



MAESTRÍA EN GERENCIA DE SISTEMAS Y TECNOLOGÍA EMPRESARIAL

Fortalecimiento de la fábrica de software in house de una entidad financiera

AUTOR

Daniel Armando Aguirre Alcívar

AÑO

2021



FACULTAD DE POSGRADOS

**TEMA**

FORTALECIMIENTO DE LA FÁBRICA DE SOFTWARE IN HOUSE DE UNA  
ENTIDAD FINANCIERA

**AUTOR**

Ing. Daniel Armando Aguirre Alcívar

**TUTOR**

MsC. Germán Pancho Carrera

**AÑO**

2021

## DECLARACION 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.”

A handwritten signature in blue ink, appearing to read 'Daniel Armando Aguirre Alcívar', is centered on the page.

Daniel Armando Aguirre Alcívar

1312311044

## Resumen

El presente proyecto muestra el análisis de la situación actual de la fábrica de software in house de una institución financiera en Ecuador, la cual pertenece al grupo de grandes bancos de Ecuador.

La institución financiera cuenta con un amplio portafolio de aplicaciones cuyo objetivo es brindar servicios y productos financieros de calidad y permitir a sus clientes la gestión de sus productos y servicios a través de los distintos aplicativos o plataformas tecnológicas disponibles para que el cliente no tenga la necesidad de acudir de manera presencial a las agencias bancarias. Para cumplir con estos objetivos es necesario que la organización cuente con herramientas tecnológicas actualizadas y flexibles, así como apoyarse en marcos de referencia que guíen su accionar al momento de realizar actividades específicas que le lleven a cumplir los objetivos.

Dentro de la operación diaria de la institución financiera se han encontrado problemas vinculados a la calidad, tiempos de entrega del software, así como la falta de adaptación ante cambios involucren nuevos desarrollos, retrasando de manera progresiva la entrega de valor a los clientes.

Después de realizar un análisis de la situación actual, necesidades y problemas, se proponen iniciativas para resolver los problemas descritos y explotar oportunidades tecnológicas y funcionales que permitan entregar valor a los clientes de la institución financiera. Estas definitivas están sustentadas y apoyadas en el marco de arquitectura empresarial Togaf, de igual forma el apoyo de marcos de trabajo complementario como son Scrum y prácticas de Devops apalancan la estrategia ideal para ser referente en la entrega continua, flexible, segura de productos de software.

## **Abstract**

This project shows the analysis of the current situation of the in-house software factory of a financial institution in Ecuador, which belongs to the group of large banks in Ecuador.

The financial institution has a wide portfolio of applications whose objective is to provide quality financial services and products and allow its clients to manage their products and services through the different applications or technological platforms available so that the client does not have the need to go in person to banking agencies. To meet these objectives, it is necessary for the organization to have up-to-date and flexible technological tools, as well as to rely on reference frameworks that guide its actions when carrying out specific activities that lead it to meet the objectives.

Within the daily operation of the financial institution, problems related to quality, software delivery times, as well as the lack of adaptation to changes involving new developments, have been found, progressively delaying the delivery of value to customers.

After carrying out an analysis of the current situation, needs and problems, initiatives are proposed to solve the problems described and exploit technological and functional opportunities that allow the delivery of value to the financial institution's clients. These definitions are supported and supported by the Togaf business architecture framework, in the same way the support of complementary frameworks such as Scrum and Devops practices leverage the ideal strategy to be a reference in the continuous, flexible, secure delivery of software products.

## ÍNDICE CONTENIDO

<b>1. Fase Preliminar</b> .....	<b>7</b>
1.1. Contexto .....	7
1.2. Organización impactada .....	8
1.2.1. Organigrama empresarial .....	8
1.2.2. Unidades impactadas .....	9
1.3. Involucrados .....	10
1.3.1. Expectativas de valor.....	10
1.3.2. Estrategias de Comunicación .....	11
1.4. Motivadores .....	12
1.4.1. Business Motivation Model (BMM) .....	12
1.5. Marcos de referencia complementarios .....	13
1.6. Gobierno de Arquitectura Empresarial .....	14
1.6.1. Equipo de Arquitectura .....	14
1.6.2. Estrategia de Gobierno .....	15
1.6.3. Catálogo de Principios.....	16
<b>2. Visión de arquitectura</b> .....	<b>18</b>
2.1. Requerimientos de alto nivel.....	18
2.2. Visionamiento y escenarios de solución .....	19
2.2.1. Empresa referente .....	19
2.2.2. Marcos de referencia .....	21
2.2.3. Visionamiento .....	23
2.3. Análisis de brechas.....	23
2.4. Definición de la arquitectura objetivo .....	25
2.4.1. Arquitectura de negocio objetivo.....	26
2.4.2. Arquitectura de aplicaciones objetivo .....	27
2.4.3. Arquitectura de datos objetivo .....	27

2.4.4.	Arquitectura de infraestructura objetivo .....	28
2.5.	Personalización de la metodología .....	29
<b>3.</b>	<b>Arquitectura de negocio .....</b>	<b>30</b>
3.1.	Arquitectura actual de negocio.....	30
3.1.1.	Estructura organizacional .....	30
3.1.2.	Personas .....	31
3.1.3.	Modelo de procesos .....	34
3.2.	Arquitectura de negocio objetivo.....	35
3.2.1.	Estructura organizacional .....	35
3.2.2.	Personas .....	37
3.2.2.1.	Roles secundarios .....	37
3.2.2.2.	Roles complementarios .....	38
3.2.2.3.	Responsabilidades .....	40
3.2.3.	Modelo de procesos .....	41
3.2.4.	Fases de Scrum.....	43
3.2.4.1.	Inicio .....	43
3.2.4.2.	Planificación y estimación.....	44
3.2.4.3.	Implementación .....	46
3.2.4.4.	Revisión y retrospectiva.....	47
3.2.4.5.	Lanzamiento .....	48
3.2.5.	Artefactos .....	49
3.2.5.1.	Product Backlog.....	49
3.2.5.2.	Sprint Backlog.....	49
3.2.5.3.	Incremento.....	50
3.2.6.	Eventos.....	50
3.2.6.1.	Sprint Planning .....	50

3.2.6.2.	Dalily Scrum.....	51
3.2.6.3.	Sprint Review.....	51
3.2.6.4.	Sprint Retrospective .....	52
3.2.6.5.	Refinamiento del Sprint .....	52
3.3.	Análisis de brechas.....	52
3.4.	Iniciativas para cerrar brechas.....	55
<b>4.</b>	<b>Arquitectura de aplicaciones y datos .....</b>	<b>57</b>
4.1.	Arquitectura de Aplicaciones y datos actual.....	57
4.1.1.	Catálogo de aplicaciones.....	57
4.1.2.	Análisis de ciclo de vida de aplicaciones .....	59
4.2.	Arquitectura de Aplicaciones objetivo .....	61
4.2.1.	Aplicaciones por procesos.....	63
4.3.	Análisis de brechas.....	64
4.4.	Iniciativas para cerrar brechas.....	66
<b>5.</b>	<b>Arquitectura de infraestructura base .....</b>	<b>68</b>
5.1.	Arquitectura de Infraestructura actual .....	68
5.2.	Arquitectura de Infraestructura objetivo .....	69
5.2.1.	Gobierno de Suscripciones.....	70
5.2.2.	Grupos de seguridad .....	71
5.3.	Análisis de Brechas .....	72
5.4.	Iniciativas para cerrar brechas.....	73
<b>6.</b>	<b>Oportunidades y soluciones .....</b>	<b>74</b>
<b>7.</b>	<b>Plan de migración.....</b>	<b>80</b>
7.1.	Análisis de impacto.....	80
7.2.	Análisis de esfuerzo.....	81
7.3.	FASES.....	81
7.4.	Roadmap.....	83



CONCLUSIONES.....	84
RECOMENDACIONES .....	85

## ÍNDICE DE TABLAS

Tabla 1 Departamentos impactados.....	9
Tabla 2 Involucrados y expectativas de valor.....	11
Tabla 3 Estrategia de comunicación .....	12
Tabla 4 Marcos de referencia.....	13
Tabla 5 Arquitectura actual vs arquitectura referente.....	24
Tabla 6 Roles actuales de la fábrica de software .....	31
Tabla 7 Procesos de la fábrica de software .....	34
Tabla 8 Responsabilidades de roles en Scrum .....	40
Tabla 9 Procesos objetivos y sus descripciones de la fábrica de software .....	41
Tabla 10 Análisis de brechas de la arquitectura de negocio actual y negocio .	53
Tabla 11 Brechas identificadas en la arquitectura de negocio .....	54
Tabla 12 Aplicaciones de la fábrica de software .....	57
Tabla 13 Aplicaciones de la arquitectura objetivo .....	62
Tabla 14 Análisis de brechas de la arquitectura de aplicaciones actual y objetivo.....	64
Tabla 15 Brechas identificadas en arquitectura de aplicaciones y datos .....	65
Tabla 16 Características de equipos de la fábrica de software .....	68
Tabla 17 Análisis de brechas de la arquitectura de infraestructura base .....	72
Tabla 18 Iniciativas por tipo de arquitectura.....	74

## ÍNDICE DE ILUSTRACIONES

Ilustración 1 Organigrama de la Unidad de Servicios Tecnológicos.....	8
Ilustración 2 Modelo de motivacional de negocio .....	13
Ilustración 3 Equipo de arquitectura. ....	15
Ilustración 4 Matriz de Roles y Responsabilidades .....	15
Ilustración 5 Procesos del marco de habilidades para la era de la información	21
Ilustración 6 Procesos del marco Scrum .....	22
Ilustración 7 Procesos Devops .....	22
Ilustración 8 Escala de un modelo de madurez de capacidades.....	23
Ilustración 9 Análisis de brechas arquitectura actual vs arquitectura referente	24
Ilustración 10 Análisis de brechas arquitectura actual vs arquitectura objetivo	25
Ilustración 11 Mapa de ruta, implementación arquitectura .....	29
Ilustración 12 Organigrama del Área de Investigación y Desarrollo .....	31
Ilustración 13 Organigrama objetivo de la Unidad de Servicios Tecnológicos .	35
Ilustración 14 Roles y áreas que participan en el proceso de Desarrollo de Aplicaciones .....	36
Ilustración 15 Descripción general de roles de Scrum .....	37
Ilustración 16 Flujo de scrum.....	43
Ilustración 17 Subprocesos de la fase de inicio de Scrum .....	43
Ilustración 18 Subprocesos de la fase de planificación y estimación de Scrum	45
Ilustración 19 Subprocesos de la fase de implementación de Scrum .....	46
Ilustración 20 Subprocesos de la fase de revisión y retrospectiva de Scrum...	47
Ilustración 21 Subprocesos de la fase de lanzamiento de Scrum .....	48
Ilustración 22 Brechas identificadas en la arquitectura de negocio actual y la objetivo.....	53
Ilustración 23 Análisis de ciclo de vida de aplicaciones. ....	60
Ilustración 24 TIME de aplicaciones.....	61
Ilustración 25 Relación aplicaciones por procesos.....	63
Ilustración 26 Brechas identificadas en la arquitectura de aplicaciones actual y la objetivo. ....	65
Ilustración 27 Suscripciones existentes y gobierno de recursos .....	70

Ilustración 28 Usuarios, grupos, identidades sincronizadas con Azure AD Connect.....	71
Ilustración 29 Brechas identificadas en la arquitectura de infraestructura base	73
Ilustración 30 Análisis de impacto (Iniciativas vs Objetivos).....	80
Ilustración 31 Análisis de esfuerzo (Iniciativas vs Criterios de esfuerzo) .....	81
Ilustración 32 Análisis de esfuerzo (Iniciativas vs Criterios de esfuerzo) .....	82
Ilustración 33 Hoja de ruta de iniciativas priorizadas.....	83

# 1. Fase Preliminar

## 1.1. Contexto

El proyecto Capstone se circunscribe a una institución financiera del Ecuador, que cuenta con 4 subsidiarias y 80 agencias a nivel nacional. En 1978 inicio operaciones atendiendo al sector corporativo; sin embargo, a partir de la primera década amplió los segmentos de negocio y la cobertura geográfica en Ecuador y a nivel internacional. En marzo de 2014, un Grupo Financiero Centroamericano, adquirió un porcentaje mayoritario de acciones de la Institución Financiera Ecuatoriana y se inició el proceso de consolidación en octubre de 2014.

En la actualidad, la calificación de riesgo de la entidad es AAA- y corresponde a una de las más altas del sistema financiero nacional emitida por dos Calificadoras de Riesgo: BankWatch Ratings y PCR-Pacific Credit Rating.

La unidad de Servicios Tecnológicos cuenta con 170 colaboradores distribuidos en los departamentos de Producción, Infraestructura, Investigación y Desarrollo y Planificación y Control.

El área de Investigación y Desarrollo da soporte a varias líneas de negocio relacionadas al mantenimiento de aplicaciones, implementaciones de productos y soluciones tecnológicos nuevos desarrollados In House o a través de proveedores; estas actividades de desarrollo, implementación y liberación se realizan utilizando como guía la metodología ágil Scrum y la Gestión de proyectos basadas en el PMI (Project Management Institute) para los proyectos con interacción con proveedores.

Todos los servicios mencionados en el párrafo anterior deben ser mantenidos, mejorados, así como el área de Investigación y Desarrollo debe adoptar nuevas tecnologías para apalancar los objetivos estratégicos de la institución.

La gran cantidad de servicios disponibles y el cambiante entorno actual del país y del mundo impulsa a liberar nuevas soluciones de manera ágil pensando siempre en el beneficio de los clientes sin dejar de lado la seguridad, continuidad y calidad. Actualmente la institución financiera no libera productos de software con la velocidad y calidad que se requiere para que los clientes prefieran esta institución sobre las otras, lo que provoca que pierda ventaja competitiva en el sector financiero que incluye a los bancos y a las cooperativas de ahorro y crédito.

## 1.2. Organización impactada

### 1.2.1. Organigrama empresarial

En la ilustración 1 se muestra el organigrama de la unidad de Servicios Tecnológicos de la institución financiera.

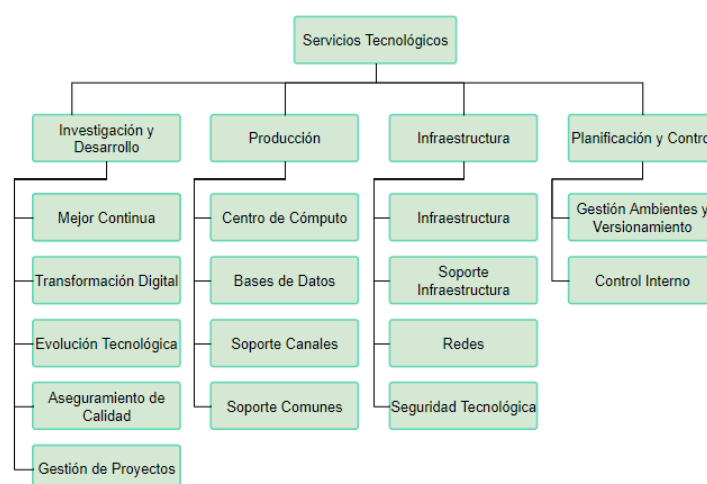


Ilustración 1 Organigrama de la Unidad de Servicios Tecnológicos

Fuente: El Autor

Como se puede evidenciar en la ilustración 1, la estructura de la unidad de Servicios Tecnológicos es jerárquica y funcional, donde se distinguen los distintos niveles de cada departamento.

### 1.2.2. Unidades impactadas

A continuación, se detallan los departamentos que serán afectados por la realización por el proyecto, estos tendrán un nivel de impacto de acuerdo a la nueva propuesta para el fortalecimiento de la fábrica de software In House y la relación que tendrán con la misma.

Tabla 1

Departamentos impactados

Unidad Organizacional	Impacto	Descripción
Canales Electrónicos	ALTO	Reducción de tiempos de entrega de productos de software, mejorar la calidad del software desarrollado, fortalecer cultura ágil.
Evolución Tecnológica	ALTO	Reducción de tiempos de entrega de productos de software, mejorar la calidad del software desarrollado, fortalecer cultura ágil.
Mejora Continua	ALTO	Reducción de tiempos de entrega de productos de software, mejorar la calidad del software desarrollado, fortalecer cultura ágil.
Aseguramiento de Calidad	ALTO	Implementar buenas prácticas de aseguramiento de calidad. Fortalecer cultura ágil.
Gestión de Proyectos	BAJO	Proyectos con dependencia de desarrollo interno entregados a tiempo.
Seguridad Tecnológica	BAJO	Disminuir vulnerabilidades en el desarrollo de aplicaciones. Mejorar controles de código seguro, optimización de prácticas de aplicaciones seguras.

Producción	MEDIO	Reducción de tiempo en la resolución de incidencias. Mejorar la calidad de software. Fortalecer cultura ágil.
Bases de Datos	ALTO	Reducir configuraciones manuales en el despliegue de aplicaciones. Gestión de configuraciones. Optimizar los procesos de monitorización de aplicaciones.
Gestión Ambientes y Versionamiento	ALTO	Mejorar procesos de integración continua, despliegue continuo, gestión de configuración.
Productividad	MEDIO	Mejorar especificaciones de historias de usuario, fortalecer cultura ágil. Reducir riesgos en los proyectos.
Seguridad de la Información	BAJO	Mejorar controles de código seguro, optimización de prácticas de aplicaciones seguras

### 1.3. Involucrados

#### 1.3.1. Expectativas de valor

En la tabla 2 se muestra la calificación de comprensión del proyecto fortalecimiento de la fábrica de desarrollo de la institución financiera y compromiso actual y esperada de los líderes de cada área afectada, la calificación es del 1 al 5, en donde 1 es la calificación más baja y 5 la calificación más alta.

Tabla 2

## Involucrados y expectativas de valor

UNIDAD	CARGO	COMPROMISIÓN ACTUAL	COMPROMISIÓN REQUERIDA	COMPROMISO ACTUAL	COMPROMISO REQUERIDO	APOYO REQUERIDO
Investigación y Desarrollo	Gerencia de Investigación y Desarrollo	2	4	3	5	3
Mejora Continua	Jefe de Mejora Continua	2	4	3	4	4
Evolución Tecnológica	Jefe de Evolución Tecnológica	3	4	3	4	4
Canales Electrónicos	Jefe de Canales Electrónicos	3	4	3	4	4
Aseguramiento de Calidad	Jefe de Aseguramiento de Calidad	3	4	3	4	4
Gestión de Proyectos	Jefe de Gestión de Proyectos	3	3	2	3	3
Gestión de Ambientes y Versionamiento	Jefe de Gestión de Ambientes y Versionamiento	3	4	3	4	4
Soporte Canales	Jefe de Soporte Canales	2	4	2	3	3
Soporte Comunes	Jefe de Soporte Comunes	2	4	2	3	3
Bases de Datos	Jefe de Bases de Datos	2	4	2	4	4
Gestión de Proyectos	Coordinador de Proyectos	2	3	2	3	3
Mejora Continua / Evolución Tecnológica / Canales Electrónicos	Coordinadores de Investigación y Desarrollo	3	4	2	4	4
Mejora Continua / Evolución Tecnológica / Canales Electrónicos	Especialistas de Investigación y Desarrollo	3	4	2	4	4
Mejora Continua / Evolución Tecnológica / Canales Electrónicos	Desarrolladores de Investigación y Desarrollo	2	3	2	3	3
Mejora Continua / Evolución Tecnológica / Canales Electrónicos	Desarrolladores Outsourcing	2	3	2	3	3
Seguridad Tecnológica	Jefe de Seguridad Tecnológica	2	3	2	3	3
Seguridad Tecnológica	Especialista Seguridad Tecnológica	2	3	2	3	3
Seguridad de la Información	Especialista Seguridad de la Información	2	3	2	3	3
Gestión de Ambientes y Versionamiento	Especialistas Gestión de Ambientes y Versionamiento	3	4	3	4	4
Gestión de Ambientes y Versionamiento	Analistas Versionamiento Outsourcing	2	3	2	3	3
Gestión de Ambientes y Versionamiento	Analistas Ambientes Outsourcing	2	3	2	3	3
Aseguramiento de Calidad	Coordinador de Aseguramiento de Calidad	2	4	2	4	4
Aseguramiento de Calidad	Especialistas Aseguramiento de Calidad	2	4	2	4	4
Aseguramiento de Calidad	Analistas Outsourcing Aseguramiento de Calidad	2	3	2	3	4
Aseguramiento de Calidad	Analistas Funcionales Aseguramiento de Calidad	2	4	2	4	4
Transformación Digital	Coordinadores Canales Digitales	2	4	2	4	4
Productividad	Coordinadores Productividad	2	4	2	4	4

En la tabla 2 se muestran los distintos roles en cada departamento dentro de la unidad de Servicios Tecnológicos, cada rol fue evaluado con una calificación del 1 al 4, donde 1 representa el nivel más bajo y 5 el nivel más alto para las categorías comprensión actual y requerido, compromiso actual y requerido y finalmente apoyo requerido. Esta información se logró obtener con base a la realización de entrevistas, encuestas, grupos focales.

### 1.3.2. Estrategias de Comunicación

En la Tabla 3 se muestra la estrategia de comunicación en función de los interesados según su nivel poder e interés. La calificación es del 1 al 4, donde 1 representa el nivel más bajo y 4 el nivel más alto.



Tabla 3

## Estrategia de comunicación

UNIDAD	CARGO	PODER	INTERÉS	ESTRATEGIA
Investigación y Desarrollo	Gerencia de Investigación y Desarrollo	4	4	Interesado clave
Mejora Continua	Jefe de Mejora Continua	3	2	Mantener satisfecho
Evolución Tecnológica	Jefe de Evolución Tecnológica	3	2	Mantener satisfecho
Canales Electrónicos	Jefe de Canales Electrónicos	3	2	Mantener satisfecho
Aseguramiento de Calidad	Jefe de Aseguramiento de Calidad	3	2	Mantener satisfecho
Gestión de Proyectos	Jefe de Gestión de Proyectos	3	2	Mantener satisfecho
Gestión de Ambientes y Versionamiento	Jefe de Gestión de Ambientes y Versionamiento	3	4	Interesado clave
Soporte Canales	Jefe de Soporte Canales	3	2	Mantener satisfecho
Soporte Comunes	Jefe de Soporte Comunes	3	2	Mantener satisfecho
Bases de Datos	Jefe de Bases de Datos	3	2	Mantener satisfecho
Gestión de Proyectos	Coordinador de Proyectos	2	3	Monitorear
Mejora Continua / Evolución Tecnológica / Canales Electrónicos	Coordinadores de Investigación y Desarrollo	2	3	Mantener informado
Mejora Continua / Evolución Tecnológica / Canales Electrónicos	Especialistas de Investigación y Desarrollo	2	3	Mantener informado
Mejora Continua / Evolución Tecnológica / Canales Electrónicos	Desarrolladores de Investigación y Desarrollo	1	3	Mantener informado
Mejora Continua / Evolución Tecnológica / Canales Electrónicos	Desarrolladores Outsourcing	1	3	Mantener informado
Seguridad Tecnológica	Jefe de Seguridad Tecnológica	3	2	Mantener satisfecho
Seguridad Tecnológica	Especialista Seguridad Tecnológica	2	2	Monitorear
Seguridad de la Información	Especialista Seguridad de la Información	2	2	Monitorear
Gestión de Ambientes y Versionamiento	Especialistas Gestión de Ambientes y Versionamiento	3	3	Interesado clave
Gestión de Ambientes y Versionamiento	Analistas Versionamiento Outsourcing	1	3	Mantener informado
Gestión de Ambientes y Versionamiento	Analistas Ambientes Outsourcing	1	3	Mantener informado
Aseguramiento de Calidad	Coordinador de Aseguramiento de Calidad	2	4	Mantener informado
Aseguramiento de Calidad	Especialistas Aseguramiento de Calidad	2	4	Mantener informado
Aseguramiento de Calidad	Analistas Outsourcing Aseguramiento de Calidad	1	2	Monitorear
Aseguramiento de Calidad	Analistas Funcionales Aseguramiento de Calidad	1	2	Monitorear
Transformación Digital	Coordinadores Canales Digitales	2	4	Mantener informado
Productividad	Coordinadores Productividad	2	4	Mantener informado

En la tabla 3 se muestra la forma en que se trata a cada interesado categorizándolo de acuerdo a su poder e interés de la siguiente manera: los interesados que tengan alto poder (3 y 4) y alto interés (3 y 4) serán involucrados clave, los interesados que tengan alto poder (3 y 4) y bajo interés (1 y 2) se los debe mantener satisfechos, los involucrados que tengan bajo poder (1 y 2) y alto interés (3 y 4) se los debe mantener informados, finalmente los involucrados que tengan bajo poder (1 y 2) y bajo interés (1 y 2) se los debe monitorear.

## 1.4. Motivadores

### 1.4.1. Business Motivation Model (BMM)

Para esquematizar los motivadores para realizar el proyecto de fortalecimiento de la fábrica de software de la institución financiera las iniciativas de mejora, en la ilustración 2 se muestra un diagrama BMM.

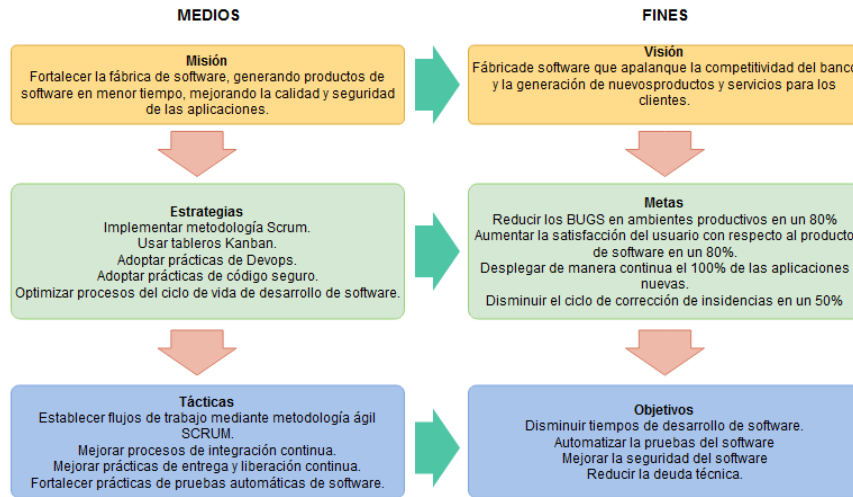


Ilustración 2 Modelo de motivacional de negocio

Fuente: El Autor

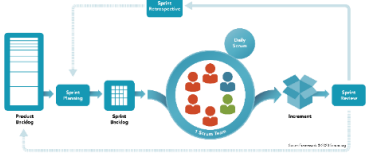

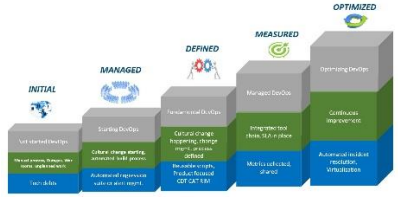
### 1.5. Marcos de referencia complementarios

Para la solución del problema empresarial complejo (concern) se utilizará los siguientes marcos de referencia para apoyar las iniciativas de mejora. A continuación, se describen en la tabla 4:

Tabla 4

Marcos de referencia

Área	Referente	Descripción
Arquitectura Empresarial	<p>Togaf 9.2</p>	<p>Es un marco de trabajo para Arquitecturas Empresariales desarrollado por The Open Group, es una metodología y un marco de arquitectura empresarial comprobados que utilizan las organizaciones líderes en el mundo para mejorar la eficiencia empresarial.</p>

<p><b>Gestión ágil de proyectos</b></p>	<p>Scrum</p> 	<p>Scrum es un marco de trabajo ágil y estructurado en el que enfoque principal es la entrega de valor en periodos cortos de tiempo.</p>
<p><b>Habilidades y competencias</b></p>	<p>SFIA 8</p> 	<p>Es un marco de trabajo que describe habilidades y competencias requeridas por el personal relacionado e inmerso en tecnologías de la información.</p>
<p><b>Devops</b></p>	<p>Devops Maturity Model</p> 	<p>Modelo de madurez de Devops creado por Salesforce, permite tener una calificación que representa el nivel de madurez de las prácticas de Devops que se usan en la organización. («DevOps Maturity- Steps To Help You Level Up in Salesforce — Plumlogix», 2020)</p>

## 1.6. Gobierno de Arquitectura Empresarial

### 1.6.1. Equipo de Arquitectura

Para la realización del proyecto se cuenta con el patrocinio de la Gerencia del área de Investigación y Desarrollo, la cual estará al tanto y participará activamente en la aplicación de marco de arquitectura empresarial.

El equipo de Arquitectura está formado por un arquitecto líder, un arquitecto de negocio, un arquitecto de aplicaciones, un arquitecto de datos y un arquitecto de infraestructura que se muestran en la ilustración 3.

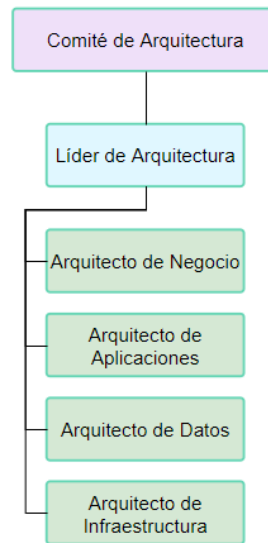


Ilustración 3 Equipo de arquitectura.

Fuente: El Autor

### 1.6.2. Estrategia de Gobierno

Para la selección de la estrategia de gobierno se utilizó como marco de referencia COBIT 2019, específicamente la herramienta de factores de diseño fundamentada en gestión de proyectos ágiles y Devops; el resultado de esta herramienta se muestra en la ilustración 4.

**RACI Matrix**

Responsabilities	Status	ROLES					
		Comité de Arquitectura	Arquitecto Líder	Arquitecto de Negocio	Arquitecto de Aplicaciones	Arquitecto de Datos	Arquitecto de Infraestructura
Gestionar la arquitectura empresarial		A	R	R	R	R	R
Gestionar los recursos humanos		I	I	R	R	R	R
Gestionar riesgos		I	A	R	R	R	R
Gestionar los programas		I	C	A	R	R	R
Gestionar la definición de requerimientos		I	C	A	R	R	R
Gestionar la identificación y construcción de soluciones		C	R	I	A	R	R
Gestionar los cambios organizativos		C	A	R	R	R	R
Gestionar los cambios de TI		A	I				A
Gestionar la aceptación y la transición de los cambios de TI		A	R	C	C	C	C
Gestionar la configuración		I	A	R	R	R	R
Gestionar los proyectos		I	A	R	R	C	C
Gestionar las operaciones		I	R	I	R	C	A
Gestionar las peticiones y los incidentes del servicio		I	I	A	C	C	I
Gestionar los problemas		I	I	A	C	C	I

Ilustración 4 Matriz de Roles y Responsabilidades

Fuente: El Autor

### 1.6.3. Catálogo de Principios

Los principios de arquitectura proporcionan un conjunto de pautas y lineamientos que le darán a la empresa y en este caso en particular a la institución financiera las pautas para dirigir tanto la toma de decisiones rápida como el análisis sólido en el camino hacia el éxito empresarial. («7 principios de arquitectura que todo arquitecto empresarial debe conocer», 2021)

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

<b>Código:</b>	<b>PRI01</b>
<b>Principio:</b>	Adopción marcos de referencia.
<b>Dominio</b>	Negocio
<b>Definición:</b>	Selección de marcos de referencia relacionados con métodos ágiles.
<b>Motivación:</b>	- Alinear la organización con metodologías ágiles.
<b>Implicaciones:</b>	- Mejora continua - Cambio cultural - Trabajo colaborativo

<b>Código:</b>	<b>PRI02</b>
<b>Principio:</b>	Arquitectura escalable, flexible y adaptable.
<b>Dominio</b>	Aplicaciones
<b>Definición:</b>	Las aplicaciones deben seguir buenas prácticas de escalabilidad, flexibilidad y adaptabilidad permitiendo que esta pueda evolucionar, integrarse ya adaptarse con otras aplicaciones.
<b>Motivación:</b>	- Aplicación fácil de mantener, mejorar, integrar y adaptar a otras aplicaciones.
<b>Implicaciones:</b>	- Uso de patrones de diseño. - Buenas prácticas de desarrollo. - Uso de tecnología actual.

<b>Código:</b>	<b>PRI03</b>
<b>Principio:</b>	Los datos son un activo institucional.
<b>Dominio</b>	Datos/Información
<b>Definición:</b>	La información es la base para el análisis del rendimiento de los equipos y despliegue de productos.
<b>Motivación:</b>	- Apoyo a la toma de decisiones.
<b>Implicaciones:</b>	- Recopilar información. - Analizar información. - Generar tableros interactivos.

<b>Código:</b>	<b>PRI04</b>
<b>Principio:</b>	Seguridad de aplicaciones
<b>Dominio</b>	Tecnología
<b>Definición:</b>	Las aplicaciones deben superar validación de código estático y dinámico.
<b>Motivación:</b>	- Aplicaciones seguras.
<b>Implicaciones:</b>	- Proceso de validación de aplicaciones. - Herramientas de análisis estático y dinámico de código.

<b>Código:</b>	<b>PRI05</b>
<b>Principio:</b>	Automatización de actividades
<b>Dominio</b>	Tecnología
<b>Definición:</b>	Automatización de tareas relacionadas con el ciclo de vida de la aplicación
<b>Motivación:</b>	- Aplicaciones robustas. - Seguimiento de versiones de las Aplicaciones.
<b>Implicaciones:</b>	- Procedimientos de control de versiones. - Pruebas continuas (unitarias, funcionales, carga). - Proceso de despliegue de aplicaciones. - Análisis de código.

## 2. Visión de arquitectura

En este capítulo define la fase inicial del de desarrollo de arquitectura (ADM), incluye información la definición del alcance, identificación de involucrados, creación de la visión de la arquitectura, definir la arquitectura actual, establecer la arquitectura deseada y objetivo, finalmente un análisis de brechas y un plan de trabajo en un alto nivel para identificar el tiempo que tomara la realización del proyecto fortalecimiento de la fábrica de software de una institución financiera. (*Phase A: Architecture Vision*, s. f.)

### 2.1. Requerimientos de alto nivel

De acuerdo con los objetivos y expectativas establecidos en el proyecto de fortalecimiento de la fábrica de software se determina el alcance y requerimientos de alto nivel.

- Fortalecer las capacidades técnicas de los distintos roles dentro de la fábrica de software.
- Fortalecer capacidades de trabajo colaborativo y gestión ágil de proyectos en el equipo de desarrollo.
- Fortalecer los procesos y procedimientos de gestión de requisitos con estándares ágiles.
- Definir políticas, procesos y procedimientos en la gestión de calidad de software.
- Implementar prácticas y herramientas de automatización de pruebas automáticas.
- La fábrica de software requiere estándares de arquitectura que facilite la interoperabilidad de las aplicaciones.

- Integrar herramientas de gestión de versión entre los ambientes de pruebas y producción.
- Reducir tiempos de despliegue de productos de software a través de prácticas de Devops.
- Optimizar tiempos de corrección de incidentes en ambientes productivos.

## **2.2. Visionamiento y escenarios de solución**

Para el análisis de visionamiento se tomó como referencia el caso de éxito de Banco BBVA en el uso de metodologías ágiles y prácticas de Devops.

### **2.2.1. Empresa referente**

La estrategia de desarrollo digital por la que BBVA apostó hace ya varios años está contribuyendo a facilitar la gestión digital de las finanzas de los clientes. La implementación de las metodologías de trabajo ágiles, las prácticas DevOps - coordinación de diferentes equipos durante el desarrollo del 'software'- y una perspectiva global, han sido clave para que las soluciones digitales aporten valor y sean escalables. (BBVA IT | DevOps, la metodología que está revolucionando los procesos de desarrollo, s. f.)

Banco BBVA obtuvo los siguientes beneficios aplicando la metodología ágil SCRUM:

- Reducción de tiempos de lanzamientos de soluciones.
- Ofrece servicios de calidad a sus clientes.
- Transversalidad
- Trabajo colaborativo.



- Mejorar la calidad del software.
- Reducción de tiempo en validación y publicación de nuevas versiones.

De igual manera implementó prácticas de Devops que le permitieron mejorar y optimizar las siguientes áreas:

- Trabajo colaborativo
- Automatización del trabajo
- Cambio de cultura
- Pruebas automáticas
- Procesos optimizados de construcción y despliegue
- Implementar herramientas Devops
  - Administrar y compilar código
  - Verificar calidad
  - Pruebas unitarias
  - Actualizar entornos de trabajo
  - Notificar versiones liberadas.
  - Uso de plataformas en nube.
  - Reducción de errores
- Reducción de tareas manuales
- Proceso transparente

Actualmente, el equipo DevOps de los canales digitales de BBVA España en la web y la 'app' de BBVA está ejecutando más de 17.000 horas de 'testing' automático al mes con cerca de 3,5 millones de pruebas. Además, se están integrando más de 200 cambios de código en cada versión de las aplicaciones, lo que permite al banco seguir avanzando en su transformación. (BBVA, 2020)

### 2.2.2. Marcos de referencia

Para evaluar el nivel de madurez de la fábrica de software se utilizará el marco SBOK 3.0 para los procesos y metodología ágil Scrum, SFIA para la dimensión personas, y Devops Maturity Model para las prácticas Devops.

Para evaluar el nivel de madurez de la fábrica de software se utilizarán los siguientes marcos con el objetivo de fortalecer las capacidades de la fábrica de desarrollo en sus distintos dominios.

En la gestión de habilidades y competencias del personal de la fábrica de software se utilizará SFIA 8 (Skill Framework for the Information Age). SFIA describe habilidades y competencias requeridas por los profesionales en roles implicados en tecnologías de información y comunicación, transformación digital e ingeniería de software. (*SFIA y la gestión de habilidades*, s. f.)

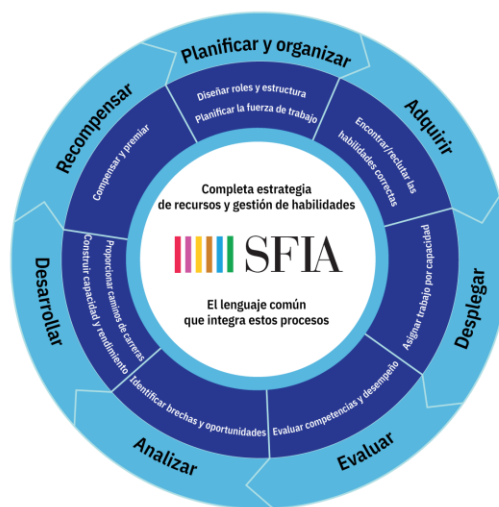


Ilustración 5 Procesos del marco de habilidades para la era de la información

Fuente: Tomado de (*SFIA y la gestión de habilidades*, s. f.)

Para la gestión de proyectos ágiles se utilizará como referencia el marco de referencia Scrum apoyado en la guía SBOK 3.0, en la siguiente ilustración se muestra el proceso de la metodología Scrum en el que describe en un alto nivel el proceso de la metodología.

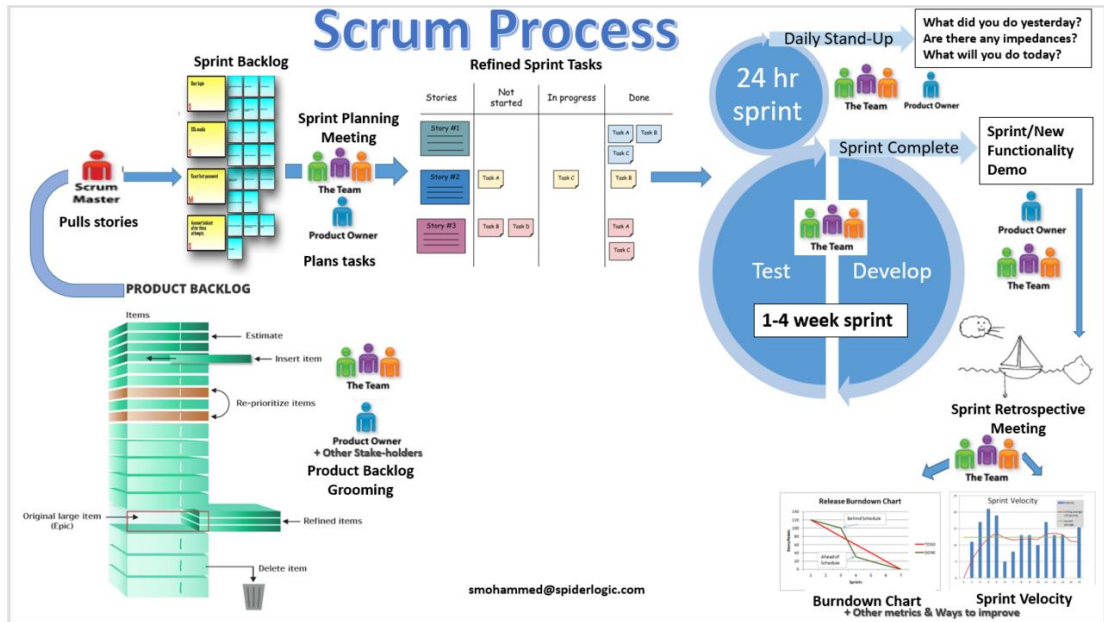


Ilustración 6 Procesos del marco Scrum

Fuente: Tomado de (Scrum Process Model as part of agile software development methodology | LinkedIn, s. f.)

En la siguiente ilustración se muestra el proceso de Devops en el cual se evidencian las etapas en las que un producto de software sigue desde que inicia el desarrollo hasta su liberación en producción y empieza a operar.

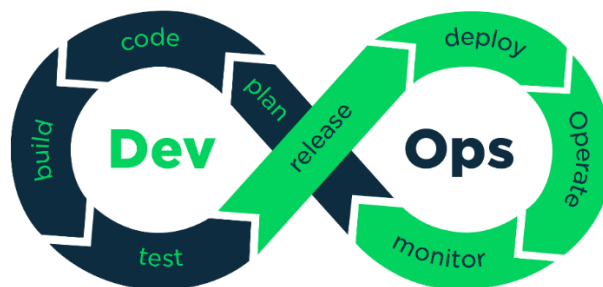


Ilustración 7 Procesos Devops

Fuente: Tomado de (DevOps is a culture, not a role!. Software is everywhere. In today's... | by Irma Harlann | Medium, s. f.)

### 2.2.3. Visionamiento

En relación al análisis del caso de éxito, se plantea que la arquitectura referente basados en los modelos de madurez en las prácticas relacionadas a Gobierno y Gestión, personas, metodología y Devops estén en nivel 5, el cual nos indica que en cada dominio la institución se encuentra en el nivel más alto del modelo de madurez mostrando que sus procesos se encuentran optimizados, lo cual quiere decir que son referentes en su sector, mejoran continuamente, aplican mejores prácticas de la industria, tienen métricas de rendimiento.

## 2.3. Análisis de brechas

Tomando como base la arquitectura de referencia y los modelos de madurez de cada marco de referencia a utilizar en el ejercicio de arquitectura planteada, se compara la arquitectura actual y la arquitectura referente con el objetivo de encontrar las brechas existentes en cada dominio.

En la siguiente ilustración se muestra el modelo de madurez de capacidades aplicado a cada uno de los dominios explicados en la sección anterior, donde 1 equivale a la calificación más baja (Inicial) y 5 la más alta (Optimizada).



Ilustración 8 Escala de un modelo de madurez de capacidades

Fuente: Tomado de (The benefits of using CMM (Capability Maturity Model) as a prelude for Process Improvement transformation | LinkedIn, s. f.)

Para obtener el nivel de madurez de cada una de las capacidades evaluadas, se han realizado encuestas y entrevistas donde el resultado de estas técnicas de obtención de información se muestra en la siguiente tabla.

Tabla 5  
Arquitectura actual vs arquitectura referente

Componentes	Actual	Referente
Gobierno y gestión ágil - Negocio	2.1	5.0
Personas - Negocio	2.1	5.0
Gestión ágil de proyectos - Negocio	2.3	5.0
Gestión ágil de proyectos - Datos	2.0	5.0
Prácticas devops - Infraestructura/Tecnología	2.2	5.0
Prácticas devops - Datos	3.0	5.0
Infraestructura base	3.0	4.0
Cumplimiento General	2.4	4.9

A continuación, se muestra en la siguiente ilustración un gráfico de radar en el que se evidencia las brechas en cada dominio de valoración.

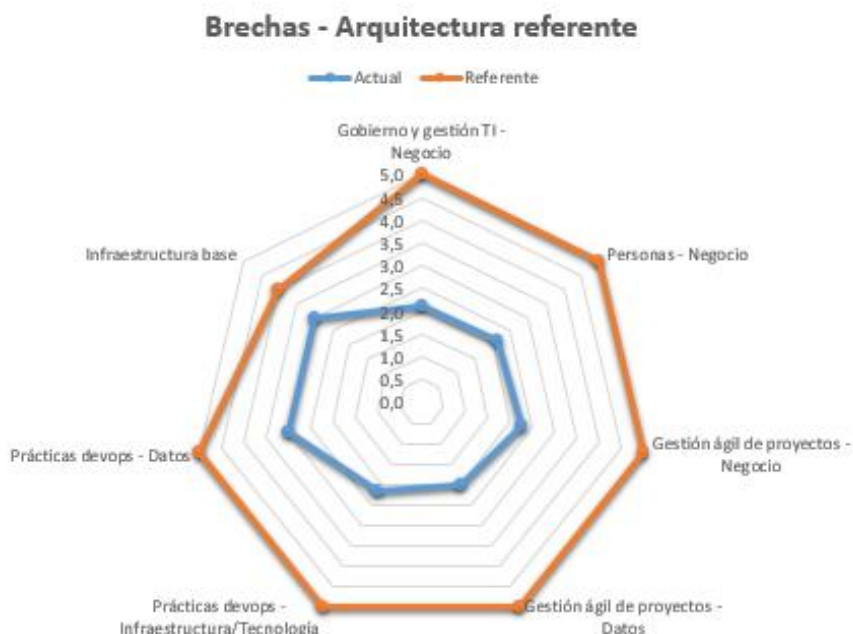


Ilustración 9 Análisis de brechas arquitectura actual vs arquitectura referente

Fuente: El Autor

De acuerdo a la tabla 5 que muestra el resultado de las encuestas y entrevistas realizadas para determinar el nivel de madurez actual en los distintos dominios de valoración, si bien cuenta con ciertos dominios en nivel 3 (Definido), muestra un promedio de valoración de 2,4 lo cual indica que el proceso se encuentra en el nivel 2 (Manejado).

## 2.4. Definición de la arquitectura objetivo

Una vez determinado el nivel de madurez actual de los componentes evaluados en la fábrica de software los cuales corresponden en promedio a “Gestionado cuantitativamente”, se define la arquitectura objetivo que sea viable en el cumplimiento de objetivos de alto nivel y el *concern* definido. Por esta razón, el nivel de madurez que la fábrica de software buscará será el nivel 4, es decir “manejado” lo que quiere decir que los distintos componentes mencionados en ilustración 12 son planificados, optimizados, medibles y controlados.

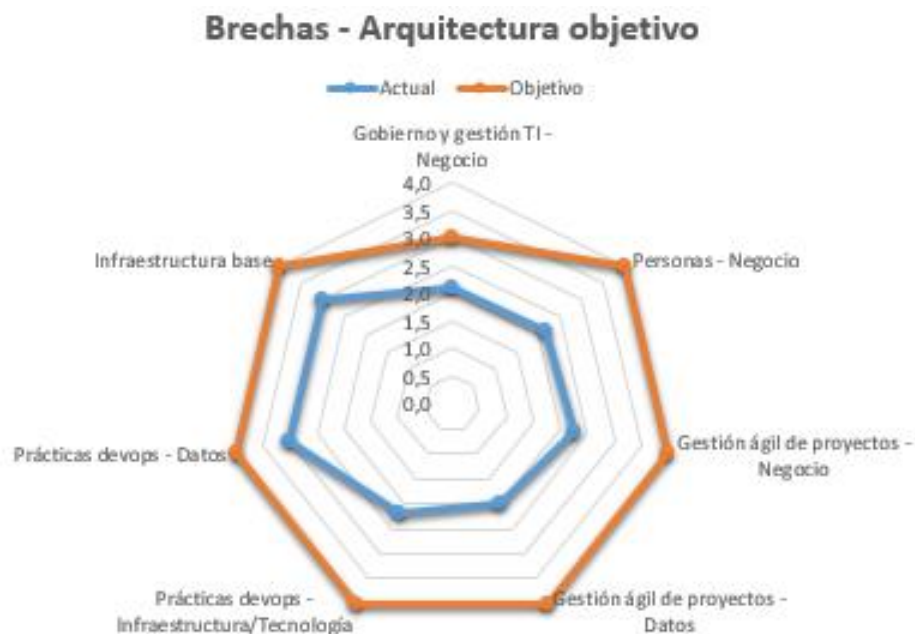


Ilustración 10 Análisis de brechas arquitectura actual vs arquitectura objetivo

Fuente: El Autor

### 2.4.1. Arquitectura de negocio objetivo

Realizando el análisis de los datos obtenidos a los distintos involucrados en base a los componentes de personas y procesos en la gestión ágil de proyectos que son referentes del dominio de negocio se evidencia que la arquitectura de negocio actual en promedio es de 2,2. Sin embargo, para lograr cumplir con la solución de los requerimientos de alto nivel y el *concern* la arquitectura de negocio para el componente de procesos de la gestión ágil de proyectos debe ser de nivel 3 y el componente personas en nivel 4. Para cumplir con la arquitectura de negocio objetivo se deberá cumplir con las siguientes características:

#### ■ Procesos gestión ágil de proyectos

- Inicio
- Planificación y estimación
- Implementación
- Revisión y retrospectiva
- Lanzamiento

#### ■ Personas

- Planificar y organizar
- Adquirir
- Desplegar
- Evaluar
- Analizar
- Desarrollar
- Recompensar

### **2.4.2. Arquitectura de aplicaciones objetivo**

Realizando el análisis de los datos obtenidos a los distintos involucrados en base al componente de “Prácticas de Devops” que es referente en el dominio de aplicaciones se evidencia que la arquitectura actual es de 2,2. Sin embargo, para lograr cumplir con la solución de los requerimientos de alto nivel y el *concern* la arquitectura de aplicaciones debe ser de nivel 4. Para cumplir con la arquitectura de aplicaciones objetivo se deberá cumplir con las siguientes características:

- Gestión de la configuración.
- Integración continua.
- Pruebas automatizadas.
- Infraestructura como código.
- Entrega continua.
- Despliegue continuo.

### **2.4.3. Arquitectura de datos objetivo**

Realizando el análisis de los datos obtenidos a los distintos involucrados en base a los componentes de “gestión ágil de proyectos” y “prácticas de Devops” que son referentes del dominio de datos se evidencia que la arquitectura actual es de 2,0 y 3,0 respectivamente. Sin embargo, para lograr cumplir con la solución de los requerimientos de alto nivel y el *concern* la arquitectura de datos debe ser de nivel 4.



Para cumplir con la arquitectura de datos objetivo se deberá cumplir con las siguientes características:

- Gestión ágil de proyectos

- Métricas

- Prácticas de Devops

- Monitoreo continuo

#### **2.4.4. Arquitectura de infraestructura objetivo**

Realizando el análisis de los datos obtenidos a los distintos involucrados en base al componente de “Infraestructura base” que son referentes del dominio de infraestructura se evidencia que la arquitectura actual es de 2,6 respectivamente. Sin embargo, para lograr cumplir con la solución de los requerimientos de alto nivel y el *concern* la arquitectura de datos debe ser de nivel 4. Para cumplir con la arquitectura de infraestructura objetivo se deberá cumplir con las siguientes características:

- Infraestructura híper convergente optimizada.

- Infraestructura como código.

- Uso de contenedores

- Auto gestión de componentes tecnológicos

- Red
- Almacenamiento
- Memoria
- Procesador
- Dominio
- Sistemas operativos
- Software base

## 2.5. Personalización de la metodología

De acuerdo con lo indicado en la sección anterior, se utilizarán los conceptos y buenas prácticas de acuerdo a los estándares de SFIA 8, SBOK 3.0 y Devops Maturity Model para el desarrollo de la arquitectura objetivo. Siguiendo el método ADM del marco de referencia Togaf se plantea un mapa de ruta en el que se muestra en un alto nivel la estimación del desarrollo de las siguientes fases y su duración en una línea de tiempo (ilustración 13):

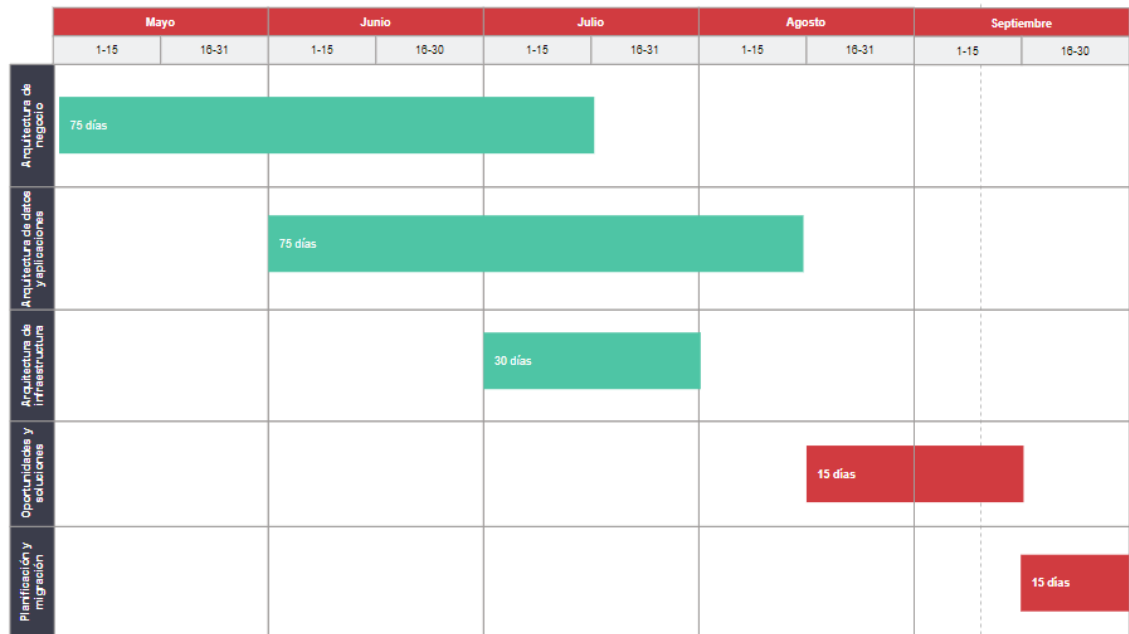


Ilustración 11 Mapa de ruta, implementación arquitectura

Fuente: El Autor

### **3. Arquitectura de negocio**

El presente capítulo describe y formula la arquitectura de negocio en términos de la estructura organizacional, personas y procesos claves de la fábrica de desarrollo de software, considerando la arquitectura de negocio actual y objetivo.

#### **3.1. Arquitectura actual de negocio**

La arquitectura actual de negocio de la institución financiera utiliza en gran medida la metodología Scrum, los distintos equipos cuentan con los roles que recomienda el marco, sin embargo, los procesos que son parte del ciclo de vida de desarrollo no están definidos correctamente, así como los perfiles de las personas y la estructura funcional que no ayuda a la generación de valor hacia la organización.

##### **3.1.1. Estructura organizacional**

A continuación, se muestra el organigrama del área de Investigación y Desarrollo que se desglosa en base a la Ilustración 2, en ambos organigramas se puede apreciar la estructura funcional y la distribución del personal de acuerdo a cada área.

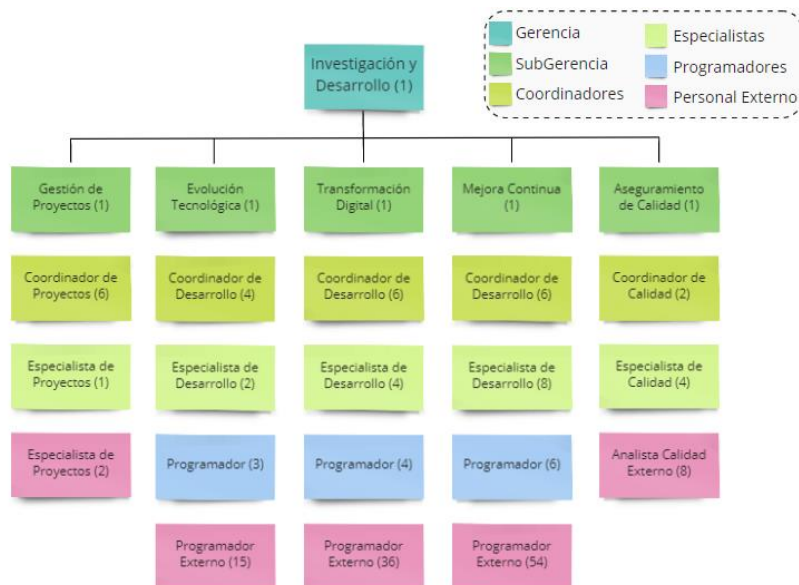


Ilustración 12 Organigrama del Área de Investigación y Desarrollo

Fuente: El Autor

Se puede evidenciar que, a pesar de estar trabajando con metodologías ágiles, la estructura organizacional es funcional y jerárquica lo que no permite que los equipos trabajen ágilmente.

### 3.1.2. Personas

A continuación, se describen las actividades y responsabilidades de los roles que actualmente existen en la fábrica de desarrollo de la institución financiera.

Tabla 6

Roles actuales de la fábrica de software

Rol	Descripción
Coordinador de proyectos Especialista de proyectos	Asume toda la responsabilidad de la definición, documentación y finalización satisfactoria de complejos proyectos (generalmente con un impacto comercial, político o de alto perfil significativo y de dependencias de alto riesgo). Adopta y adapta métodos y herramientas de gestión de proyectos, seleccionando

	<p>adecuadamente desde enfoques basados en planes / predictivos o enfoques más adaptativos (iterativos y ágiles). Asegura que el control de proyectos, el control de cambios, la gestión de riesgos y los procesos de prueba efectivos son mantenidos. Supervisa y controla los recursos, los ingresos y los costos de capital contra el presupuesto del proyecto y gestiona las expectativas de todos los interesados en el proyecto.</p>
<p>Coordinador de desarrollo Especialista de Desarrollo</p>	<p>Desarrolla políticas, estándares y pautas organizacionales para la construcción de software y refactorización. Planifica y lidera actividades de construcción de software para proyectos estratégicos, grandes y proyectos complejos de desarrollo. Desarrolla nuevos métodos y capacidades organizativas e impulsa la adopción y adherencia a las políticas y estándares.</p>
<p>Desarrollador</p>	<p>Diseña, codifica, verifica, prueba, documenta, enmienda y refactoriza programas de complejidad. Aplica estándares y herramientas acordados para lograr un resultado bien diseñado. Colabora en revisiones del trabajo con otros según corresponda.</p>
<p>Coordinador de Calidad Especialista de Calidad</p>	<p>Lidera, desarrolla y es responsable de un enfoque organizacional y un compromiso con la garantía de calidad. Garantiza que los procesos y actividades de aseguramiento de la calidad sean sólidos y se basen en mejores prácticas de la industria. Considera las implicaciones de la tecnología, enfoques, tendencias y regulaciones emergentes y legislación. Planifica y asigna recursos a las actividades de aseguramiento de la calidad organizacional. Monitorea y informa sobre las actividades de aseguramiento de la calidad, los niveles de cumplimiento y tanto organizativos como del proyecto riesgos. Revisa y analiza los resultados de las actividades de auditoría e identifica oportunidades de mejora para la organización.</p>
<p>Analista de Calidad</p>	<p>Contribuye a la recopilación de pruebas y la realización de auditorías formales o revisiones de actividades, procesos, datos, productos o servicios. Examina los</p>

	registros en busca de evidencia de que las pruebas y otras actividades de control de calidad han tenido lugar y determina el cumplimiento de la organización directivas, normas y procedimientos. Identifica no conformidades, no conformidades y ocurrencias anormales.
Analista de versionamiento y ambientes	Lidera la evaluación, análisis, planificación y diseño de paquetes de lanzamiento, incluida la evaluación de riesgo. Actúa como enlace con los socios comerciales y de TI sobre la programación de lanzamientos y la comunicación del progreso. Realiza revisiones posteriores al lanzamiento. Asegura que se apliquen los procesos y procedimientos de liberación y que las liberaciones se pueden revertir según sea necesario. Identifica, evalúa y gestiona la adopción de herramientas, técnicas y procesos de lanzamiento e implementación apropiados (incluida la automatización).
Analista de Soporte	Redacta y mantiene procedimientos y documentación para soporte de aplicaciones. Administra la aplicación mejoras para mejorar el rendimiento empresarial. Asesora sobre seguridad de aplicaciones, licencias, actualizaciones, copias de seguridad y necesidades de recuperación ante desastres. Garantiza que se tramiten todas las solicitudes de apoyo de acuerdo con los estándares y procedimientos establecidos.
Product Owner	Gestiona aspectos del ciclo de vida del producto permitiendo que el producto satisfaga las necesidades de los clientes /usuarios y alcanzar objetivos financieros o de otro tipo. Actúa como propietario del producto para uno o más productos o servicios; prioriza los requisitos del producto y posee una cartera de productos. Analiza el mercado y / o investigación de usuarios, comentarios, opiniones de expertos y datos de uso para comprender las necesidades y oportunidades. Facilita la adopción de productos mediante la planificación del desarrollo de la garantía del producto,

	apoyar y evaluar campañas, y monitorear el desempeño del producto. Implementa pruebas de productos y lanzamientos de productos.
--	---

En la tabla 6 se muestran los roles de la fábrica de software en el que se puede evidenciar la duplicidad de funciones y las jerarquías existentes en el área de Investigación y Desarrollo.

### 3.1.3. Modelo de procesos

En la actualidad en la fábrica de software el modelo de procesos para el desarrollo de productos software se lo realiza a través de metodologías ágiles. En la siguiente tabla se muestran los macro procesos utilizados en el desarrollo de software.

Tabla 7

Procesos de la fábrica de software

Proceso	Descripción
<b>Planeación y gestión</b>	Proceso responsable de la planificación y administración de los requerimientos de software desde el inicio hasta el final del ciclo de vida del producto de software.
<b>Control de versiones</b>	Responsable de administrar y monitorear el proceso de cambios de código a lo largo del tiempo.
<b>Desarrollo de software</b>	Es un conjunto de subprocesos o etapas que pueden promover rápidamente el desarrollo, la prueba y la implementación de productos de software.
<b>Gestión de pruebas</b>	Es la verificación dinámica de que los programas generan el comportamiento esperado a partir de un conjunto limitado casos de prueba, seleccionados apropiadamente del dominio de ejecución generalmente ilimitado.
<b>Trazabilidad de requerimientos</b>	La trazabilidad y los requisitos son la capacidad de registrar la relación entre los requisitos dados por las partes interesadas, los requisitos del proyecto y la solución final implementada. (pmoinformatica.com, s. f.)

<b>Seguridad</b>	Diseñado para mantener la seguridad de los sistemas y aplicaciones de información. (GTI8.pdf, s. f.)
<b>Integración continua</b>	Integra los diferentes componentes del software en desarrollo. El servidor de integración continua genera retroalimentación sobre los resultados del proceso de construcción, que es enviado a miembros designados del proyecto. (Salamon et al., 2014)
<b>Despliegue de aplicaciones</b>	Estas actividades incluyen la entrega, montaje y gestión en un sitio de los recursos necesarios para utilizar una versión de un sistema de software. (Hernández Yeja & Porven Rubier, 2016)
<b>Gestión de la configuración</b>	Gestión de la configuración abarca las actividades administrativas relacionadas con la creación, mantenimiento, cambio controlado y control de calidad de los productos. ( <i>Gestión de la configuración - Praxis Framework</i> , s. f.)

En la tabla 7 se describen los procesos actuales de la fábrica de software, lo que permite crear una línea base para comparar con los procesos objetivo.

## 3.2. Arquitectura de negocio objetivo

### 3.2.1. Estructura organizacional

A continuación, se muestra en la Ilustración 15 el organigrama objetivo para la unidad de servicios tecnológicos, en el que se describen las áreas nuevas, modificadas, reubicadas y las actuales:

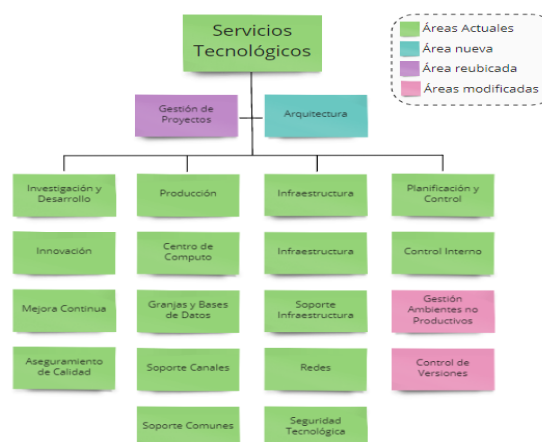


Ilustración 13 Organigrama objetivo de la Unidad de Servicios Tecnológicos

Fuente: El Autor



En la ilustración 16 se muestran las áreas que participan en el desarrollo de aplicaciones y sus roles en la arquitectura de negocio objetivo, esto desde el punto de vista de estructura organizacional.

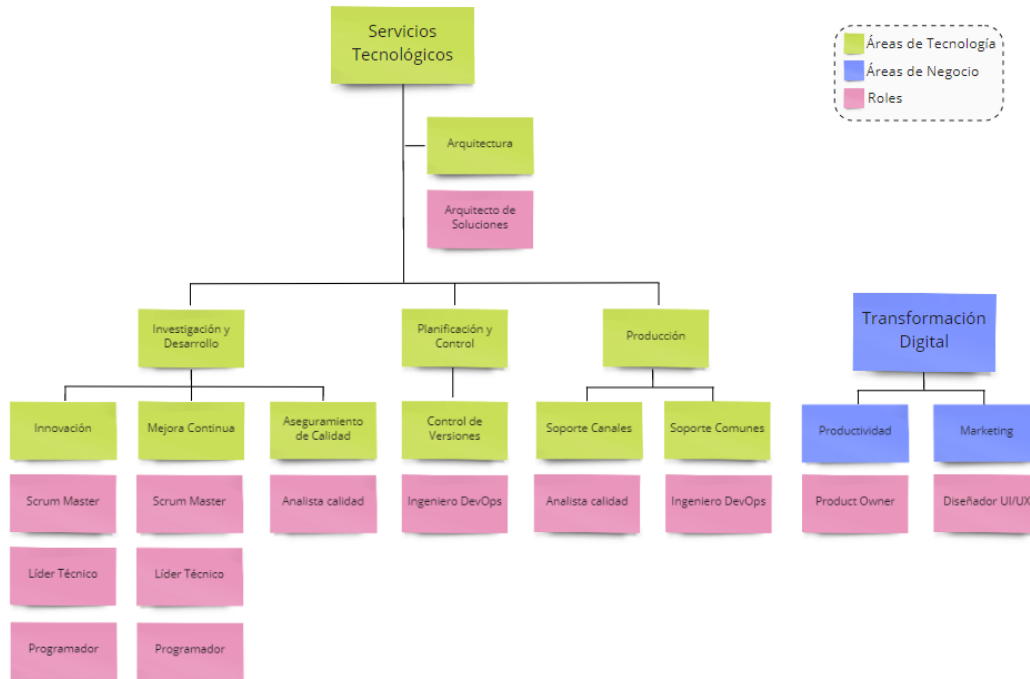


Ilustración 14 Roles y áreas que participan en el proceso de Desarrollo de Aplicaciones

Fuente: El Autor

De acuerdo a lo que se muestra en la ilustración anterior se puede evidenciar de manera rápida las áreas que participan en la generación de productos de software de manera directa y sus roles están ajustados a la metodología ágil que se implementará en la arquitectura objetivo.

### 3.2.2. Personas

A continuación, se muestran los roles necesarios para las personas que forman parte del ciclo de vida de desarrollo de aplicaciones:

#### 3.2.2.1. Roles secundarios

Existen tres roles centrales en Scrum que tienen la responsabilidad de lograr los objetivos del proyecto y son el *Product Owner*, el *Scrum Master* y el *Equipo Scrum*.

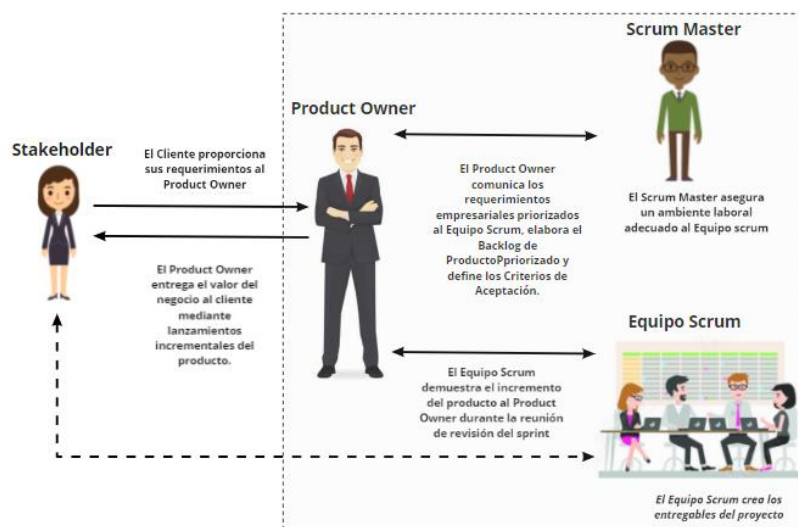


Ilustración 15 Descripción general de roles de Scrum

Fuente: El Autor

En la ilustración 15 se muestra la interacción que existe entre los roles principales de *Scrum*, es importante mencionar que ninguno de estos tres roles tiene autoridad sobre los otros.

- Product Owner define las Historias (junto con otros miembros del equipo) y prioriza la acumulación del equipo para agilizar la ejecución de las prioridades del programa. Al mismo tiempo, también mantienen la

integridad conceptual y técnica del trabajo del que es responsable el equipo. (Agile Teams - Scaled Agile Framework, s. f.)

- Scrum Master es un líder servidor y entrenador del equipo. Esta función inculca el proceso Agile acordado, ayuda a facilitar la eliminación de impedimentos para el progreso y fomenta un entorno de alto rendimiento, flujo continuo y mejora incesante. (Agile Teams - Scaled Agile Framework, s. f.)
- El Equipo Scrum, es un grupo o equipo de personas responsables de entender los requerimientos del negocio especificados por el Product Owner, de estimar las historias de usuarios y de la creación final de los entregables del proyecto. A guide to the Scrum Body of knowledge (SBOK Guide), 2013)

### **3.2.2.2. Roles complementarios**

A continuación, se detallan los roles complementarios que son necesarios para el correcto funcionamiento de la fábrica de desarrollo de software de la institución financiera:

- Diseñador UI/UX. Especifica los estándares y métodos de diseño de la experiencia del usuario para cumplir con los objetivos organizacionales para sistemas, productos y servicios y combinar digital y fuera de línea experiencias. Planifica y lidera actividades de diseño de experiencia de usuario para estrategias, grandes y complejas programas.
- Líder técnico. Construye y refactoriza software, planifica y lidera actividades de construcción de software para proyectos estratégicos, grandes y complejos proyectos de desarrollo. Desarrolla nuevos métodos y capacidades organizativas e impulsa la adopción y adherencia a las políticas y estándares.

- **Arquitecto de Soluciones.** Lidera el desarrollo de arquitecturas para soluciones complejas, asegurando la coherencia con las especificaciones de requisitos acordados con clientes externos e internos. Asume toda la responsabilidad del equilibrio entre los requisitos funcionales, de calidad del servicio y de gestión de sistemas dentro de un área significativa de la organización. Establece la política y la estrategia para la selección de la solución, componentes de la arquitectura, y coordina las actividades de diseño, promoviendo la disciplina para asegurar consistencia. (sfiaref.7.20180729.en.pdf, s. f.)
- **Analista Calidad.** Garantiza que los procesos y actividades de aseguramiento de la calidad sean sólidos y se basen mejores prácticas de la industria. Considera las implicaciones de la tecnología, enfoques, tendencias y regulaciones emergentes. Planifica y asigna recursos a las actividades de aseguramiento de la calidad organizacional. Monitorea e informes sobre las actividades de aseguramiento de la calidad, los niveles de cumplimiento y tanto organizativos como del proyecto riesgos.
- **Ingeniero Devops.** El ingeniero de DevOps incorpora procesos, herramientas y metodologías para equilibrar las necesidades durante todo el ciclo de vida del desarrollo de software, desde la codificación y la implementación hasta el mantenimiento y las actualizaciones. (¿Qué hace un ingeniero de DevOps?, s. f.)
- **Analista Soporte.** Redacta y mantiene procedimientos y documentación para soporte de aplicaciones. Administra la aplicación mejoras para mejorar el rendimiento empresarial. Asesora sobre seguridad de aplicaciones, licencias, actualizaciones, copias de seguridad y necesidades de recuperación ante desastres. Garantiza que se tramiten todas las solicitudes de apoyo de acuerdo con los estándares y procedimientos establecidos.

### 3.2.2.3. Responsabilidades

A continuación, se muestran las responsabilidades de los roles de Scrum.

Tabla 8

Responsabilidades de roles en Scrum

Rol	Responsabilidades
<b>Equipo Scrum</b>	Asume la responsabilidad colectiva y garantiza que los entregables del proyecto sean elaborados según los requerimientos. Asegura al Product Owner y el Scrum Master que el trabajo asignado se está llevando a cabo según el plan.
<b>Product Owner</b>	Elabora los requerimientos generales iniciales del proyecto y arranca el proyecto. Nombra a las personas apropiadas en los roles de Scrum Master y Equipo Scrum. Ofrece los recursos económicos iniciales y constantes para el proyecto. Determina la visión del producto y evalúa la viabilidad y asegura la entrega del producto o servicio. Asegura la transparencia y claridad de los elementos del Backlog. Priorizado del Producto. Proporciona los criterios de aceptación de las historias de usuario a ser desarrolladas en un sprint e Inspecciona los entregables.
<b>Scrum Master</b>	Asegura que todos los miembros del equipo, incluyendo el Product Owner estén cumpliendo correctamente los procesos de Scrum. Asegura de que el desarrollo del producto o servicio avance sin contratiempos y que los integrantes del Equipo Scrum cuenten con todas las herramientas necesarias para realizar el trabajo Supervisa la reunión de planificación del lanzamiento y programa otras reuniones.
<b>Program Product Owner</b>	Define los objetivos estratégicos y las prioridades de los programas.
<b>Program Scrum Master</b>	Resuelve problemas y coordina reuniones para los programas.

### 3.2.3. Modelo de procesos

Los procesos de la fábrica de desarrollo se basan en los procesos actuales y en el marco de referencia SBOK 3.0 el cual es un marco ideado para empresas grandes; a continuación, se describen los procesos de la fábrica de software.

Tabla 9

Procesos objetivos y sus descripciones de la fábrica de software

Proceso	Descripción
<b>Planeación y gestión</b>	Proceso responsable de la planificación y administración de los requerimientos de software desde el inicio hasta el final del ciclo de vida del producto de software. Incluye épicas, funcionalidades, historias de usuario y actividades.
<b>Control de versiones</b>	Responsable de administrar y monitorear el proceso de cambios de código a lo largo del tiempo. Incluye código legado y actual tanto de interno como de proveedores.
<b>Desarrollo de software</b>	Es un conjunto de subprocesos o etapas que pueden promover rápidamente el desarrollo, la prueba y la implementación de productos de software. El proceso incluye gestión ágil del proyecto con la metodología SCRUM.
<b>Gestión de pruebas</b>	Es la verificación dinámica de que los programas generan el comportamiento esperado a partir de un conjunto limitado casos de prueba, seleccionados apropiadamente del dominio de ejecución generalmente ilimitado. Incluye Gobierno de la gestión de pruebas, tipos de pruebas a realizar, procesos y procedimientos a seguir de acuerdo a la aplicación, tipo de tecnología y exposición de la misma.
<b>Trazabilidad de requerimientos</b>	Es la capacidad de registrar la relación entre los requisitos dados por las partes interesadas, los requisitos del proyecto y la solución final implementada. Incluye el

	monitoreo en ambiente productivos (pmoinformatica.com, s. f.)
<b>Seguridad</b>	Diseñado para mantener la seguridad de los sistemas y aplicaciones de información. Incluye análisis estático y dinámico del código fuente y los sitios desplegados. <b>(GTI8.pdf, s. f.)</b>
<b>Integración continua</b>	Integra los diferentes componentes del software en desarrollo. El servidor de integración continua genera retroalimentación sobre los resultados del proceso de construcción, que es enviado a miembros designados del proyecto. El proceso incluye las todas las aplicaciones de la institución a través técnicas de integración, compilación y configuración automáticas (Salamon et al., 2014).
<b>Despliegue de aplicaciones</b>	Estas actividades incluyen la entrega, montaje y gestión en un sitio de los recursos necesarios para utilizar una versión de un sistema de software. Además, el despliegue de aplicaciones usará la misma herramienta para ambientes no productivos y productivos con las autorizaciones y seguridades del caso. (Hernández Yeja & Porven Rubier, 2016)
<b>Gestión de la configuración</b>	Abarca las actividades administrativas relacionadas con la creación, mantenimiento, cambio controlado y control de calidad de los productos. ( <i>Gestión de la configuración - Praxis Framework</i> , s. f.) Incluye el cambio de parámetros y configuraciones a través de herramientas automatizadas en ambientes no productivos y productivos.

En la tabla 9 se describen a detalle los procesos de la fábrica de software de la institución financiera

### 3.2.4. Fases de Scrum

En la ilustración 16 se muestra el flujo de *Scrum* que debe seguirse para la entrega de un producto de software. A continuación, se detallan los sub procesos que son parte de las fases de *Scrum*.

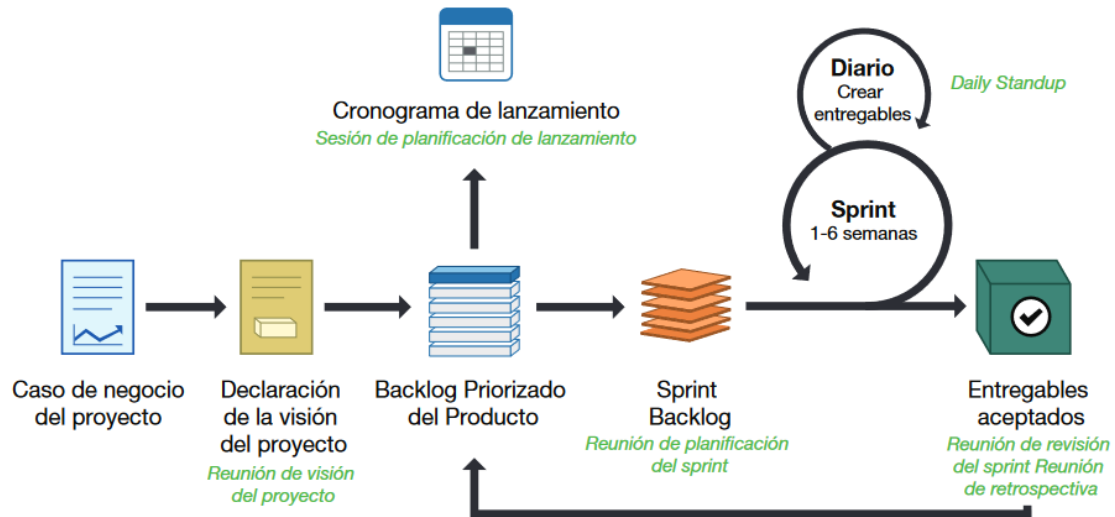


Ilustración 16 Flujo de scrum

Fuentes: Scrum Study

#### 3.2.4.1. Inicio

La fase de inicio, incluye los procesos relacionados al inicio de un proyecto gestionado con *Scrum*

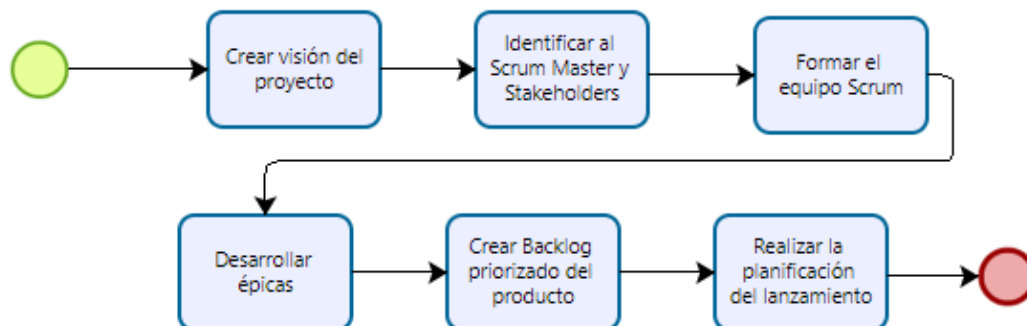


Ilustración 17 Subprocesos de la fase de inicio de Scrum

Fuente: Adaptado de SBOK 3.0



Como se puede evidenciar en la ilustración 17, se muestran los subprocesos de la fase de inicio de *Scrum*.

- Crear la visión del proyecto. En este proceso, el caso de negocio del proyecto se revisa para crear una declaración de la visión del proyecto que servirá de inspiración y proporcionará un enfoque de todo el proyecto. En este proceso se identifica al *Product Owner*.
- Identificar al Scrum Master y Stakeholders(s). En este proceso, se identifica al Scrum Master y a los *Stakeholders* utilizando criterios de selección específicos.
- Formar el Equipo Scrum. En este proceso, se seleccionan a los miembros del Equipo Scrum. Normalmente, el *Product Owner* es el responsable principal de la selección de los miembros del equipo, pero a menudo lo hace en colaboración con el Scrum Master.
- Desarrollar de épica(s). En este proceso, la declaración de la visión del proyecto (Project Vision Statement) sirve como base para el desarrollo de épicas.
- Crear el Backlog priorizado del producto. En este proceso, la épica(s) es refinada, detallada y luego priorizada para el Backlog Priorizado del Producto del proyecto. En este punto también se establecen los criterios de terminado.
- Realizar la planificación del lanzamiento. En este proceso, el equipo principal de Scrum revisa las historias de usuario en el Backlog priorizado del producto para desarrollar un cronograma de planificación del lanzamiento. La duración de los *Sprints* también se determina en este proceso.

#### **3.2.4.2. Planificación y estimación**

La fase de planificación y estimación consiste en procesos relacionados a la planificación y estimación de tareas, los cuales incluyen: Crear historias de

usuario, estimar historias de usuario, comprometer historias de usuario, Identificar tareas, estimar tareas y crear el *Sprint Backlog*.

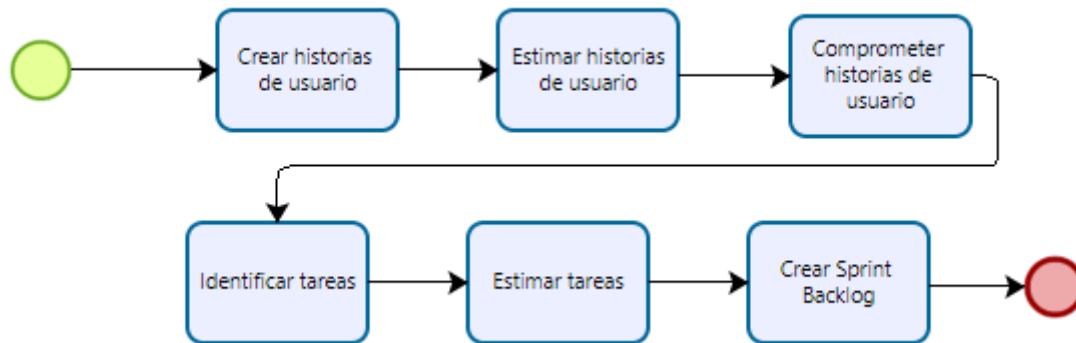


Ilustración 18 Subprocesos de la fase de planificación y estimación de Scrum

Fuente: Adaptado de SBOK 3.0

Como se puede evidenciar en la ilustración 18, se muestran los subprocesos de la fase de planificación y estimación de *Scrum*.

- Crear historias de usuario. En este proceso se crean las historias de usuario y sus respectivos criterios de aceptación. Las historias de usuario generalmente las escribe el *Product Owner* y se diseñan para garantizar que los requerimientos del cliente estén claramente representados y que todos los *Stakeholders* las pueden entender completamente. Tales historias se incorporan al Backlog priorizado del producto.
- Estimar historias de usuario. En este proceso el *Product Owner* aclara las historias de usuario a fin de que el Scrum Master y el Equipo Scrum hagan una estimación sobre el esfuerzo necesario para desarrollar la funcionalidad descrita en cada historia de usuario.
- Comprometer historias de usuario. En este proceso el Equipo Scrum se compromete a entregar al *Product Owner* las historias de usuario para un sprint.

- Identificar tareas. En este proceso las historias de usuario comprometidas se dividen en tareas específicas y se compilan en una lista de tareas.
- Estimar tareas. En este proceso, el equipo principal de Scrum estima el esfuerzo necesario para completar cada tarea en la lista de tareas.
- Crear el Sprint Backlog. En este, el equipo principal de Scrum lleva a cabo reuniones de planificación del sprint donde el grupo elabora un Sprint Backlog que contiene todas las tareas a completarse en el sprint.

### 3.2.4.3. Implementación

La fase de implementación está relacionada a la ejecución de las tareas y actividades para crear el producto de un proyecto. Estas actividades incluyen la creación de varios entregables, realizar *Daily Standups* y el refinamiento (revisiones, ajustes y actualización periódica) del Backlog Priorizado del Producto en intervalos frecuentes.

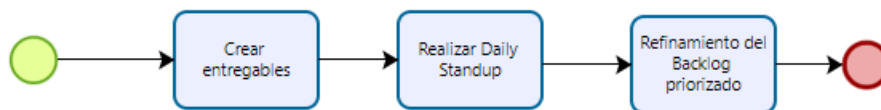


Ilustración 19 Subprocesos de la fase de implementación de Scrum

Fuente: Adaptado de SBOK 3.0

Como se puede evidenciar en la ilustración 18, se muestran los subprocesos de la fase de implementación de *Scrum*.

- Crear entregables. En este proceso, el Equipo Scrum trabaja en las tareas del Sprint Backlog para crear los entregables del Sprint. Generalmente se utiliza un *Scrumboard* para dar seguimiento al trabajo y a las actividades que se llevan a cabo.

- Realizar *Daily Standup*. En este proceso se lleva a cabo diariamente una reunión altamente focalizada con un time-box asignado y denominada: *Daily Standup Meeting*. Es un foro para que el Equipo Scrum se ponga al día sobre sus progresos y sobre cualquier impedimento que pudieran estar enfrentando.
- Refinamiento del Backlog Priorizado. En este proceso constantemente se actualiza y refina el Backlog Priorizado del Producto.

#### 3.2.4.4. Revisión y retrospectiva

La fase de revisión y retrospectiva cubre los relacionado a la revisión de los entregables y al trabajo que se ha realizado y determina las formas para mejorar las prácticas y métodos implementados para realizar el trabajo del proyecto. En las grandes organizaciones, el proceso de revisión y retrospectiva también puede incluir el convocar a reuniones de Scrum de Scrums.

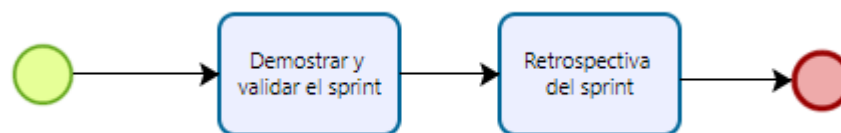


Ilustración 20 Subprocesos de la fase de revisión y retrospectiva de Scrum

Fuente: Adaptado de SBOK 3.0

Como se puede evidenciar en la ilustración 18, se muestran los subprocesos de la fase de revisión y retrospectiva de *Scrum*.

- Demostrar y validar el sprint. En este proceso, el Equipo Scrum demuestra los entregables del sprint al *Product Owner* y a los *Stakeholders* relevantes durante una reunión de revisión del sprint. El propósito de esta reunión es lograr la aprobación y aceptación del *Product Owner* respecto al producto o servicio.

- Retrospectiva del sprint. En este proceso, el Scrum Master y el Equipo Scrum se reúnen para discutir las lecciones aprendidas durante el sprint. Dicha información se documenta como lecciones aprendidas que pudieran implementarse en futuros *Sprints*.

#### 3.2.4.5. Lanzamiento

La fase de lanzamiento (*Release*) hace énfasis en la entrega al cliente de los entregables aceptados, así como en la identificación, documentación e internalización de las lecciones aprendidas durante el proyecto.

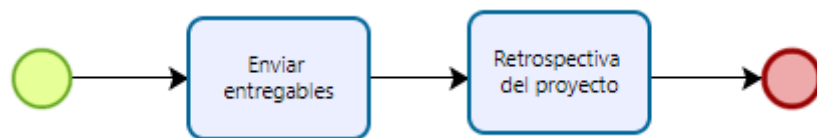


Ilustración 21 Subprocesos de la fase de lanzamiento de Scrum

Fuente: Adaptado de SBOK 3.0

Como se puede evidenciar en la ilustración 18, se muestran los subprocesos de la fase de lanzamiento de *Scrum*.

- Enviar entregables. En este proceso se hace la entrega o la transición de los entregables aceptados a los *Stakeholders* relevantes.
- Retrospectiva del proyecto. En este proceso, en el cual se concluye el proyecto, los *Stakeholders* de la organización y los miembros del equipo principal de Scrum se reúnen para hacer una retrospectiva del proyecto e identificar, documentar e internalizar las lecciones que se aprendieron.

### 3.2.5. Artefactos

Son aquellos elementos físicos resultantes de la aplicación de scrum y son *Product Backlog*, *Sprint Backlog* y el incremento.

#### 3.2.5.1. Product Backlog

El *Product Backlog* es un inventario que contiene cualquier tipo de trabajo que haya que hacer en el producto: requerimientos, casos de uso, tareas y dependencias. Es la principal fuente de información sobre el producto en Scrum, una lista, que contiene todos los requerimientos que necesitamos implementar en el producto. Esta lista es el resultado del trabajo del *Product Owner* con el cliente, los distintos *Stakeholders*, sponsors, comités, etc, y refleja el estado real del trabajo pendiente de implementar en el producto, así como el ya realizado.

Un *Product Backlog* contiene distintos elementos:

- Funcionalidades
- Bugs
- Historias de usuario
- Tareas técnicas
- Trabajo de investigación

#### 3.2.5.2. Sprint Backlog

Es una lista de elementos en los que trabajar durante la etapa de *Sprint*. Estos elementos se componen de tareas técnicas más pequeñas que permiten conseguir un incremento de software terminado. (*Artefactos Scrum: las 3 herramientas clave de gestión*, s. f.)

Este artefacto es un elemento para visualizar el trabajo a realizar durante cada *Sprint* y está gestionado por el equipo de desarrollo. Su propósito es mantener

la transparencia dentro del desarrollo, actualizándolo durante toda la iteración especialmente a través de los *Daily Scrums*. (*Artefactos Scrum: las 3 herramientas clave de gestión*, s. f.)

### **3.2.5.3. Incremento**

Incremento, es el resultado del *Sprint*, es la suma de todas las tareas, casos de uso, historias de usuario y cualquier elemento que se haya desarrollado durante el *Sprint* y que será puesto a disposición del usuario final en forma de software, aportando un valor de negocio al producto que se está desarrollando. (*Artefactos Scrum: las 3 herramientas clave de gestión*, s. f.)

### **3.2.6. Eventos**

Scrum define cinco eventos para cumplir con el control de sus procesos, y son los siguientes:

#### **3.2.6.1. Sprint Planning**

El *Sprint Planning* es una reunión que se realiza al comienzo de cada *Sprint* donde participa el equipo Scrum al completo; sirve para inspeccionar el Backlog del Producto y que el equipo de desarrollo seleccione los elementos del Backlog del producto en los que va a trabajar durante el siguiente *Sprint*. Estos elementos del Backlog del producto son los que compondrán el *Sprint Backlog*. (*Las 5 ceremonias Scrum: claves para la gestión de procesos*, s. f.)

Durante esta reunión, el *Product Owner* presenta el *Product Backlog* actualizado que el equipo de desarrollo se encarga de estimar, además de intentar clarificar aquellos ítems que crea necesarios.

En el *Sprint Planning* se inspeccionan el *Product Backlog*, los acuerdos de la Retrospectiva, la capacidad y la Definition of Done y se adaptan el *Sprint Backlog*, *Sprint Goal* y el plan para poder alcanzar la meta del *Sprint*.

El Sprint Planning se divide en dos partes. En la primera parte de la reunión se trata Qué se va a hacer en el Sprint y, en la segunda parte, se discute el Cómo. La primera parte está organizada y liderada por el *Product Owner*, mientras que de la segunda parte se encarga el *Equipo Scrum*.

### 3.2.6.2. Dalily Scrum

Es una reunión diaria de 15 minutos en la que participa exclusivamente el *Equipo Scrum*. En esta reunión todas y cada una de las personas del Development Team responden a las siguientes preguntas:

- ¿Qué hice ayer para contribuir al Sprint Goal?
- ¿Qué voy a hacer hoy para contribuir al Sprint Goal?
- ¿Tengo algún impedimento que me impida entregar?

### 3.2.6.3. Sprint Review

la reunión que ocurre al final del Sprint, donde el *Product Owner* y el *Equipo Scrum* presentan a los *Stakeholders* el incremento terminado para su inspección y adaptación correspondientes. En esta reunión organizada por el *Product Owner* se estudia cuál es la situación y se actualiza el *Product Backlog* con las nuevas condiciones que puedan afectar al negocio.

Por un lado, se revisará el incremento terminado. Se mostrará el software funcionando en producción y los *Stakeholders* tendrán la oportunidad de hacer cuantas preguntas estimen oportunas sobre el mismo. El software funcionando ha sido validado previamente por el *Product Owner*, que se ha encargado de trabajar con el equipo durante el *Sprint* para asegurarse que cumple con la definición de hecho y, efectivamente, hace que la meta del *Sprint* sea válida.



#### **3.2.6.4. Sprint Retrospective**

La retrospectiva ocurre al final del *Sprint*, justo después del *Sprint Review*. El objetivo de la retrospectiva es hacer de reflexión sobre el último *Sprint* e identificar posibles mejoras para el próximo. Aunque lo habitual es que el Scrum Master sea el facilitador, es normal que distintos miembros del equipo Scrum vayan rotando el rol de facilitador durante la retrospectiva.

Un formato común es analizar qué ha ido bien durante el *Sprint*, qué ha fallado y qué se puede mejorar.

#### **3.2.6.5. Refinamiento del Sprint**

Esta ceremonia sigue un patrón similar al resto y tiene una agenda fija específica en cada *Sprint*. Se estima su duración en 2 horas máximo por semana del *Sprint*. Es responsabilidad del *Product Owner* agendar, gestionar y dirigir esta reunión.

Los participantes de esta reunión son todo el equipo Scrum, así como cualquier recurso adicional que considere necesario el *Product Owner* y que pueda contribuir a aclarar el requerimiento. Es necesario, por tanto, que antes de la reunión todos conozcan los requerimientos o historias de usuario que van a ser tratados en la misma.

### **3.3. Análisis de brechas**

La tabla 9 muestra el análisis de brechas obtenido al comparar la arquitectura de negocio actual y la objetivo (incluida la estructura organizacional, las personas y los procesos).

Tabla 10

Análisis de brechas de la arquitectura de negocio actual y negocio

Subdominio	Categoría	Actual	Objetivo	Aspiracional
Estructura Organizacional	Tipo de Estructura	2	4	5
Estructura Organizacional	Gobernanza y mejora Continua	2	4	5
Personas	Formación académica	3	4	5
Personas	Formación complementaria	2	4	5
Personas	Experiencia	2	4	5
Personas	Habilidades	2	4	5
Procesos	Planeación y gestión	2	4	5
Procesos	Control de versiones	3	4	5
Procesos	Desarrollo de software	2	4	5
Procesos	Gestión de pruebas	1	4	5
Procesos	Trazabilidad de requerimientos	2	4	5
Procesos	Seguridad	2	4	5
Procesos	Integración continua	2	4	5
Procesos	Despliegue de aplicaciones	1	4	5
Procesos	Gestión de la configuración	2	4	5

Como se observa en la tabla 10, existen brechas significativas en la mayoría de las categorías evaluadas, para representar esta diferencia de mejor manera en la Ilustración 18 se muestra las brechas entre la arquitectura de negocio actual y la arquitectura de negocio objetivo a través de un gráfico de radar.

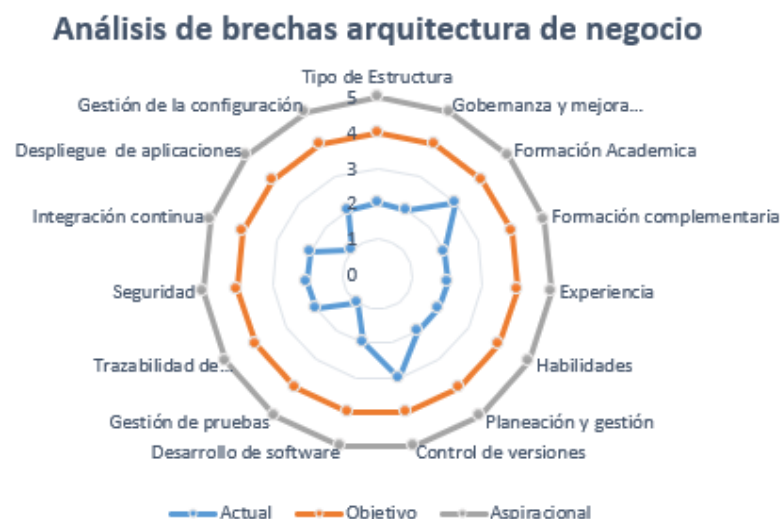


Ilustración 22 Brechas identificadas en la arquitectura de negocio actual y la objetivo.

Fuente: El Autor

En la ilustración 22 se puede ver gráficamente las brechas principales como son en el despliegue de aplicaciones, trazabilidad de requerimientos, de igual manera las atrás categorías que tienen brechas no tan notorias, así como las categorías en las que la brecha es muy leve como la Formación académica y el control de versiones.

En la tabla 11 se muestra el listado de brechas identificadas entre la arquitectura de negocio actual contrastada con la arquitectura de negocio objetivo, estas brechas incluyen.

Tabla 11

Brechas identificadas en la arquitectura de negocio

Subdominio	Categoría	Brecha
Estructura Organizacional	Tipo de Estructura	La estructura es funcional y altamente jerárquica.
Estructura Organizacional	Gobernanza y mejora Continua	No existe gobierno de la estructura organizacional y la mejora continua se la realiza de manera empírica.
Personas	Formación académica	Las personas con cargos de Coordinación y jefaturas no cuentan con la formación adecuada.
Personas	Formación complementaria	Casi el total del personal de la fábrica de software no cuenta con formación complementaria necesaria para su rol.
Personas	Experiencia	Los roles relacionados con gestión no tienen la experiencia requerida en ese ámbito.
Personas	Habilidades	Las personas no cuentan con las habilidades necesarias para ejecutar actividades de desarrollo ágil.
Procesos	Planeación y gestión	La planeación y gestión de requerimientos y/o historias de usuario no tiene métricas de rendimiento.

Procesos	Control de versiones	Existe un proceso y procedimiento formal para la mayoría de aplicaciones de la institución.
Procesos	Desarrollo de software	No existe un proceso formal para el desarrollo ágil de software ni para el control de estos.
Procesos	Gestión de pruebas	No existe un gobierno, proceso ni procedimiento para realizar las pruebas de software. Las pruebas se las realiza de manera empírica.
Procesos	Trazabilidad de requerimientos	Los requerimientos no se pueden seguir durante toda la etapa del ciclo de vida de la aplicación.
Procesos	Seguridad	Existen ciertos estándares de análisis de código estático y dinámico del código.
Procesos	Integración continua	Existen procesos y herramientas para realizar la integración continua, pero no se lo hace en todas las aplicaciones.
Procesos	Despliegue de aplicaciones	EL despliegue de productos de software es realizado por varias herramientas, incluso se lo realiza manualmente.
Procesos	Gestión de la configuración	La Gestión de la configuración en ambientes productivos es completamente manual.

### 3.4. Iniciativas para cerrar brechas

Una vez realizado el análisis de brechas entre la arquitectura de negocios actual y objetivo, se encontraron las brechas descritas en la tabla 10, las cuales pueden ser cerradas con la ejecución de las siguientes iniciativas:

- Se debe capacitar a los distintos roles que participan en la fábrica de software en la metodología SCRUM, esto incluye descripción de roles y responsabilidades, entregables, ceremonias, agilismo.

- Optimizar los procesos y procedimientos de despliegue de aplicaciones en ambientes no productivos y productivos con el objetivo que estas liberaciones sean ágiles y tengan las seguridades necesarias.
- El Scrum Master debe realizar talleres de agilidad y dejar las actividades técnicas al equipo SCRUM.
- Se debe crear una estrategia para la gestión de pruebas automáticas de software, esto debe incluir pruebas unitarias, pruebas funcionales, pruebas por comportamiento, pruebas de carga, pruebas de integración.
- Migración a la herramienta Azure Devops, de esta manera se logrará tener en una sola herramienta todo el código fuente de la institución, así como los beneficios y fortalezas de estar hospedada en nube como son resiliencia, actualizaciones, nuevos productos y características, entre otros.

## 4. Arquitectura de aplicaciones y datos

La arquitectura de aplicaciones y datos habilitan la arquitectura de negocio y la visión de arquitectura definida en capítulos previos, toma en cuenta la declaración del trabajo de arquitectura, las necesidades y preocupaciones de las partes interesadas, identifica componentes candidatos para formar parte de la hoja de ruta hacia la arquitectura destino en función de las brechas identificadas. (*The TOGAF Standard, Version 9.2 - Phase C: Information Systems Architectures - Application Architecture*, s. f.)

### 4.1. Arquitectura de Aplicaciones y datos actual

Actualmente la fábrica de software de la institución financiera cuenta con un catálogo extenso de aplicaciones. En esta sección se detallarán las aplicaciones que se utilizan dentro del ciclo de vida de desarrollo de software y de que manera almacena la información de los proyectos lo cual está relacionado con la arquitectura de datos.

#### 4.1.1. Catálogo de aplicaciones

En la tabla 12 se muestra la relación de las aplicaciones utilizadas durante la ejecución de los procesos que son parte de la fábrica de software de la institución financiera.

Tabla 12

Aplicaciones de la fábrica de software

Aplicaciones	Descripción
<b>Azure Devops Server</b>	Azure DevOps Server, antes conocido como Team Foundation Server (TFS), es un conjunto de herramientas de desarrollo de software en colaboración hospedadas en el entorno local. Azure DevOps Server se integra con su IDE o editor actual, lo que permite que los equipos multidisciplinares trabajen de un modo

	eficaz en proyectos de todos los tamaños. (Azure DevOps Server   Microsoft Azure, s. f.)
<b>Project Server</b>	Una Software para la administración de cartera de proyectos y el seguimiento diario. Permite a todos los usuarios comenzar rápidamente, priorizar proyectos y recursos y trabajar prácticamente desde cualquier lugar. (Administración de cartera de proyectos y de recursos empresariales, s. f.)
<b>Team Foundation Server</b>	Es un conjunto de herramientas de desarrollo de software en colaboración hospedadas en el entorno local. Azure DevOps Server se integra con su IDE o editor actual, lo que permite que los equipos multidisciplinarios trabajen de un modo eficaz en proyectos de todos los tamaños. (Azure DevOps Server   Microsoft Azure, s. f.)
<b>Visual Studio</b>	Es un <i>entorno de desarrollo integrado</i> que se puede usar para editar, depurar y compilar código y, después, publicar una aplicación. Un entorno de desarrollo integrado (IDE) es un programa con numerosas características que se pueden usar para muchos aspectos del desarrollo de software. (anandmeg, s. f.)
<b>Xamarin</b>	Es una plataforma de código abierto para compilar aplicaciones modernas y con mejor rendimiento para iOS, Android y Windows con .NET. Xamarin es una capa de abstracción que administra la comunicación de código compartido con el código de plataforma subyacente. (profexorgeek, s. f.)
<b>xUnit</b>	xUnit.NET es un marco de prueba popular para el código .NET que se puede usar para escribir pruebas de unitarias y de integración. Como todo lo demás, es un conjunto de paquetes NuGet que se pueden instalar en cualquier proyecto. (Pruebas unitarias - El pequeño libro de ASP.NET Core, s. f.)
<b>Microfocus UFT</b>	Automatización inteligente de pruebas con capacidades integradas basadas en la IA que acelera las pruebas en todo tipo de aplicaciones de escritorio, web, móviles, mainframe, compuestas y empaquetadas de nivel empresarial. (Automatice las pruebas funcionales desde la IU hasta la API   UFT One   Micro Focus, s. f.)
<b>Microfocus LoadRunner</b>	Realice el Performance Testing y las pruebas de carga en la nube. LoadRunner Cloud facilita la planificación, ejecución y ampliación de pruebas de rendimiento en la nube sin necesidad

	de implantar y gestionar infraestructuras. (Pruebas de carga en la nube   Performance Testing en la nube   Micro Focus, s. f.)
<b>Microfocus Fortify</b>	Solución de seguridad de aplicaciones integrales, cuenta con análisis estático y dinámico de código, así como otras características de seguridad.
<b>SonarQube</b>	Es una herramienta que permite realizar un análisis estático de código. Lo que hace es identificar los puntos susceptibles de mejora, que facilitarán la obtención de métricas necesarias para la optimización del código. («SonarQube», 2020)
<b>Gestor de despliegue a medida</b>	Herramienta de despliegue de aplicaciones desarrollada a medida, copia los paquetes o archivos compilados de un origen a un destino configurado previamente.
<b>Sharepoint online</b>	Es una herramienta para gestión documental y colaboración en la que se almacenan los documentos de los proyectos en curso.

#### 4.1.2. Análisis de ciclo de vida de aplicaciones

El análisis del ciclo de vida de aplicaciones valida estado de una aplicación con el objetivo de determina si se debe integrar, tolerar, eliminar o migrar.

En la ilustración 19 se muestra el análisis de las aplicaciones utilizadas en el ciclo de vida de desarrollo de software desde las perspectivas técnica y funcional, la rúbrica de evaluación se puede revisar en el Anexo A.

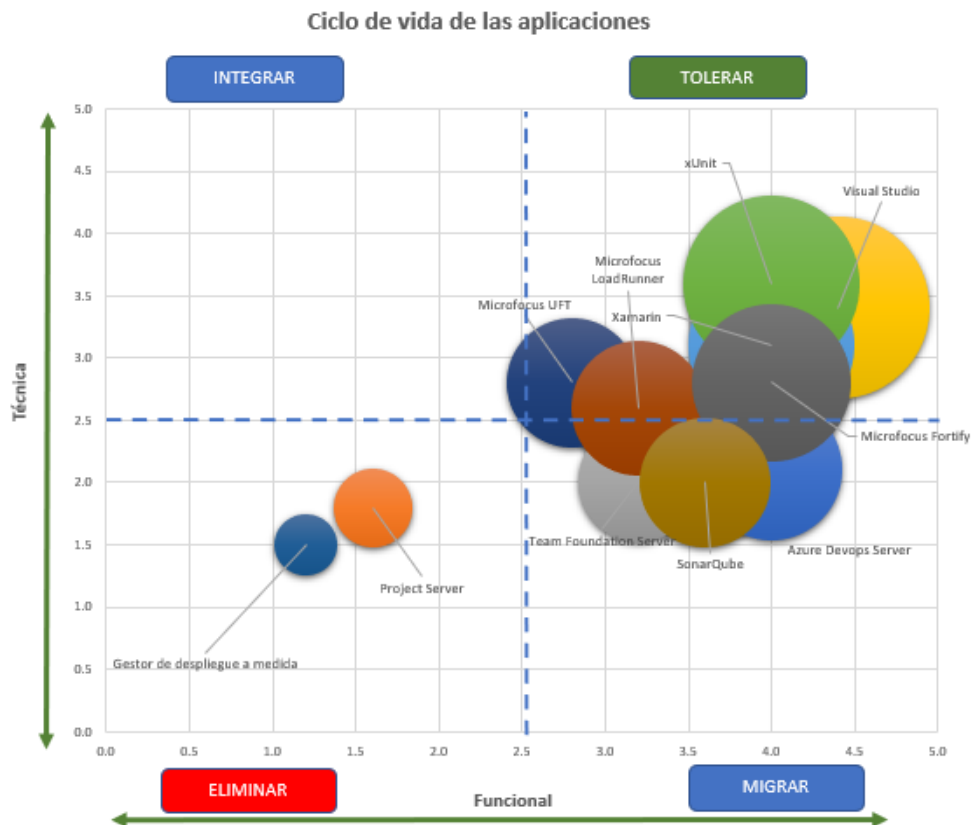


	Vigencia tecnológica	Interoperabilidad	Confianza	Arquitectura	Soporte	Seguridad	Flexibilidad / Mantenimiento	Gobernabilidad	Función de la aplicación	Rendimiento y uso	Calidad de la aplicación	Promedio Técnico	Promedio Funcional	Promedio General (Madurez)
Aplicaciones	Técnica							Funcional			40%	60%		
Azure Devops Server	2	3	4	2	3	3	2	2	4	3	3	2.1	4.0	3.1
Project Server	2	3	3	1	3	2	2	2	1	2	1	1.8	1.6	1.7
Team Foundation Server	2	3	2	2	3	4	2	2	3	2	3	2.0	3.2	2.6
Visual Studio	4	4	4	4	4	4	5	5	3	4	4	3.4	4.4	3.9
Xamarin	2	5	3	5	4	5	3	4	3	3	4	3.1	4.0	3.6
xUnit	5	4	5	4	5	4	5	4	3	3	4	3.6	4.0	3.8
Microfocus UFT	4	3	4	3	4	3	4	3	2	2	3	2.8	2.8	2.8
Microfocus LoadRunner	3	3	3	4	3	3	3	4	3	3	2	2.6	3.2	2.9
Microfocus Fortify	4	3	4	3	3	4	4	3	4	3	3	2.8	4.0	3.4
SonarQube	2	2	2	3	2	2	2	5	3	2	4	2.0	3.6	2.8
Gestor de despliegue a medida	2	2	1	2	3	2	1	2	1	1	1	1.5	1.2	1.4

Ilustración 23 Análisis de ciclo de vida de aplicaciones.

Fuente: El Autor

La ilustración 19 permite determinar el estado de las aplicaciones de acuerdo a las perspectivas funcional y técnica y con estas dimensiones entender en que parte del ciclo de vida de las aplicaciones se encuentra (Eliminar, Tolerar, Integrar, Migrar). Para lograrlo se utilizaron las rúbricas que se detallan en el Anexo A para la evaluación desde el punto de vista funcional y técnica.



Fuente: El Autor

En la Ilustración 20 se representan las aplicaciones a través de un gráfico tipo burbuja en el cual en el eje X (Funcional) se toma el valor promedio funcional de cada aplicación, en el eje Y (Técnica) se toma el valor promedio técnico de cada aplicación, finalmente se toma un tercer valor que es el promedio entre valor funcional y técnico teniendo en cuenta el peso del 60% y 40% respectivamente, este valor muestra visualmente el tamaño de la burbuja que indica el nivel de madurez de la aplicación.

## 4.2. Arquitectura de Aplicaciones objetivo

Las aplicaciones que serán parte de la arquitectura objetivo se las tomará en base al resultado de la ilustración 24, en la tabla 13 se muestra el detalle de las aplicaciones que serán parte de la arquitectura objetivo.

Tabla 13

## Aplicaciones de la arquitectura objetivo

Aplicaciones	Descripción
<b>Azure Devops</b>	Es el resultado de migrar las aplicaciones Azure Devops Server y Team Foundation Server. Esta aplicación se encargará de la gestión del ciclo de vida de la aplicación, a través de la administración del código fuente, informes, administración de requisitos, administración de proyectos, compilaciones automatizadas, administración de laboratorio, pruebas y administración de versiones.
<b>SonarCloud</b>	Es el resultado de migrar la aplicación SonarQube. SonarCloud detecta errores y vulnerabilidades en su código y ayuda a su equipo a mejorar la mantenibilidad, confiabilidad y seguridad del código. Se puede integrar a Azure Devops brindando trazabilidad del análisis de código realizado.
<b>Visual Studio</b>	Se mantiene/tolera la aplicación por la evaluación favorable tanto técnica y funcional.
<b>Xamarin</b>	Se mantiene/tolera la aplicación por la evaluación favorable tanto técnica y funcional.
<b>xUnit</b>	Se mantiene/tolera la aplicación por la evaluación favorable tanto técnica y funcional.
<b>Microfocus UFT</b>	Se mantiene/tolera la aplicación por la evaluación favorable tanto técnica y funcional.
<b>Microfocus LoadRunner</b>	Se mantiene/tolera la aplicación por la evaluación favorable tanto técnica y funcional.
<b>Microfocus Fortify</b>	Se mantiene/tolera la aplicación por la evaluación favorable tanto técnica y funcional.
<b>Azure Devops Wiki</b>	Es un repositorio de información en el que se puede almacenar la documentación de los proyectos en curso, además se puede enlazar con ítems de trabajo, historias de usuario, bugs, épicas creadas a través de Azure Devops.

#### 4.2.1. Aplicaciones por procesos

En la ilustración 25 se muestra la relación de las aplicaciones que serán parte de la arquitectura de aplicaciones objetivo y de los procesos de la fábrica de software.

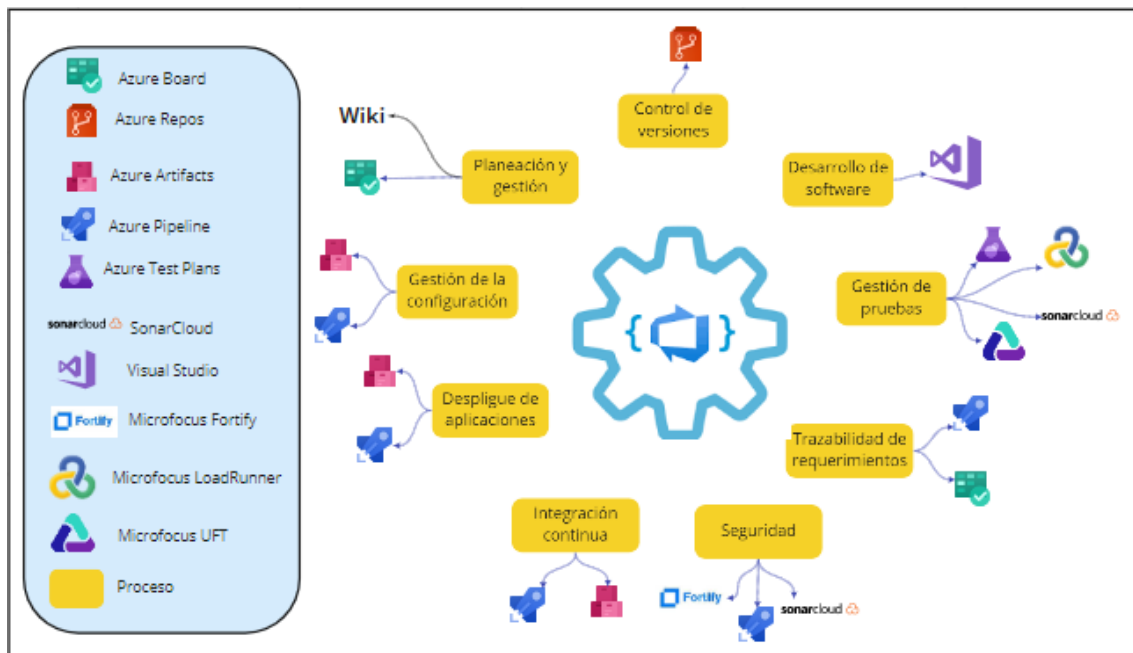


Ilustración 25 Relación aplicaciones por procesos.

Fuente: El Autor

Como se puede evidenciar en la ilustración 25, se muestran las aplicaciones que se utilizarán en cada proceso de la fábrica de software en la arquitectura de aplicaciones objetivo.

### 4.3. Análisis de brechas

La tabla 14 muestra el análisis de brechas obtenido al comparar la arquitectura de aplicaciones actual y la objetivo, el estado actual de las aplicaciones se lo obtuvo en base a la ilustración 24.

Tabla 14

Análisis de brechas de la arquitectura de aplicaciones actual y objetivo

Aplicaciones	Actual	Objetivo	Aspiracional
Azure Devops Server	3.1	4	5
Project Server	1.7	4	5
Team Foundation Server	2.6	4	5
Visual Studio	3.9	4	5
Xamarin	3.6	4	5
xUnit	3.8	4	5
Microfocus UFT	2.8	4	5
Microfocus LoadRunner	2.9	4	5
Microfocus Fortify	3.4	4	5
SonarQube	2.8	4	5
Gestor de despliegue a medida	1.4	4	5

En la Ilustración 26 se muestran las brechas entre la arquitectura de aplicaciones actual y la arquitectura de negocio objetivo, el nivel de madurez actual se tomó del análisis realizado en la sección 4.1.2 (Análisis de ciclo de vida de aplicaciones), la madurez objetivo se toma de la rúbrica del Anexo A en el que la institución financiera desea tener sus aplicaciones en una madurez de nivel 4 (Cuantitativamente manejada), finalmente la madurez aspiracional es de nivel 5 (Optimizada)

### Análisis de brechas - Arquitectura de Aplicaciones

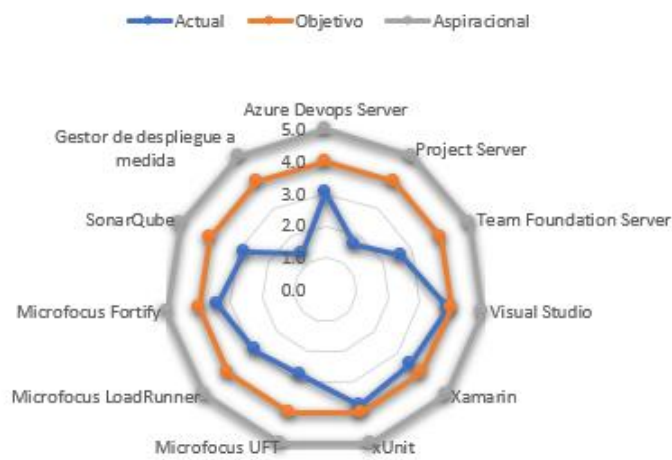


Ilustración 26 Brechas identificadas en la arquitectura de aplicaciones actual y la objetivo.

Fuente: El Autor

En la tabla 15 se muestra el listado de brechas identificadas entre la arquitectura de aplicaciones actual contrastada con la arquitectura de aplicaciones objetivo, además estas brechas incluyen la diferencia en la aplicación que es parte de la arquitectura de datos que es *Sharepoint*, la cual no es adecuada para el manejo de la documentación de los proyectos ágiles.

Tabla 15

Brechas identificadas en arquitectura de aplicaciones y datos

Aplicaciones	Brecha
Azure Devops Server	Tiene una brecha tecnológica grande al ser una herramienta OnPremises.
Project Server	Existe una brecha muy grande al ser una herramienta para la gestión de proyectos tradicionales, no brinda el soporte y ayuda necesaria para los proyectos ágiles en curso.
Team Foundation Server	Tiene una brecha tecnológica grande al ser una herramienta OnPremises.

Visual Studio	No existe una brecha significativa con esta herramienta, actualmente se está aprovechando de manera correcta las características del producto.
Xamarin	No existe una brecha significativa con esta herramienta, actualmente se está aprovechando de manera correcta las características del producto.
xUnit	No existe una brecha significativa con esta herramienta, actualmente se está aprovechando de manera correcta las características del producto.
Microfocus UFT	No existe una brecha significativa con esta herramienta, actualmente se está aprovechando de manera correcta las características del producto.
Microfocus LoadRunner	No existe una brecha significativa con esta herramienta, actualmente se está aprovechando de manera correcta las características del producto.
Microfocus Fortify	No existe una brecha significativa con esta herramienta, actualmente se está aprovechando de manera correcta las características del producto.
SonarQube	Tiene una brecha tecnológica grande al ser una herramienta OnPremises.
Gestor de despliegue a medida	La brecha es muy grande con esta aplicación, al ser hecha a medida y sin tener lo necesario para una herramienta de despliegue para una institución financiera.

#### 4.4. Iniciativas para cerrar brechas

Una vez realizado el análisis de brechas entre la arquitectura de aplicaciones actual y objetivo, se encontraron las brechas descritas en la tabla 14, las cuales pueden ser cerradas con la ejecución de las siguientes iniciativas:

- Migrar a la versión Azure DevOps la cual tiene características para cubrir las necesidades en cada parte del ciclo de vida de software de un producto.

- Utilizar las características de Azure Pipelines y Azure Artifacts de Azure Devops para realizar el despliegue y liberación continua hacia ambientes productivos.
- Utilizar la característica Azure Boards de Azure Devops para el seguimiento y gestión de proyectos ágiles, la cual permitirá tener trazabilidad de requerimientos y sus elementos, responsables y tiempo del Sprint.
- Migrar a la versión de SonarCloud la cual es una potente herramienta para análisis de código y vulnerabilidades de código, con esta migración obtendremos beneficios que van desde contar con la última versión de vulnerabilidades de código y buenas prácticas de la industria.
- Implementar Wiki en Azure Devops, de esta manera se podrá documentar los proyectos que sigan metodologías ágiles y podrán utilizar información relacionada a los elementos de trabajo, bugs, épicas, historias de usuario y cualquier información adicional que disponga la herramienta Azure Devops.



## 5. Arquitectura de infraestructura base

La arquitectura de Infraestructura base involucra la organización fundamental de un sistema de TI, que consiste en su hardware, software, tecnología de comunicaciones, las relaciones entre ellos y con el medio ambiente y los principios que rigen su diseño y evolución. («7. Phase D - Technology Architecture», s. f.)

El objetivo de la etapa de arquitectura de tecnología es desarrollar una arquitectura objetivo que habilite visión de arquitectura, la arquitectura objetivo de negocio, datos y aplicaciones se entreguen a través de componentes de tecnología y servicios de tecnología, de una manera que cubra la Declaración de trabajo de arquitectura y las inquietudes de las partes interesadas. (*The TOGAF® Standard, Version 9.2*, s. f.)

### 5.1. Arquitectura de Infraestructura actual

Actualmente la fábrica de software de la institución financiera cuenta con los siguientes equipos para sus ambientes no productivos:

Tabla 16

Características de equipos de la fábrica de software

		Características	Descripción
Nutanix	NX-	Memoria: 2048 GB	Hardware de Hiper convergencia de Nutanix, existen 4 en ambiente no productivo.
3170-G8		Procesadores: 32 cores (por socket)	
		Almacenamiento: 92 TB	

HP Elite Book 830	Memoria: 16 GB Procesadores: Intel Core i7 Almacenamiento: 1 TB	Laptop HP para uso del personal de la institución.
HP Pro Desk 800	Memoria: 16 GB Procesadores: Intel Core i7 Almacenamiento: 1 TB	Pc para equipo de desarrolladores externos.

De acuerdo a lo que se muestra en la tabla 16, se puede determinar que la infraestructura base se encuentra en un nivel de madurez bueno al usar tecnología hiperconvergente.

## 5.2. Arquitectura de Infraestructura objetivo

La infraestructura que será parte de la arquitectura de infraestructura base objetivo se la tomará con base a un referente del sector financiero como banco BBVA, el cual utiliza infraestructura en nube y contenedores tanto para ambientes productivos como los no productivos.

Actualmente la institución financiera cuenta con dos suscripciones de Azure por lo tanto se propone crear ambientes productivos y no productivos. A continuación, se muestran los pasos necesarios para lograr implementar correctamente la arquitectura de infraestructura base objetivo.

### 5.2.1. Gobierno de Suscripciones

Se recomienda utilizar las suscripciones de producción y de ambientes no productivos, en la ilustración 27 se muestran las dos suscripciones y el esquema propuesto para un correcto gobierno de los recursos.

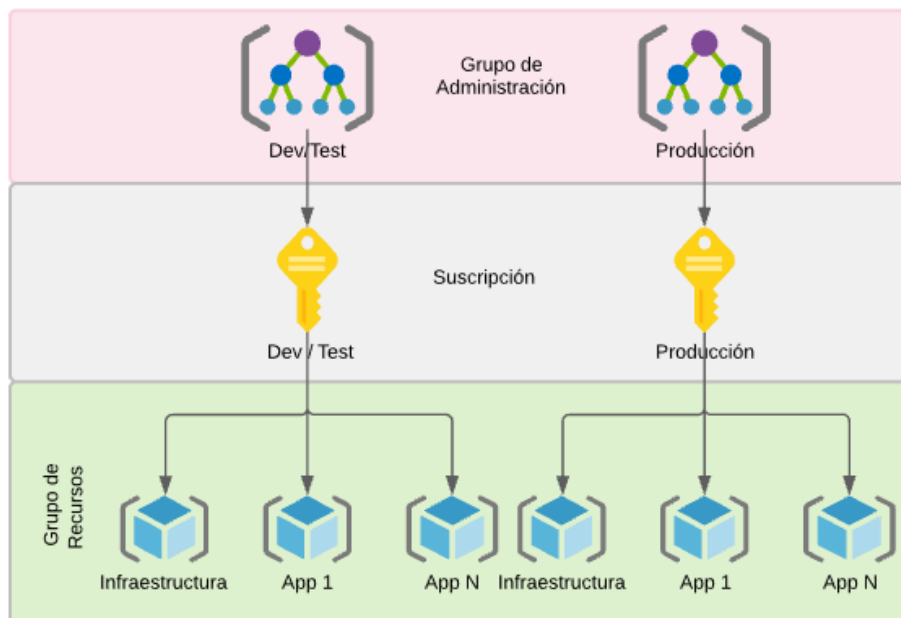


Ilustración 27 Suscripciones existentes y gobierno de recursos

Fuente: El autor

Como se puede evidenciar en la ilustración 27, actualmente la institución cuenta con dos suscripciones

- Producción, para grupos de recursos en ambiente productivos
- Dev/Test. Para ambientes no productivos, esto abarca los ambientes Testing y Desarrollo.

De igual forma se sugiere la creación de grupos de recursos específicos para infraestructura transversal, y otro conjunto de grupos de recursos relacionadas al ambiente de una aplicación específica, de tal manera que los recursos de una aplicación estén bajo el dominio de un grupo de recursos, esta organización aplica tanto para la suscripción de Producción como la de Dev/Test.

### 5.2.2. Grupos de seguridad

Se recomienda migrar los usuarios, grupos e identidades creados en el directorio activo de la organización a través del servicio Azure AD Connect de Azure de tal manera que estos usuarios estén sincronizados y seguros en la nube.

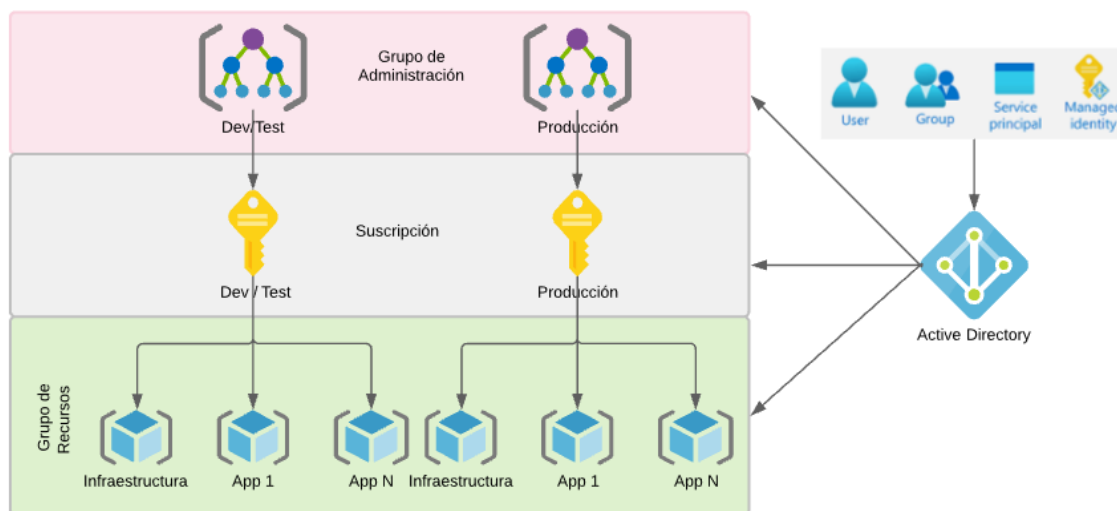


Ilustración 28 Usuarios, grupos, identidades sincronizadas con Azure AD Connect

Fuente: El autor

Como se muestra en la ilustración 28, utilizando el servicio de *Azure AD Connect* sincronizamos usuarios, grupos e identidades que necesitamos en la nube, de tal manera que en las suscripciones utilizaremos las identidades que requiramos sin exponer las credenciales que se encuentran *OnPremises*.

Adicionalmente se recomienda crear los siguientes grupos de seguridad en Azure AD para la correcta administración y auditoría de los recursos desplegados en ambas suscripciones.

- Administradores. Administración de permisos a los distintos grupos de recursos.
- Infraestructura. Creación y administración de recursos transversales.

- Seguridad. Permisos de lectura sobre todos los recursos de ambas suscripciones.
- Soporte. Permisos de lectura a grupos de recursos transversales y permisos creación y modificación sobre los grupos de recursos relacionados con las aplicaciones.

### 5.3. Análisis de Brechas

La tabla 17 muestra el análisis de brechas obtenido al comparar la arquitectura de infraestructura base actual y la objetivo, el estado actual de la infraestructura base se lo obtuvo de la tabla 16.

Tabla 17

Análisis de brechas de la arquitectura de infraestructura base

Procesos	Actual	Objetivo	Deseable
Ambiente de pruebas	3	4	5
Ambiente de producción	3	4	5

En la tabla 17 se puede evidenciar que la brecha entre la infraestructura base y actual no amplia. Sin embargo, es un punto de mejora al que se puede llegar sin mucho esfuerzo. La madurez de la infraestructura base se mide de acuerdo a la siguiente rubrica: nivel 1 (Infraestructura tradicional), nivel 2 (Infraestructura virtual), nivel 3 (Infraestructura hiper convergente), nivel 4 (Infraestructura en nube) y nivel 5 (Infraestructura en nube/contenedores).

A continuación, en la ilustración 29 se muestran las brechas identificadas de manera gráfica, tomando como insumo la tabla 17.

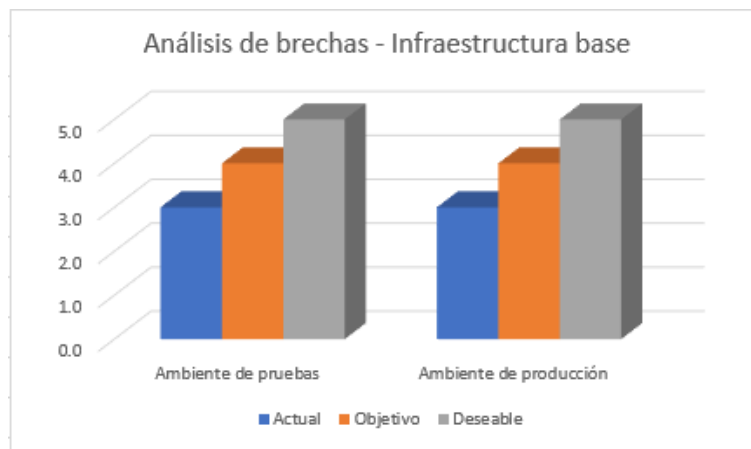


Ilustración 29 Brechas identificadas en la arquitectura de infraestructura base

Fuente: El Autor

En la ilustración 29 se puede determinar de manera rápida que la brecha entre la infraestructura base y objetivo no es amplia y por lo tanto fácil de alcanzar.

#### 5.4. Iniciativas para cerrar brechas

Una vez realizado el análisis de brechas entre la arquitectura de infraestructura actual y objetivo, se encontró una brecha relacionada al no aprovechamiento de los recursos de la nube que la institución posee.

- Utilizar los recursos de la nube de Microsoft para la creación de ambientes no productivos y exploración de nuevos productos.
- Crear un esquema de gobierno para administrar los recursos de nube en ambientes no productivos.
- Realizar una conexión entre la nube y el ambiente *OnPremises* actual a través de una VPN.

## 6. Oportunidades y soluciones

En este capítulo se consolida y se conceptualiza todas las iniciativas identificadas en etapas anteriores, con el objetivo de solucionar los problemas de la fábrica de desarrollo de la entidad financiera.

En la tabla 18 se muestran todas las iniciativas relacionadas a las arquitecturas de negocio, aplicaciones/datos e infraestructura base.

Tabla 18

Iniciativas por tipo de arquitectura

Arquitectura	Código	Iniciativa
<b>Negocio</b>	NEG-01	Fortalecer procesos ágiles de Scrum.
<b>Negocio</b>	NEG-02	Selección de personal para el equipo Scrum
<b>Negocio</b>	NEG-03	Capacitar en técnica de priorización MoSCow
<b>Negocio</b>	NEG-04	Implementar proceso de pruebas automáticas basadas en la pirámide de Cohn
<b>Aplicaciones</b>	APP-01	Migración Azure Devops Server
<b>Datos</b>	DAT-01	Implementar Wiki
<b>Infraestructura</b>	INF-01	Implementar ambientes productivos y no productivos en nube de Microsoft.

A continuación, se conceptualizan las iniciativas descritas en la tabla 18, de tal manera que muestre de manera general que se necesita para la implementación de cada iniciativa.

<b>Fortalecer procesos ágiles de Scrum.</b>			
<b>Código</b>	NEG-01	<b>Tipo</b>	Negocio
<b>Impacto</b>	Alto	<b>Esfuerzo</b>	Medio
<b>Costo</b>	\$40000	<b>Plazo</b>	< 6 meses
<b>Catalizadores COBIT</b>	Procesos Información Habilidades	<b>Roadmap</b>	Q1 2022

<b>PARTICIPANTES</b>			
Jefe de Mejora Continua (A)	Jefe de Innovación (R)	Proveedor (R)	Jefe de Aseguramiento de Calidad (C)
Gerente de Investigación y Desarrollo (I)			
<b>Situación actual y principales observaciones</b>			
Actualmente existe una brecha importante en los procesos de la metodología ágil, lo que provoca que los desarrollos tengan demoras y las entregas no sean incrementales.			
<b>Actividades fundamentales</b>			
<ul style="list-style-type: none"> <li>• Contratación consultoría. Es importante encontrar una consultoría experta en gestión ágil con Scrum, para guiar de manera correcta los procesos de la fábrica de desarrollo.</li> <li>• Capacitación. La Capacitación estará dirigida a las áreas afectadas dentro de la unidad de Servicios Tecnológicos y al área de transformación Digital.</li> <li>• Talleres. Realizar talleres para que los conceptos impartidos en la capacitación se pueden evidenciar de manera real.</li> <li>• Crear MVP, es importante tomar 1 o 2 proyectos para iniciar con todos los procesos de la metodología ágil con acompañamiento del Consultor.</li> </ul> <p>En este proyecto se necesita el trabajo el colaborativo de todas las áreas participantes y sus distintos roles.</p>			

<b>Selección de personal para el equipo Scrum</b>			
<b>Código</b>	NEG-02	<b>Tipo</b>	Negocio
<b>Impacto</b>	Medio	<b>Esfuerzo</b>	Bajo
<b>Costo</b>	\$10000 mensual	<b>Plazo</b>	< 6 meses
<b>Catalizadores COBIT</b>	Habilidades	<b>Roadmap</b>	Q2 y Q3 2022
<b>PARTICIPANTES</b>			
Gerente de Investigación y Desarrollo (I)	Jefe de Mejora Continua (R)	Jefe de Innovación (R)	Jefe de Aseguramiento de Calidad (R)
<b>Situación actual y principales observaciones</b>			
Actualmente existe una brecha importante referente a los roles necesarios para el correcto funcionamiento de la metodología ágil, de tal manera que los procesos ágiles no están formalizados y en algunos casos se los hace de manera empírica.			



<b>Actividades fundamentales</b>
<p>Crear perfiles a solicitar. Es importante detallar los perfiles a solicitar al igual que las habilidades necesarias.</p> <p>Realizar convocatoria. Se debe realizar la convocatoria abierta, para evaluar los distintos roles.</p> <p>Para la realización de esta iniciativa es importante poder obtener el financiamiento para los futuros roles que ingresaran a la institución</p>

<b>Capacitar en técnica de priorización MoSCow</b>			
<b>Código</b>	NEG-03	<b>Tipo</b>	Negocio
<b>Impacto</b>	Medio	<b>Esfuerzo</b>	Bajo
<b>Costo</b>	\$5000 mensual	<b>Plazo</b>	< 2 meses
<b>Catalizadores COBIT</b>	Tecnología Información	<b>Roadmap</b>	Q2 2022
<b>PARTICIPANTES</b>			
Gerente de Transformación Digital	Product Owner (C/I)	Proveedor (R)	
<b>Situación actual y principales observaciones</b>			
<p>Actualmente existe una brecha relacionada con la priorización de iniciativas e historias de usuario, por lo que no se entrega el valor esperado a la institución financiera, relacionada a los productos de software desarrollados por la fábrica de software.</p>			
<b>Actividades fundamentales</b>			
<ul style="list-style-type: none"> <li>• Buscar Consultor. Es importante buscar un consultor experto en priorización.</li> <li>• Capacitar personal. En la capacitación de priorización deben participar las personas con el rol de Product Owner.</li> </ul> <p>Para la correcta realización de este proyecto es necesaria la colaboración y proactividad de las personas con rol Product Owner.</p>			

<b>Implementar proceso de pruebas automáticas basadas en la pirámide de Cohn</b>			
<b>Código</b>	NEG-04	<b>Tipo</b>	Negocio
<b>Impacto</b>	Medio	<b>Esfuerzo</b>	Alto
<b>Costo</b>	\$20000	<b>Plazo</b>	< 6 meses
<b>Catalizadores COBIT</b>	Procesos Tecnología	<b>Roadmap</b>	Q3 y Q4 2022
<b>PARTICIPANTES</b>			
Jefe de Aseguramiento de Calidad (A)	Analista Calidad (C)	Proveedor (R)	Ingeniero Devops (C)
<b>Situación actual y principales observaciones</b>			
Actualmente existe una brecha importante relacionada con el proceso y gestión de pruebas automáticas, ya que actualmente estas pruebas se las realiza de manera manual sin estar apoyadas en un modelo de gobierno que ordene es estructure los procesos y procedimientos del área de Aseguramiento de Calidad.			
<b>Actividades fundamentales</b>			
<ul style="list-style-type: none"> <li>• Buscar Consultor. Buscar consultoría experta en gestión de pruebas automáticas.</li> <li>• Definir procesos a realizar. Es importante definir la hoja de ruta evolutiva que se seguirá en la implementación de este proyecto.</li> <li>• Capacitar personal. Se debe capacitar a todo el personal de Aseguramiento de Calidad.</li> <li>• Talleres. Crear talleres donde se muestre de manera practica los distintos tipos de pruebas</li> <li>• Crear MVP. Es importante crear varios escenarios de prueba y en sus distintos tipos con el área de Aseguramiento de Calidad.</li> </ul>			
Para la realización de este proyecto es importante la participación proactiva y colaborativa del área de Aseguramiento de Calidad con las demás áreas y con el Consultor.			

<b>Migración Azure Devops Server</b>			
<b>Código</b>	APP-01	<b>Tipo</b>	Negocio
<b>Impacto</b>	Alto	<b>Esfuerzo</b>	Bajo
<b>Costo</b>	\$2000 mensual	<b>Plazo</b>	< 2 meses
<b>Catalizadores COBIT</b>	Procesos Tecnología	<b>Roadmap</b>	Q4 2021
<b>PARTICIPANTES</b>			

Jefe de Control de Versiones (A)	Ingeniero Devops (R)	Proveedor (R)	Arquitectos de Solución (C)
<b>Situación actual y principales observaciones</b>			
Actualmente existe una brecha importante al no disponer de la versión en nube de Azure Devops para brindar a la fábrica de software capacidades actuales de integración continua, entrega y despliegue, gestión de la configuración, trazabilidad de requerimientos.			
<b>Actividades fundamentales</b>			
<ul style="list-style-type: none"> <li>▪ Capacitación: Nuevas características de Azure Devops. (Desarrolladores, QAs, Coordinadores, Jefaturas)</li> <li>▪ Solicitar soporte a Microsoft.</li> <li>▪ Sincronizar directorio activo.</li> <li>▪ Respalidar Base de Datos de gestores de versionamiento.</li> <li>▪ Migrar base de datos de los gestores de versionamiento actuales.</li> </ul> <p>Este proyecto requiere el trabajo colaborativo de las áreas de negocio, Tecnología para implementar correctamente la herramienta. Como parte del proyecto se incluye distintos tipos de capacitación. Relacionados a las características de Azure Devops</p>			

<b>Implementar Wiki</b>			
<b>Código</b>	DAT-01	<b>Tipo</b>	Negocio
<b>Impacto</b>	Bajo	<b>Esfuerzo</b>	Bajo
<b>Costo</b>	\$0	<b>Plazo</b>	< 2 meses
<b>Catalizadores COBIT</b>	Tecnología	<b>Roadmap</b>	Q4 2022
<b>PARTICIPANTES</b>			
Jefe de Control de Versiones (A)	Ingeniero Devops (R)	Proveedor (R)	Líder Técnico (C)
<b>Situación actual y principales observaciones</b>			
Actualmente existe una brecha relacionada a la documentación de los proyectos ágiles, esta brecha esta relacionada a la no capacidad de documentar el avance del proyecto y sus entregables.			
<b>Actividades fundamentales</b>			

- Crear Proyecto Base: EN este proyecto se creará la documentación para todos los proyectos desarrollados y por desarrollarse.
- Asignar Permisos en Wiki de Azure Devops
- Capacitación: Equipo de desarrollo, Scrum Masters, Ingenieros Devops, Analistas de Calidad.

Este proyecto requiere el trabajo operativo del equipo de Devops y el trabajo colaborativo de los demás roles participantes en la metodología ágil SCRUM.

<b>Implementar ambientes productivos y no productivos en nube de Microsoft.</b>			
<b>Código</b>	INF-01	<b>Tipo</b>	Negocio
<b>Impacto</b>	Alto	<b>Esfuerzo</b>	Medio
<b>Costo</b>	\$5000 mensual	<b>Plazo</b>	< 6 meses
<b>Catalizadores</b>	Procesos	<b>Roadmap</b>	Q1 y Q2 2022
<b>COBIT</b>	Tecnología		
<b>PARTICIPANTES</b>			
Jefe de Control de Versiones (A)	Ingeniero Devops (R)	Proveedor (R)	Arquitectos de Solución (C)
<b>Situación actual y principales observaciones</b>			
Actualmente existe una brecha relacionada a la infraestructura base utilizada en la fabrica de software, y es no poder aprovechar las capacidades de la nube al tener una suscripción empresarial y no utilizarla.			
<b>Actividades fundamentales</b>			
<ul style="list-style-type: none"> <li>• Dividir suscripciones. Se debe asignar una suscripción para ambientes productivos y otra para ambientes no productivos.</li> <li>• Crear Landing Zones. El objetivo de estas Landing Zones es que los recursos tecnológicos a crearse tengan un modelo esperado donde hospedarse (IaaS, PaaS)</li> <li>• Crear modelo autorización, Importante crear el árbol de permisos para las distintas suscripciones y los roles.</li> </ul>			
Este proyecto requiero un esfuerzo de varias áreas, incluyendo Seguridad, Infraestructura, Investigación y Desarrollo, para lograr de manera colaborativo los ambientes productivos y no productivos con un esquema de gobierno sencillo y ágil.			

## 7. Plan de migración

En este capítulo se realiza la priorización de las iniciativas, de acuerdo a dos variables que son el impacto del proyecto y el esfuerzo que representa su implementación.

### 7.1. Análisis de impacto

Para realizar el análisis de impacto se creó una matriz de correlación entre los objetivos de la fábrica de software y las iniciativas detalladas en el capítulo previo. La rúbrica que se utilizó en este ejercicio fue medir el impacto de cada iniciativa sobre los objetivos con la escala Bajo, Medio y Alto impacto como se logra muestra en la ilustración 30.

No.	Arquitectura	ID	Iniciativa	Objetivos							Valoración Cualitativa	Impacto	
				Sostenibilidad de Servicios de TI	Plataformas de TI ágiles y automatizadas	Desarrollos estables y sin errores	Integración y despliegue continuo	Automatización de pruebas	Mejorar la documentación de los proyectos y aplicaciones	Fortalecer procesos de priorización de Historia de Usuario			
1	Negocio	INI-01	Fortalecer procesos ágiles de Scrum.	Medio	Medio	Medio	No Aplica	Bajo	Medio	Medio	Medio	1.8	Medio
2	Negocio	INI-02	Selección de personal para el equipo Scrum	No Aplica	Alto	Medio	No Aplica	Alto	Alto	Medio	Medio	2.6	Alto
3	Negocio	INI-03	Capacitar en técnica de priorización MoSCow	Medio	Alto	Medio	No Aplica	No Aplica	Alto	Alto	Alto	2.6	Alto
4	Negocio	INI-04	Implementar pruebas automatizadas basadas en la pirámide de Cohn	Alto	Alto	Medio	Medio	Alto	Medio	Medio	No Aplica	2.5	Alto
5	Aplicación/Datos	INI-05	Migración Azure Devops Server a Azure Devops	Alto	Alto	Medio	Alto	Alto	Medio	Medio	No Aplica	2.7	Alto
6	Aplicación/Datos	INI-06	Implementar Wiki	Medio	Medio	Medio	No Aplica	No Aplica	Alto	No Aplica	No Aplica	2.3	Medio
7	Infraestructura	INI-07	Implementar ambientes productivos y no productivos en nube de Microsoft.	Alto	Alto	Medio	Alto	Bajo	Bajo	No Aplica	No Aplica	2.2	Medio

Ilustración 30 Análisis de impacto (Iniciativas vs Objetivos)

Fuente: El Autor

Como se ilustra en la tabla anterior se logra medir el impacto de las iniciativas relacionados con los objetivos de la fábrica de software, de tal manera que el impacto se mide de la siguiente forma: Bajo (0 - 1.5) Medio (1.6 - 2.3) Alto (2.4 - 3.0)

## 7.2. Análisis de esfuerzo

Para realizar el análisis de esfuerzo se consideraron tres criterios: Recursos económicos, complejidad, Capacidad de TI. La rúbrica para los criterios de esfuerzo se muestra en la ilustración 31.

Rúbrica			40%	30%	30%	Suma Ponderada	Esfuerzo
Recursos Económicos	Complejidad	Capacidad TI	Recursos Económicos	Complejidad	Capacidad TI		
● < \$10k	● Bajo	● Alto					
● \$10k - \$40k	● Medio	● Medio					
● > \$40K	● Alto	● Bajo					

No.	Arquitectura	ID	Iniciativa	Criterios Esfuerzo			Suma Ponderada	Esfuerzo
1	Negocio	INI-01	Fortalecer procesos ágiles de Scrum.	●	●	●	2.3	Medio
2	Negocio	INI-02	Selección de personal para el equipo Scrum	●	●	●	3.0	Bajo
3	Negocio	INI-03	Capacitar en técnica de priorización MoSCow	●	●	●	2.7	Bajo
4	Negocio	INI-04	Implementar pruebas automatizadas basadas en la pirámide de Cohn	●	●	●	1.3	Alto
5	Aplicación/Datos	INI-05	Migración Azure Devops Server a Azure Devops	●	●	●	2.3	Medio
6	Aplicación/Datos	INI-06	Implementar Wiki	●	●	●	3.0	Bajo
7	Infraestructura	INI-07	Implementar ambientes productivos y no productivos en nube de Microsoft.	●	●	●	2.3	Medio

Ilustración 31 Análisis de esfuerzo (Iniciativas vs Criterios de esfuerzo)

Fuente: El Autor

Como se ilustra en la tabla anterior se logra analizar el esfuerzo para la implementación de las iniciativas tomando como factores de evaluación los criterios: recursos económicos, complejidad, capacidad de TI, y el resultado del análisis de esfuerzo se muestra en la rúbrica de la siguiente forma: Bajo (0 - 1.5) Medio (1.6 - 2.3) Alto (2.4 - 3.0)

## 7.3. FASES

Se combina los resultados obtenidos en el análisis de impacto y el análisis de esfuerzo de las secciones 7.1 y 7.2 respectivamente, lo cual permitió priorizar y dividir las iniciativas en fases de acuerdo a su importancia.

				Rúbrica									
				Alto		Habilitante		Bajo					
				Impacto	Esfuerzo	Impacto	Esfuerzo	Impacto	Esfuerzo	Impacto	Esfuerzo	Prioridad	Fase
				Alto	Bajo	Alto	Alto	Medio	Alto				
				Alto	Medio	Medio	Medio	Bajo	Bajo				
						Medio	Bajo	Bajo	Medio				
								Bajo	Alto				

No.	No. Arquitectura	ID	Iniciativa	Impacto	Esfuerzo	Prioridad	Fase	
1	1	Negocio	INI-01	Fortalecer procesos ágiles de Scrum.	Medio	Medio	Habilitante	4
2	2	Negocio	INI-02	Selección de personal para el equipo Scrum	Alto	Bajo	Alta	1
3	3	Negocio	INI-03	Capacitar en técnica de priorización MoSCow	Alto	Bajo	Alta	1
4	4	Negocio	INI-04	Implementar pruebas automatizadas basadas en la pirámide de Cohn	Alto	Alto	Habilitante	3
5	5	Aplicación/Datos	INI-05	Migración Azure Devops Server a Azure Devops	Alto	Medio	Alta	2
6	6	Aplicación/Datos	INI-06	Implementar Wiki	Medio	Bajo	Habilitante	3
7	7	Infraestructura	INI-07	Implementar ambientes productivos y no productivos en nube de Microsoft.	Medio	Medio	Habilitante	4

Ilustración 32 Análisis de esfuerzo (Iniciativas vs Criterios de esfuerzo)

Fuente: El autor

Como se muestra en la ilustración 32 se logra determinar la prioridad de las iniciativas de acuerdo a su impacto y el esfuerzo en su implementación, resultando cuatro grupos de iniciativas organizadas de la siguiente forma.

- Fase 1: Migración Azure Devops Server a Azure Devops.
- Fase 2: Fortalecer procesos ágiles de Scrum.
- Fase 2: Implementar ambientes productivos y no productivos en nube de Microsoft.
- Fase 3: Selección de personal para el equipo Scrum.
- Fase 3: Capacitar en técnica de priorización MoSCow.
- Fase 4: Implementar pruebas automatizadas basadas en la pirámide de Cohn.
- Fase 5: Implementar Wiki

## 7.4. Roadmap

A continuación, en la ilustración 33 se muestra la hoja de ruta recomendada para la implementación de las iniciativas de acuerdo a su priorización.

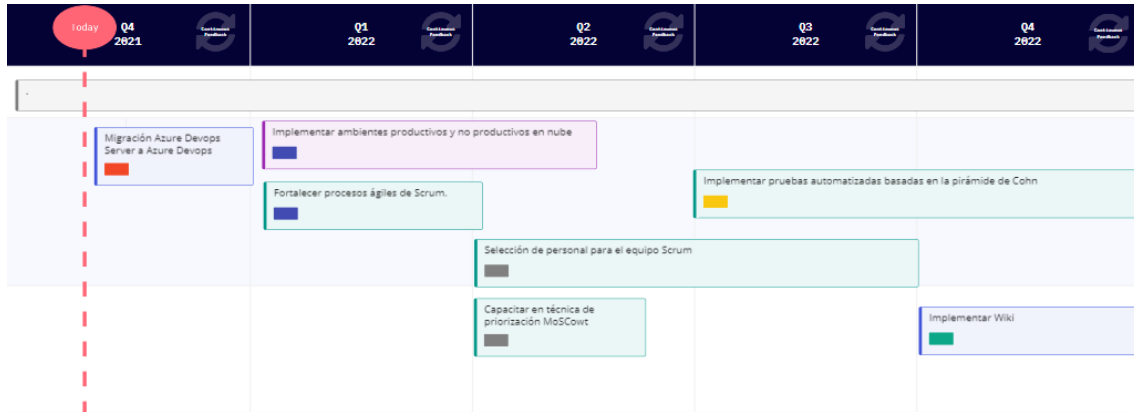


Ilustración 33 Hoja de ruta de iniciativas priorizadas

Fuente: El autor



## CONCLUSIONES

- Las metodologías ágiles permiten crear y entregar productos de software que agreguen valor a la organización, de tal manera que a través de la implantación de Scrum se logra sentar las bases para entrega de manera continua producto.
- Microsoft Azure es una de los principales proveedores de nube pública del mundo, ampliar los servicios que dispone la institución a través de la nube nos permite tener resiliencia, flexibilidad, seguridad.
- Las prácticas de Devops permiten generar seguridad, estabilidad, integración, monitoreo de las distintas aplicaciones gestionadas a través de estas herramientas, en ese sentido Azure Devops apoya y fortalece estos objetivos.
- La priorización de iniciativas es un factor fundamental al momento de entender que iniciativas generan mayor valor a la organización, de tal manera que usar la técnica MoSCow nos permite tener una visión centrada en el valor que estas iniciativas puedan generar.

## RECOMENDACIONES

- Implementar prácticas de arquitectura empresarial en la institución financiera.
- Implementar un modelo de gobierno en nube para generar lineamientos y guías de las futuras cargas de trabajo y aplicaciones que se hospeden en nube.
- Adaptar marcos de trabajo específicos para mejorar los procesos internos y prácticas relacionadas al desarrollo de software (CMMI DEV, ISTQB, Safe).
- Pensar en la estrategia nube primero (Cloud First) de tal manera que se aproveche las ventajas de la nube integrada con el CORE de la institución financiera.

## Referencias

7. Phase D - Technology Architecture. (s. f.). *Visual Paradigm Community Circle*. Recuperado 24 de octubre de 2021, de <https://circle.visual-paradigm.com/docs/togaf-adm-guide-through/phase-d-technology-architecture/>
- 7 principios de arquitectura que todo arquitecto empresarial debe conocer. (2021, julio 20). *Conexiam*. <https://conexiam.com/es/7-architecture-principles-every-enterprise-architect-should-know/>
- 2013—*A guide to the Scrum Body of knowledge (SBOK Guide.pdf)*. (s. f.). *A guide to the Scrum Body of knowledge (SBOK Guide) (2013 edition)*. (2013). SCRUMstudy, A brand of VMEdu, Inc.
- Administración de cartera de proyectos y de recursos empresariales*. (s. f.). Recuperado 16 de septiembre de 2021, de <https://www.microsoft.com/es-ww/microsoft-365/project/enterprise-project-server>
- Agile Teams—Scaled Agile Framework*. (s. f.). Recuperado 28 de agosto de 2021, de <https://www.scaledagileframework.com/agile-teams/>
- anandmeg. (s. f.). *Información general sobre Visual Studio*. Recuperado 16 de septiembre de 2021, de <https://docs.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide>
- Artefactos Scrum: Las 3 herramientas clave de gestión*. (s. f.). Recuperado 30 de octubre de 2021, de <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>
- Automatice las pruebas funcionales desde la IU hasta la API | UFT One | Micro Focus*. (s. f.). Recuperado 16 de septiembre de 2021, de <https://www.microfocus.com/en-us/products/uft-one/overview>
- Azure DevOps Server | Microsoft Azure*. (s. f.). Recuperado 16 de septiembre de 2021, de <https://azure.microsoft.com/es-es/services/devops/server/>
- BBVA. (2020, junio 25). Los nuevos procesos de desarrollo de ‘software’ garantizan la calidad de servicio de BBVA durante la crisis sanitaria. *BBVA NOTICIAS*. <https://www.bbva.com/es/es/los-nuevos-procesos-de-desarrollo-de-software-garantizan-la-calidad-de-servicio-de-bbva-durante-la-crisis-sanitaria/>

*BBVA IT | DevOps, la metodología que está revolucionando los procesos de desarrollo.* (s. f.). Recuperado 18 de abril de 2021, de <https://www.bbvaitspain.com/devops-la-metodologia-que-esta-revolucionando-los-procesos-de-desarrollo/>

*COBIT-2019-Framework-Governance-and-Management-Objectives\_res\_Spa\_0519.pdf.* (s. f.).

*¿Cuáles son los 3 pilares de Scrum y para qué sirven? - Blog de InGenio Learning.* (s. f.). Recuperado 30 de octubre de 2021, de <https://ingenio.edu.pe/blog/cuales-son-los-3-pilares-de-scrum-y-para-que-sirven/>

*Data Migration Tool for Azure DevOps Migration Guide.pdf.* (s. f.).

*DevOps is a culture, not a role!. Software is everywhere. In today's... | by Irma Harlann | Medium.* (s. f.). Recuperado 27 de abril de 2021, de

<https://neonrocket.medium.com/devops-is-a-culture-not-a-role-be1bed149b0>

*DevOps Maturity- Steps To Help You Level Up in Salesforce—Plumlogix.* (2020, mayo 14). *Plumlogix Inc.* <https://plumlogix.com/devops-maturity-for-salesforce/>

*Gestión de la configuración—Praxis Framework.* (s. f.). Recuperado 8 de septiembre de 2021, de

<https://www.praxisframework.org/es/knowledge/configuration-management>

*GTI8.pdf.* (s. f.). Recuperado 9 de septiembre de 2021, de

<https://cedia.edu.ec/dmdocuments/publicaciones/Libros/GTI8.pdf>

Hernández Yeja, A., & Porven Rubier, J. (2016). Procedimiento para la seguridad del proceso de despliegue de aplicaciones web. *Revista Cubana de Ciencias Informáticas*, 10(2), 42-56.

*Las 5 ceremonias Scrum: Claves para la gestión de procesos.* (s. f.).

Recuperado 30 de octubre de 2021, de

<https://www2.deloitte.com/es/es/pages/technology/articles/ceremonias-scrum.html>

*Mejores prácticas de testing para equipos ágiles: La Pirámide de Automatización | Abstracta Chile.* (2020, mayo 30). *Blog de Testing y Calidad*

*de Software | Abstracta Chile*. <http://cl.abstracta.us/blog/piramide-de-automatizacion/>

Método MoSCoW: Para establecer requisitos por orden de.... (2019, septiembre 25). *Toolshero*. <https://www.toolshero.es/administracion/metodo-moscow/>

*MoSCoW. ¿Qué es y cómo priorizar en el desarrollo de tu aplicación?* (s. f.). Recuperado 31 de octubre de 2021, de <https://www.itdo.com/blog/moscow-que-es-y-como-priorizar-en-el-desarrollo-de-tu-aplicacion/>

*Phase A: Architecture Vision*. (s. f.). Recuperado 8 de noviembre de 2021, de <https://pubs.opengroup.org/architecture/togaf91-doc/arch/chap07.html>

*pmoinformatica.com*, P. por. (s. f.). *¿Qué es la matriz de trazabilidad de requerimientos?* Recuperado 8 de septiembre de 2021, de <http://www.pmoinformatica.com/2018/07/que-es-matriz-trazabilidad-requerimientos.html>

*profexorgeek*. (s. f.). *¿Qué es Xamarin? - Xamarin*. Recuperado 16 de septiembre de 2021, de <https://docs.microsoft.com/es-es/xamarin/get-started/what-is-xamarin>

*Pruebas de carga en la nube | Performance Testing en la nube | Micro Focus*. (s. f.). Recuperado 16 de septiembre de 2021, de <https://www.microfocus.com/en-us/products/loadrunner-cloud/overview>

*Pruebas unitarias—El pequeño libro de ASP.NET Core*. (s. f.). Recuperado 16 de septiembre de 2021, de <https://aspnetcoremaster.com/little-aspnetcore-book/chapters/automated-testing/unit-testing.html>

*¿Qué hace un ingeniero de DevOps?* (s. f.). Recuperado 31 de agosto de 2021, de <https://www.redhat.com/es/topics/devops/devops-engineer>

Salamon, A., Maller, P. A., Boggio, A., Mira, N., Pérez, S., & Coenda, F. (2014). *La integración continua aplicada en el desarrollo de software en el ámbito científico-técnico*. XX Congreso Argentino de Ciencias de la Computación (Buenos Aires, 2014). <http://sedici.unlp.edu.ar/handle/10915/42358>

*Scrum Process Model as part of agile software development methodology | LinkedIn*. (s. f.). Recuperado 27 de abril de 2021, de

<https://www.linkedin.com/pulse/scrum-process-model-part-agile-software-development-shahab-mohammed/>

*SFIA y la gestión de habilidades*. (s. f.). [Página]. SFIA. Recuperado 19 de abril de 2021, de <https://sfia-online.org/es/about-sfia/sfia-and-skills-management/Sfiaref.7.20180729.en.pdf>. (s. f.).

SonarQube: Una herramienta útil para verificar la calidad del código -. (2020, septiembre 7). *Castor Transformación Digital*. <https://castor.com.co/sonarqube-una-herramienta-util-para-verificar-la-calidad-del-codigo/>

*The benefits of using CMM (Capability Maturity Model) as a prelude for Process Improvement transformation | LinkedIn*. (s. f.). Recuperado 27 de abril de 2021, de <https://www.linkedin.com/pulse/benefits-using-cmm-capability-maturity-model-prelude-process-barreto/>

*The TOGAF® Standard, Version 9.2*. (s. f.). Recuperado 24 de octubre de 2021, de <https://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html>

*The TOGAF Standard, Version 9.2—Phase C: Information Systems Architectures—Application Architecture*. (s. f.). Recuperado 3 de noviembre de 2021, de <https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap10.html>

User, S. (s. f.). *OVERTI*. OVERTI. Recuperado 31 de octubre de 2021, de <http://www.overti.es/>

## **ANEXOS**

Cat.	Nivel 1	Nivel 2	Nivel 3	Nivel 4	Nivel 5
Vigencia tecnológica	Obsolescencia: Tecnologías descontinuadas hace más de 10 años.	Rezagada: Tecnologías que ya no son vigentes hace más de 7 años.	Conservadora: Tecnología vigente en el mercado, 3 años de posicionamiento.	Vigente: Los proveedores efectúan innovaciones tecnológicas en todos los dominios del stack de tecnologías.	Vanguardista: Se enfocan en tendencias que serán las nuevas generaciones tecnológicas en los próximos años
Interoperabilidad	No existe: No provee ningún esquema de interoperabilidad.	Básica: Ofrece servicios básicos de consulta de información.	Colaborativa: Ofrece consulta ya actualización de información segura y confiable.	Integrada: Interoperabilidad bidireccional, trazable, seguro, administrado.	Unificada. Arquitectura SOA, catálogos de servicios, integración con Bus de servicios.
Confiabilidad	Incipiente: Incidentes recurrentes y poca consistencia de la aplicación.	Básica: Niveles de servicios razonables. No más de 10 incidentes en el año.	Consistente: Niveles de servicio estables, no más de 5 incidentes al año.	Gestionada: Niveles de servicio muy buenos, no más de 2 incidentes al año.	Optimizada: Niveles de servicio excepcionales, mejora continua.
Arquitectura	Incipiente: No existe documentación de la aplicación, ni soporte técnico.	Básica: Dispone de documentación básica, existen pocas personas que conozcan la arquitectura.	Establecida: Documentación completa y actualizada. Documentación representa arquitectura actual.	Gestionada: Documentación completa y actualizada de todos los componentes en detalle.	Optimizada: Documentación completa y actualizada de todos los componentes en detalle en herramienta, existe análisis de impacto.
Soporte	Incipiente. No existe soporte.	Básico: Existe soporte que atiende incidentes de manera correctiva.	Establecido: Soporte técnico y funcional, y con niveles de servicio.	Gestionado: Soporte técnico y funcional, y con niveles de servicio atendidos por una herramienta de HelpDesk.	Optimizado: Aplicación muy con soporte técnico y funcional, excelentes niveles de servicio gestionado con herramientas de HelpDesk y mejora continua.
Seguridad	Incipiente: No existe seguridad en la aplicación.	Básico: Dispone de algunos componentes de seguridad.	Establecido: Dispone de varios componentes de seguridad, análisis de vulnerabilidades.	Gestionado: Dispone de varios componentes de seguridad, análisis de vulnerabilidades. Reportes de seguridad bimestrales.	Optimizado: Aplicación cumple con buenas practicas de seguridad y desarrollo seguro como OWASP, análisis de vulnerabilidad recurrentes.



Flexibilidad / Mantenimiento	Inflexible. No se puede personalizar o mejorar.	Básica: Es muy costoso evolucionar la aplicación.	Establecida: Es viable su evolución a precios razonables, esto incluye cambios en código y componentes.	Gestionada: La aplicación evoluciona constantemente.	Optimizada: Aplicación en modelo SaaS (Software como servicio), siempre esta actualizada.
Gobernabilidad	Inexistente. No existe modelo de gobierno	Básica: No existe formalmente un modelo de gobierno.	Establecida: Existe un modelo de gobierno documentado.	Gestionado: Modelo formal de gobierno, documentado y actualizado.	Optimizado: Modelo formal de gobierno, documentado, actualizado y con procesos de mejora continua.

