartphones (teléfonos inteligentes), que permiten compartir contenido multimedia de una manera más eficiente y económica, impactó negativamente a la organización, generando menos ingresos.

En busca de nuevos productos, STYC se convirtió en una integradora de servicios móviles, ofreciendo servicios de SMS (servicio de mensajes cortos), una solución que les ha permitido destacar, generando ingresos suficientes para poder continuar funcionando.

Sin embargo, el servicio SMS no ha sido la única solución que se ha implementado, en la búsqueda de mayores ingresos, el área de IT ha sido participe de diferentes proyectos, como desarrollos de aplicaciones móviles de tipo bancarias, desarrollos web a medida, Mobile marketing, servicios de facturación electrónica, servicios de notificaciones (SMS, USSD, PUSH), soluciones de generación de clave única, son los productos principales con los cuales la empresa ha intentado destacar. No obstante, muchos de estos proyectos no han tenido el éxito deseado, como fueron el entretenimiento multimedia o el servicio de mensajería.

Muchos de los proyectos anteriormente mencionados han fracasado por diferentes motivos, entre los cuales podemos mencionar la falta de planificación, los costos estimados no concuerdan con los costos finales, los clientes no están de acuerdo con las soluciones entregadas, falta de liderazgo, errores en producción entre los principales. Todos estos problemas han llevado a la empresa a una situación compleja, STYC requiere aumentar sus ingresos fortaleciendo su fábrica de software, para que los proyectos que se desarrollan o están siendo desarrollados cumplan con estándares de calidad, tiempo, costo y satisfagan las expectativas del cliente.

1.1.1. Análisis Foda

A continuación, se analizan las fortalezas, debilidades, amenazas y oportunidades que tiene el área de TI:

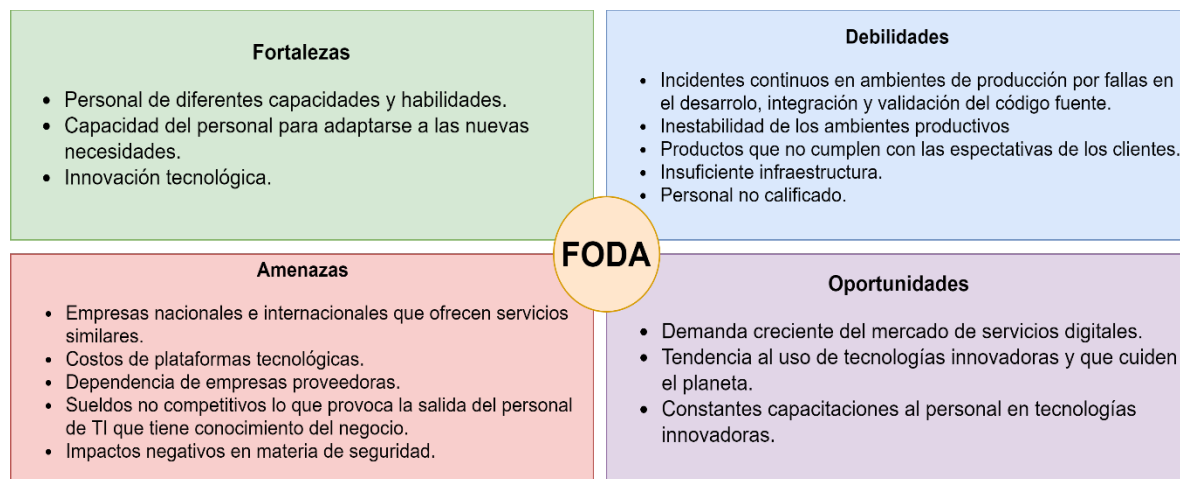


Figura 1. Análisis FODA

1.2. Organización impactada

La solución de arquitectura empresarial impactará de diferente forma a la organización en los siguientes apartados se analizarán los impactos.

1.2.1. Organigrama empresarial

A continuación, se presenta el organigrama empresarial con las principales unidades que serán afectadas.

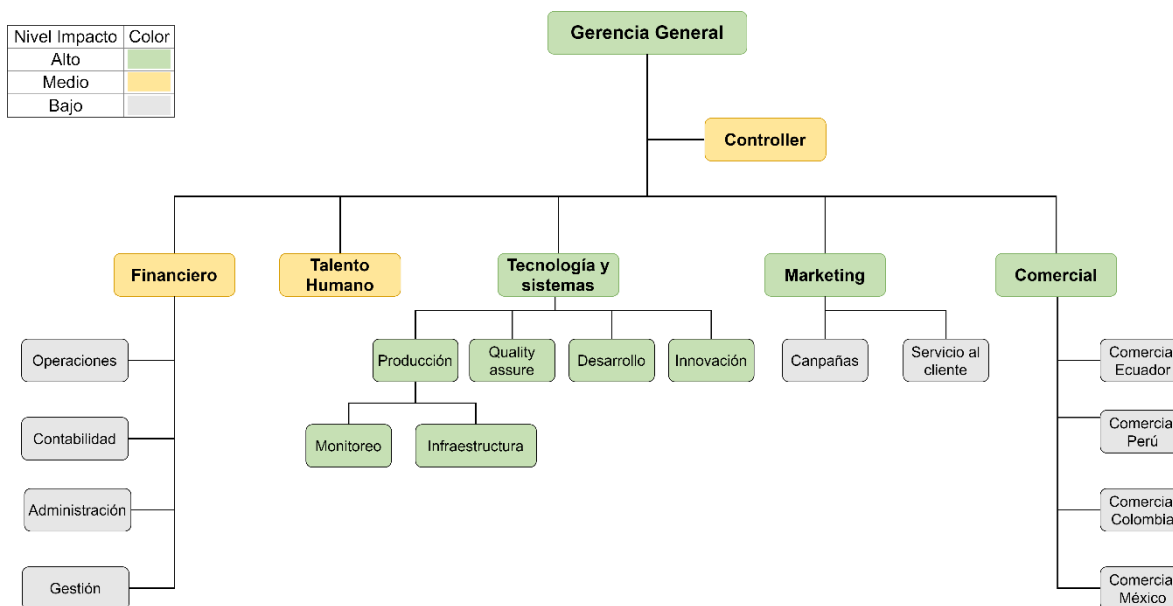


Figura 2. Organigrama empresarial y unidades impactadas

1.2.2. Nivel de impacto

La organización, se verán afectadas en diferente grado por el desarrollo del ejercicio de arquitectura empresarial, se clasifica de acuerdo al nivel de impacto en alto, medio o bajo.

Tabla 1

Unidades impactadas

Área	Nivel de impacto	Descripción de impacto
Gerencia general	Alto	<ul style="list-style-type: none"> • Cambiar los procesos de TI e institucionales. • Cambiar la estructura de la organización.
Controller	Medio	<ul style="list-style-type: none"> • Implementar políticas de calidad interna.
Financiero	Medio	Soluciones que cumplan con el presupuesto y tiempos establecidos.
Talento humano	Medio	<ul style="list-style-type: none"> • Contratar personal con las habilidades, capacidades y conocimientos necesarios para implementar metodologías ágiles. • Mejorar habilidades del personal actual.
Tecnología y sistemas	Alto	<p>Implementar metodologías ágiles en el desarrollo de software que permitan:</p> <ul style="list-style-type: none"> • Mejorar la experiencia del cliente. • Entregar productos y servicios en intervalos de tiempo cortos. • Respuesta al cambio. • Dimensionar los proyectos, minimizando los riesgos.
Marketing	Alto	Cambiar las estrategias de marketing y servicio al cliente para adaptarse al modelo de desarrollo ágil.
Comercial	Alto	Cambiar la estrategia de ventas para adaptarse al modelo de desarrollo ágil.

1.3. Stakeholders y expectativas de valor

El siguiente cuadro describe el nivel de comprensión de las iniciativas y el compromiso con las mismas, se califica de 1 a 5 donde 1 es el valor más bajo y 5 el valor más alto.

Tabla 2

Stakeholders y expectativas de valor

Unidad	Cargo	Comprensión de iniciativa		Compromiso		Expectativa de valor
		Actual	Requerida	Actual	Requerido	
Gerencia general	Gerente general	3	5	3	3	La fábrica de software genere productos informáticos de calidad, que permitan reducir los costos y maximizar los ingresos.
Controller	Gerente de control interno	2	4	3	5	Sistemas informáticos que cumplan con los estándares establecidos.
Financiero	Gerente financiero	3	4	2	4	Aplicaciones informáticas que cumplan con los requisitos establecidos por el cliente, cumpliendo tiempos y costos establecidos.
Talento humano	Director de talento humano	1	4	2	5	Personal altamente capacitado que pueda cumplir con los requerimientos del negocio.
Tecnología y sistemas	Gerente de tecnología	3	5	5	5	Desplegar aplicaciones complejas en periodos cortos de tiempo con entregables continuos que cumplan con estándares de calidad.
	Director de innovación	2	5	3	5	Implementar soluciones que sirvan de base para nuevos proyectos.
Marketing	Gerente de marketing	1	4	2	4	Identificar competencias y mercados potenciales para las nuevas aplicaciones.
Comercial	Gerente comercial	2	4	3	5	Aplicaciones funcionales, intuitivas que cumplan con las expectativas de los clientes.

1.4. Motivadores

Los motivadores para el desarrollo de la iniciativa de mejora, se exponen en un diagrama BMM.

Tabla 3

Business Motivation Model

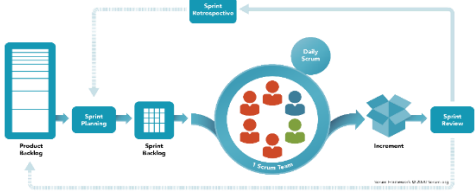

Medios	Fines
<p data-bbox="219 615 802 659">Misión</p> <p data-bbox="219 709 802 806">Fortalecer la fábrica de software implementando metodologías ágiles para el desarrollo de software.</p>	<p data-bbox="829 615 1403 659">Visión</p> <p data-bbox="829 693 1403 825">Mejorar el nivel de desempeño de la fábrica de software, minimizando los riesgos y optimizando los recursos, con la finalidad de entregar aplicaciones de calidad.</p>
Curso de acción	Resultado deseado
<p data-bbox="219 903 802 947">Estrategias</p> <ul data-bbox="219 947 802 1495" style="list-style-type: none"> • Establecer una lista de requerimientos o tareas que engloba un proyecto, llamado backlog. • Seleccionar un grupo de tareas del backlog para desarrollar, estableciendo tiempos y responsables de cada tarea. • Ejecutar la tarea y mediante iteraciones continuas obtener métricas, para conocer la velocidad del equipo de desarrollo. • Generar una entrega continua por cada sprint. • Capacidad de reacción ante cambios en los requerimientos. • Minimizar el número de errores. • Fortalecer las habilidades del equipo. 	<p data-bbox="829 903 1403 947">Objetivo</p> <ul data-bbox="829 947 1403 1325" style="list-style-type: none"> • Desarrollar aplicaciones que cumplan con estándares de calidad. • Disminuir el tiempo implementado en las diferentes fases de desarrollo, pruebas, paso a producción y soporte. • Cumplir con tiempos y presupuestos establecidos para el desarrollo de aplicaciones. • Disminuir el número de incidentes reportados por errores en las aplicaciones.
Tácticas	Meta
<ul data-bbox="219 1539 802 1852" style="list-style-type: none"> • Utilizar el marco de referencia Scrum con la finalidad de generar proyectos de manera ágil. • Utilizar el marco de referencia CMMI-DEV para manejar el ciclo de vida de los productos a ser desarrollados. • Capacitar al personal de TI en los marcos de referencia. 	<ul data-bbox="829 1539 1403 1852" style="list-style-type: none"> • Lograr que el 80% de los proyectos nuevos, sean implementados con metodologías ágiles. • Reducir en un 50% el nivel de incidentes. • Obtener una calificación mínima del 90% de satisfacción de cada uno de los nuevos proyectos.

1.5. Marcos de referencia complementarios

Dentro de la iniciativa de mejora se utilizarán los siguientes marcos de referencia.

Tabla 4

Marcos de referencia complementarios

Área	Referente	Descripción
Desarrollo de software	<p>SCRUM</p>  <p><i>Figura 3. Scrum</i></p>	<p>La metodología Scrum es un marco de trabajo o framework que se utiliza dentro de equipos que manejan proyectos complejos. Es decir, se trata de una metodología de trabajo ágil que tiene como finalidad la entrega de valor en períodos cortos de tiempo y para ello se basa en tres pilares: la transparencia, inspección y adaptación. Esto permite al cliente, junto con su equipo comercial, insertar el producto en el mercado pronto, rápido y empezar a obtener ventas. (Abellán, 2020)</p>
Desarrollo de productos y servicios	<p>CMMI-DEV</p>  <p><i>Figura 4. CMMI-DEV</i></p>	<p>Es un conjunto de mejores prácticas que cubre el ciclo de vida del producto, desde su concepción hasta su entrega y mantenimiento. CMMI-DEV organiza estas prácticas en 22 áreas de proceso y establece unos niveles de madurez y capacidad que suponen dos representaciones o caminos posibles a la hora de abordar la mejora de estas áreas de proceso. (Prestomedia, 2009)</p>

1.6. Equipo de arquitectura

Para el trabajo de arquitectura se cuenta con el patrocinio del gerente general, así como de los miembros de la organización, por tal motivo el equipo de arquitectura se formará por personas de la empresa y el equipo de arquitectura.

Tabla 5

Equipo de arquitectura

Rol	Responsabilidades
Jefe de arquitectura	<ul style="list-style-type: none"> • Proporcionar orientación a los arquitectos empresariales, sobre actividades de arquitectura. • Alinear la estrategia de TI con los objetivos empresariales.
Líder en arquitectura de negocios	Responsable de apoyar a la estrategia, y las diferentes operaciones de la organización.
Líder en arquitectura de datos	Responsable de la estructura de datos lógicos y físicos que posee la organización, así como gestionar los mismos.
Líder de aplicaciones	Encargado de conocer las aplicaciones que se deben implementar o fortalecer, así como su relación con los procesos del negocio.
Líder de tecnología	Responsable de conocer la estructura de hardware y software necesarios para la correcta implementación de los servicios de TI.
Especialistas y analistas de la organización	<ul style="list-style-type: none"> • Implementar el diseño de arquitectura empresarial. • Analizar los requisitos de arquitectura. • Participar en el contexto de la solución, alternativas y su evaluación.

1.7. Matriz Raci

A continuación, se presenta la matriz Raci con las principales funciones y responsabilidades asignadas dentro del proyecto.

Tabla 6

Raci

	Desarrollar la arquitectura empresarial	Fortalecimiento de la fábrica de software	Innovar las aplicaciones nuevas o existentes	Desarrollo de Software	Gestionar los proyectos de forma adecuada	Gestión de la calidad	Despliegue de soluciones
Jefe de arquitectura	RA	RA	RA	RA	RA	RA	RA
Líder en arquitectura de negocios	A	A	A	A	RA	A	A
Líder en arquitectura de datos	RA	A	A	A	A	A	A
Líder de aplicaciones	A	RA	A	RA	A	A	A
Líder de arquitectura de tecnología	RA	A	A	R	C	C	A
Especialistas y analistas de la organización	R	RA	RA	RA	A	A	A

1.8. Catálogo de principios

Para el presente trabajo de arquitectura se definen los siguientes principios:

Tabla 7

Principio de negocio 1

Código	PN01
Principio:	Alinear las iniciativas de TI a los procesos del negocio.
Dominio:	Negocio.
Definición:	Cualquier iniciativa de TI debe estar alineada a los objetivos empresariales.
Motivación:	Funcionamiento dentro de un marco regulatorio.
Implicaciones:	<ul style="list-style-type: none"> • Establecer procesos que garanticen la entrega de productos que generen valor • Definir una estructura organizada.

Tabla 8

Principio de negocio 2

Código	PN02
Principio:	Soluciones orientadas a la generación de valor.
Dominio:	Negocio.
Definición:	Implementar nuevas soluciones que generen valor y que puedan ser medidas.
Motivación:	<p>Desarrollar las habilidades de los empleados orientándolos a metas.</p> <p>Generar ingresos económicos a la organización.</p>
Implicaciones:	Establecimiento de indicadores para poder medir los costos generados desde el inicio de la solución hasta su seguimiento en ambiente productivos.

Tabla 9

Principio de aplicación 1

Código	PA01
Principio:	Sistemas flexibles y escalables
Dominio:	Aplicación.
Definición:	Los sistemas deben ser construidos y flexibles de tal manera que puedan ser escalables y mantenibles en el tiempo.
Motivación:	<ul style="list-style-type: none"> • Desarrollar capacidades que nos permitan adaptar a los diferentes cambios. • Soluciones que perduren en el tiempo y se adapten a las nuevas necesidades. • Reducir los tiempos de implementación de los requisitos empresariales. • Reducir los costos de mantenimiento.
Implicaciones:	<ul style="list-style-type: none"> • Contratar personal que domine las diferentes plataformas. • Contar con el apoyo de las personas involucradas en el proceso de adquisición, soporte y administración.

Tabla 10

Principio de aplicación 2

Código	PA02
Principio:	Sistemas, aplicaciones y proyectos elaborados con calidad.
Dominio:	Aplicación.
Definición:	Los nuevos y actuales sistemas y aplicaciones son uno de los activos más importantes, razón por la cual estos tienen que pasar por un estricto proceso para reducir el número de incidentes.

Motivación:	Mejorar la calidad nos permitirá, reducir al mínimo los errores en ambientes productivos, reduciendo así el costo generado por incidentes.
Implicaciones:	<ul style="list-style-type: none"> • Analizar los procesos actuales, aplicados a los diferentes sistemas. • Elaborar un mapa de habilidad de área de calidad. • Aplicar cambios en los procesos de calidad, implementando, conceptos, prácticas de empresas referentes.

Tabla 11

Principio de datos 1

Código	PD01
Principio:	La información es un activo importante.
Dominio:	Datos.
Definición:	La información es uno de los activos más importantes para la organización y debe ser adecuadamente gestionada.
Motivación:	<ul style="list-style-type: none"> • La información disponible nos permitirá analizar, clientes potenciales para nuestros nuevos productos. • En base a la información tomar decisiones que nos permitan mejorar nuestros procesos.
Implicaciones:	<ul style="list-style-type: none"> • Analizar cómo se está almacenando y que tipo de información se guarda. • Elaborar procesos para respaldar la información. • Clasificar la información para mejorar los procesos de búsqueda.

Tabla 12

Principio de datos 2

Código	PD02
Principio:	Calidad de datos
Dominio:	Datos
Definición:	Implementar planes de calidad para cada uno de los sistemas, aplicaciones y proyectos.
Motivación:	<ul style="list-style-type: none"> • Reducir costos en almacenamiento, procesamiento y búsqueda de información. • Optimizar los tiempos en procesos que dependen de la información.
Implicaciones:	<ul style="list-style-type: none"> • Transparentar la información generada por cada una de las áreas de la organización. • Generar procesos para administrar la información. • Analizar la información de manera más eficiente, para la toma de decisiones.

Tabla 13

Principio de tecnología 1

Código	PT01
Principio:	Interoperabilidad de sistemas.
Dominio:	Tecnología.
Definición:	Los sistemas, aplicaciones deben tener la capacidad para intercambiar información y utilizar dicha información.
Motivación:	<ul style="list-style-type: none"> • La información estará disponible para el sistema aplicación o persona que lo necesite. • Reducir los tiempos para obtener la información. • Eliminar dependencias por falta de información.

Implicaciones:	<ul style="list-style-type: none"> • Modificar procesos de TI. • Aplicar cambios en procesos y procedimientos. • Capacitar al personal interoperabilidad.
----------------	--

Tabla 14

Principio de tecnología 2

Código	PT02
Principio:	Implementar tecnologías que garanticen la continuidad del negocio.
Dominio:	Tecnología.
Definición:	El hardware y software que sea implementado tiene que permitir la continuidad del negocio ante fallas.
Motivación:	<ul style="list-style-type: none"> • Creación de una ventaja competitiva. • Contribuir a la resiliencia de la organización. • Reducir costos directos e indirectos por interrupciones de servicios.
Implicaciones:	<ul style="list-style-type: none"> • Incrementar costos por adquisición de plataformas que cumplan dicho requerimiento. • Mejorar la administración y monitoreo de las plataformas. • Contratar personas capacitadas en temas de continuidad del negocio.

2. VISIONAMIENTO ARQUITECTÓNICO

2.1. Requerimientos de alto nivel

Para poder fortalecer la fábrica de software, se definen los siguientes requerimientos de alto nivel.

- Implementar metodologías ágiles para el desarrollo de software que permitan:
 - Retroalimentación continua del cliente.
 - Dimensionar de mejor manera los proyectos.
 - Entrega de productos y servicios en intervalos cortos de tiempo.
 - Mejorar la calidad de software.
- Mejorar los procesos de construcción de software.
- Implementar una plataforma que permita realizar integración continua, una plataforma que unifique el desarrollo, las pruebas y el despliegue, en los diferentes ambientes.
- Capacitar al personal que forma parte de la fábrica de software.
- Mejorar los procesos que realiza control de calidad.
- Estandarizar las herramientas de desarrollo.
- Mejorar la estructura organizacional del área de desarrollo, en equipos que tengan personas con diferente perfiles y capacidades.
- Implementar aplicaciones para versionar código fuente.

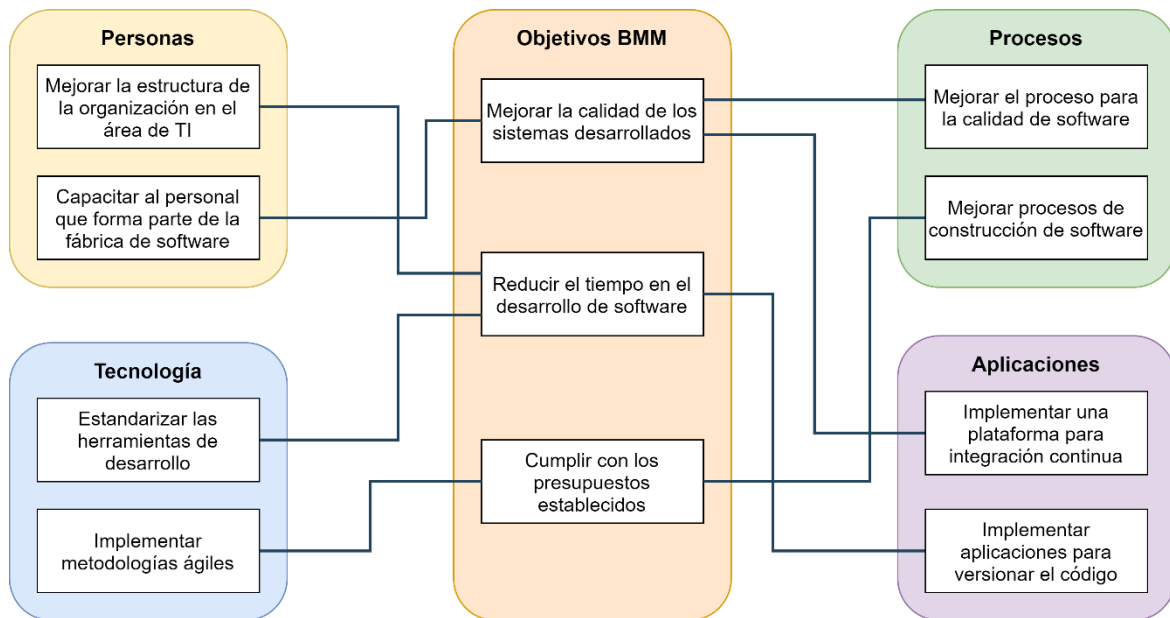


Figura 5. Diagrama conceptual requerimientos de alto nivel

2.2. Visionamiento y escenarios de la solución

La solución al problema actual de la empresa STYC, se basará en el fortalecimiento de la fábrica de software, implementando marcos de referencia como Scrum, para el desarrollo de software, se utilizará el marco de CMMI para mejorar procesos.

2.2.1. Referentes en la industria

Se toma como referencia el caso de implementación de Scrum en uno de los 3 Bancos más importantes del Ecuador y el caso de Cobis con su nivel 5 de CMMI (COBIS, 2018).

Uno de los tres principales bancos de la industria ecuatoriana, hace dos años inició un proceso de transformación digital, en este proceso se aplicó un marco de referencia como Scrum para el desarrollo de software.

El Banco con la implementación de Scrum ha logrado obtener una serie de beneficios, entre los principales podemos mencionar:

- Equipos de trabajo autoorganizados, conocidos como células.
- Optimizar el tiempo de desarrollo.
- Involucrar al cliente en cada etapa de trabajo del producto.
- Gran aceptación y respuesta al cambio.
- Entrega continua de software que cumple con los requerimientos.

Cobis es una empresa multinacional, líder de tecnología en el sector financiero, en el año 2018, alcanza el nivel 5, la valoración más alta de CMMI, este nivel también es conocido como Optimizado, lo que garantiza que la empresa constantemente mejora sus procesos, genera información y utiliza esta información para realizar proyecciones.

Alcanzar el nivel 5 le permite a la empresa:

- Ofrecer productos y servicios con garantía, que son generados con proceso de calidad.
- Alinear a la organización a cumplir con estándares internacionales.
- Tener procesos definidos que le permiten a los empleados mejorar sus actividades diarias.
- Mejorar procesos existentes.
- Analizar sus procesos utilizando herramientas estadísticas.

2.2.2. Marcos de referencia

CMMI (Capability Maturity Model Integration) son colecciones de buenas prácticas que ayudan a las organizaciones a mejorar sus procesos para el desarrollo, mantenimiento y operación de sistemas de software. (Asociación Española para la Calidad, 2019).

Se basa en 5 niveles de madurez:

Tabla 15

Niveles de madurez CMMI. Adaptado de (CMMI, 2010)

Nivel	Nombre	Descripción
Nivel 1	Inicial	Todos los procesos caóticos.
Nivel 2	Gestionado	Procesos reactivos, pero existen iniciativas para gestionar planificar procesos.
Nivel 3	Definido	Los procesos están definidos y existen estándares.
Nivel 4	Gestionado cuantitativamente	Los procesos tienen objetivos cuantitativos que permiten tomar decisiones.
Nivel 5	Optimizado	Los procesos de la organización se mejoran continuamente.

El modelo incluye una serie de áreas de procesos claves, que contienen:

- Metas específicas
Indican que es lo que persigue un área en particular.
- Metas genéricas
Engloban a las metas que persigue toda la organización.

Dentro de cada área de proceso encontraremos prácticas específicas, las que nos permiten alcanzar las metas específicas de cada área, el modelo es una colección de prácticas.

Los componentes del modelo se presentan en la siguiente ilustración

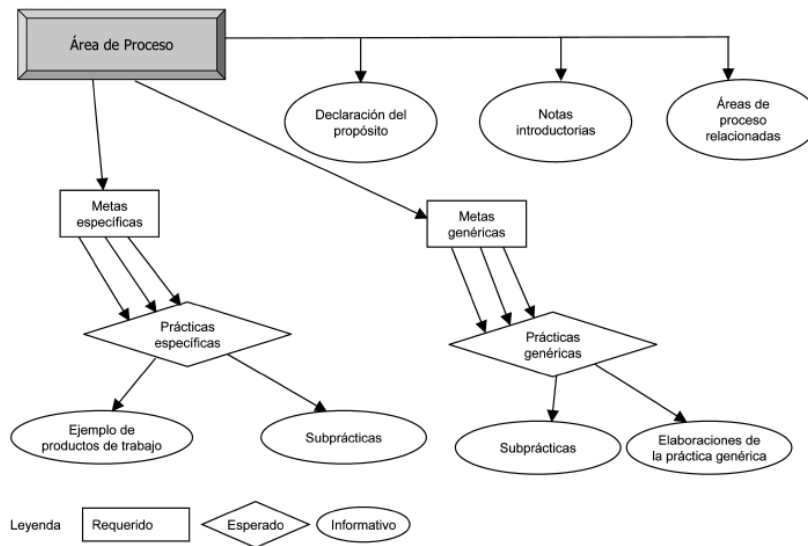


Figura 6. Componentes del modelo CMMI. (CMMI, 2010)

CMMI dispone de áreas de procesos, categorías y niveles de madurez, los cuales se exponen a mayor detalle en el Anexo 1.

Las metodologías ágiles implican cambios no solo a nivel de organización, sino también a nivel personal razón por la cual es necesario mapear las habilidades de las personas, para lo cual se utiliza SFIA.

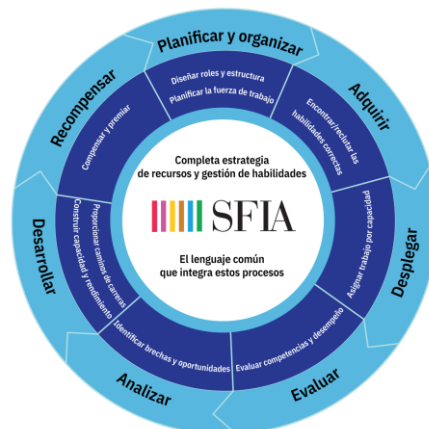


Figura 7. SFIA. (SFIA Foundation, 2021)

2.2.3. Visionamiento

La fábrica de software debe alcanzar un nivel 3, es decir definido, en la escala de CMMI, este nivel permite que los procesos estén descritos y comprendidos, convirtiéndose en un estándar dentro de la organización.

De igual manera para utilizar el marco de referencia Scrum necesitamos tener un equipo con los siguientes perfiles:

- Stakeholders
- Product Owner
- Scrum Master
- Equipo Scrum

2.3. Análisis de brechas

Se evaluó la fábrica de software actual tomando una serie de criterios, se estableció un referente y se planteó una arquitectura objetivo, lo que permite establecer las principales brechas que existen entre las diferentes arquitecturas.

Tabla 16

Análisis de brechas

Brechas		Actual	Objetivo	Referente	
Personas	Organizar	Reestructurar en base a roles y capacidades	2	3	5
	Contratar	Encontrar personal con las habilidades adecuadas	1	4	5
	Capacitar	Instruir al personal	1	3	5
	Evaluar	Medir las capacidades actuales de las personas	1	4	5

Procesos	Análisis de Decisiones y Resolución	DAR	1	4	5
	Gestión Integrada del Proyecto	IPM	2	4	5
	Definición de Procesos de la Organización	OPD	1	4	5
	Enfoque en Procesos de la Organización	OPF	2	4	5
	Formación en la Organización	OT	1	3	5
	Integración del Producto	PI	2	4	5
	Desarrollo de Requisitos	RD	2	4	5
	Gestión de Riesgos	RSKM	2	4	5
	Solución Técnica	TS	1	4	5
	Validación	VAL	2	4	5
	Verificación	VER	2	4	5
	Tecnología	Marco de referencia ágil	Scrum	1	4

La información de la tabla 16, permite generar la figura 8. Donde se analizan las principales brechas que existen.

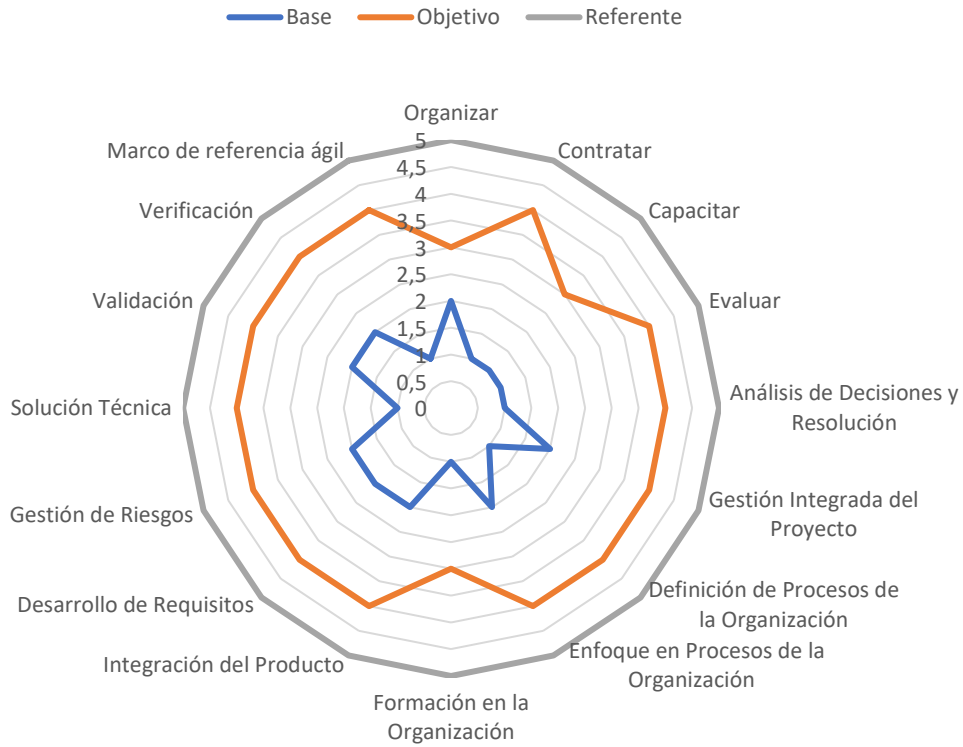


Figura 8. Análisis de brechas

La ilustración nos permite concluir que la arquitectura actual no es la adecuada, por tener una serie de inconvenientes, no se implementan buenas prácticas, para el desarrollo de software o para la gestión de procesos y no se tiene un manejo adecuado de las capacidades y habilidades de las personas del área de TI.

2.4. Definición de la arquitectura objetivo

2.4.1. Target de la arquitectura de negocio

La organización STYC, tiene que ser fortalecida ya que la estrategia actual no es la adecuada, su estructura organizacional no funciona correctamente y no existe un correcto manejo de los procesos.

Para poder corregir dichos inconvenientes es necesario:

- Alcanzar un nivel 3, en la escala de CMMI.
- Utilizar Scrum para el proceso de desarrollo de software.
- TI tiene que aportar en la ejecución de proyectos innovadores.
- Desarrollar proyectos, sistemas, aplicaciones que estén alineados a la estrategia del negocio.
- Personal de TI con las características necesarias para aportar al negocio.
- Interoperabilidad entre los diferentes aplicativos.

2.4.2. Target de la arquitectura de datos

Esta arquitectura, estructura los datos e información que se maneja en la organización, y para el objetivo que buscamos debe tener las siguientes características:

- Centralizar la información en repositorios fácilmente identificables.
- Procesos estandarizados para gestionar la información.
- Información estructurada para poder ser almacenada y respaldada.
- Desarrollar la capacidad para analizar la información.

2.4.3. Target de la arquitectura de aplicación

La arquitectura de aplicaciones permite identificar herramientas necesarias para procesar información y mantener la continuidad del negocio, actualmente la información no se encuentra orientada a servicios, el objetivo es implementar una plataforma que permita manejar el ciclo de vida de las APIs (Interfaz de programación de aplicaciones), mismo que nos permitirá centralizar la comunicación entre los diferentes servicios.

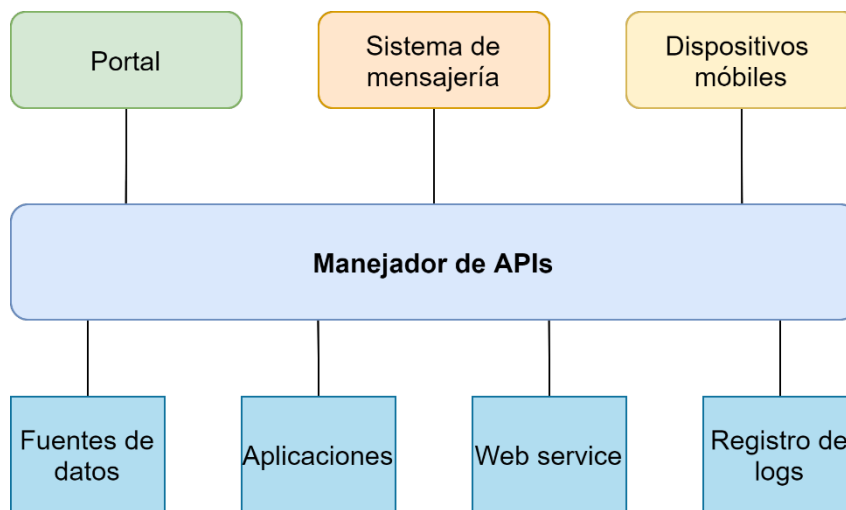


Figura 9. Arquitectura de aplicación

2.4.4. Target de la arquitectura de infraestructura base

En esta arquitectura, se agrupan los componentes de hardware, software y sus diferentes relaciones con otros recursos como bases de datos, aplicaciones, procesos de soporte, equipos de red.

Con la implementación de la arquitectura objetivo se busca que los nuevos sistemas, aplicaciones y demás desarrollos se integren a los sistemas ya existentes y lo hagan una forma transparente, sin afectar su funcionamiento. En las aplicaciones actuales es necesario fortalecer a Cloud Computing, esto nos brindará flexibilidad, reducción de costos, seguridad y escalabilidad.

2.5. Road Map

Se define el mapa de ruta para alcanzar la arquitectura objetivo, utilizando el método ADM de Togaf, iniciando en la fase preliminar llegando hasta la fase de migración.



Figura 10. Hoja de Ruta

3. ARQUITECTURA DE NEGOCIO DE LA FÁBRICA DE SOFTWARE

La arquitectura de negocio describe con detalle todos los procesos de la fábrica de software, estructura organizativa, aporte de los técnicos y especialistas con sus roles desde una visión articulada, identificando las brechas entre el nuevo modelo y la situación actual.

3.1. Modelo de procesos

3.1.1. Arquitectura de negocios base

Actualmente la fábrica de software de STYC utiliza la metodología Waterfall (cascada), que se caracteriza por el desarrollo en etapas de una superior a una inferior, para ir de un nivel a otro es necesario finalizar el anterior, utilizar este método provoca que los resultados de los diferentes proyectos puedan ser vistos en fases avanzadas del mismo.

A continuación, se detallan todos los procesos que intervienen en una metodología en cascada:

- **Análisis:** El líder de desarrollo junto a una persona del área comercial se reúnen con el cliente para definir los objetivos, requerimientos y viabilidad del nuevo proyecto, como resultado se genera el documento de especificación de requisitos.
- **Diseño:** En esta fase se modela la arquitectura de software, se establecen prototipos para interfaces del usuario y estructuras de base de datos, tomando como base el documento de especificaciones.
- **Implementación:** El equipo de desarrollo (programadores) asignados al proyecto convierten los diseños en lenguaje de software, construyen diferentes módulos y unidades, el resultado es un programa en una versión de prueba.

- **Verificación:** Los elementos que han sido programados, son analizados por el área de aseguramiento de la calidad (QA), se realizan pruebas de componentes individuales, así como de integración con todo el ecosistema. Los errores encontrados son reportados a los programadores para que sean solucionados.
- **Mantenimiento:** El software obtenido en la fase de verificación es desplegado en producción, el cliente observa el resultado del desarrollo y solicita cambios.

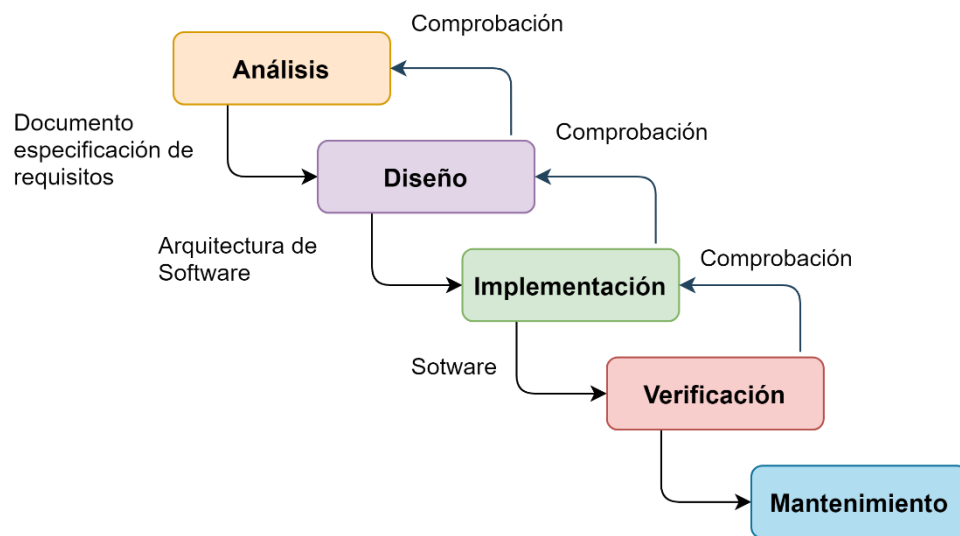


Figura 11. Modelo del proceso de desarrollo de software

3.1.2. Detalle de la arquitectura objetivo

La arquitectura objetivo toma como referente al marco de trabajo Scrum para el manejo del proyecto y para el desarrollo de software se utilizan las prácticas de ingeniería de CMMI-DEV.

3.1.2.1. Scrum

“Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos.” (Schwaber & Sutherland, 2020)

Scrum se basa en dividir un problema que ha sido expuesto por el cliente en varios pequeños problemas, denominados historias de usuario, estas se agregan a una lista priorizada denominada product backlog, el equipo Scrum va resolviendo cada una de las historias y realizando entregas continuas al cliente en intervalos de tiempo denominados sprint.

Scrum posee los siguientes artefactos (Schwaber & Sutherland, 2020):

- **Product Backlog:** Es una lista de tareas pendientes por realizar en el producto, es un artefacto administrado por el Product Owner y se obtiene del trabajo conjunto entre cliente, Stakeholder y Product Owner.
- **Sprint Backlog:** Es una lista de tareas pendientes a trabajar durante un Sprint, es administrado por el Development Team y tiene por objetivo, transparentar el estado de cada tarea en el transcurso del sprint.
- **Incremento:** El resultado de cada Sprint es un incremental, una tarea completa que genera valor al cliente.

Un Sprint es un ciclo, una iteración que se repite en cada proyecto de Scrum.

En total hay diecinueve procesos fundamentales de Scrum que se aplican a todos los proyectos. Estos procesos se agrupan en cinco fases y se muestran a continuación. (SCRUMstudy, 2017)

Tabla 17

Procesos de Scrum (SCRUMstudy, 2017)

Fase	Proceso
Inicio	<ol style="list-style-type: none"> 1. Crear la visión del proyecto 2. Identificar al Scrum Master y Stakeholder(s) 3. Formar Equipos Scrum 4. Desarrollar épica(s) 5. Crear el Backlog Priorizado del Producto 6. Realizar la planificación de lanzamiento
Planificación y estimación	<ol style="list-style-type: none"> 7. Crear historias de usuario 8. Estimar historias de usuario 9. Comprometer historias de usuario 10. Identificar tareas 11. Estimar tareas 12. Crear el Sprint Backlog
Implementación	<ol style="list-style-type: none"> 13. Crear entregables 14. Realizar Daily Standup 15. Refinar el Backlog Priorizado del Producto
Revisión y retrospectiva	<ol style="list-style-type: none"> 16. Demostrar y validar el sprint 17. Retrospectiva del sprint
Revisión y retrospectiva	<ol style="list-style-type: none"> 18. Enviar entregables 19. Retrospectiva del proyecto

En el Anexo 3 se presenta una comparación entre el método Waterfall y el marco de trabajo Scrum relacionando el valor a ser entregado con el tiempo.

3.1.2.2. Prácticas de CMMI-DEV (Área de procesos de Ingeniería)

CMMI-DEV y su área de proceso de Ingeniería cubren las actividades de desarrollo y de mantenimiento que se utilizan en todas las disciplinas de ingeniería.

Las cinco áreas de proceso de Ingeniería de CMMI-DEV son las siguientes:

- Integración del Producto (PI).
- Desarrollo de Requisitos (RD).
- Solución Técnica (TS).
- Validación (VAL).
- Verificación (VER). (CMMI, 2010)

La Tabla 18, muestra las cinco áreas de proceso de Ingeniería de CMMI-DEV, y las prácticas que se adaptarán e implementarán en la fábrica, para el desarrollo de software.

Tabla 18

Áreas de proceso de Ingeniería CMMI-DEV. Adaptado de (CMMI, 2010)

CMMI-DEV			
Áreas de Procesos	Propósito	Meta Específica (SG)	Práctica Específica (SP)
Desarrollo de Requisitos (RD)	Reducir, analizar y establecer los requisitos de cliente, de producto y de componente de producto.	SG1: Desarrollo de requisitos de cliente. SG2: Desarrollar los requisitos de producto.	SP1: Transformar las necesidades de las partes interesadas. SP1: Establecer los requisitos del producto SP2: Identificar los requisitos de interfaz.
Integración del Producto (PI)	Ensamblar el producto a partir de sus componentes, asegurar que el producto, una vez	SG1: Prepararse para la integración del producto.	SP1: Establecer una estrategia de integración. SP2: Establecer el entorno de integración del producto. SP3: Establecer los procedimientos y criterios.

	integrado, se comporta correctamente	SG3: Ensamblar los componentes y entregar el producto	SP1: Evaluar los componentes de producto ensamblado SP2: Empaquetar y entregar el producto o componente
Solución Técnica (TS)	Seleccionar, diseñar e implementar soluciones para los requisitos. Las soluciones, los diseños y las implementaciones engloban productos, componentes de producto y procesos del ciclo de vida relativos al producto	SG1: Seleccionar soluciones de componente de producto. SG2: Desarrollar el diseño.	SP1: Seleccionar las soluciones de componente de producto. SP1: Establecer un paquete de datos técnicos. SP2: Diseñar las interfaces usando criterios.
Validación (VAL)	Demostrar que un producto o componente de producto cumple con su uso previsto cuando se ubica en el entorno previsto	SG1: Preparar la validación. SG2: Validar el producto o los componentes del producto.	SP1: Seleccionar los productos para la validación. SP2: Establecer el entorno de validación. SP3: Establecer los procedimientos y criterios de validación. SP1: Realizar la validación. SP2: Analizar los resultados de la validación.
Verificación (VER)	Asegurar que los productos de trabajos seleccionados cumplen los requisitos especificados	SG1: Preparar la verificación. SG2: Realizar las revisiones entre pares	SP1: Seleccionar los productos de trabajo para verificación. SP2: Establecer el entorno de verificación. SP3: Establecer los procedimientos y los criterios de verificación. SP1: Preparar las revisiones entre pares. SP2: Realizar la revisión entre pares. SP3: Analizar los datos de la revisión entre pares.

3.1.2.3. Integración de Scrum y prácticas CMMI-DEV

La fábrica de software actualmente no tiene un manejo adecuado de los proyectos, y el marco de trabajo Scrum nos permite fortalecer este proceso, otro problema de la fábrica es el mejorar la calidad del software para poder reducir los errores en producción, para optimizar el desarrollo del software se implementa en cada Sprint las prácticas de las áreas de proceso de Ingeniería de CMMI-DEV, donde se aplican procesos enfocados en el desarrollo de software, la integración de Scrum junto con las prácticas de CMMI-DEV nos permitirán fortalecer la fábrica tanto en la gestión de proyectos como en el desarrollo de software.

A continuación, se presenta un gráfico con la arquitectura objetivo, donde se integra el marco de trabajo Scrum y las prácticas de las áreas de procesos de Ingeniería de CMMI-DEV, definidos en la tabla 18.

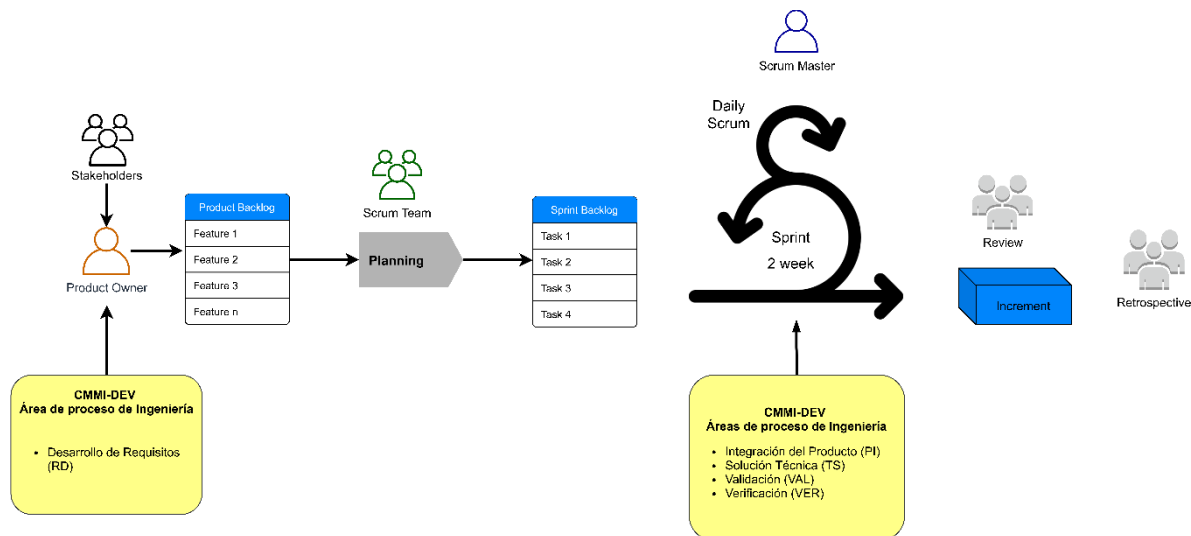


Figura 12. Arquitectura objetivo Scrum y Cmmi-Dev

3.1.3. Análisis de brechas

Al comparar la arquitectura actual vs la arquitectura objetivo se generan las brechas, así como las iniciativas que las cierran y se resumen en el siguiente gráfico.

Tabla 19

Análisis de brechas procesos actuales vs objetivo

Proceso	Estado actual (AS IS)	Estado deseado (TO BE)	Existe brecha	Brecha a cerrar	Iniciativa
Crear la visión del proyecto	El proyecto nace de interacciones entre clientes y personas del área comercial	Crear una visión global y detallada del proyecto	SI	Manejo adecuado desde el inicio del proyecto, levantando las necesidades del cliente	Crear un equipo dedicado, para elaborar el caso de negocio, así como un enfoque para todo el proyecto.
Identificar al Scrum Master y Stakeholder(s)	No existe este proceso	Seleccionar al Scrum Master y Stakeholders mediante un proceso adecuado	SI	Seleccionar al Scrum Master y Stakeholders.	Implementar un proceso para identificar al Scrum Master y Stakeholder, que se adapten a las necesidades del negocio.
Formar Equipos Scrum	No existe este proceso	Disponer de Equipo Scrum, con los perfiles adecuados, para cualquier proyecto	SI	Selección un equipo Scrum	El Product Owner, con la colaboración del Scrum Master seleccionan a los miembros del Equipo Scrum
Desarrollar épica(s)	No existe este proceso	Establecer un repositorio donde se pueda desarrollar las épicas (Historias macro del proyecto)	SI	Desarrollo de las épicas	Implementar Jira para describir y gestionar las épicas, que permitirán tener una visión global del proyecto.
Crear el Backlog Priorizado del Producto	No existe este proceso	Organizar y priorizar las épicas e historias de usuario	SI	Elaborar el Backlog, para la priorización de historias	Utilizar Jira para incorporar todas las épicas e historias de usuario en un Backlog, que tiene que ser priorizado por el Product Owner
Realizar la planificación de lanzamiento	No existe este proceso	Disponer de un cronograma de implementación	SI	Creación de un cronograma, que permita, planificar la implementación.	Elaborar un cronograma de planificación, para la implementación

		ón y planificación			por fases del proyecto.
Crear historias de usuario	No existe este proceso	Elaborar las historias de usuario, que permitirán plasmar claramente los requisitos del cliente.	SI	Implementar el proceso para la creación de historias de usuario	Capacitar al Product Owner y Equipo Scrum para crear las historias de usuario y criterios de aceptación.
Estimar historias de usuario	No existe este proceso	Estimar las historias de usuario, para obtener establecer un cronograma.	SI	Desarrollar las habilidades para poder estimar	Implementar el proceso de planning poker para poder estimar las historias de usuario.
Identificar tareas	No existe este proceso	Disponer de tareas que se tengan que realizar en cada historia de usuario	NO		
Crear el Sprint Backlog	No existe este proceso	Disponer de la lista de tareas que se están desarrollando en el Sprint	SI	Implementar el Sprint Backlog	Utilizar Jira para elaborar el Sprint Backlog para que las diferentes historias y responsables puedan ser observados.
Realizar Daily Standup	Se realiza una reunión cada mañana donde los miembros del equipo exponen en que van a estar trabajando		NO		
Refinar el Backlog Priorizado del Producto	No existe este proceso	Revisa cada una de las historias del backlog para poder solventar dudas e inquietudes	SI	Implementar una actividad para poder revisar el Backlog	Desarrollar reuniones durante el sprint para priorizar el Backlog del producto
Demostrar y validar el sprint	No existe este proceso	Presentar al Product Owner y	SI	Implementar proceso para mostrar resultados	Crear una reunión donde se revisen los entregables del

		Stakeholders los resultados del Sprint			Sprint al Product Owner y a los Stakeholders
Retrospectiva del sprint	No existe este proceso	Exponer y documentar los aciertos y errores obtenidos durante el Sprint	SI	Implementar la retrospectiva	Crear una reunión entre el Product Owner, Scrum Master y Development Team, para analizar las lecciones aprendidas durante el Sprint
Enviar entregables	No existe este proceso	Entregar a los Stakeholders documentos con observaciones de cada sprint	SI	Implementar un entregable y enviarlo a los involucrados	Elaborar un documento que será entregado a los Stakeholders relevantes, con las conclusiones más importantes del Sprint.
Retrospectiva del proyecto	No existe este proceso	Documentar las lecciones aprendidas durante el proyecto por parte de todos los involucrados.	SI	Implementar la retrospectiva del proyecto	Elaborar una reunión para concluir el proyecto, donde Stakeholders y el Equipo Scrum, documentan las lecciones aprendidas
Desarrollo de requisitos de cliente	Las necesidades del cliente son atendidas por una persona comercial	Identificar las necesidades de las partes interesadas.	SI	Las necesidades del cliente no son elaboradas correctamente, lo que genera objetivos no muy claros.	Utilizar Jira para levantar las necesidades del cliente, trabajando de forma conjunta, para generar los requisitos y los resultados deben ser visibles para todas las partes interesadas
Desarrollar los requisitos de producto	No existe un flujo para mapear de manera correcta los requisitos del producto, esta actividad es realizada al inicio del	Establecer los requisitos del producto Identificar los requisitos de interfaz	SI	Las necesidades del producto no son mapeadas de manera adecuada, lo que genera en ocasiones que el diseño de soluciones no esté de acuerdo a lo que esperaba el cliente	Utilizar Jira para levantar las necesidades del producto, trabajando con el cliente como principal fuente de información, los resultados deben ser visibles para

	proyecto por las personas comerciales				todas las partes interesadas
Prepararse para la integración del producto.	En las etapas tempranas no se contempla una posible integración	Establecer una estrategia de integración. Establecer el entorno de integración del producto. Establecer los procedimientos y criterios	SI	La fase de integración es importante ya que cubre estrategias, entorno integral, procedimientos y criterios, que ayudarán a llevar el producto correctamente desde el inicio del mismo.	Crear reuniones, grupos de trabajo y documentos para la integración del producto, donde se consideren todos los procesos relevantes del producto
Ensamblar los componentes de producto y entregar el producto	Los componentes del producto se ensamblan, de acuerdo a las definiciones y no existe un diseño que cumpla con las necesidades	Evaluar los componentes del producto ensamblado. Empaquetar y entregar el producto o componente	SI	Los componentes previo a su ensamble y entrega deben cumplir con el diseño establecido	Implementar herramientas de automatización como SonarQube que garantizan la calidad del código
Seleccionar soluciones de componente de producto	No se generan ni analizan posibles alternativas de solución	Seleccionar las soluciones de componente de producto	SI	Generar varias soluciones y seleccionar la que mejor cumpla con los requerimientos del cliente	Crear reuniones para evaluación de las soluciones y como resultado se obtendrá la solución más adecuada.
Desarrollar el diseño	Actualmente si se realizan los diseños utilizando criterios e interfaces que sirven de guía	Establecer un paquete de datos técnicos. Diseñar las interfaces usando criterios.	NO	No existe brecha	
Preparar la validación	El área de Qa prepara sus pruebas, tiene un ambiente para las mismas y establece criterios para	Seleccionar los productos para la validación. Establecer el entorno de validación. Establecer los procedimientos	NO	No existe brecha	

	cada una de las pruebas	os y criterios de validación			
Validar el producto o los componentes del producto	Se valida el producto y en base a los resultados se aprueba o niega el producto, la validación asegura que se construyó lo correcto	Realizar la validación. Analizar los resultados de la validación.	NO	No existe brecha	
Preparar la verificación	Actualmente no se realiza un proceso de verificación, este nos permite comprender que lo construido se hizo correctamente	Seleccionar los productos de trabajo para verificación. Establecer el entorno de verificación. Establecer los procedimientos y los criterios de verificación	SI	No existe un proceso definido, que certifique que lo construido se lo hizo correctamente, actualmente solo se comprueba funcionalidad.	Elaborar un documento que permita conocer los diferentes pasos a seguir para la verificación correcta de cualquier componente.
Realizar las revisiones entre pares	No se realiza una revisión cruzada de actividades entre los miembros del equipo	Preparar las revisiones entre pares. Realizar la revisión entre pares. Analizar los datos de la revisión entre pares	SI	Cada una de las tareas realizadas tiene que ser revisada por otro miembro del equipo para mejorar la calidad del código o encontrar bugs.	Crear reuniones entre pares previo a la entrega de cada tarea, para poder obtener puntos de mejora y corregir de forma temprana.

Las principales brechas a cerrar se presentan en la implementación de Scrum, principalmente en la planificación, creación de historias, formación de equipo, manejo del backlog y las diferentes ceremonias dichas brechas se cierran con la formación de equipos capacitados, implementación de herramientas colaborativas, creación de reuniones especializadas según la guía del marco Scrum.

En lo relacionado al desarrollo de software existen brechas en los requerimientos, integración, ensamblar componentes, selección, verificación y revisión entre pares para lo cual se tomarán medidas como utilizar Jira para un manejo correcto de los requerimientos, herramientas como SonarQube para mejorar la calidad del software, la creación de reuniones para revisión entre pares.

3.2. Modelo organizacional y talento humano

3.2.1. Arquitectura base

Actualmente la fábrica de software se encuentra conformada por 20 personas que cumplen diferentes actividades:

Tabla 20

Talento humano

Cargo	Cantidad	Actividades
Gerente de Tecnología y Sistemas	1	<ul style="list-style-type: none"> • Gestionar y controlar los proyectos. • Verificar que se cumpla la planificación y tiempos de entrega. • Solventar los riesgos que se presenten en los proyectos.
Líder de producción	1	<ul style="list-style-type: none"> • Controlar al equipo de monitoreo e infraestructura
Equipo de monitoreo	3	<ul style="list-style-type: none"> • Identificar problemas en las aplicaciones en producción. • Manejar y solventar los incidentes reportados.
Equipo de infraestructura	2	<ul style="list-style-type: none"> • Mantenimiento y soporte de la infraestructura, servidores, bases de datos. • Solucionar problemas de infraestructura. • Accesos a los servidores, para poder realizar cambios en ambientes productivos.
Quality assurance (QA)	2	<ul style="list-style-type: none"> • Verificar que los desarrollos cumplan los requerimientos del cliente. • Controlar la calidad de los diferentes desarrollos. • Aprobar desarrollos para despliegues en ambientes de producción.
Líder de desarrollo	2	<ul style="list-style-type: none"> • Liderar los proyectos asignados • Buscar soluciones a las necesidades planteadas. • Controlar los plazos, fechas y presupuestos establecidos. • Asignar y controlar actividades al equipo de desarrollo.
Equipo de desarrollo	6	<ul style="list-style-type: none"> • Transformar los diseños en software funcional • Cumplir con las tareas asignadas en el tiempo establecido.

		<ul style="list-style-type: none"> • Liberar un software funcional para el área de calidad.
Líder de innovación	1	<ul style="list-style-type: none"> • Investigar nuevas tecnologías que contribuyan con el área de tecnología. • Asignar y controlar actividades al equipo de innovación. • Mostrar resultados de los proyectos de investigación e innovación desarrollados.
Equipo de Innovación	2	<ul style="list-style-type: none"> • Implementar, desarrollar y probar diferentes tecnologías. • Generar documentación con los resultados de las investigaciones

En la fábrica de software las personas se encuentran distribuidas de acuerdo al siguiente organigrama, cumpliendo con las actividades de la tabla 20.

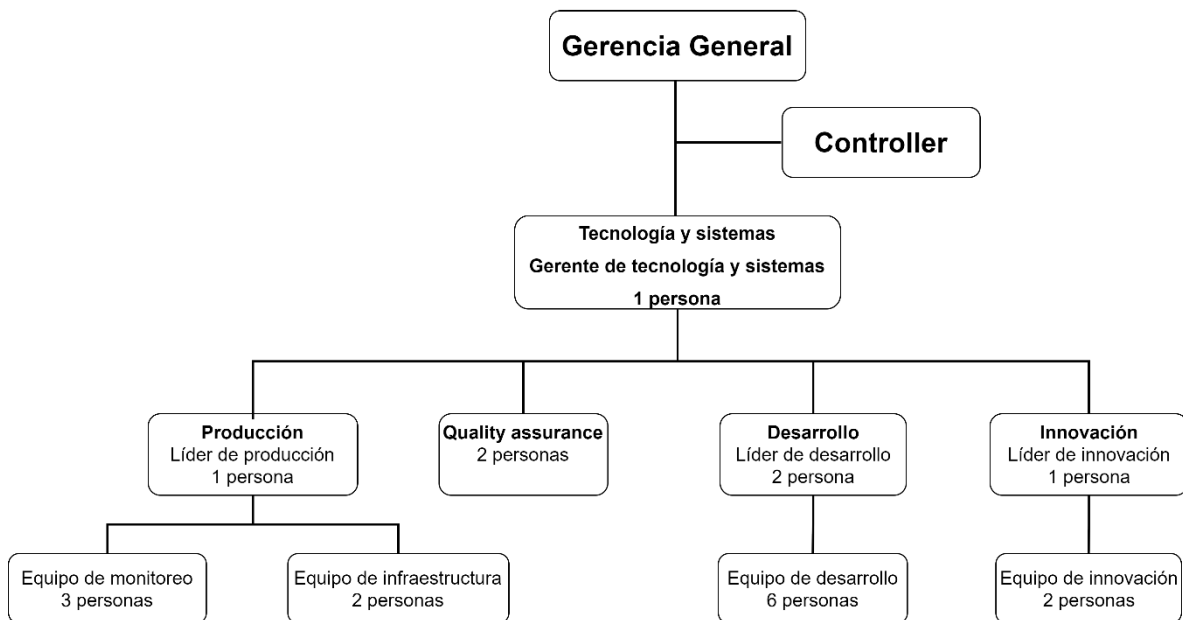


Figura 13. Estructura organizativa y talento humano

3.2.2. Detalle de la arquitectura objetivo

Para la implementación de Scrum es necesario adaptarnos a los roles del marco de trabajo, y necesitamos crear un Scrum Team el cual está conformado por:

- **Product Owner:** Es la persona responsable de maximizar el valor del producto resultante del trabajo del Scrum Team. Se relaciona estrechamente con el cliente para poder plasmar sus necesidades en un artefacto denominado Product Backlog, asegurando que sean priorizadas, transparentes, visibles y puedan ser entendidas.
- **Scrum Master:** Es la persona responsable de establecer el marco de trabajo como se define en la Guía de Scrum. Lo hace ayudando a todos a comprender la teoría y la práctica de Scrum, tanto dentro del Scrum Team como de la organización.
- **Developers:** Son las personas del Scrum Team que se comprometen a crear un incremento utilizable en cada Sprint y son responsables de adaptar su plan para conseguir el objetivo, cumplir con los requerimientos establecidos en las historias de usuario y gestionar el Sprint Backlog. (Schwaber & Sutherland, 2020)

Para conformar el equipo developers, se toma como referencia la estructura de uno de los bancos más grandes del Ecuador.

- Arquitecto
- Desarrolladores
- Control de calidad (QA)
- Devops
- Experiencia de Usuario (Ux/Ui)

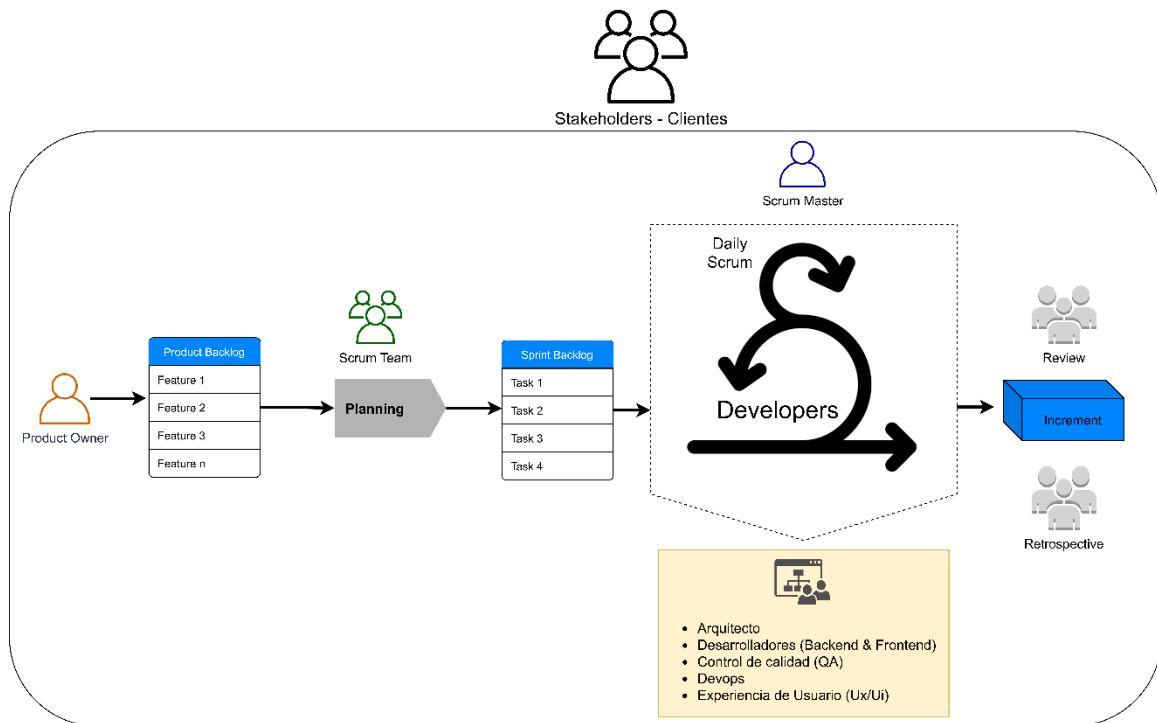


Figura 14. Scrum y equipo developers

En la arquitectura de negocios se propuso una estructura ágil, el equipo developers estará conformado por especialistas en las áreas antes mencionadas, multidisciplinarios y auto organizados, con capacidad de entregar incrementos continuos al cliente, con la calidad y tiempo planificados.

La estructura organizativa se verá afectada, en la siguiente imagen se muestra la estructura objetivo con el número de personas en cada área.

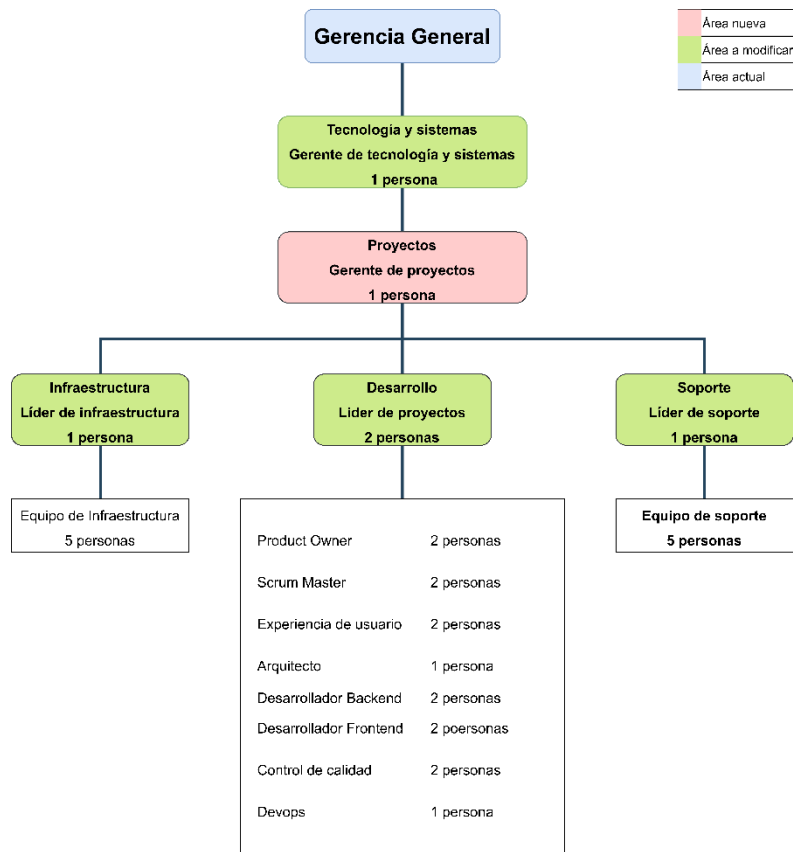


Figura 15. Estructura organizativa objetivo

La fábrica de software dentro de la estructura de la organización necesita contar con nuevos roles, la siguiente tabla muestra los roles, número de recursos y responsabilidades, que se necesitan para iniciar las actividades, para definir los perfiles y responsabilidades se utiliza el modelo de referencia SFIA.

Tabla 21

Perfil personal para la arquitectura objetivo (SFIA Foundation, 2021)

Rol	Número de recursos	Perfil profesional	Responsabilidades
Gerente de tecnología y sistemas	1	<ul style="list-style-type: none"> Habilidades de liderazgo, comunicación, negociación. 	<ul style="list-style-type: none"> Gestionar al equipo que pertenece al área de TI

		<ul style="list-style-type: none"> • Dominio de Sistemas Informáticos, redes y diseño de software • Habilidades de comunicación verbal y escrita. • Conocimiento de arquitectura, diseño de software y marcos ágiles. • Prioriza actividades • Habilidades de negociación. • Conocimiento de metodologías ágiles, y marcos de referencia para el diseño y desarrollo de software 	<ul style="list-style-type: none"> • Implementar procedimientos de seguridad. • Coordinar y controla las áreas bajo su dominio. • Administrar el presupuesto asignado a TI
Gerente de Proyectos	1	<ul style="list-style-type: none"> • Entiende el impacto de las nuevas tecnologías. • Comunica con eficacia en todos los niveles a audiencias técnicas y no técnicas. • Promueve el cumplimiento de la legislación y la necesidad de servicios, productos y prácticas de trabajo. • Toma la iniciativa para mantener actualizadas las habilidades propias y de sus colegas. • Gestiona y mitiga el riesgo. • Desempeña un rol de liderazgo en sus áreas de responsabilidad. 	<ul style="list-style-type: none"> • Asumir la responsabilidad total por la definición, la documentación y la finalización exitosa de proyectos complejos. • Adoptar y adapta los métodos y las herramientas de gestión de proyectos. • Asegurar que se mantengan procesos efectivos de control de proyectos, control de cambios, gestión de riesgos y pruebas. • Supervisar y controla los recursos.
Líder de Proyectos	2	<ul style="list-style-type: none"> • Habilidad para manejar proyectos. • Conocimientos de modelos de madurez CMMI • Habilidad para elaborar documentación. • Habilidad para gestionar y asignar requisitos. • Habilidades de comunicación verbal y escrita • Conocimiento de arquitectura, diseño de software y marcos ágiles. 	<ul style="list-style-type: none"> • Liderar el proyecto • Control y seguimiento de tareas. • Cumplir con los costos y tiempos de entrega establecidos. • Elaborar informes del avance y problemas. • Identificar riesgos • Estimar tiempos y responsables para el desarrollo del proyecto.
Product Owner	2	<ul style="list-style-type: none"> • Habilidad de gestionar proyectos. • Dominio de metodologías ágiles 	<ul style="list-style-type: none"> • Construir una visión del producto y luego plasmarla en historias de usuario.

		<ul style="list-style-type: none"> • Habilidades de comunicación verbales y escritas. • Dominio en de herramientas para la elaboración y seguimiento de flujos • Conocimientos en ingeniería de software, marketing, ofimática • Manejo de herramientas para prototipado 	<ul style="list-style-type: none"> • Transmitir esa visión al equipo de desarrollo que va a construir el software. • Conocimientos sólidos respecto de los usuarios, el mercado, la competencia y las tendencias de futuro para el dominio o el tipo de sistema que se está desarrollando. (Estayno & Meles, 2014)
Scrum Master	2	<ul style="list-style-type: none"> • Dominio de metodologías, marcos de referencia relacionadas con agilidad como Scrum, Kanban, XP, Lean. • Experiencia trabajando con equipo. • Habilidades blandas, facilitación y resolución de problemas. • Conocimiento en actividades relacionadas con Desarrollo de Software. • Resolución de conflictos 	<ul style="list-style-type: none"> • Gestionar y asegura que el marco de referencia Scrum se cumpla. • Eliminar impedimentos que se presentan en los equipos. • Apoyar al producto Owner. • Asegurar que el equipo genere valor sprint tras sprint • Identificar acciones de mejora. • Llevar métricas del equipo
Experiencia de Usuario	2	<ul style="list-style-type: none"> • Dominio de herramientas para el desarrollo de prototipos • Dominio de herramienta de diseño. • Conocimientos de diseño web y móvil. • Conocimientos de metodologías, marcos de trabajo orientados al cliente como Desing Thinkig. • Habilidades blandas para interactuar con clientes • Dominio de herramientas para manejo de flujos • Formación técnica en diseño gráfico, diseño industrial, informática o carreras afines. 	<ul style="list-style-type: none"> • Diseñar y desarrolla tareas, interacción e interfaces digitales y fuera de línea para cumplir con los requisitos de usabilidad y accesibilidad acordados. • Traducir conceptos en productos y prototipos y captura los comentarios de los usuarios para mejorar los diseños. • Especificar las herramientas, los métodos y los patrones de diseño apropiados. • Evaluar opciones de diseño alternativas y recomienda diseños. • Interpretar y sigue las pautas y lineamientos de diseño visual de la marca para crear una experiencia de usuario consistente e impactante.

Arquitecto	1	<ul style="list-style-type: none"> • Demuestra un liderazgo claro. • Experiencia en diseño y construcción de software. • Dominio de arquitectura, diseño y desarrollo de software para backend, frontend. • Experiencia en el diseño, modelamiento y manejo de bases de datos. • Formación técnica en Ingeniería de sistemas, informática o carreras afines. • Dominio de metodologías ágiles • Dominio de tecnologías nuevas e innovadoras. 	<ul style="list-style-type: none"> • Liderar el desarrollo de arquitecturas para sistemas complejos. • Asumir toda la responsabilidad del equilibrio entre requerimientos funcionales, de calidad de servicio y de gestión de sistemas dentro de un ámbito importante de la organización. • Establecer la política y la estrategia para la selección de componentes de la arquitectura de sistemas y coordina las actividades de diseño, fomentando la disciplina para asegurar la consistencia. • Asegurar que se cumplan los estándares apropiados
Desarrollador Backend	2	<ul style="list-style-type: none"> • Experto en el desarrollo e integración de APIS, servicios. • Conocimientos de metodologías ágiles y desarrollo de proyectos. • Formación técnica en Ingeniería de sistemas, informática o carreras afines. • Dominio de lenguajes de programación, bases de datos, pruebas. • Trabajo en equipo 	<ul style="list-style-type: none"> • Desarrollar políticas organizacionales, estándares y directrices para la construcción y refactorización de software. • Planificar y lidera actividades de construcción de software para proyectos de desarrollo estratégico, grandes y complejos. • Desarrollar nuevos métodos y capacidades organizacionales e impulsa la adopción y el cumplimiento de políticas y estándares.
Desarrollador Frontend	2	<ul style="list-style-type: none"> • Especialista en el desarrollo de aplicaciones web y móviles. • Dominio de lenguajes de programación para Frontend, React, Angular, Html, Javascript, React Native. • Conocimientos de metodologías ágiles y desarrollo de proyectos. 	<ul style="list-style-type: none"> • Desarrollar políticas organizacionales, estándares y directrices para la construcción y refactorización de software. • Planificar y lidera actividades de construcción de software para proyectos de desarrollo estratégico, grandes y complejos.

		<ul style="list-style-type: none"> • Formación técnica en Ingeniería de sistemas, informática o carreras afines. 	<ul style="list-style-type: none"> • Desarrollar nuevos métodos y capacidades organizacionales e impulsa la adopción y el cumplimiento de políticas y estándares.
Control de calidad (QA)	2	<ul style="list-style-type: none"> • Dominio de metodologías, marcos de referencia y modelos para garantizar la calidad del software. • Experiencia en testing con la pirámide de pruebas E2E, integración y unitarias. • Conocimientos de lenguajes de programación backend y frontend. • Conocimientos del lenguaje SQL, para consultas de base de datos. • Conocimientos de metodologías ágiles y desarrollo de proyectos. • Formación técnica en Ingeniería de sistemas, informática o carreras afines. • Dominio de herramientas para testing. 	<ul style="list-style-type: none"> • Determinar la política de prueba y es el dueño de los procesos de soporte, incluidas las pruebas de seguridad del software. • Asumir la responsabilidad de la gestión de todas las actividades de prueba. • Administrar todos los riesgos asociados con las pruebas. • Evaluar y aconseja sobre la practicidad de las alternativas del proceso de prueba. • Evaluar las capacidades de los procesos de desarrollo y prueba de los proveedores. • Determinar los estándares de pruebas del proyecto para todas las fases.
Devops	1	<ul style="list-style-type: none"> • Formación técnica en Ingeniería de sistemas, informática o carreras afines. • Dominio de plataformas para despliegue continuo Azure, Google Cloud • Experiencia en lenguajes de programación backend, frontend, procesos de pruebas. • Conocimientos de arquitectura y diseño de software. • Manejo de bases de datos. 	<ul style="list-style-type: none"> • Implementa prácticas automatizadas para asegurar entrega continua. • Control de versiones. • Despliegue continuo. • Compilación y monitoreo automático.

3.2.3. Impacto de la arquitectura objetivo en la situación actual

Al implementar marcos de trabajo como Scrum junto con CMMI-DEV en la fábrica de software los procesos descritos en la arquitectura objetivo permiten desarrollar la capacidad para reaccionar de forma inmediata a los diferentes cambios, así como también se garantiza la entrega de software de calidad.

La estructura organizativa, así como los perfiles de cada miembro de la fábrica deben adaptarse a la nueva arquitectura objetivo, lo que permitirá a la fábrica optimizar el trabajo de cada una de las personas, generará nuevas oportunidades de empleo lo cual motivará a los empleados a mejorar sus habilidades en busca de alcanzar uno de estos puestos.

3.2.4. Road Map

Se define un plan de implementación de la arquitectura de negocio objetivo que impactará a procesos, personas y organización para lo cual se elabora el siguiente diagrama.

HOJA DE RUTA	2021			
	Q2			
	mar	abr	may	jun
Diagnostico Inicial	15 Días			
Planificación y estructuración	15 Días			
Adaptación de procesos		30 Días		
Validación de procesos			10 Días	
Implementación			10 Días	
Verificación			5 Días	
Cierre				5 Días

Figura 16. Hoja de ruta arquitectura de negocio objetivo

4. ARQUITECTURA DE APLICACIONES Y DE DATOS

La arquitectura de aplicaciones está conformada por programas clasificados en diferentes categorías que se describen a continuación.

4.1. Arquitectura de aplicaciones

4.1.1. Arquitectura actual de aplicaciones base

En la siguiente tabla se describe un inventario de las aplicaciones que se manejan actualmente en la fábrica de software.

Tabla 22

Arquitectura de aplicaciones actual. Catálogo de aplicaciones

Proceso	Aplicación	Versión	Descripción
Diseño	Draw.io	13.9.9	Herramienta para crear diagramas de diferentes tipos, posee diferentes plantillas para elaborar flujos, diagramas de secuencia, uml entre otros
	Proto.io	2.11	Entorno para la creación de prototipos, utilizado para el diseño y modelamiento en dispositivos móviles
	Adobe Illustrator	8	Software que permite la creación de elementos gráficos.
Desarrollo	Visual Studio Code	1.54.1	Editor de código fuente, que permite trabajar con diferentes lenguajes de programación como Php, Java, Javascript, Css, HTML, Python
	NetBeans	8.2	Entorno de desarrollo integrado (IDE), se utiliza para la creación de aplicaciones web, móviles y de escritorio, incluye soporte para trabajar con lenguajes de programación como Java, Php, C/C++
	IntelliJ IDEA Community Edition	2021.1.3	Es un IDE desarrollado por JetBrains, que permite la codificación para gran variedad de lenguajes, especialmente optimizado para el lenguaje Java.

	Android Studio	4.2.2	Entorno de desarrollo para aplicaciones Android, es una plataforma que se utiliza para desarrollar aplicaciones móviles que se desean liberar en el Play Store de Google.
	Xcode	12.5	Es un entorno de desarrollo integrado para la creación, depuración y despliegue de aplicaciones para Mac, iPhone, iPad.
	Xampp	8.0.9	Es un conjunto de herramientas de desarrollo que consta de una base de datos un servidor web e intérpretes de varios lenguajes.
	Chrome	93.0.4577.62	Es Navegador de web, utilizado para hacer debug de las aplicaciones web e híbridas, consta de una serie de herramientas especializadas en la interfaz gráfica.
Verificación (QA)	Postman	8.10.0	Herramienta para realizar peticiones REST, utilizada para probar los diferentes APIs o webs services de proveedores o desarrollos internos.
	Github Desktop	2.9.2	Aplicación para la creación de proyectos de manera colaborativa, almacenados en repositorios remotos, que permite el manejo de cambios y versiones en los proyectos.
	Microsoft Office	2019	Suite de ofimática, es un conjunto de programas desarrollado por Microsoft Office, se utiliza principalmente Word, Excel, PowerPoint
Soporte	Incident Manager	4.1	Es una aplicación utilizada para la gestión de incidentes, reportes de errores, así como el seguimiento de cada ticket generado
Datos	Pentaho	Community	Es una herramienta de BI (Business Intelligence), que permite extraer, analizar y cargar datos, esta información puede ser utilizada para la generación de informes.

	PgAdmin	5.4	Es una herramienta open source, utilizada para administrar, gestionar y mantener bases de datos
	DBeaver	21.1.5	Es una herramienta utilizada para la gestión de gran cantidad de bases de datos

A continuación, se ilustran las aplicaciones que tiene actualmente la fábrica de software.

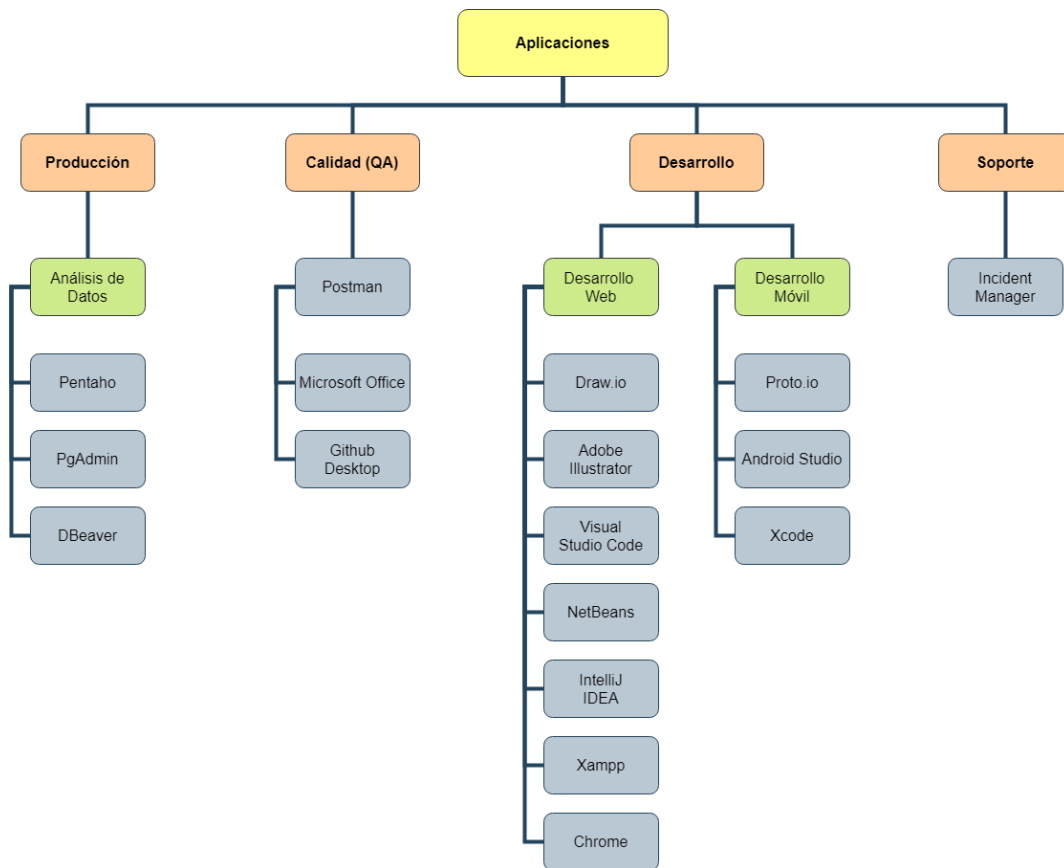


Figura 17. Aplicaciones existentes

4.1.1.1. Matriz de procesos vs aplicaciones

A continuación, se detallan los principales procesos de CMMI-DEV enfocados al desarrollo y cómo influye en cada una de las aplicaciones de la fábrica de software.

Tabla 23

Procesos vs aplicaciones

Aplicación	Áreas de Procesos de Arquitectura Objetivo (CMMI-DEV)				
	Desarrollo de Requisitos (RD)	Integración del Producto (PI)	Solución Técnica (TS)	Validación (VAL)	Verificación (VER)
Jira	X	X			
Pentaho			X	X	X
Power BI			X	X	
PgAdmin			X		X
DBeaver			X		X
SonarQube		X	X	X	X
Draw.io	X	X	X		
Adobe Illustrator	X		X	X	
Visual Studio Code			X		
IntelliJIDEA			X		
Xampp		X	X	X	
Chrome		X	X		X
Microsoft Office	X				
Github Desktop		X		X	X
Postman		X		X	X
Cypress				X	X
Proto.io	X		X		
Android Studio			X	X	
Xcode			X	X	
Incident Mangment				X	X

4.1.1.2. Matriz de roles vs aplicaciones

En la tabla 24 se presentan los principales roles del equipo Scrum y como se utiliza cada una de las aplicaciones existentes.

Tabla 24.

Roles vs aplicaciones

Aplicación	Roles		
	Product Ower	Scrum Master	Developers
Jira	X	X	X
Pentaho	X		X
Power BI	X		X
PgAdmin			X
DBeaver			X
SonarQube			X
Draw.io	X	X	X
Adobe Illustrator	X		X
Visual Studio Code			X
IntelliJIDEA			X
Xampp			X
Chrome	X	X	X
Microsoft Office	X	X	
Github Desktop			X
Postman			X
Cypress			X
Proto.io	X	X	X
Android Studio			X

Xcode			X
Incident Mangment	X	X	X

4.1.2. Arquitectura de aplicaciones objetivo

Los aplicativos a ser utilizados en la arquitectura objetivo serán los existentes, y para mejorar el flujo de trabajo y la calidad del código se implementará:

- Jira
- SonarQube
- Cypress
- PowerBI
- CircleCI

4.1.3. Trazabilidad de nuevas aplicaciones

Actualmente existen muchas aplicaciones que ayudan en la gestión de proyectos, pero una herramienta que se adapta perfectamente al marco Scrum es Jira.

4.1.3.1. Jira

Es una herramienta de gestión de trabajo, orientada al modelo ágil que permite manejar la gestión de requisitos, casos de pruebas, incidentes y el desarrollo de software ágil.

Entre las características principales de Jira se puede mencionar:

- Diseñado para que los miembros del equipo puedan planificar, publicar y supervisar el desarrollo de software.
- Seguimiento al detalle, gracias a que Jira centra todo el flujo en un solo tablero, es fácil distinguir en qué actividades se encuentran cada uno de los miembros del equipo.

- Creación de informes, tiene una serie de opciones y reportes que nos permiten obtener reportes.
- Integración con herramientas, por su fácil adaptación con otras herramientas Jira se vuelve una herramienta ideal para adaptarse a cualquier tipo de ambiente.
- Personalización, es una herramienta en la cual podemos personalizar muchas opciones para que se adapte a cada una de nuestras necesidades.

4.1.3.2. SonarQube

Plataforma para evaluar la calidad el código fuente, permite realizar un análisis estático del código, con el principal objetivo de advertirnos sobre diferentes puntos de calidad:

Entre sus principales características tenemos:

- Detección de código duplicado, permite identificar métricas relacionadas con la duplicidad de nuestro código.
- Detección de código muerto, permite identificar código dentro de nuestra aplicación que nunca es utilizado.
- Estándares de codificación, establece normas para la escritura de código fuente, dependiendo del lenguaje de programación se puede establecer dichas normas.
- Bugs, son defectos en el software o problemas que hacen que un programa no funcione de manera adecuada.
- Complejidad ciclomática, permite identificar el número de caminos independientes que tiene un código, mientras más compleja es la lógica más difícil se vuelve de entender.
- Comentarios, permite analizar el nivel de documentación del código.

- Test unitarios y de integración, la forma de identificar el correcto funcionamiento de un código es mediante la utilización de pruebas unitarias y de integración los cuales son una forma de garantizar el código.
- Cobertura de código, la cobertura nos permite identificar la cantidad de líneas cubiertas por los diferentes test.

4.1.3.3. Cypress

Es un framework de testing, rápido y fácil de usar, ideal para el desarrollo de pruebas automatizadas a nivel de frontend.

- Incluye una serie de librerías que permiten realizar pruebas e2e automáticas.
- Se pueden realizar diferentes casos de prueba orientándonos al funcionamiento de cada componente de la interfaz.
- Fácil de entender, es una herramienta basada en Javascript cuya curva de aprendizaje no es muy alta.

4.1.3.4. Power BI

Es una solución desarrollada por Microsoft, que permite obtener datos de diferentes fuentes para ser analizados y presentados en informes dinámicos.

- Permite centralizar la información de la organización
- Dispone de numerosas plantillas para presentar los resultados de los datos en informes.
- Conexión a gran cantidad de fuentes de datos.
- Permite implementar inteligencia artificial.

4.1.3.5. CircleCI

Es una herramienta de integración continua (CI) que se integra a la perfección con Github, una de las herramientas en la nube para control de versiones. CircleCI

puede alojarse de forma local o utilizarse a través de un servicio en la nube, sus principales características son:

- Desencadenadores de notificaciones a partir de eventos de CI.
- Rendimiento optimizado para compilaciones rápidas.
- Depuración sencilla a través de compilaciones locales y SSH.
- Análisis para medir el rendimiento de compilación. (Atlassian, 2021b)

En la figura 18 se presenta la arquitectura objetivo, en color rojo se muestran las aplicaciones que van a ser implementadas, en gris las aplicaciones que serán eliminadas y en color azul las que se mantienen.

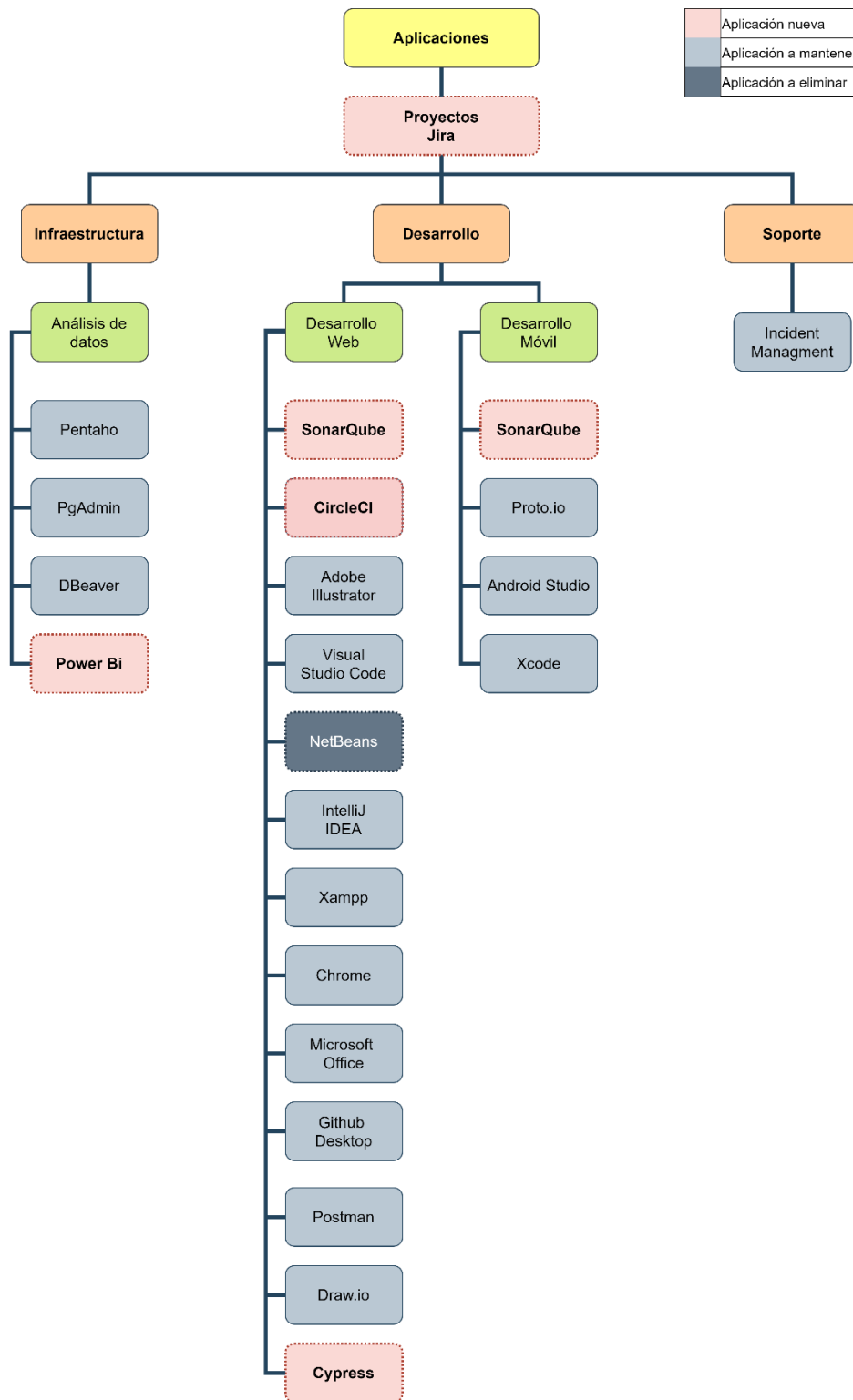


Figura 18. Arquitectura de aplicaciones objetivo

4.1.4. Análisis de brechas e iniciativas

Para el análisis de brechas se tomaron en cuenta todas las aplicaciones que tiene la fábrica de software, en busca de herramientas que permitan solucionar los problemas que actualmente posee la fábrica de software.

Para el análisis de la madurez se utilizará la siguiente rúbrica:

Tabla 25

Rúbrica de madurez

Calificación	Detalle
5	Se evidencia un proceso exponencial de innovación el cual garantiza liderazgo en la empresa presentando un componente arquitectónico maduro y referente en el área de conocimiento.
4	Presentan tecnologías maduras y reconocidas por el área de conocimiento presentando un buen desempeño donde el componente arquitectónico se encuentra guiado por prácticas del sector.
3	Dispone de documentación formal teniendo las herramientas e indicadores de gestión adecuados y mejorados periódicamente. El componente arquitectónico presenta un tiempo superior a 6 meses de implementación.
2	No se cuenta con documentación formal ni herramientas de gestión. El componente arquitectónico es básico, desplegado en los distintos actores siendo de cumplimiento obligatorio, pero el desempeño no es el estándar y adecuado al sector.
1	No se cuenta con apoyo tecnológico y en el caso de elaborarse es únicamente por iniciativa de empleados. Presenta un nivel de componentes arquitectónicos incompleto.
0	No presenta evidencia actual ni tampoco se encuentran componentes arquitectónicos.

Una vez realizado el análisis se identificó el nivel de madurez de las diferentes aplicaciones, y se obtuvieron los siguientes resultados.

Tabla 26

Análisis de brechas

Área	Aplicación	Estado Actual	Mediano Plazo	Referente Largo Plazo	Brecha por cerrar	Iniciativa que cierra brecha
Gestión	Gestión de proyectos	0	4	5	No existe una herramienta para gestionar las actividades de cada miembro del equipo de trabajo.	Implementar una plataforma para gestionar proyectos de manera colaborativa centralizada, para este caso se recomienda Jira
Análisis de datos	Pentaho	4	4	5	Actualmente se utiliza Pentaho como única solución para análisis de datos y existe una persona que domina la herramienta, lo que produce retraso y acumulación de tareas	Capacitar a más personas del área de TI en el uso de Pentaho.
	Business Intelligence	0	5	5	Pentaho ha sido la principal herramienta utilizada para análisis de datos, se la puede reforzar con una aplicación para mostrar resultados de una manera más ilustrativa.	Implementar Power Bi permitirá la creación de informes dinámicos, que pueden ser obtenidos de numerosas fuentes de datos
	PgAdmin	5	5	5		
	DBeaver	5	5	5		
Desarrollo Web	Calidad de código	1	5	5	No existe un proceso automatizado para la validación del código.	Implementar SonarQube, permitirá mejorar la calidad del software desarrollado, disminuyendo la cantidad de errores, código

						duplicado o no utilizado.
	Draw.io	5	5	5		
	Adobe Illustrator	5	5	5		
	Visual Studio Code	3	4	5	No es utilizado por todos los miembros del equipo	Implementar VScode, como herramienta estándar para el desarrollo frontend de aplicaciones web e híbridas
	IntelliJIDEA	3	4	5	No es utilizada por todos los miembros del equipo	Implementar IntelliJIDEA, como herramienta estándar para el desarrollo de aplicaciones backend como son web service, endpoints, procesos batch.
	Xampp	5	5	5		
	Chrome	5	5	5		
	Microsoft Office	4	4	5	No todas las personas que disponen del Office, cuentan con las licencias para su uso.	Comprar licencias para el área comercial y líderes de áreas.
	Github Desktop	3	4	5	Se inició con el proceso de versionamiento, pero no todas las personas tienen un conocimiento del manejo de esta herramienta	Capacitar al equipo de desarrollo, sobre el uso adecuado de la aplicación.
	Postman	3	4	5	No existen licencias para el equipo de calidad	Realizar la compra de licencias para las personas encargadas del aseguramiento de la calidad QA
	Automatizar pruebas	0	4	5	Mejorar la calidad de las pruebas con la implementación	Implementar una herramienta como cypress.js, que permite realizar

					de pruebas automatizadas	pruebas automatizadas.
Desarrollo móvil	Proto.io	3	4	5	Se cuenta con la versión gratuita, que no permite utilizar todas las opciones proporcionadas por la herramienta	Comprar licencias para el equipo de diseño.
	Android Studio	3	4	5	Ideal para el desarrollo de aplicaciones Android, pero existen muy pocas personas capacitadas para proyectos de este estilo	Capacitar al equipo de desarrollo, en la implementación de aplicaciones móviles con Android Studio.
	Xcode	2	4	5	No existe una persona especializada en el desarrollo para iOS o MacOS	Capacitar al equipo de desarrollo, en la implementación de aplicaciones móviles con Xcode.
Soporte	Incident Mangment	4	4	5	No existe un proceso claro para la capacitación en dicha herramienta	

La fábrica de software actualmente tiene problemas definidos en fases previas, para mejorar dichos inconvenientes es necesario implementar herramientas que permitan mejorar la gestión del trabajo colaborativo, incrementar la calidad del código, reducir el tiempo en pruebas y realizar análisis de la información que dispone la fábrica, es por esto que se necesita implementar:

- Gestionar proyectos y trabajo colaborativo con Jira
- Automatizar pruebas con Cypress
- Mejorar la calidad del código con SonarQube
- Business Intelligence con PowerBI

Es necesario eliminar Netbeans como IDE para el desarrollo ya que actualmente se dispone de VsCode y de IntelliJIDEA, que son herramientas especializadas en frontend y backend respectivamente, y cubren por completo las funciones ofrecidas por Netbeans.

A continuación, se muestra un gráfico con las principales brechas:

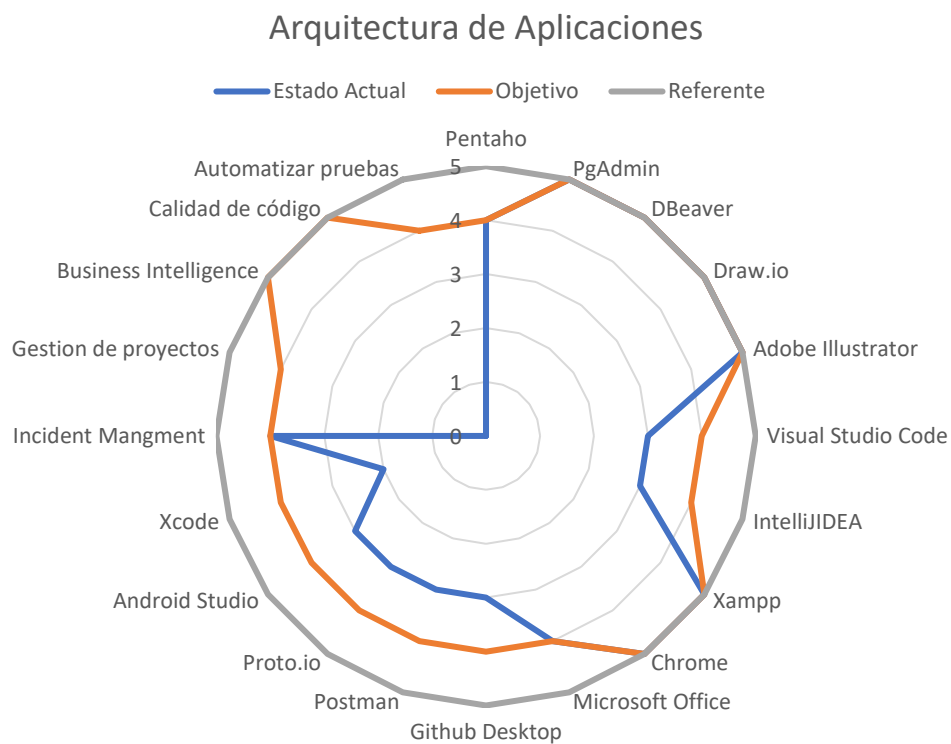


Figura 19. Análisis de brechas arquitectura actual vs arquitectura objetivo

4.2. Arquitectura de datos

La arquitectura de datos describe a través de un conjunto de requerimientos, principios, modelos la forma en que se manejan los datos dentro de la organización.

4.2.1. Arquitectura actual de datos base

La información generada por la fábrica de software es de dos tipos:

- Datos estructurados: Datos que pueden ser clasificados, ordenados y procesados con facilidad.
- Datos no estructurados: Datos con estructuras que no se pueden identificar, razón por la cual son más difíciles de procesar.

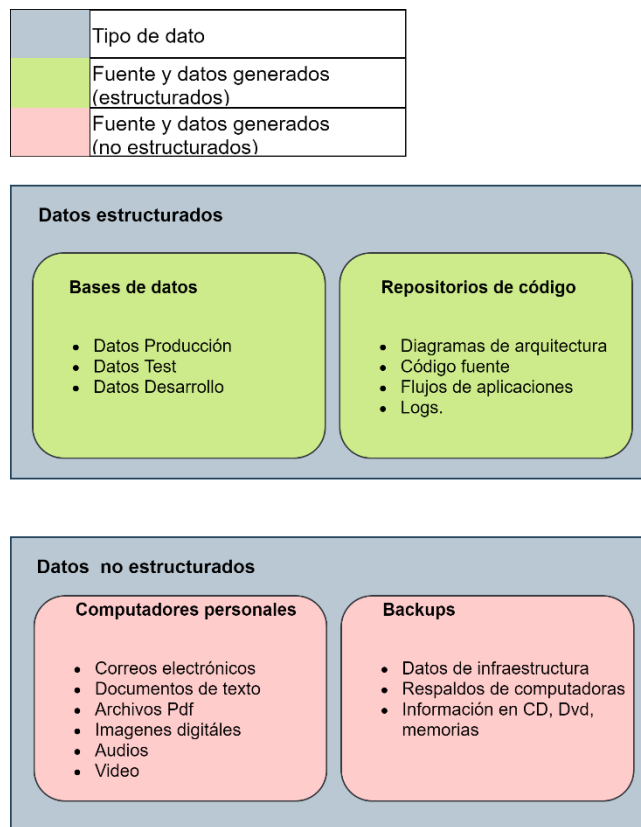


Figura 20. Fuentes de datos y su relación con la información

Existe gran cantidad de información almacenada en diferentes fuentes, estructuras complejas y con una interoperabilidad muy baja. Cada aplicación genera información en el espacio asignado para almacenar.

4.2.2. Arquitectura de datos objetivo

Se mantendrá el manejo actual de la información de datos estructurados, para datos no estructurados es necesario migrar hacia una plataforma colaborativa que permita:

- Centralizar la información.
- Crear, colaborar y comentar los datos.
- Compartir información entre las diferentes personas.

4.2.3. Análisis de brechas

La información se encuentra almacenada en diferentes repositorios con un bajo nivel de intercambio de datos. Se requiere establecer un repositorio centralizado que cumpla con las características descritas en target de arquitectura de datos de capítulos anteriores.

Existe gran cantidad de información no estructurada duplicada en algunos repositorios, es necesario depurar dicha información, previo a cualquier proceso de mejora.

4.2.4. Iniciativa para cerrar la brecha

Para centralizar la información no estructurada es necesario implementar Confluence, una herramienta colaborativa que permite centralizar el trabajo de los equipos. Para una correcta implementación de Confluence es necesario que los nuevos documentos sean generados considerando el rol y su responsabilidad.

La siguiente tabla presenta una matriz de documentos versus responsables.

Tabla 27

Documentación versus responsables

Documento	Gerente de Tecnología	Gerente de proyectos	Líder de proyectos	Product Owner	Scrum Master	Developers
Presentaciones	X	X	X	X	X	X
Indicadores / métricas	X	X	X	X	X	
Actas	X	X	X	X	X	
Diagramas		X	X	X	X	X
Flujos		X	X	X	X	X
Documentos de control de calidad					X	X
Acuerdos de equipo				X	X	X
Historias de usuario				X	X	X
Catálogos / documentos desarrollo				X	X	X

5. ARQUITECTURA DE INFRAESTRUCTURA BASE

Actualmente la fábrica de software dispone de servidores virtualizados en la plataforma de Google y servidores físicos.

5.1. Infraestructura de la fábrica de software

5.1.1. Infraestructura base

Características de servidores, equipos portátiles y dispositivos móviles de la fábrica de software.

Tabla 28

Infraestructura actual

Tipo	Estado	Marca	Cantidad	Características	Detalle de uso
Servidores	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • (TWS) 2 CPU virtuales • 13 GB RAM • Disco de 120 GB • Sistema Operativo CentOS 7.9 	Servidor utilizado en ambiente de producción para el manejo de aplicaciones web
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • (TAP) 2 CPU virtuales • 13 GB RAM • Disco de 120 GB • Sistema Operativo CentOS 7.9 	Servidor utilizado en ambiente de producción para el manejo de procesos y aplicaciones
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • (TDB) 4 CPU virtuales • 15 GB RAM • Disco de 250 GB • Sistema Operativo CentOS 7.9 	Servidor utilizado en ambiente de producción para el almacenamiento de información
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • (SMS) 2 CPU virtuales • 7.5 GB RAM • Disco de 120GB • CentOS 7.9 	Servidor utilizado en ambiente de producción para procesos relacionados con mensajería (SMS).
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • (TWS-TEST) 1 CPU virtuales • 7 GB RAM • Disco de 80 GB 	Servidor utilizado en ambiente de test para el manejo de aplicaciones web

				<ul style="list-style-type: none"> • Sistema Operativo CentOS 7.9 	
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • (TAP-TEST) 1 CPU virtuales • 7 GB RAM • Disco de 80 GB • Sistema Operativo CentOS 7.9 	Servidor utilizado en ambiente de test para el manejo de procesos y aplicaciones
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • (TDB-TEST) • 1 CPU virtuales • 7.5 GB RAM • Disco de 80 GB • Sistema Operativo CentOS 7.9 	Servidor utilizado en ambiente de test para el almacenamiento de información
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • SMS-TEST • 1 CPU virtuales • 7.5 GB RAM • Disco de 60GB • CentOS 7.9 	Servidor utilizado en ambiente de test para procesos relacionados con mensajería (SMS).
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • (TWS-DEV) 1 CPU virtuales • 2 GB RAM • Disco de 40 GB • Sistema Operativo CentOS 7.9 	Servidor utilizado en ambiente desarrollo para el manejo de aplicaciones web
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • (TAP-DEV) 1 CPU virtuales • 2 GB RAM • Disco de 40 GB • Sistema Operativo CentOS 7.9 	Servidor utilizado en ambiente desarrollo para el manejo de procesos y aplicaciones
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • (TDB-DEV) • 1 CPU virtuales • 2 GB RAM • Disco de 40 GB • Sistema Operativo CentOS 7.9 	Servidor utilizado en ambiente desarrollo para el almacenamiento de información
	Nube - Google	Hewlett-Packard (HP)	1	<ul style="list-style-type: none"> • SMS-DEV • 1 CPU virtuales • 3 GB RAM • Disco de 40GB • CentOS 7.9 	Servidor utilizado en ambiente de desarrollo para procesos relacionados con mensajería (SMS).
Computador	Físico	Asus	3	<ul style="list-style-type: none"> • Procesador Intel(R) Core i7-4790 • Memoria RAM de 16GB DDR4 • Almacenamiento 500 GB • Tarjeta gráfica Intel Graphics 630 	Computador destinado para el desarrollo de software

				<ul style="list-style-type: none"> • Sistema Operativo Windows 10 Pro 	
	Físico	Dell	3	<ul style="list-style-type: none"> • Intel Core i7-8700 de Seis núcleos a 3,2 GHz • Memoria RAM DDR4 de 16GB • Almacenamiento 500 GB • UHD Graphics 630 • Windows 10 Pro (64 bits) 	Computador destinado para el desarrollo de software
	Físico	Genérica	2	<ul style="list-style-type: none"> • Procesador Core i3 2da generación, • Memoria RAM 4 GB • Disco Duro 500 GB • Windows 10 Pro (64 bits) 	Computador destinado para el desarrollo de software
CPU	Físico	Mac Mini	2	<ul style="list-style-type: none"> • Apple Mac Mini 3.6GHz quad-core • Memoria RAM 8GB of 2666MHz DDR4 SO-DIMM • 256GB PCIe-based SSD1 • Tarjeta gráfica Intel UHD Graphics 630 	Computador destinado para el desarrollo de software
Portátil	Físico	HP	4	<ul style="list-style-type: none"> • Intel(R) Core (TM) i7-4700MQ CPU @ 2.40GHz 2.40 GHz • Memoria RAM 8,0 GB • SATA de 750 GB 5400 rpm • Windows 10 Home Single Language 	Equipo destinado para líderes de área de la fábrica
	Físico	Toshiba	2	<ul style="list-style-type: none"> • Core I5-10210U 1.60Ghz • Memoria RAM 8,0 GB DDR3 • SATA de 500 GB • Windows 10 Home Single Language 	Equipo destinado para las personas del área de calidad (QA)
Dispositivos móviles	Físico	LG	1	<ul style="list-style-type: none"> • Modelo K10 • Android OS, v5.1 Lollipop • LTE chip quad-core de 1.3GHz 	Dispositivo para probar aplicaciones Android

				<ul style="list-style-type: none"> • 2GB de RAM • Almacenamiento interno es de 16GB • Cámaras son de 13 MP atrás y 8MP al frente 	
Físico	Tablet Samsung	1	<ul style="list-style-type: none"> • Android 4.4.2 KitKat • 1.5 GB de RAM / 16 GB de ROM • Pantalla WXGA de 8.0" • Procesador de cuádruple núcleo, de 1.2 GHz • Cámara trasera de 3 MP / cámara frontal de 1.3 MP 	Dispositivo para probar aplicaciones Android	
Físico	Iphone	1	<ul style="list-style-type: none"> • Apple A10 Fusión de cuatro núcleos a 2.34 GHz • Memoria RAM: 3GB • Memoria interna de 128GB • Conectividad: 3G y 4G, NFC, Bluetooth y A-GPS 	Dispositivo para probar aplicaciones iOS	
Físico	Ipad	1	<ul style="list-style-type: none"> • Apple iPad • Procesador: Apple A9 • Memoria RAM: 2GB • Almacenamiento interno: 128GB • Pantalla: 9.7", 1536 x 2048 pixels 	Dispositivo para probar aplicaciones iOS	

La fábrica de software dispone de un apalancamiento tecnológico en la nube de Google de tipo LAAS (infraestructura como servicio), que le permite virtualizar, aprovisionar y administrar su infraestructura por medio de internet, lo que le permite comprar, instalar, administrar y configurar cualquier software de forma independiente, se caracteriza por ser una solución escalable limitándose a pagar por el uso.

A continuación, se ilustra el diagrama de la infraestructura actual de la fábrica de software:

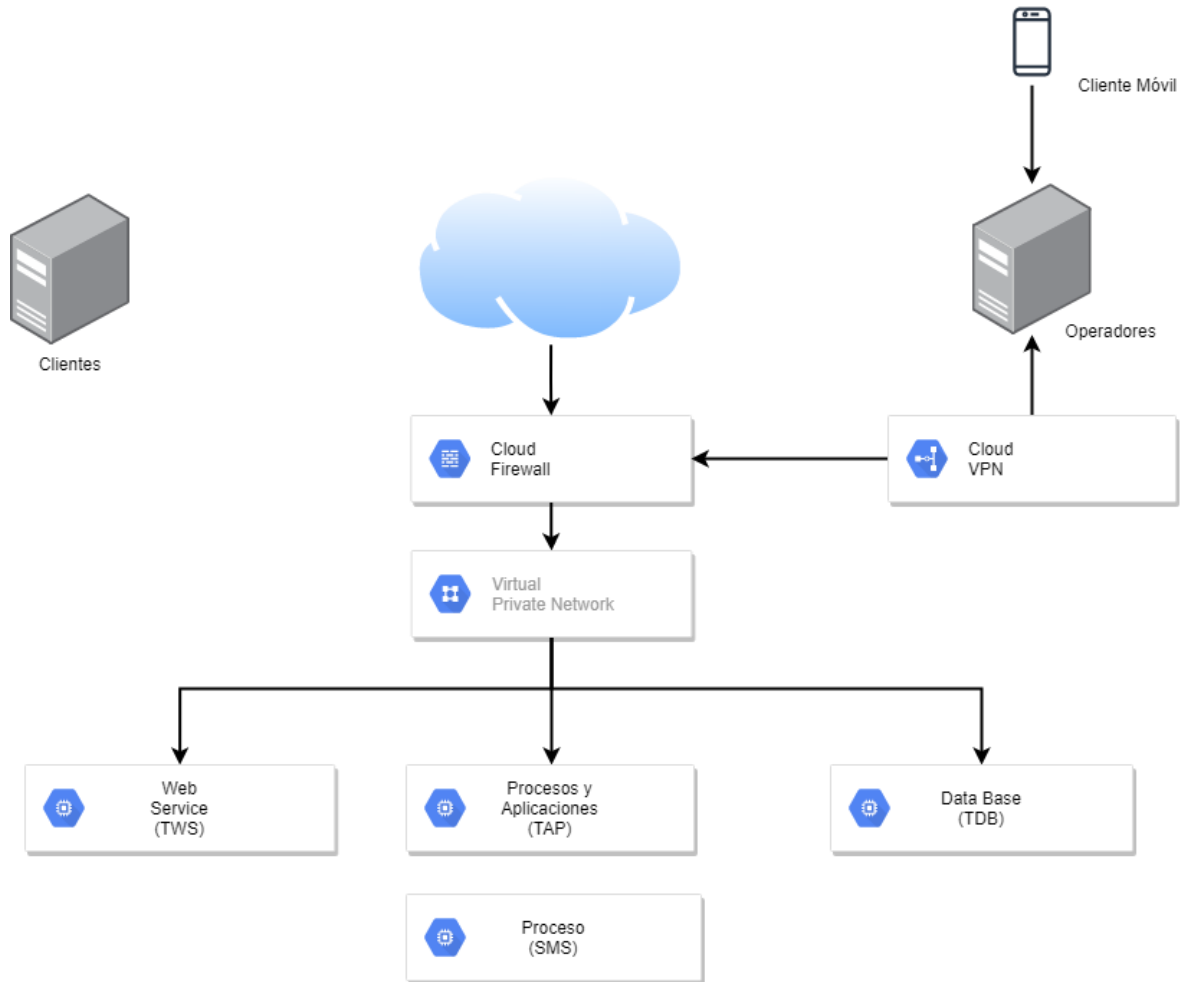


Figura 21. Infraestructura actual

5.1.2. Entornos de desarrollo

La fábrica de software actualmente dispone de 3 ambientes para el desarrollo, los cuales son: desarrollo, test y producción, cada ambiente es independiente y posee los recursos y capacidades.

- Ambiente de desarrollo: En este ambiente el equipo de desarrolladores inicia el ciclo del desarrollo de software, se implementan los requerimientos del cliente y se transforman en una versión estable o un conjunto de módulos.
- Ambiente de test: El ambiente de test es muy similar al ambiente de producción, el equipo de calidad (QA) toma el software desarrollado en el ambiente anterior y lo despliega en este ambiente para realizar una serie de pruebas que garanticen la calidad del software, así como el cumplimiento de requisitos. (Ortiz Vivas, 2017)
- Ambiente de producción: En el ambiente de producción el software que ha sido certificado por el equipo de calidad (QA), es desplegado y se convierte en la versión activa. En este ambiente el cliente podrá ver los diferentes requerimientos solicitados, y de ser necesario solicitar cambios que serán ingresados como un nuevo requerimiento.

El proceso para desplegar en producción se lo realiza de forma manual entre el equipo de desarrollo y el equipo de producción, lo que genera errores que afectan a todo el ambiente generando gran cantidad de incidentes en el área de soporte. (Ortiz Vivas, 2017)

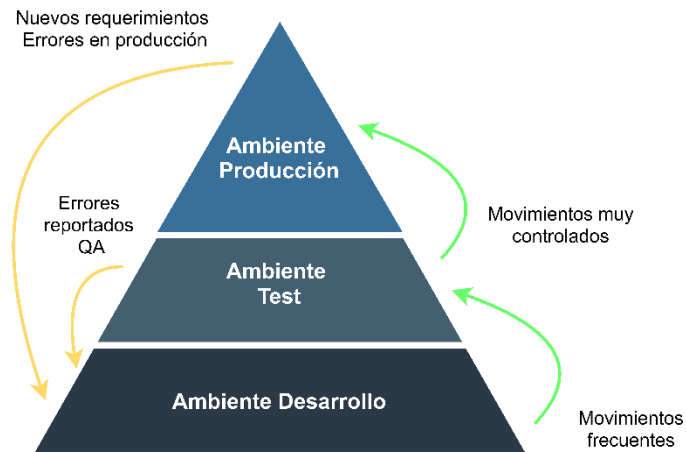


Figura 22. Entornos de desarrollo. Adaptada de
(Gilart Iglesias & Capella D'alton, 2005)

En el diagrama se puede observar cómo se mueve el software desarrollado en cada ambiente, en caso de encontrar algún tipo de error el programa o módulos son devueltos al ambiente de desarrollo, para iniciar el ciclo.

5.2. Arquitectura de infraestructura base objetivo

Para implementar la arquitectura de infraestructura tecnológica objetivo se mantendrá la infraestructura actual que se encuentra en la nube de Google Cloud Platform (GCP), el informe de Gartner indica que GCP es uno de los líderes en el mercado de servicios, plataforma e infraestructura en la nube.



Figura 23. Cuadrante de Gartner, plataforma e infraestructura en la nube.

(Gartner, 2021)

Tomando como referencia las prácticas recomendadas y arquitectura de referencia para el diseño de nubes privadas virtuales con Google se realiza el análisis de la infraestructura actual bajo los siguientes parámetros: (Cloud VPN, 2021)

- Seguridad
- Disponibilidad
- Monitoreo
- Registro de logs
- Integración continua

5.2.1. Seguridad

Cloud VPN extiende la red de intercambio de tráfico de forma segura hacia la red de Google a través de un túnel VPN IPsec. El tráfico se encripta y fluye entre las dos redes a través de la Internet pública. (*Cloud VPN*, 2021)

5.2.2. Disponibilidad

En un nodo de usuario único, puedes aprovisionar varias VM en tipos de máquinas de varios tamaños, lo que te permite usar de manera eficaz los recursos subyacentes del hardware del host dedicado. Además, debido a que no compartes el hardware de host con otros proyectos, puedes cumplir con los requisitos de seguridad o de cumplimiento con cargas de trabajo que requieran aislamiento físico de otras cargas de trabajo o VM. (*Documentación de Compute Engine*, 2021)

5.2.3. Monitoreo

Permite obtener una mayor visibilidad del rendimiento, la disponibilidad y el estado de las aplicaciones y de la infraestructura y ofrece características como:

- Monitorización de objetivos de nivel de servicio
- Métricas personalizadas
- Integración de Google Cloud (*Cloud Monitoring*, 2021)

5.2.4. Loggin

Es parte de la plataforma de Google Cloud y permite administrar los registros (colección estructurada de datos), de manera programática. Logging te permite leer y escribir entradas de registro, consultar registros y controlar cómo enrutar, almacenar y usar los registros. (*Cloud Logging*, 2021)

5.2.5. Integración continua

La integración continua es una práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas. La integración continua se refiere en su mayoría a la fase de creación o integración del proceso de publicación de software y conlleva un componente de automatización. (Amazon, 2021).



Figura 24. Beneficios de la integración continua. (SmartNodus, 2021)

En el proceso de integración continua se utiliza Git como repositorio central para manejar las versiones del código, se crea una rama master que tiene la última versión del desarrollo e irá viajando entre los diferentes ambientes con la supervisión del área de calidad. Adicionalmente se implementará la automatización de la construcción, testeo, despliegue y el proceso de puesta a producción utilizando una herramienta como CircleCI.

5.3. Análisis de brechas

Al comparar la arquitectura de infraestructura base actual contra la arquitectura de infraestructura base objetivo se obtienen las siguientes brechas e iniciativas que cierran esas brechas:

Para el análisis de la madurez se utilizará la siguiente rúbrica:

Tabla 29.

Rúbrica de madurez

Calificación	Detalle
5	Se evidencia un proceso exponencial de innovación el cual garantiza liderazgo en la empresa presentando un componente arquitectónico maduro y referente en el área de conocimiento.
4	Presentan tecnologías maduras y reconocidas por el área de conocimiento presentando un buen desempeño donde el componente arquitectónico se encuentra guiado por prácticas del sector.
3	Dispone de documentación formal teniendo las herramientas e indicadores de gestión adecuados y mejorados periódicamente. El componente arquitectónico presenta un tiempo superior a 6 meses de implementación.
2	No se cuenta con documentación formal ni herramientas de gestión. El componente arquitectónico es básico, desplegado en los distintos actores siendo de cumplimiento obligatorio, pero el desempeño no es el estándar y adecuado al sector.
1	No se cuenta con apoyo tecnológico y en el caso de elaborarse es únicamente por iniciativa de empleados. Presenta un nivel de componentes arquitectónicos incompleto.
0	No presenta evidencia actual ni tampoco se encuentran componentes arquitectónicos.

Tabla 30.

Análisis de brechas e iniciativas

Tipo	Área	Estado Actual	Mediano Plazo	Referente Largo Plazo	Brecha por cerrar	Iniciativa que cierra brecha
Infraestructura	Seguridad	2	4	5	El servidor de Base de datos TDB se encuentra en la misma red virtual que los servidores web y de procesos lo que genera una brecha de seguridad	Migrar la base de datos a una red privada virtual independiente
	Disponibilidad	3	4	5	Actualmente existe un solo servidor para cada proceso importante de la fábrica de software	Crear un balanceador de carga hacia diferentes nodos para que se pueda garantizar una alta disponibilidad
	Monitoreo	1	4	5	Se realiza un monitoreo básico y muy manual de los diferentes servidores.	Implementar el servicio de Cloud Monitoring para recoger diferentes métricas de la infraestructura
	Registro de logs	0	3	5	Se registran logs muy básicos de ciertos servidores	Complementar el servicio de Monitorig con Cloud Logging para la gestión y análisis de registros en tiempo real.
Despliegue	Integración continua	0	4	5	El pasar el código entre los diferentes ambientes es una tarea manual y propensa a errores	Implementar una herramienta de despliegue continuo como CircleCi, que elimina procesos manuales y permite integrar el código en los diferentes ambientes de una manera más eficiente.

La siguiente ilustración muestra el diagrama del impacto de la infraestructura objetivo en la infraestructura base.

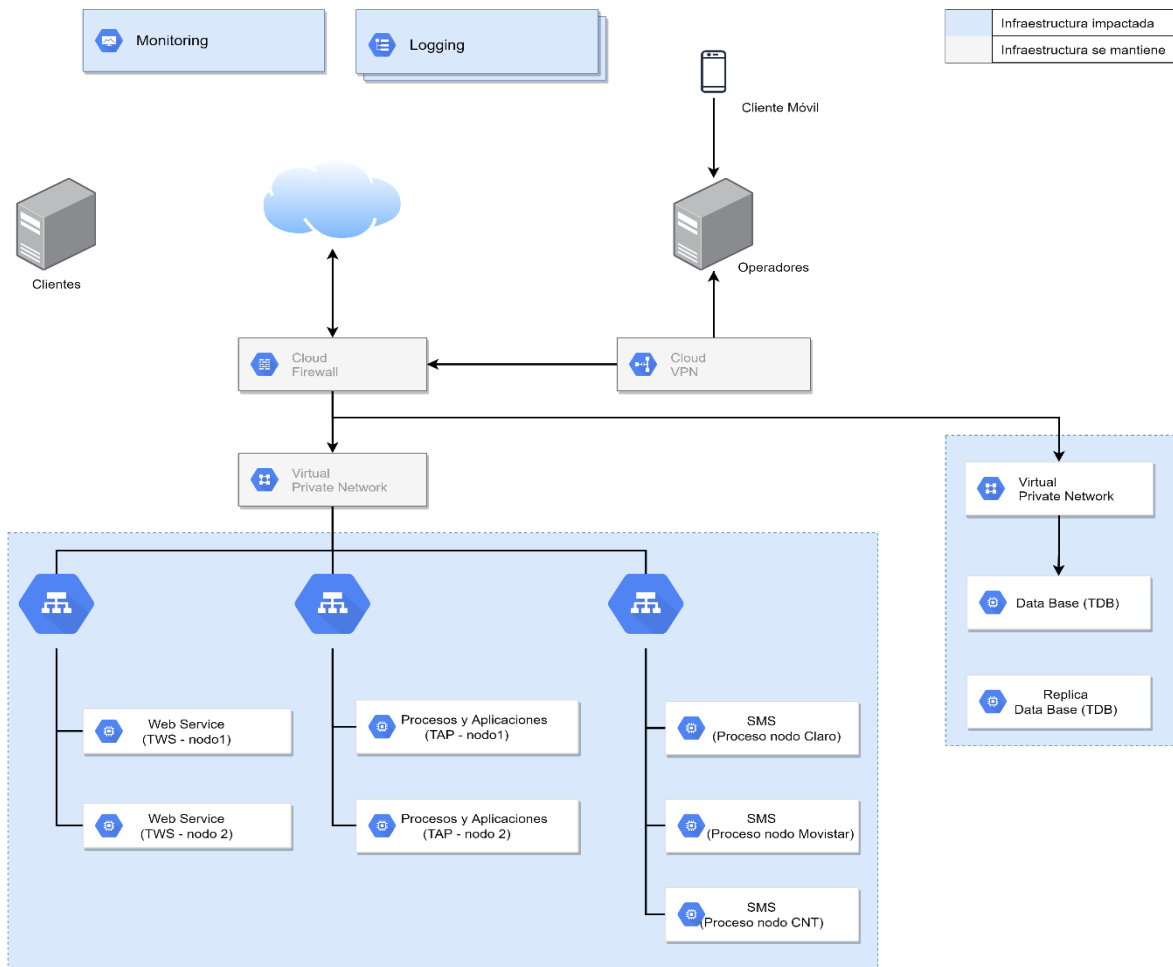


Figura 25. Impacto de la infraestructura objetivo

5.4. Impacto de la arquitectura objetivo en la situación actual

La infraestructura actual de la fábrica de software se verá beneficiada por las mejoras a ser implementadas en temas de seguridad, disponibilidad, monitoreo, logs, que actualmente son problemas que la fábrica de software necesita solucionar.

Al utilizar el despliegue continuo entre los diferentes ambientes, se optimizará el tiempo, reduciendo los errores ya que la herramienta CircleCi está optimizada para estas tareas, lo cual permitirá cumplir con un requerimiento solicitado para fortalecer la fábrica de software.

6. OPORTUNIDADES Y SOLUCIONES

En el presente capítulo se consolidan las iniciativas presentadas en las fases anteriores para cerrar las brechas entre la situación actual y objetivo que se ha planteado para la fábrica de software.

En la siguiente tabla se muestra para cada arquitectura sus respectivas iniciativas, las cuales más adelante serán desarrolladas con mayor detalle.

Tabla 31

Arquitecturas e Iniciativas

Arquitectura	Iniciativa
Negocio	Implementar el marco de trabajo Scrum
	Seleccionar el personal para formar el equipo Scrum
	Implementar las áreas de proceso de ingeniería de CMMI-DEV para el desarrollo de software
	Implementar la herramienta colaborativa Jira
Aplicaciones	Implementar CircleCi para reducir los tiempos de integración y despliegue en los diferentes ambientes
	Implementar SonarQube como herramienta para evaluar la calidad del código
	Implementar Cypress como herramienta para automatizar pruebas
	Implementar Confluence para gestión documental
Datos	Fortalecer el proceso de Business Intelligence con la aplicación PowerBI
Infraestructura base	Fortalecer la infraestructura en la nube que actualmente dispone la fábrica de software

6.1. Arquitectura de negocio

Actualmente la fábrica de software no dispone de un marco de trabajo para el manejo de proyectos de forma ágil, además se necesita fortalecer el conocimiento de todo el personal de la fábrica con relación a Scrum. Otro aspecto a considerar es que el equipo de desarrollo necesita reforzar sus prácticas actuales para mejorar la calidad del software.

6.1.1. Iniciativa: Implementar el marco de trabajo Scrum

Esta iniciativa propone implementar el marco de trabajo Scrum, dentro de la fábrica de software para la gestión adecuada de proyectos, el mismo consta de las siguientes fases (Sanabria, 2021):

a) Inicio

En la fase de inicio se detalla y analiza el proyecto, identificando las necesidades requeridas en el sprint.

En esta fase es necesario tener en cuenta las siguientes preguntas:

¿Qué quiero?

¿Cómo lo quiero?

¿Cuándo lo quiero?

b) Planificación y estimación

La estimación y planificación del sprint, es uno de los aspectos más importantes para la gestión de proyectos, en esta fase se definen los diferentes objetivos del proyecto. Se crea una lista priorizada con todas las actividades que necesitan ser implementadas.

c) Implementación

El equipo Scrum se reúne en una sala, esta reunión tiene por objetivo discutir cómo se desarrollará el sprint, optimizando el trabajo del equipo, en la fase de

implementación no deben existir cambios significativos ya que en fases previas como la de planificación se definieron los aspectos más importantes a ser desarrollados en el sprint.

d) Revisión y retrospectiva

Finalizada la implementación, se realiza una inspección del proceso donde se encuentran los aciertos y equivocaciones del equipo, sobre las equivocaciones se plantean acciones de mejora.

e) Lanzamiento

Corresponde a la última fase del marco de trabajo Scrum, donde se entrega el producto y el proyecto llega a su etapa final.

6.1.2. Conceptualización de la implementación

Para la implementación del marco de trabajo Scrum y las fases antes indicadas es necesario considerar los aspectos de la figura 26.

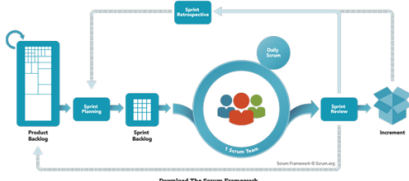
Implementar el marco de trabajo Scrum							
Código	Situación Actual y Principales Observaciones	Roadmap					
PN01	Actualmente la fábrica de software no dispone de una correcta gestión de proyectos, lo que ha provocado que muchos de ellos no cumplan con los presupuestos, costos ni tiempos de entrega. El marco de trabajo Scrum es un marco de referencia que es utilizado con grandes industrias como es el caso de Google, Spotify, que han logrado solventar estos problemas con la correcta implementación de metodologías ágiles que han aplicado prácticas como: 1. Dotar a los equipos de autonomía. 2. Formar grupos pequeños de trabajo (entre 3 y 8 personas). 3. Coordinar a los equipos para colaborar entre sí. 4. Escuchar a los colaboradores y capacitarlos para seguir la metodología de manera efectiva.	Fase: 2022 Área: Negocio					
Requerimientos de implementación	Actividades Fundamentales: <ul style="list-style-type: none"> Disponer del equipo Scrum completo. Taller de capacitación sobre el marco de referencia Scrum. https://www.aibes.org/certificaciones/official-scrum-fundamentals-certification/ Levantar en Confluente un espacio para almacenar información relacionada a Scrum. Seleccionar e iniciar un proyecto de prueba implementando Scrum. Taller de evaluación para analizar resultados del Sprint de prueba. Socializar resultados. 	Plazo de Implementación					
<input checked="" type="checkbox"/> Normativa <input type="checkbox"/> Hardware <input type="checkbox"/> Software <input type="checkbox"/> Servicios		<input checked="" type="checkbox"/> < 2 meses <input type="checkbox"/> < 6 meses <input type="checkbox"/> < 12 meses <input type="checkbox"/> > 12 meses					
Acerca de la Iniciativa	Actividades Macro	Participantes					
Impacto : Alto Esfuerzo: Bajo		<table border="1"> <tr> <td>Líder de proyectos</td> <td>A</td> </tr> <tr> <td>Scrum Master</td> <td>AR</td> </tr> <tr> <td>Equipo Scrum</td> <td>R</td> </tr> </table>	Líder de proyectos	A	Scrum Master	AR	Equipo Scrum
Líder de proyectos	A						
Scrum Master	AR						
Equipo Scrum	R						
Costos referenciales	 <p>Download The Scrum Framework</p>						
\$ 3000							

Figura 26. Conceptualización de iniciativa implementar Scrum

6.1.3. Iniciativa: Seleccionar el personal adecuado para formar el equipo Scrum

El equipo Scrum también es conocido como equipo de desarrollo, ya que este grupo de personas es el encargado de elaborar el producto o servicio, el objetivo de este equipo es trabajar en las historias de usuario del Sprint Backlog para crear los diferentes entregables.

6.1.3.1. Responsabilidades del Equipo Scrum

La siguiente tabla presenta las principales responsabilidades del equipo Scrum en los procesos del marco de trabajo ágil.

Tabla 32

Responsabilidades del equipo Scrum. (SCRUMstudy, 2017)

Procesos	Responsabilidades
Desarrollar épica(s)	Asegura una comprensión clara de la épica(s) y prototipos (<i>Personas</i>)
Crear el Backlog Priorizado del Producto	Entiende las historias de usuario en el Backlog Priorizado del Producto
Realizar la planificación del lanzamiento	Está de acuerdo con los demás miembros del equipo principal de Scrum sobre la duración del sprint Busca clarificación sobre los nuevos productos o cambios, si los hay, en los productos existentes en el Backlog Priorizado del Producto.
Crear historias de usuario	Proporciona aportes al Product Owner en la creación de historias de usuario
Estimar historias de usuario	Estima las historias de los usuarios aprobadas por el Product Owner
Comprometer historias de Usuario	Compromete las historias de usuario a realizarse en un sprint
Identificar tareas	Desarrolla una lista de tareas con base en las historias de usuario y dependencias acordadas
Estimar tareas	Estima el esfuerzo de las tareas identificadas y, de ser necesario, actualiza la lista de tareas.
Crear el Sprint Backlog	Desarrolla el Sprint Backlog y el Sprint Burndown Chart
Crear entregables	Elabora los entregables. Identifica riesgos y ejecuta acciones de mitigación de riesgos, si los hay. Actualiza el Impedimento y las dependencias.
Realizar el Daily Standup	Actualiza el Burndown Chart, el Scrum Board y el Impedimento Discute los problemas que enfrenta cada miembro y busca soluciones para motivar al equipo Identifica riesgos, si lo hay.

	Presenta solicitudes de cambio, si se requieren.
Refinar el Backlog Priorizado del Producto	Participa en las reuniones de revisión del Backlog Priorizado del Producto
Demostrar y validar sprints	Muestra los entregables completados al Product Owner para su aprobación
Retrospectiva del sprint	Identifica oportunidades de mejora, si las hay, del Sprint actual y decide si está de acuerdo sobre las posibles mejoras viables para el próximo sprint
Retrospectiva del proyecto	Participa en la reunión de retrospectiva del proyecto

El Equipo Scrum debe tener habilidades técnicas como blandas para poder trabajar en un proyecto, este equipo tiene de seis a diez personas y para seleccionarlas se tiene que considerar las siguientes características:

Tabla 33

Roles y características de Scrum. Adaptado (SCRUMstudy, 2017)

Rol	Características	Estado
Product Owner	<ul style="list-style-type: none"> • Experto en Scrum • Conocimiento de dominio del negocio • Excelentes habilidades de comunicación • Conocimiento de procesos Scrum • Habilidad para manejar las incertidumbres • Habilidades de negociación • Accesible • Proactivo • Decisivo 	Activo en nómina

	<ul style="list-style-type: none"> • Pragmático • Orientado a las metas 	
Scrum Master	<ul style="list-style-type: none"> • Experto en Scrum, Certificación PROFESSIONAL SCRUM MASTER™ III (Scrum.org, 2021) • Líder servicial • Moderador • Solucionador de problemas • Accesible • Motivador • Perceptivo • Mentor • Habilidades de coordinación • Introspectivo 	Activo en nómina
Equipo Scrum	<ul style="list-style-type: none"> • Conocimiento de Scrum • Colaborativos • Auto-organizables • Altamente motivados • Proactivos • Expertos técnicos • Perspectiva interfuncional • Saben trabajar en equipo • Independientes • Responsables • Intuitivos • Enfocados a los objetivos • Introspectivos 	<p>Se necesita la contratación de:</p> <p>1 persona para manejo de experiencia de usuario (UX)</p> <p>1 persona para diseños de interfaz de usuario (UI)</p> <p>1 desarrollador Frontend</p> <p>1 desarrollador Backend</p> <p>1 persona para Devops</p>

6.1.3.2. Conceptualización de la implementación

A continuación, se detallan los principales aspectos a considerar en la implementación de la iniciativa.

Seleccionar personal para el equipo Scrum					
Código	Situación Actual y Principales Observaciones	Roadmap			
PA02	Actualmente la fábrica de software no implementa ninguna metodología ágil, además el personal que dispone no tiene las competencias adecuadas para implementar el marco de trabajo Scrum; no obstante existen pocas personas que se adaptan perfectamente al nuevo marco de trabajo, ya que se han capacitado de forma personal en el marco de trabajo propuesto, las demás personas deben entrar a un proceso de capacitación. Por los antecedentes descritos es necesario contratar nuevos perfiles que se incorporen para completar el equipo Scrum, ya que algunos perfiles de la fábrica de software pueden cubrir ciertos cargos en necesario reforzar con perfiles que no existen actualmente	Fase: 2022 Área: Negocio			
Requerimientos de implementación	Actividades Fundamentales: <ul style="list-style-type: none"> Analizar perfiles actuales de la fábrica de software. Consolidar perfiles adecuados con perfiles por contratar. Iniciar proceso de contratación de perfiles que no existen en la fábrica. Contratar a perfiles adecuados Proceso de inducción a la empresa y al nuevo marco de trabajo. 	Plazo de Implementación			
<input checked="" type="checkbox"/> Normativa <input type="checkbox"/> Hardware <input type="checkbox"/> Software <input type="checkbox"/> Servicios		<input checked="" type="checkbox"/> < 2 meses <input type="checkbox"/> < 6 meses <input type="checkbox"/> < 12 meses <input type="checkbox"/> > 12 meses			
Acerca de la Iniciativa	Actividades Macro	Participantes			
Impacto: Alto Esfuerzo: Medio		<table border="1"> <tr> <td>Líder de proyectos</td> <td>A</td> </tr> <tr> <td>Recursos humanos</td> <td>AR</td> </tr> </table>	Líder de proyectos	A	Recursos humanos
Líder de proyectos	A				
Recursos humanos	AR				
Costos referenciales					
\$ 5000					

Figura 27. Conceptualización de iniciativa seleccionar equipo Scrum

6.1.4. Iniciativa: Implementar las áreas de proceso de Ingeniería de CMMI-DEV para el desarrollo de software

Las áreas de proceso de ingeniería se aplican al desarrollo de cualquier producto o servicio dentro del dominio de desarrollo (productos de software, productos de hardware, servicios, proceso). Las cinco áreas de proceso de Ingeniería de CMMI-DEV son las siguientes (CMMI, 2010):

- Integración del Producto (PI)
- Desarrollo de Requisitos (RD)
- Solución Técnica (TS)
- Validación (VAL)

- Verificación (VER)

6.1.4.1. Detalles de la implementación

El área de proceso de Desarrollo de Requisitos identifica las necesidades del cliente y las convierte en requisitos de producto, todos los requisitos son analizados para elaborar una solución en alto nivel. (CMMI, 2010)

El conjunto de requisitos de producto describe claramente las características, diseño y atributos, en términos que el desarrollador puede comprender y usar. El área de Desarrollo de Requisitos envía los requerimientos al área de proceso de Solución Técnica, donde se transforman en arquitectura y diseño. Los requisitos son enviados también al área de Integración del Producto donde se combinan los componentes del producto y se verifican las interfaces. (CMMI, 2010)

El área de proceso de Solución Técnica desarrolla los componentes para que sean utilizados por el área de proceso de Integración. La Solución Técnica se basa en prácticas específicas del área de procesos de Verificación, quienes se encargan de asegurar que los productos de trabajo cumplen con los requisitos especificados, la revisión utiliza diferentes métodos, y se puede utilizar la técnica de colaboración entre pares como un método para eliminar defectos, este tipo de verificación se lo puede realizar en un ambiente operacional o simulado. (CMMI, 2010)

La Integración del Producto utiliza las prácticas específicas para generar una estrategia de integración de los componentes del producto y entregar el producto directamente al cliente. La integración utiliza prácticas tanto de Verificación como de Validación en su proceso de integración. (CMMI, 2010)

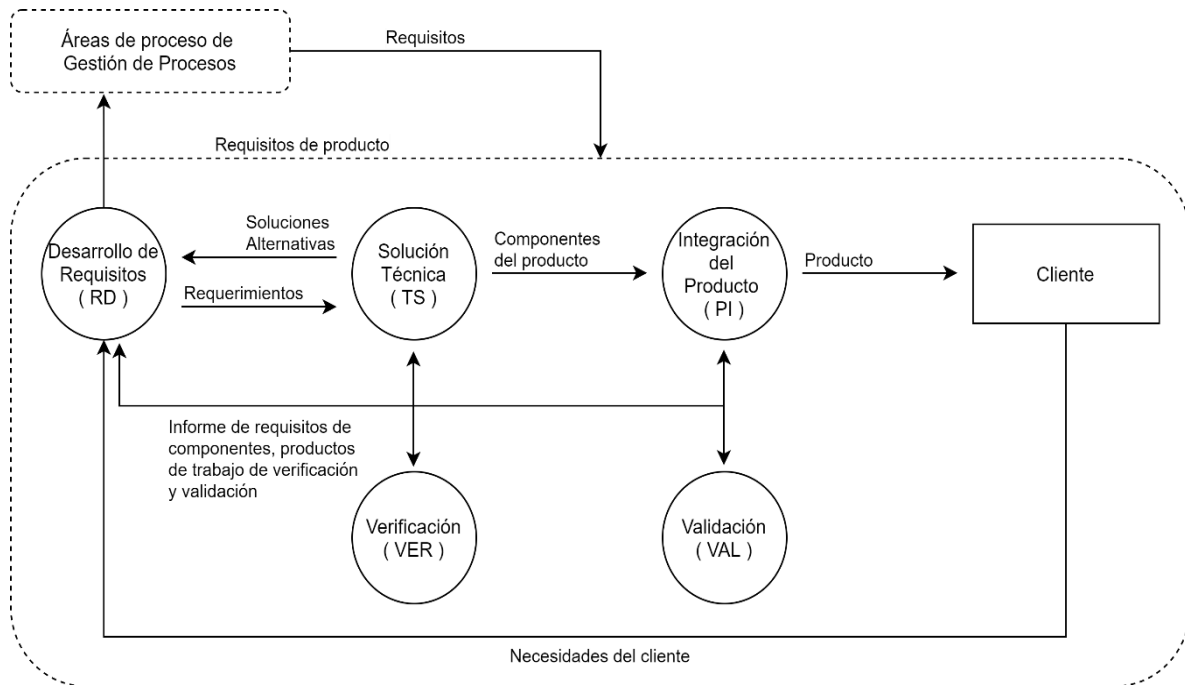


Figura 28. Áreas de proceso de Ingeniería. (CMMI, 2010)

6.1.4.2. Conceptualización de la implementación

La siguiente figura presenta los principales aspectos de la implementación.

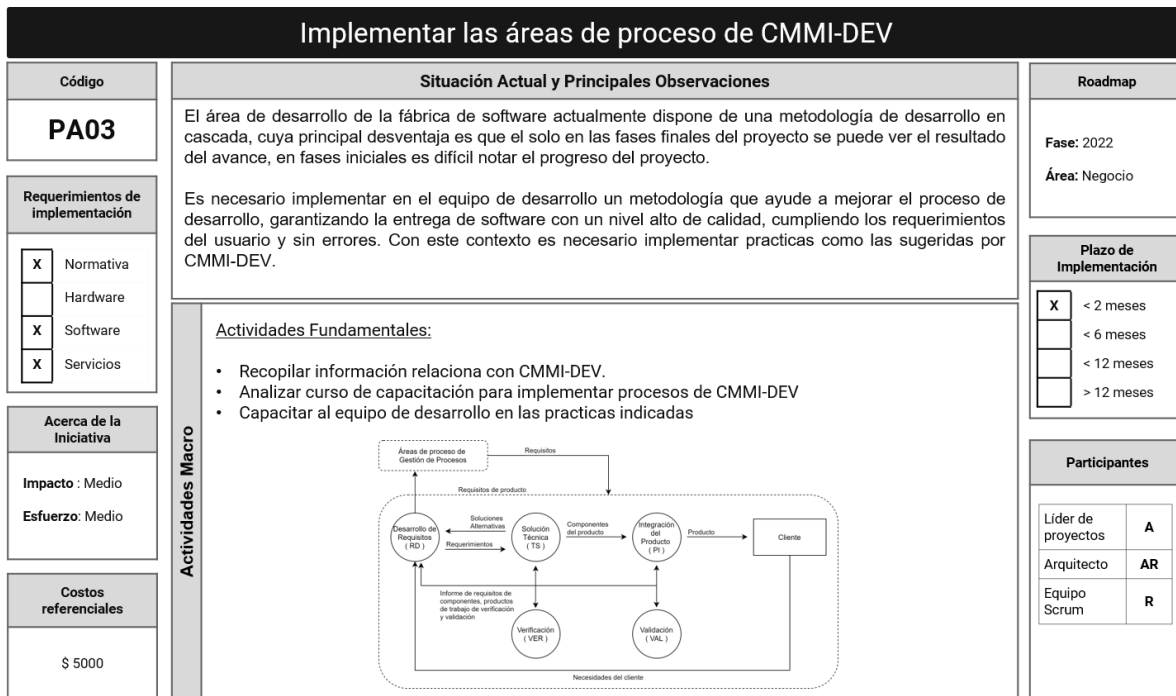


Figura 29. Conceptualización de iniciativa implementar áreas de proceso de CMMI-DEV

6.2. Arquitectura de Aplicaciones y Datos

La fábrica de software actualmente no cuenta con aplicaciones que les permita optimizar sus procesos entre los principales se tiene:

- Herramientas colaborativas para manejo del proyecto.
- Herramientas para el despliegue continuo.
- Herramientas para análisis de código.
- Herramientas para la gestión de datos e información.
- Herramientas para automatización de pruebas.
- Plataforma para fortalecer el análisis de datos.

6.2.1. Iniciativa: Implementar la herramienta colaborativa Jira

Jira permite a los miembros del equipo gestionar las tareas y actividades del sprint, desde la planificación, gestión de requisitos, casos de prueba, es una herramienta ideal para equipos que trabajan con metodologías ágiles.

6.2.1.1. Características de la implementación

La implementación de Jira, tiene una participación relevante en diferentes aspectos de la fábrica de software:

Usuarios:

- Product Owner
- Scrum Master
- Equipo Scrum
- Líder de proyecto
- Gerente de proyectos

Casos de uso:

- Supervisión de errores
- Gestión de proyectos
- Gestión de productos
- Gestión de procesos
- Gestión de tareas
- Desarrollo de software
- Desarrollo de software ágil
- Gestión de pruebas

Integraciones importantes:

- Confluence
- Bitbucket
- Trello
- Slack
- GitHub
- Microsoft
- Google

Opciones de alojamiento:

- Cloud

Licencia:

- Todos los usuarios de Jira Software pueden acceder a las funciones de Jira.
- Número de licencias a gestionar 20.

Costos:

- Pago mensual Jira Software Standard \$150. (Atlassian, 2021c)

6.2.1.2. Conceptualización de la implementación

Para la configuración del primer proyecto se realiza los siguientes pasos (Atlassian, 2021d):



Figura 30. Conceptualización de iniciativa implementar Jira

6.2.2. Iniciativa: Implementar CircleCI

La integración continua (CI) es un método para probar código continuamente, se monitorea automáticamente cada una de las versiones de código los cambios permiten ejecutar eventos como suites de pruebas, despliegues en ambientes, notificaciones.

Funciones:

- Desencadenadores de notificaciones a partir de eventos de CI
- Rendimiento optimizado para compilaciones rápidas
- Depuración sencilla a través de compilaciones locales y SSH
- Análisis para medir el rendimiento de compilación

6.2.2.1. Características de la implementación

La implementación de CircleCI permite utilizar integración continua dicho método influye en aspectos de fábrica de software como:

Usuario:

- Equipo Scrum

Costos:

- Pago mensual CircleCI Performance Standard \$30.

6.2.2.2. Conceptualización de la implementación

A continuación, se detallan los pasos necesarios para poder poner en marcha la herramienta CircleCi:

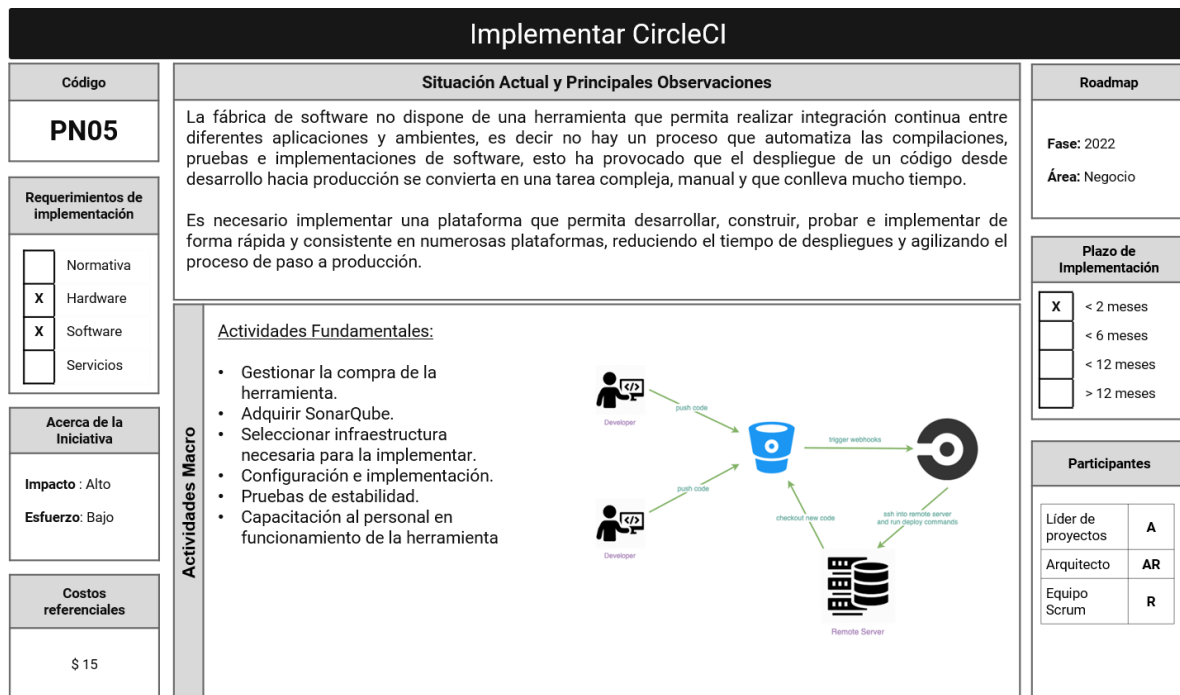


Figura 31. Conceptualización de iniciativa implementar CircleCI

6.2.3. Iniciativa: Implementar SonarQube

La fábrica de software necesita implementar una plataforma para evaluar la calidad del código fuente, para solventar dicho requerimiento se puede implementar la plataforma SonarQube que permite realizar un análisis estático y permite encontrar puntos de mejora, así como métricas de diferente índole.

6.2.3.1. Características de la implementación

Fortalecer el análisis del código fuente desarrollado por cada uno de los miembros de la fábrica de software, contribuye a solucionar una de sus debilidades, la implementación afecta a los siguientes aspectos:

Usuarios:

- Equipo de desarrollo

Funciones principales:

- Detección de código duplicado
- Detección de código muerto
- Análisis de estándares de codificación
- Búsqueda y reporte de Bugs
- Complejidad ciclomática
- Comentarios
- Test unitarios y de integración
- Cobertura de código

Integraciones:

- Github
- AzureDevops
- Bitbucket
- Gitlab

- Docker

Costos:

- Pago anual \$150

6.2.3.2. Conceptualización de la implementación

La implementación de SonarQube tiene una serie de aspectos importantes a considerar los que se sintetizan en la figura 32.

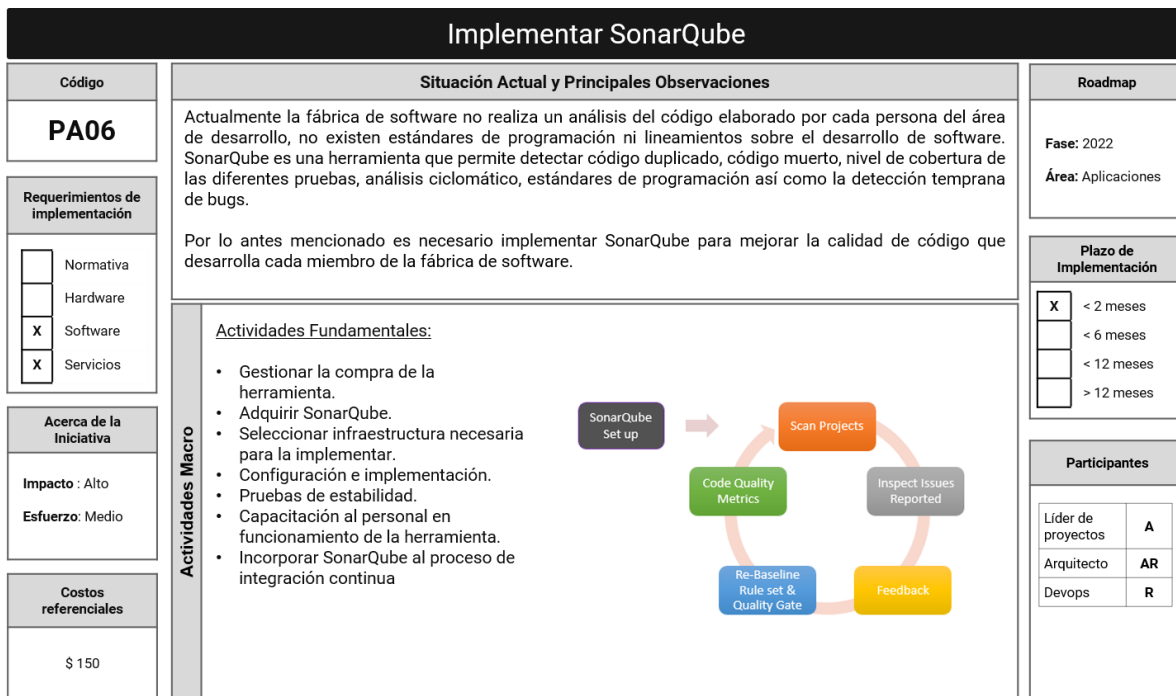


Figura 32. Conceptualización de iniciativa implementar SonarQube

La implementación de SonarQube requiere instalar la herramienta del mismo nombre y un scanner que permite analizar el código fuente.

6.2.4. Iniciativa: Implementar Confluence

Confluence es una plataforma que permite almacenar en un único lugar el contenido y documentación generada por el equipo de trabajo, permitiendo a todos los miembros del equipo tener visibilidad y acceso a la información.

6.2.4.1. Características de la iniciativa

Se requiere manejar un correcto flujo de la información generada por los miembros del equipo de desarrollo, Scrum master y Product owner, dicha implementación tendrá afectaciones en:

Usuarios:

- Product Owner
- Scrum master
- Equipo Scrum

Costos:

- Pago mensual Confluence estándar \$120 USD.

6.2.4.2. Detalles de implementación

La implementación de Confluence tiene una serie de pasos que son necesarios para poder utilizar la herramienta de forma adecuada.

Paso 1: Conocer los espacios

Los espacios son la forma de organizar la información, los espacios son grupos de páginas que guardan relación, en las que trabajan usuarios que tengan los accesos, se puede crear la cantidad de espacios necesarios.

Para la fábrica de software es necesario crear los siguientes espacios:

- Documentación técnica

- Documentación de pasos a producción
- Documentación de reuniones
- Diccionario de datos
- Acuerdos de equipos

Paso 2: Crear un espacio

Una vez identificados los espacios, es necesario crearlos, para lo cual se siguen los siguientes pasos (Atlassian, 2021a):

- Ir al sitio de Confluence de la organización.
- En la pantalla de inicio, seleccionar Crear espacio.
- Seleccionar el tipo de espacio a crear.
- Rellenar los campos Nombre del espacio, Clave del espacio y demás datos.
- Establecer los permisos para tu espacio.
- Seleccionar Crear.

Paso 3: Personalizar la presentación de tu espacio

Cada espacio proviene de una presentación por defecto que nos permite explorar las diferentes funciones, por tal motivo es necesario personalizar con el objetivo y funciones requeridas por el equipo de trabajo.

Paso 4: Organizar tu contenido

Para organizar el contenido se recomienda:

- Usar las páginas principales para agrupar contenidos similares
- Crear atajos para las páginas importantes
- Etiquetar páginas y adjuntos
- Mantener el contenido organizado

Paso 5: Gestionar usuarios y permisos

Los administradores de Confluence pueden gestionar usuarios grupos y permisos, es responsabilidad de cada administrador gestionar dichos accesos, los permisos globales permiten:

- Crear espacios
- Gestionar accesos a los perfiles de usuario
- Gestionar accesos a usuarios sin licencia
- Permitir aplicaciones que accedan al sitio

6.2.4.3. Conceptualización de la iniciativa

La siguiente figura nos muestra las principales características de la implementación:


Implementar Confluence																	
Código	Situación Actual y Principales Observaciones	Roadmap															
PA08	<p>La fábrica software actualmente no dispone una correcta gestión de datos y la información generada se almacena en diferentes lugares haciendo difícil su búsqueda cuando se la necesite. Han existido esfuerzos aislados por intentar manejar la información, pero por falta de apoyo y organización no se han consolidado esas ideas, existe información como acuerdos de equipos, informes de reuniones, documentos de pasos a producción entre otros, que se han extraviado por la falta de una correcta gestión.</p> <p>Por los datos anteriormente misionados la fábrica de software busca:</p> <ul style="list-style-type: none"> • Construir una cultura de trabajo colaborativa. • Organizar todo en un solo sitio. • Convertir cada conversación en una acción. 	<p>Fase: 2022</p> <p>Área: Aplicaciones</p>															
Requerimientos de implementación	Actividades Macro	Plazo de Implementación															
<table border="1"> <tr><td><input type="checkbox"/></td><td>Normativa</td></tr> <tr><td><input type="checkbox"/></td><td>Hardware</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Software</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Servicios</td></tr> </table>		<input type="checkbox"/>	Normativa	<input type="checkbox"/>	Hardware	<input checked="" type="checkbox"/>	Software	<input checked="" type="checkbox"/>	Servicios	<table border="1"> <tr><td><input checked="" type="checkbox"/></td><td>< 2 meses</td></tr> <tr><td><input type="checkbox"/></td><td>< 6 meses</td></tr> <tr><td><input type="checkbox"/></td><td>< 12 meses</td></tr> <tr><td><input type="checkbox"/></td><td>> 12 meses</td></tr> </table>	<input checked="" type="checkbox"/>	< 2 meses	<input type="checkbox"/>	< 6 meses	<input type="checkbox"/>	< 12 meses	<input type="checkbox"/>
<input type="checkbox"/>	Normativa																
<input type="checkbox"/>	Hardware																
<input checked="" type="checkbox"/>	Software																
<input checked="" type="checkbox"/>	Servicios																
<input checked="" type="checkbox"/>	< 2 meses																
<input type="checkbox"/>	< 6 meses																
<input type="checkbox"/>	< 12 meses																
<input type="checkbox"/>	> 12 meses																
Acerca de la Iniciativa	<p><u>Actividades Fundamentales:</u></p> <ul style="list-style-type: none"> • Gestionar la compra de la herramienta. • Adquirir Confluence. • Seleccionar infraestructura necesaria para la implementar. • Configuración e implementación. • Pruebas de estabilidad. • Capacitación al personal en funcionamiento de la herramienta. 	Participantes															
<p>Impacto: Bajo</p> <p>Esfuerzo: Bajo</p>		<table border="1"> <tr><td>Líder de proyectos</td><td>A</td></tr> <tr><td>Scrum Master</td><td>AR</td></tr> <tr><td>Arquitecto</td><td>R</td></tr> </table>	Líder de proyectos	A	Scrum Master	AR	Arquitecto	R									
Líder de proyectos	A																
Scrum Master	AR																
Arquitecto	R																
Costos referenciales																	
\$ 110																	

Figura 33. Conceptualización de iniciativa implementar Confluence

6.2.5. Iniciativa: Implementar PowerBI

Es una herramienta que se utiliza para crear cuadros de mando que permiten tomar decisiones. La información se actualiza de forma automática y permite compartirla dentro de la propia aplicación.

Las principales capacidades de PowerBI son (Duque, 2019):

- Permitir importar datos y modelarlos de la manera que creamos conveniente
- Facilitar cruzar datos, hacer cálculos, así como gráficos sofisticados de manera relativamente sencilla.
- Reutilizar los cálculos, lo que permite un manejo correcto de la información reduciendo los errores.
- Representar la información en tablas y gráficos a los cuales se le puede implementar una serie de filtros, para poder agrupar la información.

6.2.5.1. Características de la implementación

Usuarios:

- Product Owner
- Líder de proyecto
- Gerente de proyectos

Costos:

- Suscripción con costo \$10 USD por usuario al mes.

6.2.5.2. Detalles de la Implementación

Los pasos principales para crear un informe en Power Bi son (Microsoft, 2021):

- Iniciar sesión en la cuenta de Power BI en línea (o registrarse, si todavía no tiene).

- Abrir el servicio Power BI.
- Obtener algunos datos y abrirlos en la vista de informe.
- Usar esos datos para crear visualizaciones y guardarlos como un informe.
- Crear un panel anclando iconos desde el informe.
- Agregar otras visualizaciones al panel mediante la herramienta de lenguaje natural Preguntas y respuestas.
- Cambiar el tamaño, reorganizar y editar los detalles de los iconos en el panel.
- Limpiar los recursos mediante la eliminación del conjunto de datos, el informe y el panel.

6.2.5.3. Conceptualización de la iniciativa

La implementación de PowerBI, tiene una serie de características a ser consideradas como:

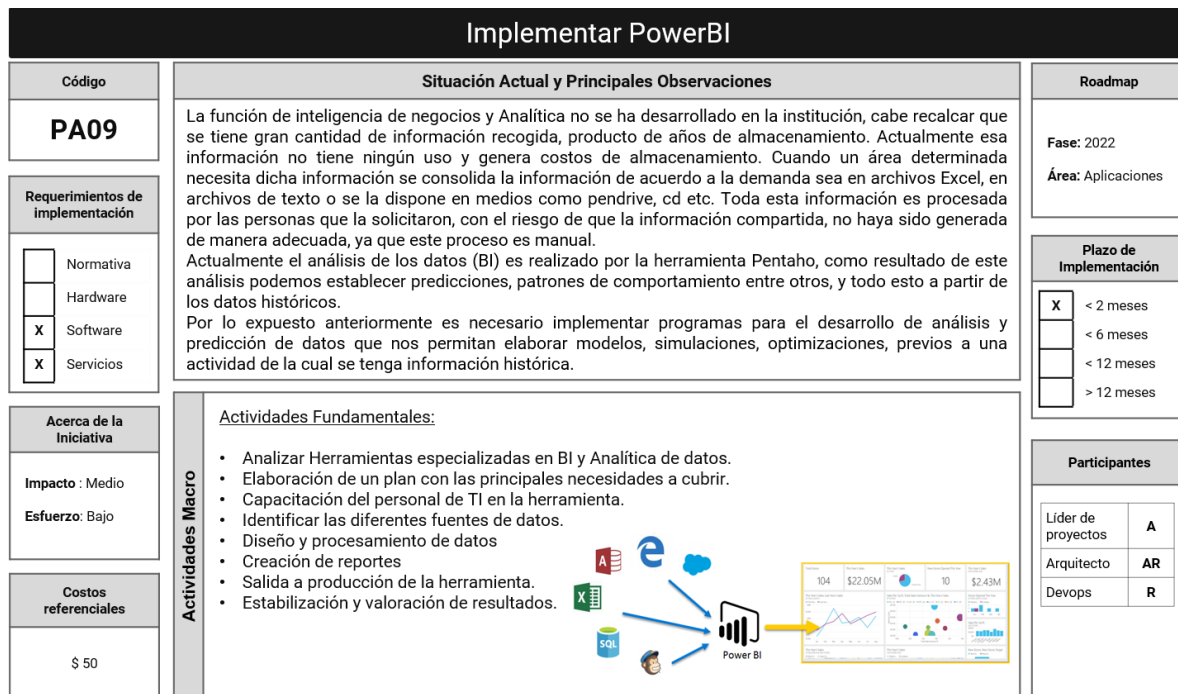


Figura 34. Conceptualización de iniciativa PowerBI

6.2.6. Iniciativa: Implementar Cypress

Las pruebas End-to-End es una metodología que permite asegurar la calidad del software para probar una aplicación desde el inicio hasta el final. Su objetivo es simular al máximo el comportamiento de un usuario real. (Desna, 2020).

Cypress es un framework de testing que se caracteriza por ser rápido y fácil de usar que se utiliza para realizar pruebas sobre aplicaciones web. Cypress maneja una sola dependencia además dispone de una interfaz gráfica que permite disponer en tiempo real cada uno de los pasos y las diferentes acciones realizadas en la prueba, así como los test que no fueron exitosos.

6.2.6.1. Características de la implementación

Usuarios:

- Equipo Scrum

Costos:

- La versión gratuita permite obtener las funcionalidades necesarias.

6.2.6.2. Detalles de la implementación

Para instalar Cypress se requiere tener instalado NodeJS y NPM, una vez instalados se prosigue con los siguientes pasos.

Paso1: Instalar Cypress en el proyecto NPM o en un proyecto desde el inicio

- `npm install cypress -D`

Para un proyecto desde el inicio:

- `mkdir nombre-del-proyecto && cd nombre-del-proyecto`
- `npm init -y`

- `npm install cypress -D`

Paso 2: Añadir script

En el fichero `package.json` agregar

```
"cy:open": "./node_modules/.bin/cypress open"
```

Paso 3: Iniciar Cypress

Ejecutar el comando `npm run cy:open`

Crear un nuevo test

Paso 4: Ejecutar el test

En la pantalla levantada por Cypress (Test Runner), seleccionar el test a ejecutar.

6.2.6.3. Conceptualización de la iniciativa

La figura 35 permite identificar las principales características de la implementación:

Implementar Cypress																		
Código	Situación Actual y Principales Observaciones	Roadmap																
PA07	<p>La fábrica de software desde su nacimiento no ha manejado una cultura de pruebas en el área de calidad, ejecuta pruebas de enfocadas en el desarrollo y de forma manual, pero cuando las pruebas tienen muchos procesos integrados o dependientes estas pruebas se pueden alargar en tiempo y complejidad dando pasos a posibles errores.</p> <p>Por lo expuesto anteriormente es necesario implementar una herramienta que permita automatizar las pruebas de diferentes flujos, pruebas de las páginas web simulando la manera de interactuar de las personas con los desarrollos web de la fábrica, generar escenarios de pruebas, validar que los componentes de la página se encuentran funcionando de acuerdo a lo esperado.</p>	<p>Fase: 2022</p> <p>Área: Aplicaciones</p>																
Requerimientos de implementación	<p>Actividades Fundamentales:</p> <ul style="list-style-type: none"> • Gestionar la compra de la herramienta. • Adquirir Cypress. • Seleccionar infraestructura necesaria para la implementar. <ul style="list-style-type: none"> • Configuración e implementación. • Capacitación al personal del área de calidad. • Programar y analizar un escenario de prueba. 	Plazo de Implementación																
<table border="1"> <tr><td><input type="checkbox"/></td><td>Normativa</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Hardware</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Software</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Servicios</td></tr> </table>		<input type="checkbox"/>	Normativa	<input checked="" type="checkbox"/>	Hardware	<input checked="" type="checkbox"/>	Software	<input checked="" type="checkbox"/>	Servicios	<table border="1"> <tr><td><input checked="" type="checkbox"/></td><td>< 2 meses</td></tr> <tr><td><input type="checkbox"/></td><td>< 6 meses</td></tr> <tr><td><input type="checkbox"/></td><td>< 12 meses</td></tr> <tr><td><input type="checkbox"/></td><td>> 12 meses</td></tr> </table>	<input checked="" type="checkbox"/>	< 2 meses	<input type="checkbox"/>	< 6 meses	<input type="checkbox"/>	< 12 meses	<input type="checkbox"/>	> 12 meses
<input type="checkbox"/>		Normativa																
<input checked="" type="checkbox"/>		Hardware																
<input checked="" type="checkbox"/>	Software																	
<input checked="" type="checkbox"/>	Servicios																	
<input checked="" type="checkbox"/>	< 2 meses																	
<input type="checkbox"/>	< 6 meses																	
<input type="checkbox"/>	< 12 meses																	
<input type="checkbox"/>	> 12 meses																	
Acerca de la Iniciativa	<p>Actividades Macro</p>	Participantes																
<p>Impacto : Medio</p> <p>Esfuerzo: Bajo</p>		<table border="1"> <tr><td>Lider de proyectos</td><td>A</td></tr> <tr><td>Arquitecto</td><td>AR</td></tr> <tr><td>Devops</td><td>R</td></tr> </table>	Lider de proyectos	A	Arquitecto	AR	Devops	R										
Lider de proyectos	A																	
Arquitecto	AR																	
Devops	R																	
Costos referenciales																		
\$0																		

Figura 35. Conceptualización de iniciativa implementar Cypress

6.3. Arquitectura de infraestructura base

La infraestructura actual de la fábrica de software ha iniciado un proceso de migración a la nube, actualmente dispone de infraestructura en la plataforma de Google (GCP).

6.3.1. Iniciativa: Fortalecer la infraestructura en la nube

El proceso de migración a la nube se encuentra en proceso y en una fase estable, pero es necesario fortalecer algunos aspectos como:

- Seguridad: Existe la necesidad de generar una red virtual exclusiva para la gestión y administración de la base de datos, ya que almacena información privada de los usuarios.
- Disponibilidad: Es necesario generar nodos adicionales de diferentes servidores, para poder balancear el número de transacciones y tener una alta disponibilidad en los servicios

- Monitoreo: Implementar Monitoring de la plataforma de Google para establecer un monitoreo de la infraestructura.
- Registro de logs: Implementar Logging de la plataforma de Google que junto con la plataforma de monitoreo se complementan para poder generar métricas del comportamiento de la infraestructura.

Los costos estimados de estas implementaciones se detallan en la siguiente tabla:



Edison Chancusig.

Your Estimated Bill *

Estimated Monthly Cost: USD 1,033.42

 1 x	e2-standard-4	730 total hours per month	USD 117.52
 1 x boot disk	Persistent Disk - Compute Engine	100 GiB	USD 20.40
 2 x	e2-standard-2	1460 total hours per month	USD 117.52
 2 x boot disk	Persistent Disk - Compute Engine	100 GiB	USD 40.80
db-standard-2	500 GB	730 total hours per month	USD 450.12
Operations	Volume of logs	100 GiB	USD 25.00
 2 x	e2-medium	1460 total hours per month	USD 58.75
 2 x boot disk	Persistent Disk - Compute Engine	50 GiB	USD 20.40
 2 x	e2-standard-2	1460 total hours per month	USD 117.52
 2 x boot disk	Persistent Disk - Compute Engine	10 GiB	USD 0.96
Forwarding rules	Forwarding rules	5	USD 20.44
Load Balancer ingress	Ingress	20 GiB	USD 0.20
Forwarding rules	Forwarding Rules	5	USD 43.80
Total Estimated Monthly Cost			USD 1,033.42

Figura 36. Costos estimados del fortalecimiento de la infraestructura. (Calculadora de precios de Google Cloud Platform, 2021)

6.3.1.1. Conceptualización de la iniciativa

La infraestructura en la nube tiene aspectos como seguridad, disponibilidad, monitoreo y registro de logs que tienen que ser implementados, en la figura 37 se muestra los principales aspectos.


Fortalecer la infraestructura en la nube																		
Código	Situación Actual y Principales Observaciones	Roadmap																
PA10	<p>Actualmente la fábrica de software no implementa ninguna metodología ágil, además el personal que dispone no tiene las competencias adecuadas para implementar el marco de trabajo Scrum; no obstante existen pocas personas que se adaptan perfectamente al nuevo marco de trabajo, ya que se han capacitado de forma personal en el marco de trabajo propuesto, las demás personas deben entrar a un proceso de capacitación.</p> <p>Por los antecedentes descritos es necesario contratar nuevos perfiles que se incorporen para completar el equipo Scrum, ya que algunos perfiles de la fábrica de software pueden cubrir ciertos cargos en necesario reforzar con perfiles que no existen actualmente.</p>	<p>Fase: 2022</p> <p>Área: Infraestructura</p>																
<p>Requerimientos de implementación</p> <table border="1"> <tr><td><input type="checkbox"/></td><td>Normativa</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Hardware</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Software</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Servicios</td></tr> </table>	<input type="checkbox"/>	Normativa	<input checked="" type="checkbox"/>	Hardware	<input checked="" type="checkbox"/>	Software	<input checked="" type="checkbox"/>	Servicios	<p>Actividades Macro</p> <p><u>Actividades Fundamentales:</u></p> <ul style="list-style-type: none"> • Analiza costos de implementación. • Asignar impacto de las iniciativas. • Ejecución de mejora en seguridad. • Ejecución de mejora en disponibilidad. • Implementar monitoring y Logging. • Fase de estabilización. • Fase de liberación y socialización • Monitoreo de cambios en infraestructura. 	<p>Plazo de Implementación</p> <table border="1"> <tr><td><input type="checkbox"/></td><td>< 2 meses</td></tr> <tr><td><input type="checkbox"/></td><td>< 6 meses</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>< 12 meses</td></tr> <tr><td><input type="checkbox"/></td><td>> 12 meses</td></tr> </table>	<input type="checkbox"/>	< 2 meses	<input type="checkbox"/>	< 6 meses	<input checked="" type="checkbox"/>	< 12 meses	<input type="checkbox"/>	> 12 meses
<input type="checkbox"/>	Normativa																	
<input checked="" type="checkbox"/>	Hardware																	
<input checked="" type="checkbox"/>	Software																	
<input checked="" type="checkbox"/>	Servicios																	
<input type="checkbox"/>	< 2 meses																	
<input type="checkbox"/>	< 6 meses																	
<input checked="" type="checkbox"/>	< 12 meses																	
<input type="checkbox"/>	> 12 meses																	
<p>Acerca de la Iniciativa</p> <p>Impacto: Alto</p> <p>Esfuerzo: Alto</p>		<p>Participantes</p> <table border="1"> <tr><td>Líder de proyectos</td><td>A</td></tr> <tr><td>Arquitecto</td><td>AR</td></tr> <tr><td>Devops</td><td>R</td></tr> </table>	Líder de proyectos	A	Arquitecto	AR	Devops	R										
Líder de proyectos	A																	
Arquitecto	AR																	
Devops	R																	
<p>Costos referenciales</p> <p>\$ 1100</p>																		

Figura 37. Conceptualización de iniciativa fortalecer la nube de la fábrica de software

7. PLAN DE MIGRACIÓN

El presente capítulo propone una priorización de las diferentes iniciativas. Se analizará el impacto y el esfuerzo de cada una de ellas para elaborar un plan estructurado con las fases de implementación.

7.1. Análisis de impacto

Para el análisis de impacto se realizará una matriz que relaciona los objetivos de la fábrica de software y las iniciativas desarrolladas en capítulos anteriores. Se considera una valoración de 0 a 2 puntos según el impacto y utilizará la siguiente rúbrica:

Tabla 34

Escala de Impacto

Escala de Impacto		
Bajo	0 - 0,7	
Medio	0,7 - 1,4	
Alto	1,4 - 2	

Tabla 35

Análisis de impacto

No	Dominio	Id	Iniciativa	Objetivos						Valoración cualitativa	Impacto
				25%	15%	15%	12%	13%	20%		
				Sostenibilidad de los servicios de TI	Contribuir a la experiencia del cliente interno - externo	Fortalecer los controles de calidad y desarrollo de software	Implementar una plataforma de TI segura, resiliente e iteroperable	Desarrollar, operar y actualizar los proyectos teniendo en cuenta buenas	Generar mayores ingresos (reduciendo costos y obteniendo mayores		
1	Negocio	PN01	Implementar el marco de trabajo Scrum							1,87	Alto
2	Negocio	PN02	Seleccionar el personal para el equipo Scrum							1,76	Alto
3	Negocio	PN03	Implementar proceso CMMI-DEV							1,38	Medio
4	Aplicaciones	PA04	Implementar Jira							1,15	Medio
5	Aplicaciones	PA05	Implementar CircleCi							1,55	Alto
6	Aplicaciones	PA06	Implementar SonarQube							1,56	Alto
7	Aplicaciones	PA07	Implementar Cypress							0,98	Medio
8	Aplicaciones	PA08	Implementar Confluence							0,63	Bajo
9	Datos	PA09	Implementar PowerBI							1,13	Medio
10	Infraestructura base	PI10	Fortalecer la infraestructura en la nube							1,52	Alto

La tabla de análisis de impacto, permite concluir que existen cinco iniciativas de alto impacto, las cuales tienen que ser priorizadas sobre las que tienen un menor impacto.

7.2. Análisis de esfuerzo

El análisis de esfuerzo considera tres criterios, recursos económicos, complejidad y capacidad de TI.

Se considera una valoración de 0 a 3 puntos y se utiliza la siguiente rúbrica:

Tabla 36

Escala de esfuerzo

Escala de Esfuerzo		
Bajo	1 - 1,7	
Medio	1,7 - 2,4	
Alto	2,4 - 3	

Tabla 37

Escala de ponderación, recursos, complejidad, capacidad de TI.

Valor	Recursos económicos	Complejidad	Capacidad de TI	
1	Menos \$10000	Baja	Listo	
2	\$10000 y \$150000	Media	En proceso	
3	Más de \$150000	Alta	A futuro	

Tabla 38

Análisis de esfuerzo

No	Área	Id	Iniciativa	Criterios Esfuerzo			Suma Ponderada	Esfuerzo
				40% Recursos Económicos	30% Complejidad	30% Capacidad TI		
1	Negocio	PN01	Implementar el marco de trabajo Scrum	1	2	2	1,6	Bajo
2	Negocio	PN02	Seleccionar el personal para el equipo Scrum	2	2	2	2	Medio
3	Negocio	PN03	Implementar proceso CMMI-DEV	2	1	3	2	Medio
4	Aplicaciones	PA04	Implementar Jira	2	2	3	2,3	Bajo
5	Aplicaciones	PA05	Implementar CircleCi	1	2	2	1,6	Bajo
6	Aplicaciones	PA06	Implementar SonarQube	2	2	1	1,7	Medio
7	Aplicaciones	PA07	Implementar Cypress	1	1	1	1	Bajo
8	Aplicaciones	PA08	Implementar Confluence	1	1	2	1,3	Bajo
9	Datos	PA09	Implementar PowerBI	1	1	1	1	Bajo
10	Infraestructura base	PI10	Fortalecer la infraestructura en la nube	3	3	2	2,7	Alto

Fortalecer la infraestructura en la nube de la fábrica de software es la iniciativa que mayor esfuerzo representa ya que es una iniciativa costosa, con un alto grado de complejidad y el personal de la fábrica no tiene el conocimiento indicado para realizar dicho proceso.

7.3. Fases

En base a los resultados obtenidos en el análisis de impacto y de esfuerzo se clasifican las diferentes iniciativas en fases. Cada fase será implementada según un cronograma de actividades.

Tabla 39

Fases de implementación

No	Dominio	Id	Iniciativa	Impacto	Esfuerzo	Prioridad	Fase
1	Negocio	PN01	Implementar el marco de trabajo Scrum	Alto	Bajo	Habilitante	1
2	Negocio	PN02	Seleccionar el personal para el equipo Scrum	Alto	Medio	Habilitante	1
3	Negocio	PN03	Implementar proceso CMMI DEV	Medio	Medio	Habilitante	3
4	Aplicaciones	PA04	Implementar Jira	Medio	Bajo	Medio	4
5	Aplicaciones	PA05	Implementar CircleCi	Alto	Bajo	Habilitante	2
6	Aplicaciones	PA06	Implementar SonarQube	Alto	Medio	Habilitante	2
7	Aplicaciones	PA07	Implementar Cypress	Medio	Bajo	Bajo	5
8	Aplicaciones	PA08	Implementar Confluence	Bajo	Bajo	Bajo	5
9	Datos	PA09	Implementar PowerBI	Medio	Bajo	Bajo	4
10	Infraestructura base	PI10	Fortalecer la infraestructura en la nube	Alto	Alto	Alto	6

La figura 38 presenta un diagrama de dispersión donde se analiza el esfuerzo y el impacto de cada una de las iniciativas priorizadas.

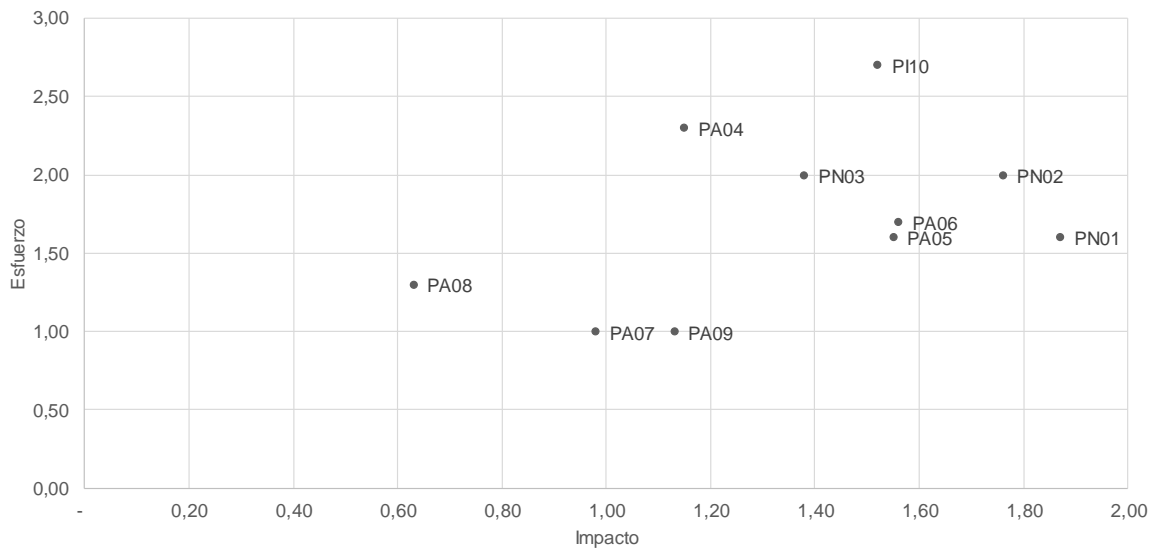


Figura 38. Priorización de iniciativas

7.4. Análisis de dependencias

A continuación, se presenta en la tabla 40 las iniciativas y sus correspondientes dependencias, se cruza cada iniciativa con todas las iniciativas que se priorizaron.

El diagrama del análisis de dependencias se lo presenta en el Anexo 2.

7.5. Mapa de ruta

El mapa de ruta permite ordenar los proyectos que se han priorizado con base al análisis de impacto, análisis de esfuerzo y las diferentes dependencias que existen entre ellos, con esa premisa se propone el siguiente cronograma de actividades.

7.5.1. Cronograma de actividades

Se establece un cronograma de actividades, con las estimaciones de tiempo para cada una de las iniciativas mencionadas, de acuerdo a la priorización obtenida en la tabla 38 y tabla 39 de la fase anterior.

Tabla 40

Cronograma de implementación por fases

Nombre de tarea	Duración	Comienzo	Fin
Fase 1 - Marco de trabajo Scrum	40 días	lun 24/1/22	vie 18/3/22
Implementar el marco de trabajo	40 días	lun 24/1/22	vie 18/3/22
Kickoff de la iniciativa	3 días	lun 24/1/22	mié 26/1/22
Establecer flujos de información	4 días	jue 27/1/22	mar 1/2/22
Curso de Scrum	31 días	vie 4/2/22	vie 18/3/22
Seleccionar el personal adecuado para el equipo Scrum	40 días	lun 24/1/22	vie 18/3/22
Identificar las capacidades actuales del equipo	10 días	lun 24/1/22	vie 4/2/22
Publicar vacantes	10 días	lun 7/2/22	vie 18/2/22
Seleccionar del personal	5 días	lun 21/2/22	vie 25/2/22

Procesos de inducción	15 días	lun 28/2/22	vie 18/3/22
Fase 2 - Desarrollo y Operaciones (Devops)	12 días	lun 21/3/22	mar 5/4/22
Implementar CircleCi	12 días	lun 21/3/22	mar 5/4/22
Kickoff de la iniciativa	3 días	lun 7/3/22	mié 9/3/22
Configuración	5 días	jue 10/3/22	mié 16/3/22
Integración	3 días	jue 17/3/22	lun 21/3/22
Pruebas de aplicación	3 días	vie 1/4/22	mar 5/4/22
Implementar SonarQube	11 días	mar 22/3/22	mar 5/4/22
Kickoff de la iniciativa	2 días	mar 22/3/22	mié 23/3/22
Configuración	3 días	jue 24/3/22	lun 28/3/22
Integración	3 días	mar 29/3/22	jue 31/3/22
Pruebas de aplicación	3 días	vie 1/4/22	mar 5/4/22
Fase 3 - Fortalecer el desarrollo de software	10 días	mié 6/4/22	mar 19/4/22
Implementar las áreas de proceso de Ingeniería de CMMI-DEV	10 días	mié 6/4/22	mar 19/4/22
Kickoff de la iniciativa	2 días	mié 6/4/22	jue 7/4/22
Socializar el marco de referencia CMMI-DEV	2 días	vie 8/4/22	lun 11/4/22
Implementar prácticas	3 días	mar 12/4/22	jue 14/4/22
Generación de documentos	3 días	vie 15/4/22	mar 19/4/22
Fase 4 - Herramienta colaborativa	26 días	mié 20/4/22	mié 25/5/22
Implementar Jira	22 días	mié 20/4/22	jue 19/5/22
Kickoff de la iniciativa	2 días	mié 20/4/22	jue 21/4/22
Configuración y licencias	2 días	vie 22/4/22	lun 25/4/22
Integración	4 días	mar 26/4/22	vie 29/4/22
Pruebas de aplicación	3 días	lun 2/5/22	mié 4/5/22
Gestionar accesos	2 días	jue 5/5/22	vie 6/5/22
Implementar PowerBI	10 días	vie 6/5/22	jue 19/5/22
Kickoff de la iniciativa	2 días	vie 6/5/22	lun 9/5/22
Configuración	3 días	mar 10/5/22	jue 12/5/22
Integración	2 días?	vie 13/5/22	lun 16/5/22
Pruebas de aplicación	3 días	mar 17/5/22	jue 19/5/22
Fase 5 - Gestión de la información	11 días	jue 26/5/22	jue 9/6/22
Implementar Confluence	11 días	jue 26/5/22	jue 9/6/22
Kickoff de la iniciativa	2 días	vie 20/5/22	lun 23/5/22
Configuración	4 días	mar 24/5/22	vie 27/5/22
Integración	5 días	lun 30/5/22	vie 3/6/22
Gestión de accesos	3 días	lun 6/6/22	mié 8/6/22
Implementar Cypress	7 días	jue 26/5/22	vie 3/6/22
Kickoff de la iniciativa	3 días	jue 26/5/22	lun 30/5/22
Configuración	2 días	mar 31/5/22	mié 1/6/22
Integración	2 días	jue 2/6/22	vie 3/6/22

Fase 6 - Fortalecer la infraestructura	26 días	vie 10/6/22	vie 15/7/22
Fortalecer la infraestructura	15 días	vie 10/6/22	jue 30/6/22
Análisis de costos	3 días	vie 10/6/22	mar 14/6/22
Priorización de actividades	3 días	mié 15/6/22	vie 17/6/22
Ejecución de actividades	9 días	lun 20/6/22	jue 30/6/22

En la siguiente figura se relacionan las actividades y los diferentes tiempos para implementar las iniciativas priorizadas.

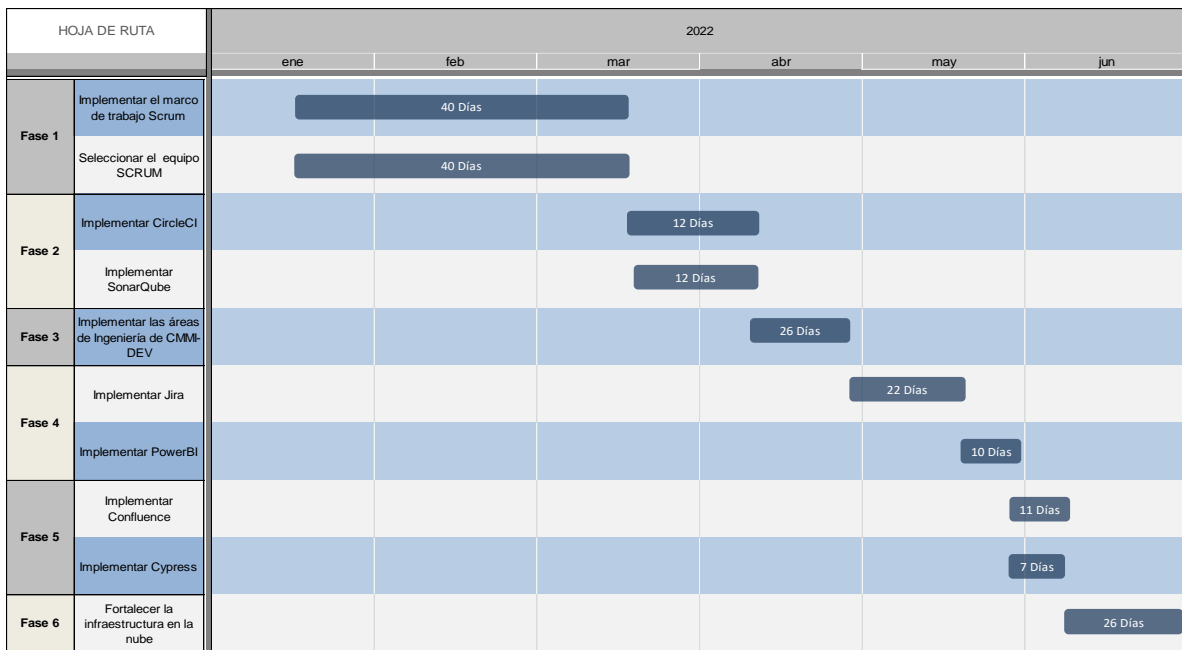


Figura 39. Road Map de iniciativas

Conclusiones

En base se realiza un análisis final tomando como base las observaciones y análisis desarrollados a lo largo de los anteriores capítulos:

- La gestión de proyectos de la fábrica de software se va a basar en Scrum considerando que es un marco de referencia que permite la generación incremental de valor en los proyectos de desarrollo tecnológico, esa generación incremental de valor puede generar resultados parciales en un proyecto que pueden entrar en operación y eso responde a la expectativa de la empresa de beneficiarse de aplicativos que apoyen a la empresa, además de que la generación en tiempos cortos habilita el mantener la confianza y patrocinio de los niveles ejecutivos en el personal de TI.
- El compromiso de STYC y su fábrica de software, de ofrecer servicios digitales óptimos y de excelente calidad se ve cubierto gracias a la ejecución de pruebas automatizadas, así como las prácticas de desarrollo recomendadas por la metodología CMMI-DEV, en cada una de estas actividades el equipo de desarrollo tiene que participar activamente.
- Debido al gran número de usuarios de canales digitales, es necesario contar con aplicaciones que ayuden al equipo de la fábrica de software a reducir los trabajos manuales, permitiendo enfocar sus esfuerzos en actividades que optimicen su trabajo.

Recomendaciones

La fábrica de software tiene una serie de inconvenientes que deben ser mejorados, razón por la cual se sugiere:

- Es indispensable contar con personal capacitado y comprometido para la implementación de las iniciativas propuestas, ya que estas personas cuentan con conocimientos, herramientas y habilidades que les permiten cumplir con las tareas asignadas, en tiempos reducidos, con la calidad esperada y sobre todo aportando con diferentes visiones para solucionar los problemas.
- La fábrica de software inició un proceso de migración a la nube de GCP (Google Cloud Platform), y se encuentra funcionando de manera estable, sin embargo, existen una serie de optimizaciones importantes que se tienen que implementar en busca de cubrir parámetros de seguridad, continuidad del negocio y monitoreo.
- Para poder implementar las iniciativas relacionadas a la arquitectura de negocio es indispensable contar con el equipo humano completo, de manera que todos los miembros de la fábrica puedan partir de un punto en común en busca del mismo objetivo la generación de valor.

Referencias

Abellán, E. (2020). *Scrum: Qué es y cómo funciona esta metodología*.

<https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>

Amazon. (2021). *Integración continua del software | Pruebas automatizadas |*

AWS. Amazon Web Services, Inc.

<https://aws.amazon.com/es/devops/continuous-integration/>

Asociación Española para la Calidad. (2019). *AEC - CMMI*. CMMI.

<https://www.aec.es/web/guest/centro-conocimiento/cmmi>

Atlassian. (2021a). *Configura tu sitio y tus espacios | Atlassian*. Atlassian.

<https://www.atlassian.com/es/software/confluence/guides/get-started/set-up>

Atlassian. (2021b). *Herramientas de integración continua*. Atlassian.

<https://www.atlassian.com/es/continuous-delivery/continuous-integration/tools>

Atlassian. (2021c). *Jira Pricing—Monthly and Annual Subscription Cost per User*.

Atlassian. <https://www.atlassian.com/software/jira/pricing>

Atlassian. (2021d). *Primeros pasos con Jira Software | Tutorial gratuito*. Atlassian.

<https://www.atlassian.com/es/software/jira/guides/getting-started/basics>

Calculadora de precios de Google Cloud Platform. (2021). Google Cloud.

<https://cloud.google.com/products/calculator?hl=es>

Cloud Logging. (2021). Google Cloud. [https://cloud.google.com/logging/docs/basic-](https://cloud.google.com/logging/docs/basic-concepts?hl=es)

[concepts?hl=es](https://cloud.google.com/logging/docs/basic-concepts?hl=es)

Cloud Monitoring. (2021). Google Cloud.

<https://cloud.google.com/monitoring?hl=es>

Cloud VPN. (2021). Google Cloud. [https://cloud.google.com/network-](https://cloud.google.com/network-connectivity/docs/vpn?hl=es)

[connectivity/docs/vpn?hl=es](https://cloud.google.com/network-connectivity/docs/vpn?hl=es)

CMMI, P. T. (2010, noviembre). *CMMI for Development, Version 1.3.*

<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9661>

COBIS. (2018). *COBIS alcanza el nivel 5 en el modelo CMMI.*

<https://blog.cobiscorp.com/cobis-excelencia-modelo-cmmi>

Desna, D. (2020). *Cypress: Instala y ejecuta tu primera prueba e2e en 5 minutos.*

Santander Global Tech. <https://santanderglobaltech.com/cypress-instala-y-ejecuta-tu-primera-prueba-e2e-5-minutos/>

DEV Community. (2021). *Profesionaliza tus proyectos personales con CircleCi.*

DEV Community. <https://dev.to/sebalinares/profesionaliza-tus-proyectos-personales-con-circleci-netlify-github-parte-1-esp-3l5j>

Documentación de Compute Engine. (2021). Google Cloud.

<https://cloud.google.com/compute/docs/nodes/sole-tenant-nodes?hl=es>

Duque, C. (2019). *Para qué se utiliza Power BI, qué es capaz de hacer y dónde*

descargarlo. - BIMÁTICO. <https://www.bimatico.com/es/bi-news/para-que-se-utiliza-power-bi-que-es-capaz-de-hacer-y-donde-descargarlo>

Estayno, M. G., & Meles, J. (2014). *El desafío de ser un Product Owner.* XX

Congreso Argentino de Ciencias de la Computación (Buenos Aires, 2014).

<http://sedici.unlp.edu.ar/handle/10915/42283>

- Gartner. (2021). *Gartner Reprint*. <https://www.gartner.com/doc/reprints?id=1-271OE4VR&ct=210802&st=sb>
- Gilart Iglesias, V., & Capella D'alton, A. (2005). *Entornos para el desarrollo de grandes aplicaciones de gestión de redes*. Alicante.
<https://www.dtic.ua.es/grupoM/recursos/articulos/JDARE-04-B.pdf>
- Microsoft. (2021). *Tutorial: Introducción a la creación en el servicio Power BI - Power BI*. <https://docs.microsoft.com/es-es/power-bi/fundamentals/service-get-started>
- Ortiz Vivas, A. (2017, julio 10). Ambientes de trabajo para software: La forma básica para desarrollar, probar y lanzar. *ortizvivas.com*.
<https://ortizvivas.com/blog/ambientes-de-trabajo-para-software/>
- Prestomedia. (2009). *CMMI-DEV, MODELO DE MEJORA DE PROCESOS DE DESARROLLO DE SOFTWARE*. <http://www.prestomedia.es/es/pag/68>
- Sanabria, E. (2021). Fases de la Metodología Scrum. *ComparaSoftware Blog*.
<https://blog.comparasoftware.com/fases-metodologia-scrum/>
- Schwaber, K., & Sutherland, J. (2020). *La Guía de Scrum*.
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- Scrum.org. (2021). *Professional Scrum Master™ III*. Scrum.Org.
<https://www.scrum.org/professional-scrum-master-iii-certification>
- SCRUMstudy. (2017). *Cuerpo de Conocimiento de Scrum*.
<https://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-3rd-edition-spanish.pdf>

SFIA Foundation. (2021). *SFIA y la gestión de habilidades*. SFIA. <https://sfia-online.org/es/about-sfia/sfia-and-skills-management>

SmartNodus. (2021). *Beneficios de la Integración Continua “CI” – SmartNodus*. <https://www.smartnodus.cl/integracion-continua/>

SonarSource S.A. (2021). *SonarScanner | SonarQube Docs*. <https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>

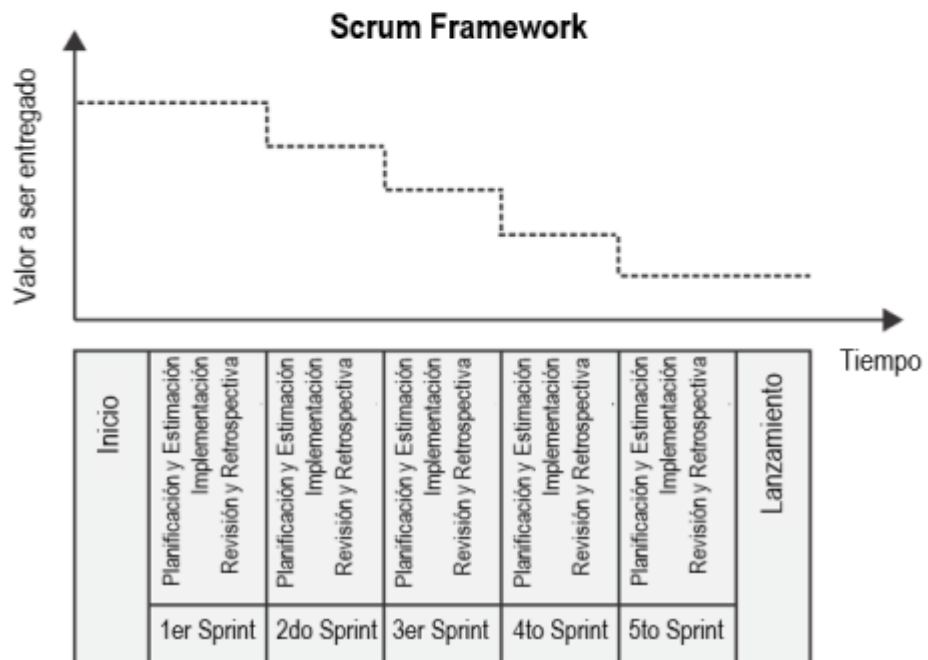
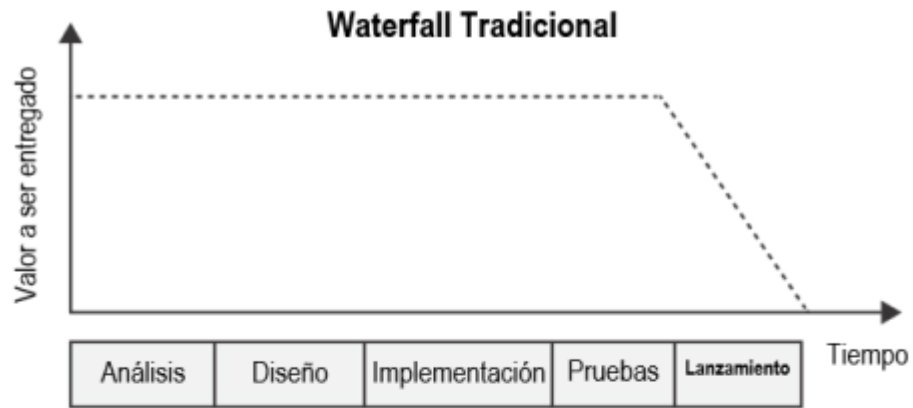
ANEXOS

Anexo 1

Área de Proceso	Categoría	Nivel de Madurez
Análisis Causal y Resolución (CAR)	Soporte	5
Gestión de Configuración (CM)	Soporte	2
Análisis de Decisiones y Resolución (DAR)	Soporte	3
Gestión integrada del Proyecto (IPM)	Gestión de proyectos	3
Medición y análisis (MA)	Soporte	2
Definición de Procesos de la Organización (OPD)	Gestión de procesos	3
Enfoque en Procesos de la Organización (OPF)	Gestión de procesos	3
Gestión del Rendimiento de la Organización (OPM)	Gestión de procesos	5
Rendimiento de Procesos de la Organización (OPP)	Gestión de procesos	4
Formación en la organización (OT)	Gestión de procesos	3
Integración del Producto (PI)	Ingeniería	3
Monitorización y Control del Proyecto (PMC)	Gestión de proyectos	2
Planificación del Proyecto (PP)	Gestión de proyectos	2
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)	Soporte	2
Gestión cuantitativa del Proyecto (QPM)	Gestión de proyectos	4
Desarrollo de Requisitos (RD)	Ingeniería	3
Gestión de Requisitos (REQM)	Gestión de proyectos	2
Gestión de Riesgos (RSKM)	Gestión de proyectos	3
Gestión de Acuerdos con Proveedores (SAM)	Gestión de proyectos	2
Solución Técnica (TS)	Ingeniería	3
Validación (VAL)	Ingeniería	3
Verificación (VER)	Ingeniería	3

Adaptado de (CMMI, 2010)

Anexo 3



(SCRUMstudy, 2017)

