



FACULTAD DE POSTGRADOS

FORTALECIMIENTO DE LA FÁBRICA DE SOFTWARE EN UNA EMPRESA QUE
PROVEE SOLUCIONES PARA EL SECTOR FINANCIERO

AUTOR

MARCO ANTONIO GUAÑUNA LINCANGO

AÑO

2022



FACULTAD DE POSTGRADOS

TEMA

FORTALECIMIENTO DE LA FÁBRICA DE SOFTWARE EN UNA
EMPRESA QUE PROVEE SOLUCIONES PARA EL SECTOR
FINANCIERO

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Magister en Gerencia de
Sistemas y Tecnología Empresarial.

TUTOR

GERMÁN ERNESTO PANCHO CARRERA

AUTOR

MARCO ANTONIO GUAÑUNA LINCANGO

AÑO

2022

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido el trabajo, Fortalecimiento de la fábrica de software en una empresa que provee soluciones para el sector financiero, a través de reuniones periódicas con el estudiante Marco Antonio Guañuna Lincango en los talleres de trabajo capstone, orientando sus conocimientos y competencias para un adecuado desarrollo del tema seleccionad, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

Germán Ernesto Pancho Carrera
CI. 0601918253

DECLARACIÓN DE AUTORÍA

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

A handwritten signature in blue ink, appearing to read 'Marco Antonio Guañuna Lincango', enclosed within a blue oval scribble.

Marco Antonio Guañuna Lincango
CI. 1713270963

AGRADECIMIENTOS

A Dios, por su infinita bondad, amor y amparo en los momentos difíciles, además, por haberme dado salud y la provisión para lograr mis objetivos.

A Don José Alabarda Ortega y a su esposa Zhéfora, quienes inculcaron en mi infancia valores indispensables para mi crecimiento personal.

A mis maestros, quienes me brindaron su conocimiento, su experiencia y sobre todo su amistad.

DEDICATORIA

A mis padres, Luz María y Manuel de Jesús por siempre brindarme su confianza e impulso incondicional para poder cumplir con todos los proyectos emprendidos.

A mi esposa, por su comprensión y paciencia, siendo compañera y apoyo en cada jornada.

A mi hijo, Ariel Alejandro, importante motivación y aliciente para cumplir a cabalidad con el objetivo que hace tiempo me había propuesto.

RESUMEN

El presente estudio, está estructurado en ocho capítulos que describen de forma secuencial el procedimiento para el análisis integral del estado de una fábrica de software, en una empresa que tiene por objetivo proveer aplicaciones seguras y de calidad, que contribuyan al desarrollo de la innovación en las instituciones financieras de diferentes tipos y tamaños mediante un amplio catálogo de soluciones que pueden ser desplegadas en la infraestructura del cliente o incluso en la nube, por profesionales capacitados que hacen uso de las herramientas y tendencias tecnológicas más actuales.

La metodología utilizada es ADM propuesta por TOGAF que consta de una fase preliminar y otras seis fases que se ejecutan de forma cíclica e iterativa que permiten identificar las arquitecturas Base y Objetivo y que se adaptan según la realidad de la empresa y la complejidad del caso, que se complementan con marcos de referencia internacionales para incorporar las buenas prácticas.

Finalmente, y como resultado se propone un mapa de ruta donde se establecen soluciones para mejorar la gestión y la capacidad del área de ingeniería en las perspectivas de personas, procesos, aplicaciones y tecnología, iniciativas analizadas considerando el impacto en la organización, el esfuerzo requerido para su ejecución y la prioridad definida conforme su aporte al cumplimiento de los objetivos empresariales.

ABSTRACT

This study is structured in eight chapters that sequentially describe the procedure for the integral analysis of the state of a software factory, in a company that aims to provide secure and quality applications that contribute to the development of innovation in financial institutions of different types and sizes through a wide range of solutions that can be deployed in the client's infrastructure or even in the cloud, by trained professionals who make use of the most current tools and technological trends.

The methodology used is ADM proposed by TOGAF, which consists of a preliminary phase and six other phases that are executed in a cyclical and iterative way that allow identifying the Base and Target architectures and that are adapted according to the reality of the company and the complexity of the case, which are complemented with international reference frameworks to incorporate good practices.

Finally, and as a result, a roadmap is proposed where solutions are established to improve the management and capacity of the engineering area in the perspectives of people, processes, applications and technology, initiatives analyzed considering the impact on the organization, the effort required for its execution and the priority defined according to its contribution to the fulfillment of the business objectives.

Índice del contenido

1.	FASE PRELIMINAR	1
1.1.	DESCRIPCIÓN GENERAL DE LA ORGANIZACIÓN	1
1.1.1.	Organigrama.....	2
1.1.2.	Misión.....	2
1.1.3.	Visión	3
1.1.4.	Principios.....	3
1.1.5.	Valores	3
1.2.	ANÁLISIS DEL CONTEXTO	3
1.3.	PROBLEMÁTICA	4
1.4.	ORGANIZACIÓN IMPACTADA.....	6
1.5.	STAKEHOLDERS.....	6
1.5.1.	Expectativas de Valor	6
1.5.2.	Estrategia de Comunicación.....	7
1.6.	BMM	7
1.7.	MARCOS DE REFERENCIA Y ESTÁNDARES	8
1.8.	EQUIPO DE ARQUITECTURA	8
1.9.	CATÁLOGO DE PRINCIPIOS	10
2.	VISIONAMIENTO ARQUITECTÓNICO	11
2.1.	REQUERIMIENTOS DE ALTO NIVEL.....	11
2.2.	VISIONAMIENTO Y ESCENARIOS DE LA SOLUCIÓN.....	12
2.3.	ANÁLISIS DE BRECHAS	17
2.4.	DEFINICIÓN DE LA ARQUITECTURA OBJETIVO	19
2.4.1.	Arquitectura de negocio objetivo	19
2.4.2.	Arquitectura de datos objetivo	20
2.4.3.	Arquitectura de aplicación objetivo	21
2.4.4.	Arquitectura de infraestructura base	22
2.4.5.	Road Map.....	23
3.	ARQUITECTURA DE NEGOCIO	25
3.1.	ARQUITECTURA DE NEGOCIO OBJETIVO	25
3.1.1.	Procesos asociados a la estimación de software.....	25
3.1.2.	Procesos asociados al desarrollo y mejora de productos de software	30
3.1.3.	Procesos asociados a la valoración del desempeño de los miembros de la fábrica de software.	31
3.2.	ORGANIZACIÓN Y PERSONAL	33
3.2.1.	Perfiles.....	34
3.3.	ARQUITECTURA ACTUAL Y ANÁLISIS DE BRECHAS	35
4.	ARQUITECTURA DE APLICACIONES Y DE DATOS .	37
4.1.	ARQUITECTURA DE APLICACIONES	37

5. ARQUITECTURA DE INFRAESTRUCTURA BASE.....	52
5.1. INFRAESTRUCTURA DE LA FÁBRICA DE SOFTWARE	52
6. OPORTUNIDADES Y SOLUCIONES	59
6.1. ARQUITECTURA DE NEGOCIO	60
6.2. ARQUITECTURA DE APLICACIONES Y DATOS.....	65
7. PLANIFICAR LA MIGRACIÓN.....	68
7.1. PRIORIZACIÓN.....	68
7.1.1. Análisis de impacto	68
7.1.2. Análisis de Esfuerzo	68
7.1.3. Fases.....	69
7.2. ANÁLISIS DE DEPENDENCIAS.....	69
7.3. MAPA DE RUTA	70
8. CONCLUSIONES Y RECOMENDACIONES	71
8.1. CONCLUSIONES	71
8.2. RECOMENDACIONES.....	72
9. REFERENCIAS.....	73
ANEXOS	75

Lista de Figuras

Figura 1. Clientes en Latinoamérica	1
Figura 2. Estructura de la organización en Latam	2
Figura 3. Descripción del Impacto	6
Figura 4. Expectativas de Valor	6
Figura 5. Estrategia de comunicación.	7
Figura 6. Equipo de arquitectura	9
Figura 7. Responsabilidades del equipo de arquitectura.	9
<i>Figura 8.</i> Diagrama conceptual de requerimientos de alto nivel. Fuente: Creación propia.	12
Figura 9. Arquitectura y organización del Área Práctica. Figura tomada de “CMMi v2.0 Development Model”.	13
Figura 10. Aspectos de evaluación. Figura tomada de “SFIA 7: The complete reference”.	17
<i>Figura 11.</i> Situación actual.	18
<i>Figura 12.</i> Análisis de brechas situación actual.	18
Figura 13. Mapa de capacidades requeridas.	19
Figura 14. Arquitectura de datos objetivo.	21
Figura 15. Arquitectura de aplicaciones objetivo.	22
Figura 16. Componentes mínimos para un ambiente de desarrollo.	23
Figura 17. Estimación de la arquitectura objetivo.	24
<i>Figura 18.</i> Diagrama IPO correspondiente al proceso de estimación de software. Fuente: Creación propia.	29
<i>Figura 19.</i> Diagrama IPO correspondiente al proceso de valoración de desempeño. Fuente: Creación propia.	33
<i>Figura 20.</i> Estructura organizacional del área de ingeniería. Fuente: Creación propia.	34
<i>Figura 21.</i> Perfiles del área de ingeniería. Fuente: Creación propia.	35
<i>Figura 22.</i> Iniciativas que cierran brechas. Fuente: Creación propia.	36
Figura 23. Aplicaciones existentes en la fábrica de software del área de ingeniería.	39
Figura 24. Aplicaciones utilizadas por los roles ágiles definidos en el área de ingeniería.	41
Figura 25. Aplicaciones definidas para el fortalecimiento del área de ingeniería.	43
Figura 26. Origen de datos según su tipo.	46
Figura 27. Relación entre los tipos de documentos y roles.	47
Figura 28. Clasificación por confidencialidad.	49
Figura 29. Criterios de confidencialidad de activos de información.	49
Figura 30. Valoración de la confidencialidad de activos de información.	50
Figura 31. Análisis de brechas de los activos de información.	51
Figura 32. Iniciativas que cierran brechas de los activos de información.	51
Figura 33. Arquitectura actual equipos de un entorno de desarrollo.	53
Figura 34. Distribución de un ambiente de desarrollo.	53
Figura 35. Posición de AWS en el cuadrante de Gartner.	55
Figura 36. Conceptualización de la iniciativa INEG1.	61
Figura 37. Conceptualización de la iniciativa INEG2.	62
Figura 38. Conceptualización de la iniciativa INEG3.	63

Figura 39. Conceptualización de la iniciativa INEG4.	65
Figura 40. Detalle de la iniciativa IAPL1.	66
Figura 41. Conceptualización de la iniciativa IAPL1.	66
Figura 42. Conceptualización de la iniciativa IAPL2.	67
Figura 43. Análisis de impacto.	68
Figura 44. Análisis de esfuerzo.	69
Figura 45. Proyectos y fases.	69
Figura 46. Mapa de ruta.	70

Lista de Tablas

Tabla 1. Estándares y metodologías utilizadas.	8
Tabla 2. Principio PR1.	10
Tabla 3. Principio PR2.	10
Tabla 4. Adaptación de técnicas ágiles al proceso de estimación.	28
Tabla 5. Áreas prácticas de Ingeniería. Adaptado.	30
Tabla 6. Arquitectura de aplicaciones actual. Catálogo de aplicaciones	37
Tabla 7. Aplicaciones y procesos CMMI.	40
Tabla 8. Rúbrica de madurez	44
Tabla 9. Iniciativas identificadas por arquitectura	59

1. FASE PRELIMINAR

1.1. Descripción general de la organización

Es una empresa multinacional que provee servicios y soluciones de software para el sector bancario.

Fundada en 1953, busca identificar las necesidades tecnológicas, las tendencias de la banca y potenciar el éxito digital de las instituciones financieras.

Tiene su sede central en Estados Unidos y su software (Cooperative Open Banking Information System) está instalado y operativo en más de 70 prestigiosas instituciones financieras de diferentes tamaños en 17 países.



Figura 1. Clientes en Latinoamérica

Imagen tomada desde <https://www.cobiscorp.com/nosotros/>.

1.1.1. Organigrama

La estructura organizativa de la empresa para Latinoamérica contempla 4 áreas:

- Área de Operaciones. - brinda servicios de soporte al software operativo.
- Área de Negocios. - está orientada a la gestión de clientes.
- Área de Marketing. - se orienta al desarrollo de negocios y descubrimiento de prospectos, además del posicionamiento de los productos.
- Área de Ingeniería. - cuenta con servicios y recursos especializados para el desarrollo, implementación y personalización de productos de software de la empresa ya construidos.

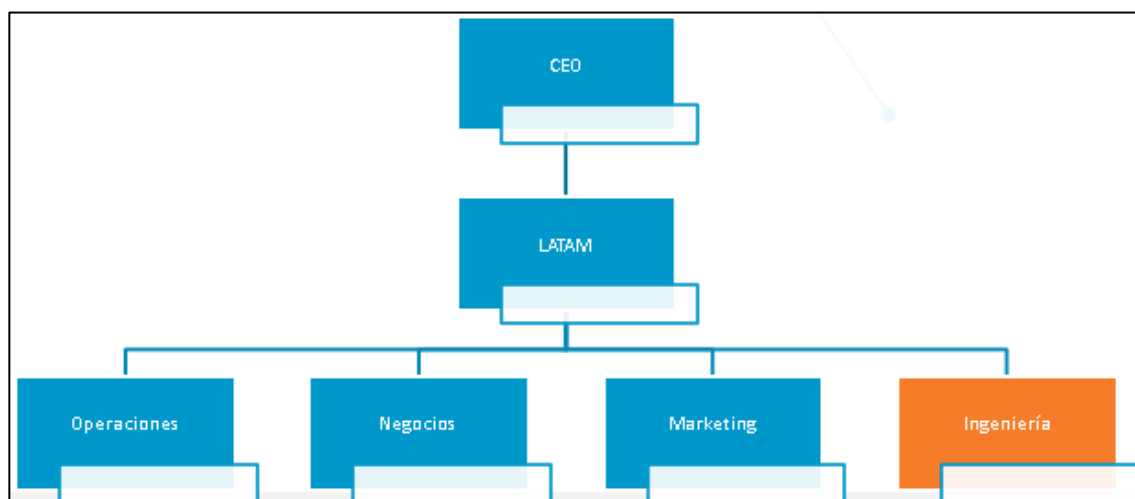


Figura 2. Estructura de la organización en Latam

1.1.2. Misión

“Potenciar el éxito digital de las instituciones financieras”.

1.1.3. Visión

“Acelerar el crecimiento de la red de valor de nuestros clientes”.

1.1.4. Principios

- confiabilidad
- satisfacción del cliente
- liderazgo e innovación

1.1.5. Valores

- honestidad
- compromiso
- empatía
- calidad
- recursividad
- enfoque en soluciones
- comunicación
- trabajo en equipo

1.2. Análisis del Contexto

La empresa mantiene vigentes aproximadamente 172 proyectos de implementación de software y 17 contratos de servicios de mantenimiento de aplicaciones e infraestructura, soportados por más de 530 profesionales con vasta experiencia y conocimiento en procedimientos bancarios y financieros, herramientas de desarrollo y metodologías ágiles.

Cuando un proyecto de implementación de software es aprobado por un cliente para su ejecución, es asignado a un gerente del área de operaciones para que realice los preparativos para su gestión con un rol de Business Owner. Los Business Owners o BOs por sus siglas en inglés son “un pequeño grupo de partes interesadas que tienen la

responsabilidad principal de la gobernanza comercial y técnica, el cumplimiento y el retorno de la inversión (ROI) para una solución desarrollada por un Agile Release Train (ART)” (Scaled Agile, Inc., 2022).

El área de ingeniería asigna recursos para el proyecto con el fin de conformar el ART de equipos o células que tendrán la misión de planificar, estimar, estructurar el backlog con historias de usuario y desarrollar el producto o funcionalidades en iteraciones conocidas como sprints según la metodología Agile.

Al final de cada mes, se genera una factura al área de operaciones con el valor de los recursos o servicios asignados al proyecto. Los recursos son colaboradores con diferente perfil y servicios como arrendamiento de infraestructura, catalogación y el mantenimiento de ambientes de prueba o certificación.

1.3. Problemática

En la empresa es un logro significativo incrementar las ventas de los productos cada año porque de ellas dependerá su rentabilidad y sostenibilidad a futuro, sin embargo, este incremento precisa una mayor demanda de recurso humano principalmente del área de ingeniería que no están disponibles en un futuro cercano porque están asignados a los proyectos en marcha.

Ante esta situación, el departamento de Capital Humano ejecuta procesos de reclutamiento para atraer al mejor personal, en la mayoría de los casos, tardan mucho hasta ser contratados. Como alternativa, se ha optado en buscar empresas externas que provean los recursos según los perfiles requeridos, pero se señala que tienen experiencia en el uso de las herramientas de desarrollo, pero ninguna o muy poca en conocimientos en el negocio bancario, en procedimientos financieros, o en productos de la empresa.

En la fase de planeación, se han advertido inconsistencias al no estimar correctamente el esfuerzo y el presupuesto para la ejecución del proyecto.

En la fase de construcción, se han evidenciado productos con muy poca calidad, con excesivo número de errores que obligan a ejecutar retrabajos o refactorizaciones repercutiendo en las fechas de entrega y en los compromisos adquiridos con el cliente.

Se ha tergiversado la metodología ágil al no ejecutarse pruebas de completitud que garanticen la eficacia del release centrándose solamente en el alcance de cada iteración. Cuando se prueba la situación ideal (happy path) solamente evitando las situaciones incómodas que comúnmente se presentan en entornos productivos.

Se evidencia la falta de compromiso de los desarrolladores al no dar a conocer condiciones adversas a la funcionalidad que se construye; por desarrollos inconclusos que no se advierten en la fase de pruebas y que exigen el esfuerzo adicional para que se completen o corrijan, incluso como horas suplementarias cuando deberían ser orientadas a la construcción de nuevas funcionalidades.

En la facturación mensual que se reporta hacia el área de servicios se reportan variaciones considerables entre meses anteriores como consecuencia de rubros que no fueron advertidos, justificados o acordados previamente y que afectan el presupuesto destinado para el proyecto.

1.4. Organización Impactada

El área de ingeniería es la principal interesada en mejorar el nivel de satisfacción de las áreas internas relacionadas con el cliente.

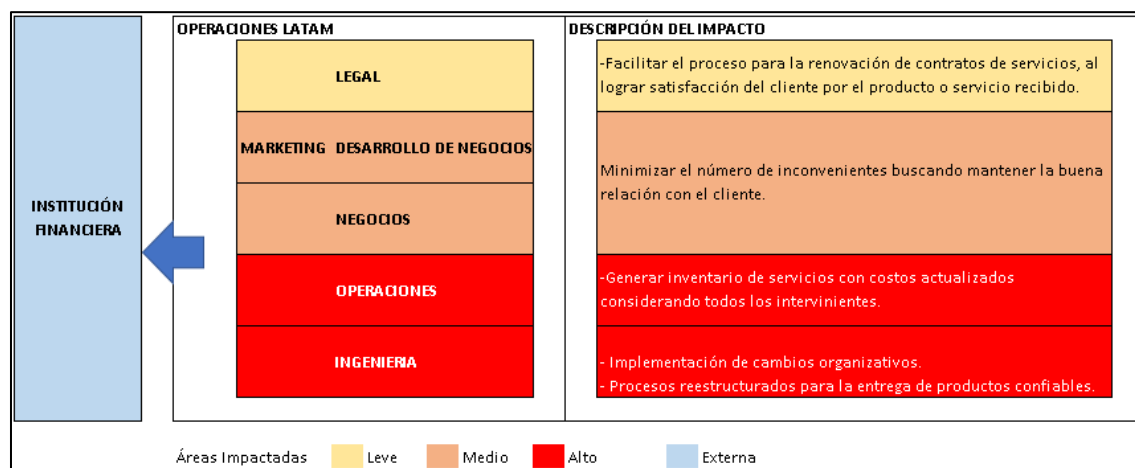


Figura 3. Descripción del Impacto

1.5. Stakeholders

1.5.1. Expectativas de Valor

UNIDAD	CARGO	Comprensión actual	Comprensión Requerida	Compromiso actual	Compromiso requerido	Expectativas en relación al Concern
Gerencia General	Gerente General	3	4	2	4	Eficiencia Operativa
Legal	Especialista Legal	1	2	1	1	Cliente predispuesto a la renovación contractual
Calidad y Procesos	Director de Procesos y Calidad	4	5	4	5	Programas de implementación rentables
	Analista de Procesos	4	5	4	5	Índices de rendimiento favorables
	Líder Ágil	2	3	1	3	Mayor cumplimiento metodología
Tesorería	Subgerencia Financiera	2	3	2	3	Agilidad de Pagos y reducción de costos
	Tesorera	1	2	1	2	Aceptación de facturas y pagos acreditados
Capital Humano	Directora de Capital Humano	3	4	3	5	Procesos de selección ágiles
Ingeniería	Director de Operaciones	1	4	1	4	Eficiencia Operativa
	Gerente de Proyecto RTE	1	3	1	4	Proyectos Rentables
	Equipo Ágil	2	3	1	4	Entregas con calidad, disminución de errores
Servicios	Director Operaciones	3	3	2	5	Rentabilidad
	Gerente de Proyecto BO	3	3	2	5	Control Presupuestario

Figura 4. Expectativas de Valor

Los representantes de las áreas que tienen alguna relación con el cliente mantienen altas expectativas, porque su eficiencia se incrementará si el cliente se encuentra satisfecho con el servicio que paga o por el producto de calidad que recibe.

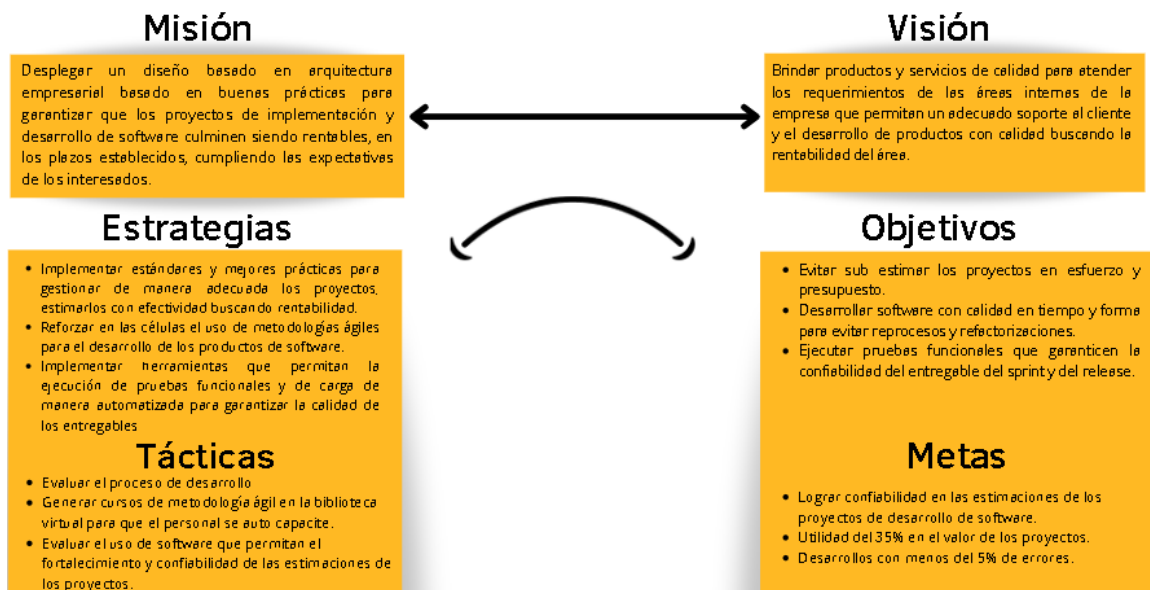
1.5.2. Estrategia de Comunicación

UNIDAD	CARGO	PODER	NIVEL INTERÉS	ESTRATÉGIA	CUADRANTE
Directorio	Presidente del Directorio	Alto	Bajo	Mantenerlo satisfecho	3
	Miembro del Directorio	Alto	Bajo	Mantenerlo satisfecho	3
Gerencia General	Gerente General	Alto	Medio	Mantenerlo satisfecho	3
Legal	Director Legal	Alto	Bajo	Mantenerlo informado	3
Calidad y Procesos PMO	Director de Procesos y Calidad	Alto	Alto	Interesado clave	4
Tesorería	Subgerencia Financiera	Alto	Medio	Mantenerlo informado	3
Capital Humano	Subgerencia Administrativa	Alto	Bajo	Mantenerlo Informado	3
	Directora de Capital Humano	Medio	Alto	Mantenerlo informado	3
Ingeniería	Director de Operaciones	Alto	Alto	Interesado clave	4
	Gerente de Proyecto RTE	Alto	Alto	Interesado clave	4
Servicios	Director Operaciones	Alto	Alto	Interesado clave	4
	Gerente de Proyecto BO	Alto	Alto	Interesado clave	4

Figura 5. Estrategia de comunicación.

Los directores de Servicios e Ingeniería han mostrado su interés en el resultado del ejercicio porque buscan eficiencia operativa y rentabilidad para sus áreas, y lo principal que la fidelización del cliente dependerá de su gestión.

1.6. BMM



1.7. Marcos de Referencia y Estándares

A continuación, se listan los estándares más importantes que se han identificado en concordancia con la problemática identificada.

Tabla 1.

Estándares y metodologías utilizadas.

Área	Referente	Descripción
OPERACIONES	SCRUM/ÁGILE	Conjunto de reglas, roles y procedimientos utilizados para acelerar el lanzamiento de productos con calidad.
GESTIÓN DE PROYECTOS	PMBOK	Describe una guía con los fundamentos para la Dirección de Gestión de Proyectos, otorga una serie de herramientas que permiten identificar procesos y generar resultados óptimos.
OPERACIONES	Devops	Conjunto de prácticas para abordar la gestión del desarrollo y el despliegue de funcionalidades para la operación con agilidad.
OPERACIONES	CMMI-DEV	Modelo de Madurez de Capacidad Integrado, sirve de guía para la mejora de las diferentes áreas de los procesos de desarrollo de software.
OPERACIONES	ISO/IEC 12207	Norma que establece el ciclo de vida del software, constituye una estructura y lenguaje en común para los intervinientes en el desarrollo de software como compradores, desarrolladores, proveedores, operadores, técnicos y gestores.
OPERACIONES	IEEE 1008	Método estándar para las pruebas unitarias de software que brindan información del fracaso, análisis y corrección de fallas en el software.
NEGOCIO	BABOK	Esta Guía describe las prácticas generalmente aceptadas por los profesionales y sus grupos de interés para ofrecer valor en el campo del Análisis de Negocios.
NEGOCIO	KANBAN	Método para gestionar el trabajo de manera visual que permite identificar las actividades críticas de un proceso garantizando que el trabajo sea continuo e ininterrumpido.
OPERACIONES	eSCM	Es un modelo de capacidad que establece las mejores prácticas para ayudar a las empresas a mejorar los procesos de adquisición de productos y servicios.

1.8. Equipo de Arquitectura

Para el trabajo de arquitectura se considera el patrocinio del vicepresidente de Ingeniería de Producto y el apoyo de una célula de

arquitectura que provee del conocimiento base para la toma de decisiones, es responsable de conducir a la organización hacia la mejora operativa al proveer información real, confiable y oportuna.

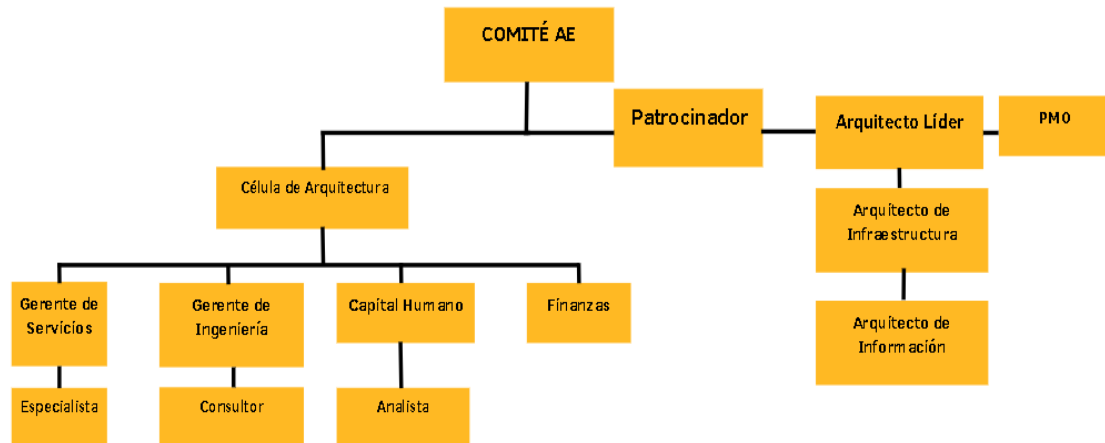


Figura 6. Equipo de arquitectura

Rol	Responsabilidad
Arquitecto Líder	- Proporcionar los lineamientos necesarios que orienten a los arquitectos empresariales sobre las actividades de arquitectura. - Mantener la alineación con los objetivos empresariales ejerciendo un adecuado gobierno.
Arquitecto de Infraestructura	Diseñar y definir el software, hardware y protocolos a usar para soportar las operaciones, mantenerlas estables y disponibles.
Arquitecto de Información	Modelamiento adecuado de datos para soportar las funcionalidades del negocio.
Especialista PMO	Brindar lineamientos y evaluar procesos conforme los marcos de referencia existentes en la organización.
Célula de Arquitectura	- Analizar los requisitos y brindar información para el ejercicio de arquitectura. - Implementar el diseño de la arquitectura. - Responder de manera efectiva al diseño de la arquitectura identificando alternativas conforme el contexto de la empresa.

Figura 7. Responsabilidades del equipo de arquitectura.

1.9. Catálogo de Principios

Para establecer los objetivos que debe considerar el ejercicio de arquitectura, en la siguiente tabla se establecen los principios que la empresa ha instaurado.

Tabla 2.

Principio PR1.

Código	PR1
PRINCIPIO	Adopción de Metodología.
DOMINIO	GUÍA.
DEFINICIÓN	Uso de metodología ágil scrum para el desarrollo de software y PMBOK para el gobierno del proyecto.
MOTIVACIÓN	Beneficiarse de las mejores prácticas de ambas metodologías.
IMPLICACIONES	Los especialistas deberán tener certificación PMP y/o Agile.

Tabla 3.

Principio PR2.

Código	PR2
PRINCIPIO	Adopción de Metodología.
DOMINIO	GUÍA.
DEFINICIÓN	Las herramientas de desarrollo serán Java, HTML5, CSS3, ANGULAR.
MOTIVACIÓN	Minimizar la inversión en licencias de herramientas de desarrollo.
IMPLICACIONES	El perfil de conocimientos del desarrollador deberá contemplar el conocimiento en esas herramientas.

2. VISIONAMIENTO ARQUITECTÓNICO

2.1. Requerimientos de alto nivel

Para fortalecer los servicios que brinda la fábrica de software a la empresa, se definen los siguientes requerimientos:

- Establecer un modelo de estimación que permita un dimensionamiento de los esfuerzos asociados a los proyectos considerando:
 - Alcance. – identificando los límites, objetivos, plazos y entregables.
 - Complejidad. – identificando el número de interfaces, funcionalidades e integraciones con proveedores internos o externos.
 - Conformación del equipo. - cantidad, nivel de conocimiento y experiencia de los recursos, velocidad de desarrollo de las células, curvas de aprendizaje.
 - Costo. - directos e indirectos de recursos propios o externos.
- Ejecutar una nueva valoración de madurez del proceso de desarrollo del software que permita mejorarlos dando lugar a productos de calidad.
- Capacitar al personal técnico que conforman las células de desarrollo del área de ingeniería.
- Fortalecer los procesos de valoración del desempeño del personal técnico involucrado en los proyectos de desarrollo de software.
- Implementar aplicaciones que aporten al dimensionamiento de los proyectos de software.

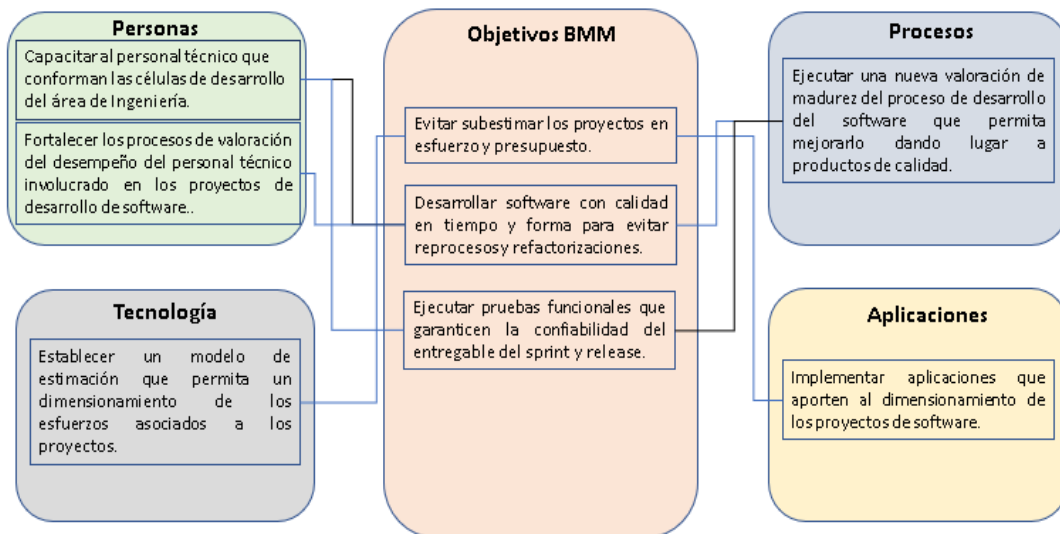


Figura 8. Diagrama conceptual de requerimientos de alto nivel. Fuente: Creación propia.

2.2. Visionamiento y escenarios de la solución

La estimación del esfuerzo para construir software es un paso desafiante y muy importante, porque su efectividad determina el éxito de un proyecto normalmente enmarcado por el cumplimiento de las restricciones de tiempo y presupuesto, que repercute directamente en la calidad del producto desarrollado.

Un esfuerzo sobre estimado generalmente implica un alto presupuesto de modo que el costo del software se vuelve oneroso innecesariamente y pierde la oportunidad de la ventaja competitiva con la competencia.

Por el contrario, el sub dimensionamiento es un problema muy común en los proyectos de desarrollo y repercute en la calidad del software al destinarse recursos insuficientes.

Para lograr una mejora en la estimación de costos, se utilizará uno de los referentes sugeridos para proyectos ágiles basados en scrum, este método es el de estimación mediante tallas.

Para garantizar la calidad del producto construido en el área de Ingeniería es indispensable medir el nivel de madurez del proceso de desarrollo de software utilizando como referencia el modelo de evaluación de CMMI-Dev, que contiene las mejores prácticas que ayudan a las organizaciones a mejorar sus procesos.

CMMi es administrado por el SEI (Software Engineering Institute) instituto federal estadounidense de investigación y desarrollo, fue fundado en 1984 y se encarga de valorar a las organizaciones que aplican este modelo evaluando la madurez de los procesos y proporciona una orientación referente de cómo mejorarlos para dar lugar a mejores productos y servicios.

La versión CMMi Dev 2.1 en particular establece una lista de áreas prácticas del modelo que intervienen cuando se aplican procesos con técnicas ágiles con SCRUM. Para la evaluación se contemplarán las siguientes Áreas de Capacidad:



Figura 9. Arquitectura y organización del Área Práctica. Figura tomada de "CMMi v2.0 Development Model".

A continuación, se describen las áreas prácticas que corresponden a cada área de capacidad y su relación con Scrum.

1.- Área de capacidad Asegurando la Calidad (ENQ)

1.1 Área práctica Desarrollo y Gestión de Requisitos (RDM)

Tiene como propósito el obtener los requerimientos, garantizar un entendimiento común por parte de los grupos involucrados e interesados y alinear los requisitos, planes y productos de trabajo.

En un proyecto ágil con Scrum las prácticas de esta área permiten reforzar el alcance mediante una mayor comprensión, claridad, identificación de problemas o riesgos en relación con los requerimientos y se relacionan con las actividades:

- planificación de liberaciones.
- refinamiento de backlog.
- planificación de un sprint.
- ejecución de un sprint.
- revisión o demostración de un sprint.
- retrospectiva.

1.2 Área práctica Verificación y Validación (VV)

La verificación (se construyó el producto correctamente) consiste en comprobar que lo que se desarrolló cumple con todas las especificaciones de sus requisitos funcionales y no funcionales mediante pruebas unitarias, pruebas de código, pruebas de regresión, y la validación (se construyó el producto correcto) se ocupa de controlar si el producto satisface los requerimientos y expectativas del cliente espera mediante las pruebas funcionales y de certificación con el cliente.

En un proyecto ágil con Scrum se relacionan con las actividades que aseguran que la historia de usuario cumple con sus criterios de aceptación establecidos y si la funcionalidad el producto en las demostraciones se

comporta satisfactoriamente en su entorno deseado y si cumple con sus expectativas.

1.3 Área práctica Revisión hechas por colegas (PR)

Tiene por objetivo el identificar y resolver problemas de los productos de trabajo a través de revisiones realizadas por pares del producto o expertos en la materia.

En un proyecto ágil Scrum las revisiones pares se realizan en:

- el refinamiento de backlog
- demostrar historias completas durante la revisión del sprint con el propietario del producto (Product Owner).
- usar programación en pares.
- revisar el diseño, planes de pruebas, casos de pruebas.

1.4 Área práctica Proceso Aseguramiento de Calidad (PQA)

Se orienta a verificar y habilitar la mejora de la calidad de los procesos ejecutados y sus productos de trabajo resultantes.

En proyectos ágiles pueden evaluar los productos de trabajo cuando se examina los requisitos en la revisión del backlog, en la revisión del sprint en la retrospectiva donde se identifican y registran además las lecciones aprendidas.

2.- Área de capacidad Productos de Ingeniería y Desarrollo (EDP)

2.1 Área Práctica Solución Técnica (TS)

El objetivo es diseñar y construir soluciones con los requerimientos del cliente.

Las prácticas aumentan el valor de un proyecto ágil con Scrum cuando se evalúan los requerimientos en:

- planeación de la liberación. – comprensión más anticipada y completa de los riesgos de la solución.

- refinamiento / revisión del backlog. – validar la asignación de requerimientos para diseñar los componentes que se deben desarrollar para las historias de usuario conocidas durante el sprint, ayuda a identificar los requerimientos adicionales que pueden haberse omitido.
- planeación de Sprint. – Conocimiento más amplio para el contexto de los diseños y las interfases o conexiones para el próximo sprint.
- ejecución del sprint. – La mayor parte de la ejecución del sprint se emplea en desarrollar una solución funcional en lugar de refactorizar.
- revisión de sprint. – Permite tener una comprensión más profunda de lo que ha conseguido durante el sprint.
- retrospectiva del sprint. – Proporciona una comprensión exhaustiva de qué componentes de diseño quedan.

2.2 Área práctica Integración del Producto (PI)

El objetivo es integrar y entregar la solución que aborda los requisitos de calidad y funcionalidad que cumplen con la expectativa del cliente.

Esta práctica enmarca a las herramientas que los equipos ágiles con Scrum pueden utilizar:

- Emplear automatización y procesos de Devops.- con la premisa de reducir el esfuerzo humano tanto como sea posible en pruebas unitarias, pruebas de regresión, pruebas de sistemas e integración continuas.
- Asegura que las herramientas y ambientes necesarios se hayan planeado oportunamente.
- Asegura que las funcionalidades, conexiones e interfases se hayan validado con éxito.

Para fortalecer los procesos de valoración del desempeño del personal técnico que conforman los proyectos de desarrollo de software de la fábrica en el área de Ingeniería se considerará una evaluación de capacidades a los perfiles:

- Product Manager (PM)
- Product Owner (PO)
- Scrum Master (SM)
- Célula Scrum (ST)

El marco de referencia sFIA permite identificar el grado de aplicación del conocimiento para resolver situaciones inherentes a la actividad profesional.

En la siguiente figura se muestran los aspectos que serán considerados en la evaluación.

AUTONOMÍA	INFLUENCIA	COMPLEJIDAD	CONOCIMIENTO	HABILIDADES PARA NEGOCIOS
<ul style="list-style-type: none"> • En qué nivel el recurso puede trabajar por su cuenta, sin necesidad de ayuda e instrucciones. 	<ul style="list-style-type: none"> • En qué nivel se puede dirigir a sus compañeros y cambiar la mentalidad de las personas. 	<ul style="list-style-type: none"> • En qué nivel el recurso es apto para comprender problemas, requisitos e instrucciones difíciles. • En qué nivel el recurso puede completar las tareas complicadas. 	<ul style="list-style-type: none"> • Posee conocimientos genéricos básicos adecuados al área de trabajo. • Aplica los conocimientos recién adquiridos para desarrollar nuevas habilidades. 	<ul style="list-style-type: none"> • En qué nivel puede trabajar con otras personas y clientes. • En qué nivel puede usar los sistemas de software de la empresa para su beneficio o el colectivo.

Figura 10. Aspectos de evaluación. Figura tomada de “SFIA 7: The complete reference”.

2.3. Análisis de brechas

Para proceder con el análisis de las brechas es primordial analizar la situación actual alineada con las actividades que se realizan en el proceso de desarrollo de software:

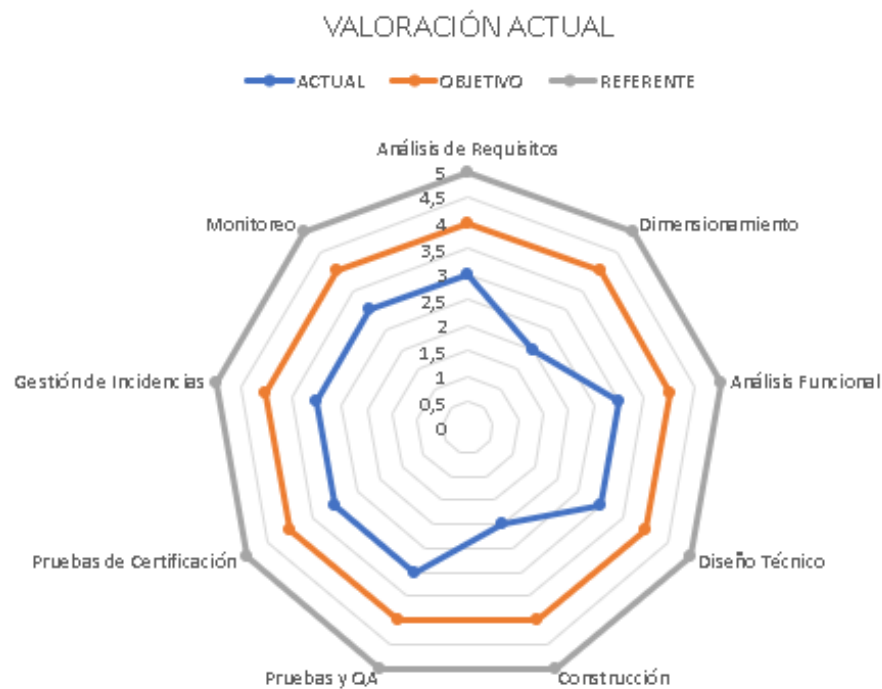


Figura 11. Situación actual.

VALORACIÓN ACTUAL	ACTUAL	OBJETIVO	REFERENTE	ANÁLISIS DE LAS BRECHAS
Análisis de Requisitos	3	4	5	La funcionalidad se describe a muy alto nivel.
Dimensionamiento	2	4	5	Sub dimensionamiento de los proyectos en términos del alcance, complejidad y/o esfuerzo.
Análisis Funcional	3	4	5	La actividad se ejecuta pero el documento es generado en función de los lineamientos a alto nivel.
Diseño Técnico	3	4	5	Se ejecuta la actividad unicamente para historias de más de 80 puntos.
Construcción	2	4	5	Existen problemas en la priorización del backlog, el refinamiento no advierte esfuerzos adicionales por refactorizaciones que impactan los tiempos de entrega.
Pruebas y QA	3	4	5	Se ejecutan pruebas de completitud muy básicas
Pruebas de Certificación	3	4	5	Se ejecutan con desviación para garantizar la funcionalidad.
Gestión de Incidencias	3	4	5	Se gestionan conforme a la disponibilidad del equipo de soporte, si es un grupo reducido el tiempo de solución se incrementa ocasionando malestar al cliente.
Monitoreo	3	4	5	

Figura 12. Análisis de brechas situación actual.

2.4. Definición de la arquitectura objetivo

Realizado el análisis de la arquitectura actual de la fábrica de desarrollo del área de Ingeniería, identificada la referencia y el análisis de brechas existente a continuación se define una arquitectura para dar respuesta a las necesidades.

2.4.1. Arquitectura de negocio objetivo

En este dominio se prevén los procesos y el fortalecimiento de las capacidades relevantes que impulsarán el cambio en la fábrica de software.

Es preciso indicar que las capacidades de la planificación y dimensionamiento son prioritarias, de su efectividad depende de la consecución y calidad del producto entregado.

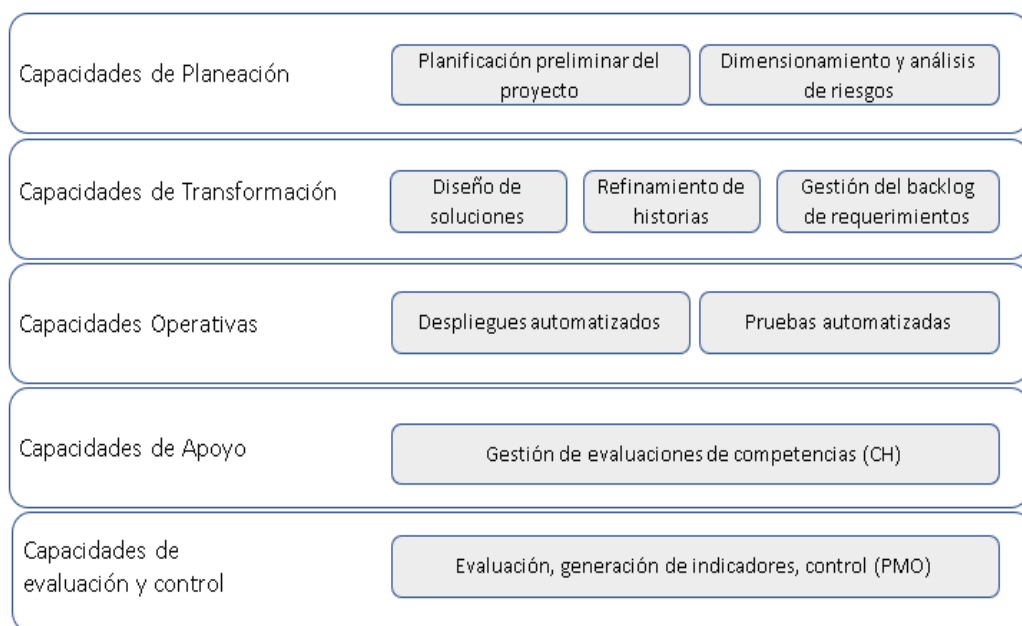


Figura 13. Mapa de capacidades requeridas.

Para soportar las capacidades requeridas se han identificado los siguientes procesos, roles y competencias necesarias para impulsar el cambio en el área de ingeniería.

PROCESOS	FASES	ROLES	COMPETENCIAS
<ul style="list-style-type: none"> • Migración. • Parametrización. • Capacitación. • Administración de la configuración. • Gestión de proyectos. 	<ul style="list-style-type: none"> • Estrategia y Visionamiento. • Desarrollo Iterativo. • Despliegue y Entrega. 	<ul style="list-style-type: none"> • Roles de Portafolio. • Roles de diseño y construcción • Roles de gestión de proyectos. 	<ul style="list-style-type: none"> • Comunicación efectiva. • Trabajo en equipo. • Resolución de problemas. • Adaptabilidad. • Orientación al logro.

2.4.2. Arquitectura de datos objetivo

Para contribuir de manera adecuada el proceso de dimensionamiento de los proyectos, será necesario establecer un almacenamiento de información donde se acumulen las características adicionales del producto construido, como por ejemplo, el tipo de tecnología utilizada para el desarrollo, número de componentes por pantalla, número de recursos asignados, experiencia de los recursos, cantidad de historias de usuario construidas de todos los proyectos del área de ingeniería y que nos permitirán establecer parámetros y relaciones para identificar comportamientos.

Esta información alimentará al modelo de dimensionamiento para establecer un parámetro de velocidad y estimaciones de esfuerzo basadas en los históricos.

En la siguiente figura se muestran las relaciones entre los repositorios de datos existentes y los nuevos.

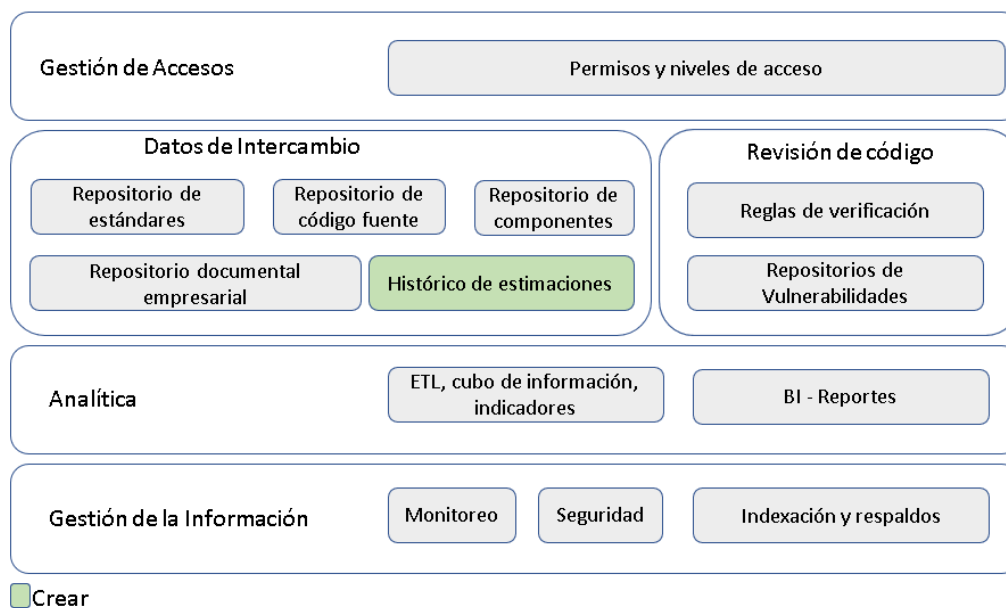


Figura 14. Arquitectura de datos objetivo.

2.4.3. Arquitectura de aplicación objetivo

El enfoque principal que se necesita es el de automatizar el método de estimación implementando una plataforma que contemple la información de los distintos procesos de la empresa, en especial del proceso desarrollo de software para asegurar un dimensionamiento adecuado.

En la siguiente figura se muestran las aplicaciones existentes y su influencia en las fases del desarrollo de software de la fábrica.

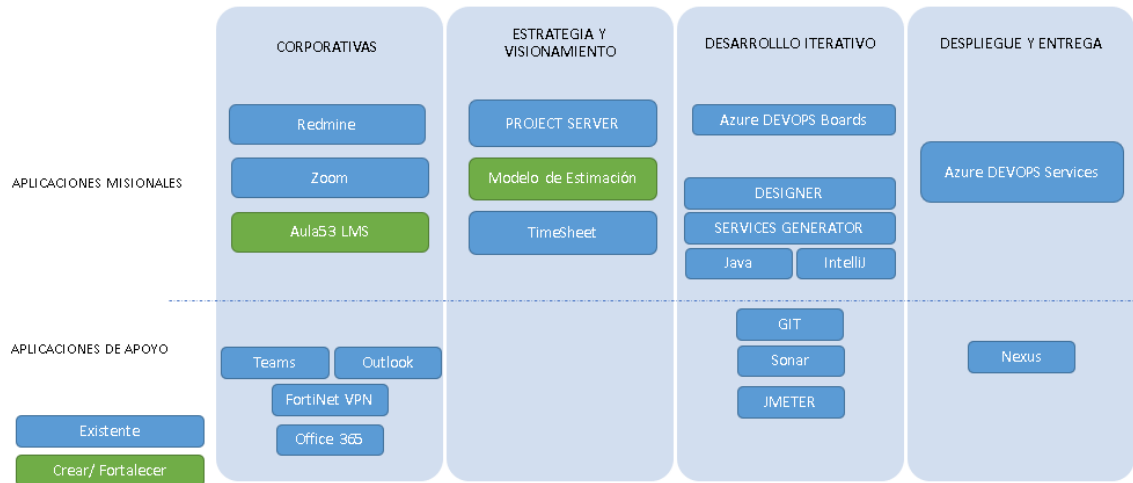


Figura 15. Arquitectura de aplicaciones objetivo.

2.4.4. Arquitectura de infraestructura base

En la siguiente figura se agrupan componentes de hardware y software requeridos para la adquisición de las capacidades para una disposición física instalada en sitio no en nube conocida como on-premise.

Al realizar un análisis de la infraestructura actual es imprescindible la existencia de un ambiente de desarrollo local instalado en la máquina del desarrollador y que tiene acceso a los servicios corporativos en la nube para generar, almacenar y probar componentes o funcionalidades que posteriormente serán desplegados en los servidores.

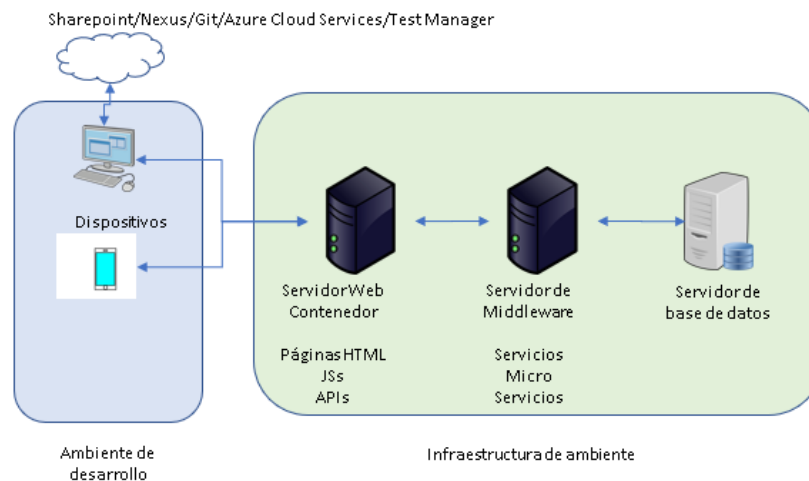


Figura 16. Componentes mínimos para un ambiente de desarrollo.

La disposición indicada en la figura 10, dependiendo de las características y capacidades de los servidores, podrían incluir el ambiente de desarrollo y el ambiente de pruebas.

Para las pruebas de certificación, las funcionalidades que forman parte del entregable serán desplegadas en un ambiente que disponga el cliente, que al menos tenga la disposición de la figura 10.

2.4.5. Road Map

A continuación, se describe la ruta propuesta para alcanzar el objetivo arquitectónico mediante la consecución de los proyectos identificados:

HOJA DE RUTA	Q1			Q2			Q3		
	M1	M2	M3	M4	M5	M6	M7	M8	M9
GENERAR Y PUBLICAR CONTENIDO DE CAPACITACIÓN ACTUALIZADO	20d								
REVISIÓN DE VALORACIÓN DE DESEMPEÑO		20d							
EJECUTAR VALORACIÓN DE DESEMPEÑO			20d						
NUEVA VALORACIÓN DE MADUREZ PROCESO DE DESARROLLO				40d					
IMPLEMENTAR APLICACIONES PARA EL DIMENSIONAMIENTO								40d	
EVALUACIÓN Y MEJORA						10d			
IMPLEMENTACIÓN DEL MODELO DE ESTIMACIÓN							60d		

Figura 17. Estimación de la arquitectura objetivo.

3. ARQUITECTURA DE NEGOCIO

Según Andrew Josey “La Fase B aborda el desarrollo de una Arquitectura de Negocio que apoye la Visión de la Arquitectura acordada” (Andrew Josey,2013, *TOGAF Versión 9.1 Guía de Bolsillo*, Capítulo 2, p. 38), y su objetivo describir cómo la empresa tiene que transformarse para alcanzar los objetivos de negocio, responder a los impulsores estratégicos de manera que se solventen las preocupaciones de los interesados.

3.1. Arquitectura de negocio objetivo

Como primer marco de referencia se establece el framework ágil de desarrollo SAFe, que permite escalar los beneficios y prácticas de Agile y Lean hacia todos los niveles de la organización -equipos, programas, portafolios- a través de prácticas, roles y herramientas aplicadas, con el objetivo de obtener entregas frecuentes de software con calidad, maximizando el valor y beneficio al negocio y permitiendo mayor agilidad organizacional.

El segundo referente es CMMI-DEV v2.0 y sus áreas de capacidad que permiten establecer lineamientos de referencia para la planificación y gestión del trabajo y el aseguramiento de la calidad de las soluciones de software que se construirán con metodologías ágiles.

3.1.1. Procesos asociados a la estimación de software

Las empresas frecuentemente aplican técnicas de mejora de procesos (reingeniería de procesos) a las operaciones del negocio para buscar la eficiencia y eficacia. Estas mismas técnicas se pueden aplicar para mejorar la eficiencia y eficacia del proceso Ágil de estimación.

A continuación, se describe el proceso donde se integran el marco de trabajo Scrum y las áreas prácticas de CMMI para estimar el tamaño, el esfuerzo, el costo del trabajo, la duración y los recursos

necesarios para desarrollar y proporcionar la solución a alto nivel, mediante la definición de historias de usuario que considerarán y priorizarán todo lo que genera valor a la organización.

Es preciso indicar que, en los proyectos ágiles con Scrum, las prácticas de CMMI-Dev se ejecutan de manera implícita en las ceremonias que describimos a continuación.

3.1.1.1. Planificación de Liberaciones (Release Planning)

El objetivo de esta actividad es identificar las historias del backlog que se trabajarán a futuro, para establecer una fecha aproximada de entrega al cliente de un conjunto de funcionalidades o productos utilizables.

Se debe elaborar un documento (conocido como release plan) que contemple las responsabilidades, los recursos y las actividades o prerequisites necesarios para cada entrega.

Establecer la prioridad para el desarrollo de las historias e identificar los riesgos que podrían interferir en la finalización del desarrollo y la consecución de la funcionalidad o producto.

3.1.1.2. Refinamiento del backlog

En esta actividad se revisarán cada una de las historias de usuario para lograr el entendimiento suficiente en todos los miembros del equipo, respecto al alcance de la funcionalidad que se irá a desarrollar, profundizando los detalles según el propósito y los criterios de aceptación.

Se realizará una estimación de esfuerzo de las historias utilizando técnicas como el Scrum póker, o por tallas de camisetas para identificar un tamaño, las tareas y el esfuerzo y los supuestos de las tareas.

3.1.1.3. Planificación de un sprint

Es el punto de inicio de un sprint y cuenta con la colaboración de todos los miembros del equipo para establecer un objetivo y un plan de trabajo, se establecen los responsables de la ejecución de las historias de usuario, se define lo que se puede entregar en la iteración (incremento) y las tareas que se ejecutarán para llegar a cumplir con los criterios de aceptación definidos, se realiza una estimación conjunta más refinada del esfuerzo necesario para cada tarea especificada.

3.1.1.4. Ejecución de un sprint

Definido el objetivo y el plan en la actividad anterior, en cada sprint se trabaja para ejecutar las tareas establecidas y obtener un incremento en la funcionalidad de un producto o un nuevo producto.

Se irá registrando el logro obtenido en el tablero Kanban conforme finalicen las tareas en el transcurso del sprint.

Una historia de usuario no se dará por aceptada hasta que se hayan cumplido todos sus criterios de aceptación, este control es fundamental para garantizar que la funcionalidad contenga todo lo establecido.

3.1.1.5. Revisión y demostración del sprint (Sprint Review)

Se constituye en primer lugar por una revisión o demostración del incremento del producto o servicio que un equipo ágil ha realizado y en segundo lugar permite adaptar el Product Backlog a las condiciones actuales del negocio.

En esta revisión los interesados pueden inspeccionar el incremento para obtener una visión adecuada de cómo se avanza con la construcción de las funcionalidades y recibir retro alimentación.

3.1.1.6. Retrospectiva

Es la ceremonia que le permite al equipo reflexionar sobre su forma de trabajar en el sprint, promueve una mejora continua en lo que se hace identificando acciones que se mantendrán o corregirán en el siguiente sprint.

Se revisan además la cultura, el proceso utilizado para la construcción del software y el rendimiento del equipo ágil.

A continuación se muestran las relaciones entre el proceso de estimación y los procesos según la metodología ágil.

Tabla 4.

Adaptación de técnicas ágiles al proceso de estimación.

PROCESO DE ESTIMACIÓN	PROCESOS ÁGILES					
	PLANIFICACIÓN DEL RELEASE	REFINAMIENTO DE BACKLOG	PLANIFICACIÓN DEL SPRINT	EJECUCIÓN DEL SPRINT	REVISIÓN DEL SPRINT	RETROSPECTIVA
Revisar los requerimientos y el alcance.	X	X				
Dimensionar el trabajo y asignar recursos.	X	X	X			X

Elaborar el primer estimado base.	X	X	X			
Identificar y analizar riesgos.	X	X	X	X	X	X
Validar y revisar el estimado.	X	X	X			X

Para extender y organizar la información del proceso de estimación de software, a continuación, se muestran con más detalle los datos necesarios para establecer los lineamientos del proyecto, los procesos se ejecutan con los datos y los responsables de su manejo y la información que se obtiene será el insumo para el inicio de la construcción del software.

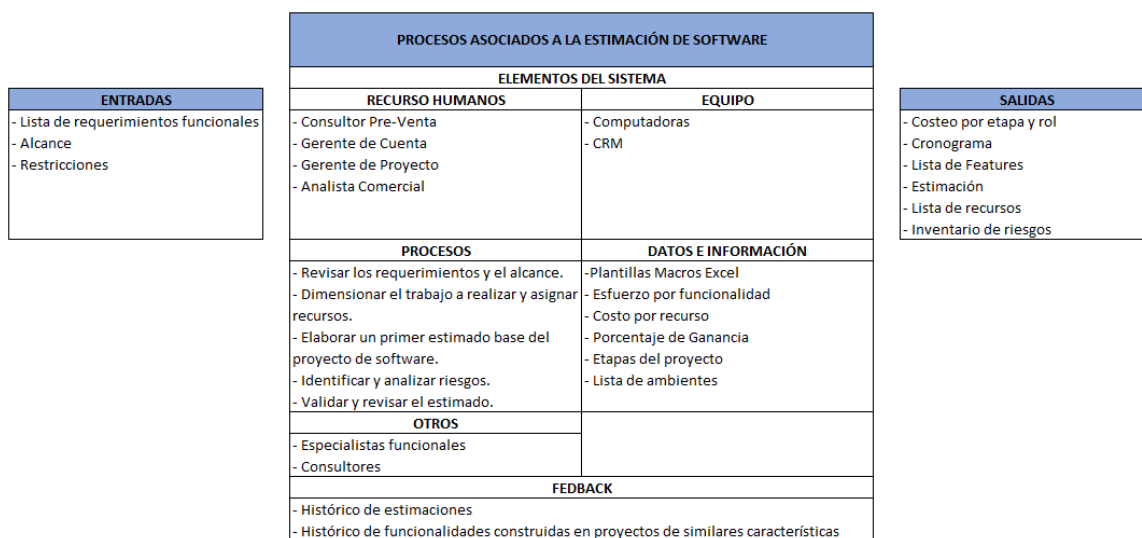


Figura 18. Diagrama IPO correspondiente al proceso de estimación de software. Fuente: Creación propia.

3.1.2. Procesos asociados al desarrollo y mejora de productos de software

El CMMI ayuda a las organizaciones a mejorar sus procesos al ofrecer un conjunto de buenas prácticas. En el proceso de desarrollo de servicios y productos de software, se considera el área de capacidad Asegurando la Calidad y sus áreas prácticas, que permitirán asegurar la construcción de productos de alta calidad.

A continuación, se describen las áreas prácticas de ingeniería de los procesos CMMI-DEV que se fortalecerán en la fábrica de software.

Tabla 5.

Áreas prácticas de Ingeniería. Adaptado.

ÁREA DE CAPACIDAD	AREA PRÁCTICA	OBJETIVO ESPECÍFICO
Asegurando la Calidad (ENQ)	Desarrollo y Gestión de Requisitos (RDM)	3.7 Validar los requerimientos para garantizar que la solución resultante va a funcionar como se lo esperaba en el ambiente destino.
	Proceso de Aseguramiento de Calidad (PQA)	2.3 Comunicar los problemas de calidad y no-conformidades y asegurar su resolución.
		2.4 Registrar y utilizar los resultados de las actividades de aseguramiento de la calidad
	Validación y Verificación (VV)	3.1 Identificar y registrar las oportunidades de mejora durante las actividades de aseguramiento de la calidad. 2.1 Seleccionar los componentes y métodos para la verificación y validación
Revisiones Hechas por Colegas (PR)	3.1 Desarrollar, actualizar y utilizar los criterios para la verificación y la validación 3.2 Analizar y comunicar los resultados de la verificación y la validación 2.2 Seleccionar los productos de trabajo para ser revisados por pares	

		<p>2.3 Preparar y realizar revisiones por pares en productos de trabajo seleccionados aplicando procedimientos establecidos</p> <p>2.4 Resolver los problemas identificados en las revisiones por pares</p> <p>3.1 Analizar resultados y datos de las revisiones por pares</p>
Productos de Ingeniería y Desarrollo (EDP)	Solución Técnica (TS)	<p>3.3 Realizar un análisis de creación, compra o reutilización</p> <p>3.4 Seleccionar soluciones basadas en los criterios de diseño</p> <p>3.6 Diseñar interfaces o conexiones de la solución con los criterios establecidos</p>
	Integración del Producto (PI)	<p>2.5 Evaluar los componentes integrados para garantizar el cumplimiento de los requisitos y el diseño de la solución</p> <p>2.6 Integrar soluciones y componentes según la estrategia de integración</p> <p>3.2 Confirmar, antes de la integración, que las interfaces de los componentes o conexiones cumplen con las descripciones de interfaz o conexión</p> <p>3.3 Evaluar los componentes integrados en cuanto a compatibilidad de interfaz o conexión.</p>

3.1.3. Procesos asociados a la valoración del desempeño de los miembros de la fábrica de software.

Un reto muy importante para las organizaciones es el contar con el mejor personal calificado que les permita de manera activa, la consecución de los objetivos estratégicos de la compañía.

Es por tanto imprescindible para la fábrica de desarrollo del área de ingeniería, la aplicación de una valoración que permita detectar las fortalezas y debilidades de su equipo de trabajo para ejecutar procesos que potencialicen su conocimiento y rendimiento.

Dentro de los objetivos principales de la evaluación es el medir el rendimiento de los colaboradores y con base a eso determinar incentivos logrando una gestión justa, fomentar la comunicación líder-colaborador-equipo para lograr sinergia, estimular a una mayor productividad, detectar la necesidad de capacitaciones que refuercen las habilidades y conocimientos requeridos para su rol.

A continuación, se describe el proceso de evaluación:

- Establecer objetivos y aspectos de evaluación
Se requiere considerar criterios de evaluación para los diferentes objetivos o aprendizajes esperados, deben entenderse como indicadores concretos para compararlos con referentes o estándares de desempeño.
- Planificar la ejecución de la evaluación
Se debe establecer un cronograma para las actividades de evaluación de tal manera que su impacto al negocio sea el mínimo conforme a las características físicas de los espacios o disponibilidad y capacidad de las plataformas.
- Ejecutar la evaluación
Son las actividades que realizan los colaboradores donde se espera una activa participación.
- Presentación de resultados
Se recopilan evidencias que permitan establecer métricas e indicadores de tal manera que se puedan determinar las circunstancias que afectan el desempeño de los colaboradores.
- Identificación de actividades de mejora
Con la información relevante es preciso generar retroalimentación a los participantes e incorporar actividades que refuercen las actitudes o conocimientos.

En el siguiente diagrama se muestran las relaciones y el entorno del proceso de valoración.

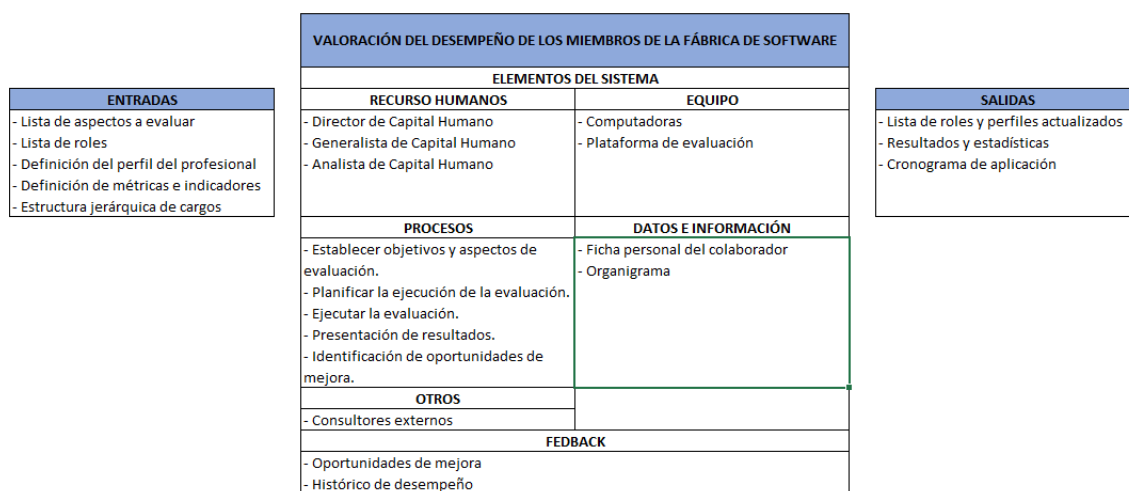


Figura 19. Diagrama IPO correspondiente al proceso de valoración de desempeño. Fuente: Creación propia.

3.2. Organización y personal

La fábrica de software del área de ingeniería por su magnitud tiene la capacidad de gestionar varios equipos ágiles que se encuentran asignados a diferentes proyectos y que en múltiples ocasiones pueden ejecutarse al mismo tiempo.

Para la gestión de equipos la metodología SAFe contempla 3 niveles. En el primer nivel correspondiente a equipo, se encuentran los equipos ágiles conocidos también como células están conformados por un Product Owner, un Scrum Master, los desarrolladores y los testers.

El segundo Nivel se llama Capa Programa y está contemplado por un Product Manager que puede estar a cargo de varias cédulas ágiles, un arquitecto de sistemas y un RTE (Release Train Engineer) quien es el encargado de generar un entorno adecuado y reunir las personas adecuadas para la toma de decisiones basadas en datos correctos.

Finalmente, el tercer nivel corresponde al portafolio que establecen la visión de los proyectos y se encargan de garantizar la relación con los objetivos estratégicos organizacionales.

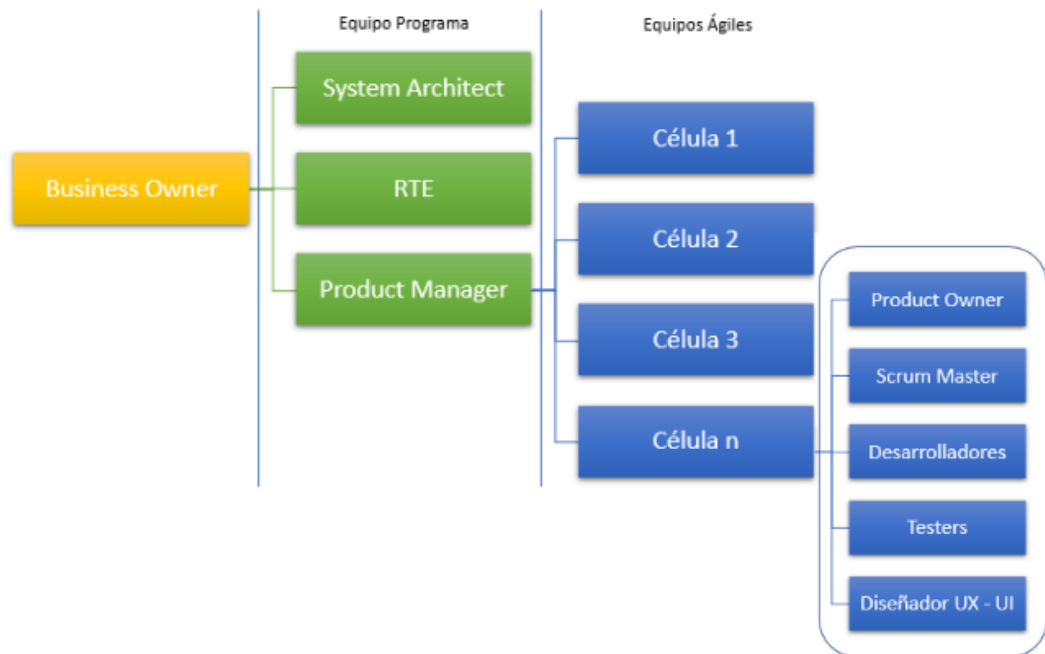


Figura 20. Estructura organizacional del área de ingeniería. Fuente: Creación propia.

3.2.1. Perfiles

La fábrica de software del área de ingeniería necesita fortalecer los roles existentes, razón por la cual, se establecen los perfiles necesarios que contribuyan a las exigencias del negocio.

ROL	PERFIL PROFESIONAL	RESPONSABILIDAD
Product Owner	<ul style="list-style-type: none"> - Habilidad de gestionar proyectos y priorización de tareas - Dominio de metodologías ágiles - Habilidades de comunicación efectiva 	<ul style="list-style-type: none"> - Construir la visión del producto y transmitirla al equipo. - Mantenimiento del backlog. - Creación de historias de usuario, estimación y priorización.
Scrum Máster	<ul style="list-style-type: none"> - Dominio de las metodologías, marcos de referencia relacionadas con el agilismo Scrum, Lean, Kanban, SAFe. - Capacidad de resolución de conflictos. - Experiencia trabajando con equipos. 	<ul style="list-style-type: none"> - Asegurar que la metodología se cumpla. - Eliminar y gestionar impedimentos. - Generar métricas de gestión del equipo ágil. - Identificar acciones de mejora.
Desarrolladores	<ul style="list-style-type: none"> - Formación técnica en ingeniería de sistemas. - Dominio de lenguajes de programación java, kendo, HTML5, CSS3, java script. - Trabajo en equipo. - Conocimiento y experiencia trabajando con metodologías ágiles. - Conocimiento en gestores de bases de datos SQL, Sybase, Oracle. 	<ul style="list-style-type: none"> - Construcción de soluciones de software altamente parametrizables. - Desarrollar funcionalidades con base en la metodología y estándares definidos.
Arquitecto de Sistemas	<ul style="list-style-type: none"> - Experiencia en diseño y construcción de software. - Experiencia en manejo de patrones de diseño. - Formación técnica en ingeniería de sistemas, informática o afines. - Dominio de tecnologías ágiles. 	<ul style="list-style-type: none"> - Dirigir la creación o la revisión de las capacidades de infraestructura para soportar la demanda de servicios estratégicos del negocio. - Responsable de la revisión de código generado por los desarrolladores.
Release Train Engineer(RTE)	<ul style="list-style-type: none"> - Habilidades para dirigir y gestionar equipos. - Dominio sólido de metodologías ágiles. - Comunicación efectiva. - Capacidad de resolución de conflictos. 	<ul style="list-style-type: none"> - Establecer y comunicar los calendarios anuales de iteraciones e incrementos de programa. - Escalar y rastrear impedimentos. - Proporcionar información de los recursos para abordar los cuellos de botellas críticos. - Capacitar a líderes, equipos, y Scrum Masters en prácticas y mentalidades Lean-Agile.

Figura 21. Perfiles del área de ingeniería. Fuente: Creación propia.

3.3. Arquitectura actual y análisis de brechas

Luego de analizar los procesos de la fábrica de desarrollo de software del área de ingeniería de la empresa, y considerando el desempeño deseado, se han identificado brechas muy específicas y sus correspondientes brechas que nos permitirán solventarlas. Estas iniciativas se muestran en la figura número 5.

PROCESO	BRECHA	INICIATIVA
ESTIMACIÓN DE SOFTWARE	Aplicación errónea o no-utilización de la información histórica de la organización.	Crear un alcance al proceso para incluir el procedimiento que describa el almacenamiento de los datos correspondientes a las estimaciones reflejando el alcance, cantidad de recursos, tiempos y funcionalidades involucradas, número de componentes, el número de características, el número de requisitos, los puntos de función personalizadas o estándar, número de líneas de código.
ESTIMACIÓN DE SOFTWARE	Certificación de cumplimiento a lo planificado.	Establecer un procedimiento para el registro de la información necesaria para lograr la certeza de que el desarrollo de software se realizó de acuerdo al plan, limitaciones y supuestos, .
DESARROLLO Y MEJORA DE PRODUCTOS DE SOFTWARE	Garantizar que el producto construido cumpla con el alcance en tiempo y forma.	<ul style="list-style-type: none"> - Fortalecer la actividad de validación y verificación para garantizar que el producto de software construido funcione correctamente y que la solución establecida cumpla con el uso previsto en el entorno de destino. - Fortalecer el proceso de verificación para enfatizar las pruebas de completitud, las pruebas automáticas, demostraciones y simulaciones.
VALORACIÓN DEL DESEMPEÑO DE LOS MIEMBROS DE LA FÁBRICA DE SOFTWARE	Poca o nula valoración de desempeño a los miembros del equipo.	Fortalecer el proceso de valoración del desempeño para incorporar los aspectos de autonomía, influencia, complejidad, habilidad para negocios.

Figura 22. Iniciativas que cierran brechas. Fuente: Creación propia.

4. ARQUITECTURA DE APLICACIONES Y DE DATOS

La arquitectura de aplicaciones es uno de los cuatro dominios fundamentales de la arquitectura empresarial, está conformada por programas clasificados en diferentes categorías que contribuyen y facilitan las actividades de los funcionarios, permite establecer cuándo amerita adquirir software comercial, cuándo usar SaaS, cuándo actualizar o adquirir grandes soluciones o cuándo optar por un desarrollo personalizado y cómo se debe plantear el ciclo de vida del software en la organización.

4.1. Arquitectura de aplicaciones

4.1.1. Arquitectura de aplicaciones base

A continuación, se describe un inventario de las aplicaciones que se utilizan en la fábrica de software del área de ingeniería.

Tabla 6.

Arquitectura de aplicaciones actual. Catálogo de aplicaciones

PROCESO	NOMBRE	VERSIÓN	DESCRIPCIÓN
Diseño	Draw..iu	20.2.8	Herramienta libre para la creación de diferentes tipos de diagramas (flujo, UML, y más) que permite integración con varias plataformas.
Desarrollo	Visual Studio Profesional	2022	Entorno de desarrollo de software multiplataforma que permite trabajar con proyectos de cualquier tamaño y complejidad.
Desarrollo	HTML	5	Lenguaje de marcado utilizado en el desarrollo web para crear contenido y la estructura de una página web.
Desarrollo	CSS	3	Lenguaje de hojas de estilos creado para establecer y controlar el aspecto o presentación de los documentos definidos con HTML5, se utiliza para crear la presentación del contenido en páginas web complejas.
Desarrollo	JavaScript	ES12	Lenguaje de programación interpretado por los navegadores que permite dinamismo e interactividad a las páginas web.
Desarrollo	Angular	12	Framework de JavaScript de código abierto muy potente orientado al desarrollo de aplicaciones web modernas, de complejidad media o elevada.
Desarrollo	Kendo UI	2.0.0	Librería de JavaScript para el desarrollo de aplicaciones web modernas, contiene diversas interfaces de usuario dinámicas.
Desarrollo	Java	8	Lenguaje de programación altamente utilizado para codificar aplicaciones web.
Desarrollo	Visual Studio Code	1.69.2	Editor de código fuente personalizable desarrollado por Microsoft que permite trabajar con diversos lenguajes de programación.
Desarrollo	Notepad++	8.4.4	Editor de texto de código fuente libre que permite visualizar el contenido de los archivos de varios lenguajes de programación.

Desarrollo	Brackets	2.1.2	Es un editor de código diseñado para mejorar la productividad mediante una interfaz que soporta distintos lenguajes de programación.
Desarrollo	Android Studio	2021.2.1	Entorno de desarrollo integrado para la plataforma Android ideal para la creación de aplicaciones móviles.
Desarrollo	Microsoft Azure Devops	Cloud	Es un producto de Microsoft que proporciona herramientas para el control de versiones, la gestión de equipos, compilaciones automatizadas cubriendo todo el ciclo de vida de la aplicación y habilita las capacidades de DevOps.
Validación y Verificación (QA)	SonarQube	8	Plataforma para evaluar código fuente, genera métricas que ayudan a mejorar la calidad del código de software programado.
Validación y Verificación (QA)	Microsoft Test Manager	2012	Herramienta para administrar y gestionar todo el ciclo de pruebas funcionales de software.
Validación y Verificación (QA)	Postman	10	Herramienta que sirve de gran ayuda al equipo de desarrollo y pruebas al permitir la simulación de peticiones a servicios web o invocaciones a APIs de forma sencilla.
Control de Versiones	GIT	2.35.1	Software de control de versiones, eficiente, confiable y compatible que permite el mantenimiento de las versiones de los archivos fuente.
Control de Versiones	Nexus	2.15.1	Gestor de repositorios de artefactos o librerías de compilación interno y compartido por toda la organización, creados para la integración continua y que pone a disposición para la implementación automatizada en los ambientes de prueba, certificación y producción.
Soporte Infraestructura	IBM WebSphere Application Server	9.0.5	Es un servidor de aplicaciones de software Java, provee de un entorno de ejecución de servidor flexible y seguro.
Soporte Infraestructura	IBM WebSphere MQ	7.0	Middleware de mensajería por medio de colas que facilita el intercambio de información garantizado, seguro y fiable entre aplicaciones.
Soporte Infraestructura	JBOSS	EAP 7.2	Plataforma de código abierto que permite crear, ejecutar, implementar y administrar aplicaciones Java empresariales ofreciendo seguridad, rendimiento, escalabilidad.
Soporte Infraestructura	Apache Tomcat	9.0.0	Es un servidor web que permite el despliegue dinámico de aplicaciones Java.
Análisis de Datos	Power BI	3.0.0	Conjunto de servicios para el análisis de datos e inteligencia empresarial de Microsoft, que permite unir diferentes fuentes de datos, analizarlos y generar visualizaciones interactivas a través de paneles e informes.

En la figura 23, se ilustran las aplicaciones que se utilizan la fábrica de software del área de ingeniería.

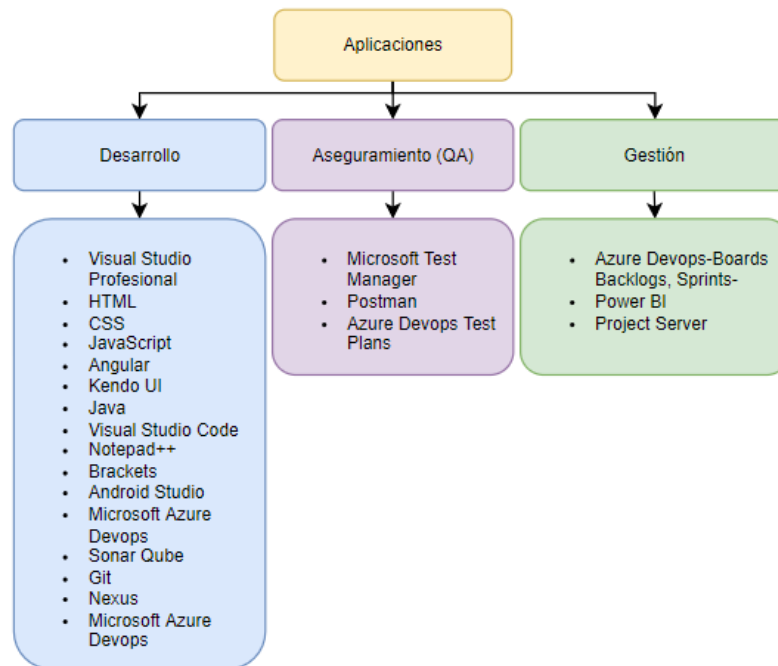


Figura 23. Aplicaciones existentes en la fábrica de software del área de ingeniería.

4.1.1.1. Matriz de procesos y aplicaciones

A continuación, se detallan las aplicaciones que son utilizadas en el área de ingeniería y su relación con las principales áreas prácticas de CMMI-DEV enfocadas al desarrollo y al aseguramiento de la calidad de los productos o servicios desarrollados.

Tabla 7.

Aplicaciones y procesos CMMI.

Aplicación	Áreas Prácticas de la Arquitectura Objetivo (CMMI-DEV)					
	Desarrollo y Gestión de Requisitos (RDM)	Proceso de Aseguramiento de la Calidad (PQA)	Verificación y Validación (VV)	Revisiones Hechas por Colegas (PR)	Solución Técnica (TS)	Integración del Producto (PI)
Draw.iu	X				X	X
Visual Studio Profesional	X				X	
HTML	X					X
CSS	X					X
JavaScript	X					X
Angular	X					X
Kendo UI	X					X
Java	X					X
Visual Studio Code	X				X	X
Notepad++	X	X	X	X	X	X
Brackets	X			X	X	X
Android Studio	X				X	
Microsoft Azure Devops	X	X	X	X	X	X
SonarQube	X					X
Microsoft Test Manager		X	X			
Postman	X	X	X	X	X	X
GIT	X			X	X	X
Nexus	X					X
IBM WebSphere Application Server						X
IBM WebSphere MQ						X
JBOSS						X
Apache Tomcat	X				X	X
Power BI	X					
Microsoft Office 365	X	X	X	X	X	X
Google Chrome	X	X	X	X	X	X
Microsoft Project Plan 3	X					

4.1.1.2. Matriz de roles y aplicaciones

En la figura 3 se presenta la frecuencia de uso de las aplicaciones existentes por los principales roles ágiles.

APLICACIÓN	Business Owner (BO)	Product Manager (PM)	Product Owner (PO)	Scrum Master (SM)	Desarrollador (DEV)	Tester (TES)	Arquitecto de Sistemas (SA)
Draw.iu							
Visual Studio Profesional							
HTML							
CSS							
JavaScript							
Angular							
Kendo UI							
Java							
Visual Studio Code							
Notepad++							
Brackets							
Android Studio							
Microsoft Azure Devops							
SonarQube							
Microsoft Test Manager							
Postman							
GIT							
Nexus							
IBM WebSphere Application Server							
IBM WebSphere MQ							
JBOSS							
Apache Tomcat							
Power BI							
Microsoft Office 365							
Google Chrome							
Microsoft Project Plan 3							



Uso muy frecuente 
 Uso poco frecuente 

Figura 24. Aplicaciones utilizadas por los roles ágiles definidos en el área de ingeniería.

4.1.2. Arquitectura de aplicaciones objetivo

Los siguientes aplicativos son considerados por la arquitectura objetivo para mejorar la efectividad en el flujo de la construcción del software del área de ingeniería.

4.1.2.1. JUnit

Es un Framework de código abierto para ejecutar pruebas unitarias y de integración de proyectos de software con Java, contiene un conjunto de herramientas, clases y métodos que facilitan la labor de realizar pruebas de los sistemas y así comprobar la funcionalidad y consistencia del producto de software desarrollado.

Las principales características de la plataforma Junit son:

- Facilita la definición de casos de prueba orientados a validar una funcionalidad en particular de manera específica.
- Permite iniciar la ejecución de los casos de pruebas ya definidos y escritos utilizando una ventana de comandos.
- Posee una interfaz que permite automatizar la ejecución de grupos de casos de prueba.
- Genera estadísticas que verifican y evalúan el cumplimiento de los casos de prueba seleccionados.
- Ha sido incorporado de manera nativa en los ambientes más populares de desarrollo IDE (integrated development environment) como Visual Studio Code, IntelliJ IDEA, Eclipse, NetBeans y en herramientas de compilación como por ejemplo Gradle, Maven, Hormiga.

4.1.2.2. JMeter

Es una herramienta ideal para realizar pruebas de rendimiento de aplicaciones desarrolladas para la web.

Permite grabar las interacciones de los tipos de usuario con la aplicación para generar casos de prueba que podrán ser ejecutados de manera concurrente por parte de un número de usuarios previamente definido, con el fin de validar su eficacia y completitud.

Una vez definidos los casos de prueba, JMeter permite su agrupación mediante la definición de planes de prueba que se almacenan en un archivo que podrá ser ejecutado en los diferentes ambientes de pruebas o certificación UAT para medir su rendimiento.

Finalmente, permite generar los indicadores que resumen la ejecución automática en las dimensiones de completitud, correctitud y confiabilidad.

4.1.3. Trazabilidad de nuevas aplicaciones

En la figura 25, se establecen las aplicaciones necesarias para el fortalecimiento del área de ingeniería.

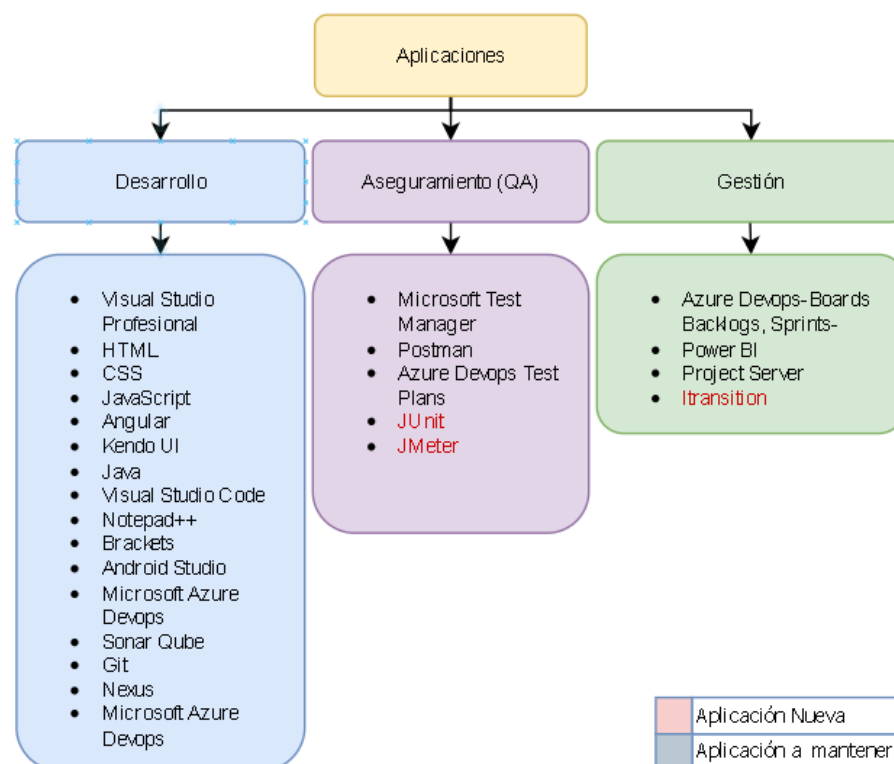


Figura 25. Aplicaciones definidas para el fortalecimiento del área de ingeniería.

4.1.4. Análisis de brechas e iniciativas

Para identificar las brechas y las iniciativas que las solucionan, es necesario primero evaluar las aplicaciones utilizadas en la fábrica de software del área de arquitectura mediante una rúbrica personalizada que mostramos a continuación.

Tabla 8.

Rúbrica de madurez

Calificación	Detalle
5	Se comprueba un proceso de innovación exponencial que justifica el liderazgo corporativo al presentar componentes de arquitectura maduros y altamente soportados por las mejores prácticas en el dominio del conocimiento.
4	Ofrecen técnicas probadas y tecnologías maduras reconocidas en el dominio del conocimiento, los componentes arquitectónicos muestran un buen desempeño considerando las guías y prácticas de la industria.
3	Se documenta formalmente con herramientas y métricas de gestión adecuadas y mejoradas periódicamente. Los componentes arquitectónicos se implementaron en un tiempo superior a 6 meses.
2	No existe documentación oficial ni herramienta de gestión. Los componentes arquitectónicos son básicos y obligatorios para la implementación en varios clientes, pero el rendimiento no es estándar ni apropiado para la industria.
1	Frecuentemente no existe soporte técnico, y cuando existe, responde por iniciativa de los empleados. Muestra un nivel incompleto de componentes arquitectónicos.
0	No proporciona evidencia actual y no se encontraron componentes arquitectónicos.

4.2. Arquitectura de datos

Según The Open Group Architecture Framework(TOGAF) es la fase que describe el desarrollo de los modelos, políticas, reglas y estándares que gobiernan la recopilación, almacenamiento, disposición, integración y el uso de los datos dentro de la organización.

La arquitectura de datos tiene por objetivo el identificar las necesidades del negocio y transformarlas en requisitos de datos y sistemas y como deben ser administrados dentro de los procesos de una empresa.

4.2.1. Arquitectura actual de datos

En la fábrica de software del área de ingeniería, se generan tres grandes tipos de datos:

- Datos estructurados: son aquellos que se encuentran ordenados mediante una serie de filas y columnas bien definidas, usados de comúnmente en la mayor parte de las bases de datos relacionales (RDBMS).
- Datos semiestructurados: poseen un nivel medio de estructura, jerarquía y organización, como por ejemplo los archivos comprimidos, XML u otros lenguajes de marcado, por tanto, su gestión y automatización se dificulta.
- Datos no estructurados: son aquellos que no utilizan modelos o esquemas de datos fijos y predefinidos, pueden ser almacenados en bases de datos no relacionales ya que sería muy complejo ajustarlos a las filas y columnas estandarizadas de una base de datos tradicional.



Figura 26. Origen de datos según su tipo.

Los datos que se utilizan y generan en la fábrica de software del área de ingeniería se almacena en data centers locales con respaldo en la nube corporativa con mayor prioridad, mientras que los datos semi estructurados y no estructurados también son almacenados en la nube pero tienen limitaciones de espacio.

4.2.2. Arquitectura de datos objetivo

Con base en la norma ISO 27001 se deben considerar ciertos puntos respecto al cuidado y manejo de los datos que se generan en el área de ingeniería.

Por tanto, se considera la necesidad de fortalecer el método de generación de respaldos y disponibilidad de la información.

4.2.3. Análisis de brechas

La fábrica de software se conforma por colaboradores internos que forman parte de la organización, y colaboradores externos que acceden a los recursos empresariales durante la ejecución de los proyectos.

Considerando lo indicado, es imprescindible mantener una adecuada gobernanza de los datos para controlar la forma de obtención, el almacenamiento y la compartición de los datos.

Las amenazas a la seguridad de los datos pueden repercutir en:

- Accesos no autorizados a un conjunto de datos críticos o confidenciales.
- Corrupción o alteración de datos de manera parcial o pérdida total, de forma accidental o maliciosa.
- Faltas a los acuerdos de confidencialidad o leyes de protección de datos.
- Fuga de información o prematura por métodos físicos o digitales.
- Secuestro de datos por ransomware, software malicioso que encripta la información a cambio de una recompensa.

En la siguiente imagen se pueden visualizar los activos de información que se generan y utilizan en la fábrica de software y los roles responsables de su administración.

Tipos de Documentos	Business Owner (BO)	Product Manager (PM)	Product Owner (PO)	Scrum Master (SM)	Desarrollador (DEV)	Tester (TES)	Arquitecto de Sistemas (SA)
Presentaciones	X	x	x	X	X	X	X
Plantillas de estimación	X	X	X				X
Cronogramas de proyectos	X						
Actas de reuniones	X	X	X	X			
Acuerdos	X	X	X	X			
Contratos	X						
Métricas e indicadores	X	X	X	X			
Diagramas		X	X		X		X
Flujos		X	X				X
Repositorio documental	X	X	X	X	X	X	X
Repositorio de fuentes		X	X		X		X
Repositorio de componentes					X		X
Evidencia de pruebas QA						X	
Evidencia de pruebas unitarias					X		X
Historias de usuario			X				
Lista de features		X	X				
Actas de refinamiento		X	X	X			

Responsable



Figura 27. Relación entre los tipos de documentos y roles.

4.2.3.1. Clasificación de los activos de la información

Un activo de información según la norma ISO27001 se define como “cualquier cosa que tenga valor para la organización”. De acuerdo con esto, en la fábrica de desarrollo del área de ingeniería, se generan y utilizan diferentes tipos de datos de varios niveles de importancia que se requiere identificar y mantener dentro de un inventario de activos para asignarles una clasificación que permita un manejo y distribución seguro.

La clasificación de los activos tiene por objetivo asegurar que la información recibe los niveles de protección adecuados ya que amerita un tipo de manejo especial según su valor y otras características particulares.

Para ejecutar una adecuada clasificación se consideran las siguientes propiedades de seguridad de la información:

- Confidencialidad. – propiedad que determina que la información no sea revelada a entidades, procesos o usuarios no autorizados.
- Integridad. – propiedad encargada de salvaguardar la exactitud, confiabilidad y completitud de los datos.
- Disponibilidad. – propiedad de la información para determinar si es accesible y utilizable por una entidad con autorización.

Para la evaluación de los activos de la fábrica de software de la organización, se contemplará a continuación el primer dominio de confidencialidad solamente.

4.2.3.1.1. Clasificación de acuerdo con la confidencialidad

Se debe definir de acuerdo con las características de los activos que se manejan en cada organización.

Confidencial	Información disponible sólo para un proceso y usuarios específicos de la organización, que restringe su tratamiento, no está sujeta a publicidad o divulgación no autorizada.
Restringida	Esta información es propiedad de la empresa o de un tercero y puede ser utilizada por los empleados dentro de los procesos de la organización, pero no puede divulgarse sin el consentimiento del propietario.
De uso interno	Información que puede ser utilizada en todos los procesos por los funcionarios de la organización y que podría verse afectada negativamente en sus intereses si es conocida por terceros no autorizados.
Público	Información que puede ser entregada, publicada o distribuida sin restricciones a cualquier persona dentro y fuera de la entidad sin que esto implique daños a terceros ni a las actividades y procesos de la entidad.
No Clasificada	Activos de información que no han sido clasificados porque no forman parte del inventario. Deben ser tratados como confidenciales.

Figura 28. Clasificación por confidencialidad.

4.2.3.1.2. Criterios de confidencialidad

El criterio de confidencialidad determina que la información no sea accesible o divulgada a personas, en dispositivos o procesos no autorizados.

Tipo Información	La clasificación de la confidencialidad en este tipo de activos se debe realizar con base en su contenido (judiciales, financiero, datos personales...) y teniendo en consideración quienes deben tener acceso a estos.
Tipo Sistemas, Servicios y Hardware	La valoración se realiza considerando que los sistemas, servicios y hardware son elementos que almacenan, procesan o transfieren información y que contienen configuraciones, parámetros e información interna y de proveedores de interés para la organización.
Tipo Personas	La valoración de la confidencialidad de las personas se ejecuta con base en la información a la que tiene acceso por medios físicos o digitales, o conocen en función a su rol, función o actividad que ejecuta dentro de la organización.
Tipo Instalaciones	La valoración de este tipo de activos considera y hereda la clasificación de la información que almacenan o resguardan, así por ejemplo, si almacenan información restringida entonces la valoración de confidencialidad de las instalaciones a nivel de confidencialidad será igual.

Figura 29. Criterios de confidencialidad de activos de información.

4.2.3.1.3. Valoración de los criterios de confidencialidad

En la fábrica de software actualmente se tienen problemas en la manera de categorizar los activos de información, en muchos casos inclusive no se realiza esta actividad, lo que ocasiona que funcionarios dentro de distintos procesos tengan acceso no autorizado a la información relevante de los proyectos.

Es preciso indicar que contractualmente existe un acuerdo de confidencialidad para colaboradores internos y externos, sin embargo, es necesario reforzar la importancia de una adecuada clasificación.

CRITERIO	ESTADO ACTUAL	MEDIANO PLAZO	REFERENTE LARGO PLAZO
Tipo Información	2	4	5
Tipo Sistemas, Servicios y Hardware	3	4	5
Tipo Personas	2	4	5
Tipo Instalaciones	4	4	5

Figura 30. Valoración de la confidencialidad de activos de información.

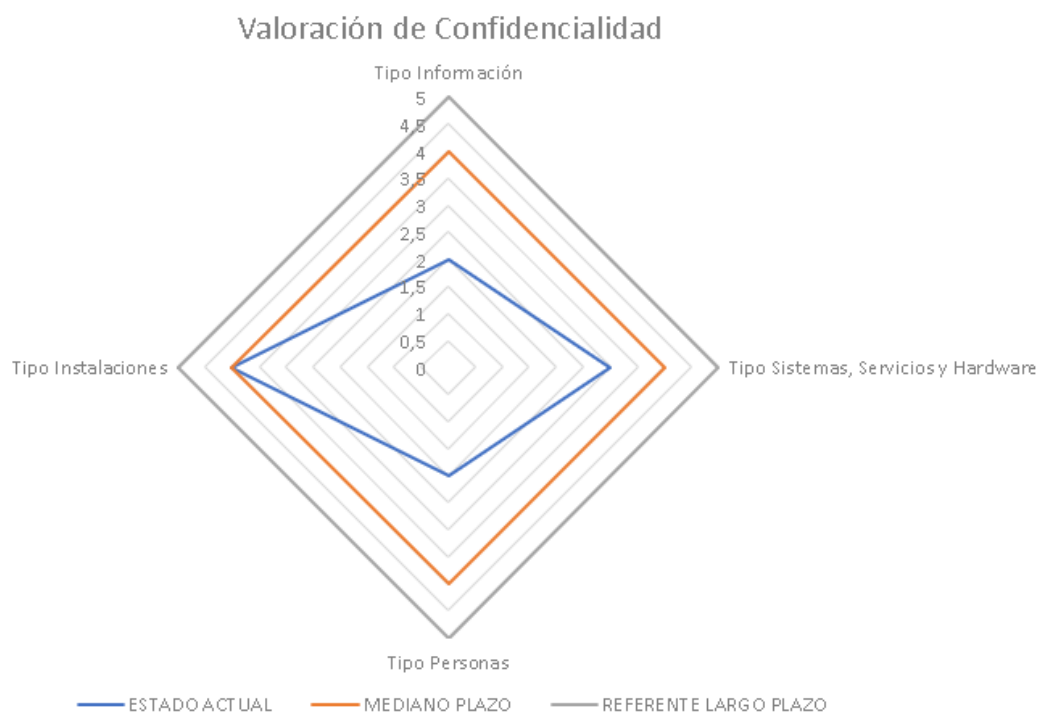


Figura 31. Análisis de brechas de los activos de información.

4.2.4. Iniciativas para cerrar brechas

Para lograr confidencialidad de los datos que se generan en la fábrica de software, se contempla la implantación de las siguientes medidas técnicas u organizacionales.

CRITERIO	INICIATIVA
Tipo Información	Generar procedimientos formales para clasificar la información en función de su valor, sensibilidad para su debido tratamiento.
Tipo Sistemas, Servicios y Hardware	Fortalecer los controles de acceso a equipos, servidores, almacenamiento y sistemas local o en nube, donde reside la información, a la red a través de la cual viaja la información para evitar el acceso no autorizado.
Tipo Personas	Crear campañas de formación y sensibilización sobre confidencialidad y protección de la información. Mantener los acuerdos de confidencialidad formalizados y firmados con los colaboradores que acceden y manejan información confidencial.

Figura 32. Iniciativas que cierran brechas de los activos de información.

5. ARQUITECTURA DE INFRAESTRUCTURA BASE

La arquitectura de infraestructura tecnológica permite identificar las capacidades de hardware necesarias para respaldar la implementación de servicios del negocio, de datos y de aplicaciones. Esto incluye infraestructura de TI, redes, comunicaciones, procesamiento y estándares.

5.1. Infraestructura de la fábrica de software

En la actualidad la fábrica de software dispone de servidores virtualizados en modalidad *on premise*, lo que significa que están almacenados dentro de las instalaciones propias de la organización para el desarrollo de productos propios.

Una vez que los productos están listos para su venta y de acuerdo con las características de los proyectos que pueden tener como objetivo la provisión de servicios o de implementación, se considerará la instalación en la infraestructura *on premise* del cliente o en la nube.

Los productos desarrollados pueden ser desplegados en la nube para aprovechar sus beneficios, considerando la alianza estratégica que la organización mantiene con AWS (Amazon Web Services), sin embargo, se ve limitada por temas regulatorios de los países donde serán instaladas las soluciones, en relación con el manejo y seguridad de la información financiera de los clientes, por consiguiente, se contempla el uso de una nube híbrida para sortear dichas limitaciones.

5.1.1. Infraestructura base

A continuación, se detallan las características de los servidores y equipos que se utilizan en la fábrica de software por proyecto en modo *on premise*.

TIPO	OBJETO	DETALLE	MARCA	CANTIDAD	MEMORIA RAM	CORES	PROCESADOR	ESPACIO HD	SO
Servidores	MIDDLEWARE 1	Servidor donde se despliegan los componentes que contienen la lógica de negocio.	CISCO HCI	1	8Gb	3	E52690 2.6GHz	300Gb	RedHat 7.5
	CONTENEDOR 1	Servidor que permite el despliegue de funcionalidades web.	CISCO HCI	1	4Gb	1	E52690 2.6GHz	300Gb	RedHat 7.5
	ACTIVE MQ MASTER	Servidor de gestión de colas de mensajes.	CISCO HCI	1	8Gb	1	E52690 2.6GHz	300Gb	RedHat 7.5
	GESTOR DOCUMENTAL	Servidor de gestión documental, permite almacenamiento y categorización de los activos digitales de la empresa.	CISCO HCI	1	16Gb	2	E52690 2.6GHz	300Gb y 1Tb	RedHat 7.5
	BASE DATOS SYBASE	Servidor que administra la base de datos relacional.	CISCO HCI	1	12Gb	3	UltraSpark3.4GHz	43GB libres	Solaris 10
Equipos de desarrollo y pruebas	COMPUTADOR TIPO1	Ambiente personal de desarrollo de software.	DELL	1/R	16Gb	5	Intel i5	1Tb	Windows 10
	COMPUTADOR TIPO2	Equipo destinado para el equipo de aseguramiento de calidad	DELL	1/R	8Gb	5	Intel i5	500 Gb	Windows 10

Figura 33. Arquitectura actual equipos de un entorno de desarrollo.

A continuación, se muestra un diagrama de la infraestructura actual on premise de la fábrica de software.

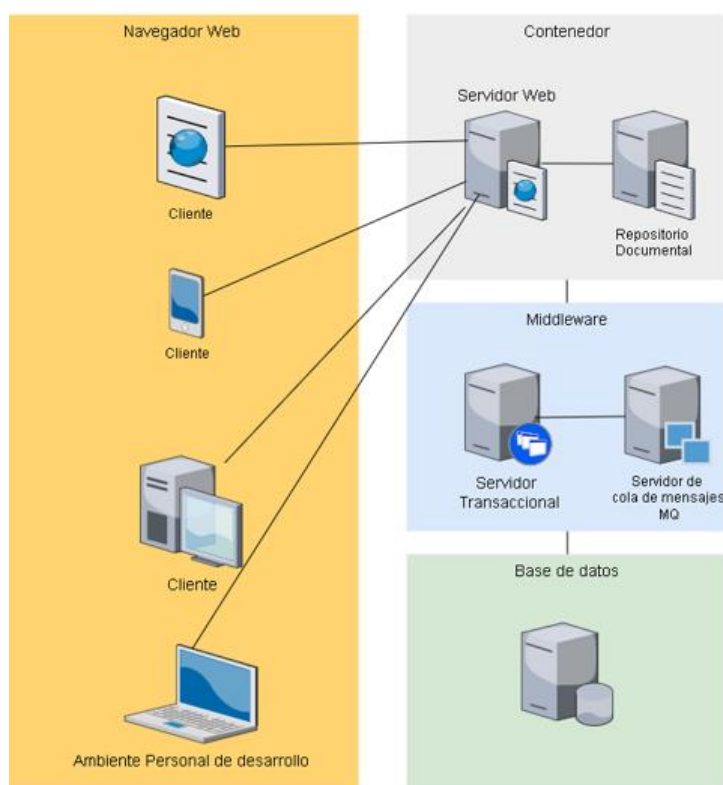


Figura 34. Distribución de un ambiente de desarrollo.

5.1.2. Entornos de desarrollo y pruebas

La fábrica de desarrollo dispone de dos ambientes para el desarrollo y evaluación de productos:

- el primero, dirigido a tareas de construcción de soluciones de software donde se despliegan los servicios desarrollados de manera incremental y que son catalogados conforme se van construyendo.
- el segundo, el ambiente de prueba que contiene las soluciones estables construidas que se someterán a la evaluación funcional por parte del equipo de calidad (QA) para garantizar que cumplen con las expectativas del cliente con un rendimiento favorable.

5.1.3. Entornos de certificación y producción

Son ambientes de propiedad del cliente donde se despliegan de manera manual los componentes que ya superaron la etapa de evaluación de manera satisfactoria y que están listos para la evaluación del usuario final.

Las características técnicas de este ambiente son superiores y están relacionados estrechamente con la demanda de servicios que requiere la entidad financiera.

5.2. Arquitectura de infraestructura base objetivo

Para implementar la infraestructura tecnológica objetivo se mantendrá la arquitectura actual basada en un modelo de nube híbrida.

La computación en la nube es un estilo de computación que considera las capacidades de TI como escalables y elásticas, que se brindan como un servicio mediante las tecnologías del internet.

Una nube híbrida es una combinación de nubes públicas y privadas que permite a las organizaciones de TI convertirse en intermediadores de servicios, algunos de los cuales son ofrecidos por cuenta propia (on premise) y otros adquiridos en orígenes de nube pública.

Es importante citar que el proveedor de servicios en nube de la organización se ubica como líder del mercado de plataformas e infraestructura, según Gartner 2021.



Figura 35. Posición de AWS en el cuadrante de Gartner.

Con base en la arquitectura de referencia, se realiza el análisis de la infraestructura actual bajo los siguientes parámetros.

5.2.1. Agilidad

Permite acceso sencillo a un amplio conjunto de tecnologías que le facilita el innovar con rapidez al crear e implementar servicios tecnológicos que marquen la diferencia en las expectativas de los clientes en cuestión de minutos.

5.2.2. Seguridad

La infraestructura de AWS cuenta con certificaciones en las principales normas y certificaciones internacionales como ISO 27001-9001, PCI nivel 1, entre otras que garantizan el buen manejo de los datos.

5.2.3. Disponibilidad

La plataforma brinda la mayor disponibilidad de red que cualquier otro proveedor en la nube, garantiza un 99.95% mensual.

Se encuentra distribuida geográficamente en regiones aisladas unas de otras, tanto de forma geográfica como es software.

En el caso de incurrir en algún incumplimiento en los niveles de servicios establecidos, se brinda compensaciones mediante un descuento en el cobro del servicio mensual.

5.2.4. Monitoreo

Proporciona varias herramientas de monitoreo que pueden configurarse para que de manera automática efectúen validaciones, se identifiquen inconvenientes y se intente solucionarlos eliminando y volviendo crear servicios, además de ofrecer también monitoreo manual que va a requerir de la acción del administrador para su reparación.

5.2.5. Elasticidad

Es una propiedad importante dentro de la arquitectura en los servicios, se centra en la posibilidad de crecer (adquisición de recursos) o decrecer (liberación de recursos) de manera automática según la demanda de los servicios y que consideran una serie de condiciones o métricas establecidas de antemano, que se ocupan como disparadores, garantizando el aprovisionamiento de recursos que realmente necesita.

5.2.6. Costos

Los servicios en la nube le permiten reemplazar los gastos fijos (que se incurren por la adquisición y mantenimiento de centros de datos y servidores físicos) por gastos variables que se van a calcular a medida que se utilizan, por tanto, los gastos variables son mucho menores que los gastos fijos en adquisiciones que muchas veces son sobredimensionadas al principio.

5.3. Análisis de brechas

Luego de analizar la arquitectura de infraestructura actual en la fábrica de software, podemos concluir que ya cumple con las características de la arquitectura objetivo en los parámetros establecidos. Se aprovecha las ventajas de la plataforma de nube para ofrecer la posibilidad de un despliegue conforme a la necesidad y especificaciones del cliente y lo que permite la regulación, mientras que, en despliegues on premise la infraestructura base la provee la entidad financiera bajo las especificaciones que la empresa provee y la demanda de servicios lo exige.

5.4. Impacto de la arquitectura objetivo en la situación actual

Al concluir que la arquitectura actual coincide con la arquitectura objetivo, el impacto no existe, sin embargo, si se puede determinar

como una acción de mejora la organización debe buscar el máximo aprovechamiento de los componentes necesarios por cuanto representan un costo operativo, que al lograr minimizarse podría convertirse en una ventaja competitiva y contribuir a un mayor nivel de ventas de las soluciones en nube.

6. OPORTUNIDADES Y SOLUCIONES

Este capítulo tiene por objetivo establecer un plan para la implementación de las iniciativas que surgieron como resultado del análisis de las brechas entre la situación actual de la organización y los referentes, propios de cada arquitectura analizada en las fases anteriores.

En la siguiente tabla, se listan las arquitecturas y las iniciativas identificadas que serán desarrolladas a mayor detalle.

Tabla 9.

Iniciativas identificadas por arquitectura

Código	Arquitectura	Iniciativa
INEG1	Negocio	Crear un repositorio histórico que permita almacenar los datos de las estimaciones realizadas en cada proyecto.
INEG2	Negocio	Fortalecer el proceso de estimación mediante el registro de riesgos inherentes a la construcción de software para mantener un histórico.
INEG3	Negocio	Fortalecer la actividad de validación y verificación para garantizar que el producto de software construido funcione correctamente.
INEG4	Negocio	Fortalecer el proceso de valoración del desempeño para incorporar los aspectos de autonomía, influencia, complejidad, habilidad para negocios.
IAPL1	Aplicaciones	Implementar JUnit como herramienta para pruebas automatizadas.
IAPL2	Aplicaciones	Implementar JMeter como herramienta para pruebas de carga y rendimiento.
IDAT1	Datos	Fortalecer el procedimiento para clasificar adecuadamente la información en función de su valor, sensibilidad para su debido tratamiento.
IDAT2	Datos	Fortalecer los controles de acceso a equipos, servidores, al almacenamiento en sistemas locales o en nube y a la red para evitar acceso no autorizado.
IDAT3	Datos	Fortalecer campañas de formación y sensibilización sobre la importancia de la confidencialidad y protección de la información.
IDAT4	Datos	Mantener los acuerdos de confidencialidad formalizados con los colaboradores que acceden y manejan información confidencial.

6.1. Arquitectura de negocio

La fábrica del área de ingeniería no cuenta con un procedimiento estandarizado para la estimación de los proyectos de implementación de software, que aproveche la experiencia de otros proyectos con similares características. Depende en gran medida de las técnicas de estimación basadas en la analogía o el juicio de expertos que en la mayoría de los casos son subjetivas y derivan en una estimación sub dimensionada.

6.1.1. Iniciativa: Crear un repositorio histórico de estimaciones

Esta iniciativa surge como respuesta a la necesidad de manejar un sistema centralizado que permita el registro de las condiciones, restricciones y características de los proyectos de desarrollo en la fábrica y que sea un medio para obtener datos que contribuyan a mejorar el proceso de estimación de los siguientes proyectos.

6.1.1.1. Conceptualización de la implementación

Para implementar el repositorio que almacenará información referente a las estimaciones de los proyectos es necesario considerar los aspectos de la figura 36.

INEG1		Crear un repositorio histórico de estimaciones													
Catalizadores COBIT relacionados <input type="checkbox"/> Políticas <input checked="" type="checkbox"/> Procesos <input type="checkbox"/> Organización <input type="checkbox"/> Cultura <input type="checkbox"/> Habilidades <input checked="" type="checkbox"/> Información <input checked="" type="checkbox"/> Tecnología		Situación Actual y Principales Observaciones <p>La fabrica de software no cuenta con una plataforma unificada y en línea que permita estimar proyectos de implementación de software. Es necesario buscar rentabilidad de los proyectos evitando la tendencia a subestimar el esfuerzo de desarrollo atribuido al exceso de optimismo y a la falta de experiencia de los miembros del equipo. Formar una base histórica que contribuya al refinamiento de la estimación de los próximos proyectos a partir de los datos generados.</p>													
Requerimientos de implementación <input type="checkbox"/> Normativa <input type="checkbox"/> Hardware <input checked="" type="checkbox"/> Software <input checked="" type="checkbox"/> Servicios		Actividades Fundamentales: <ul style="list-style-type: none"> Desarrollar un modelo de estimación que contemple la cantidad de recursos, tiempos, funcionalidades, número de componentes, características, requisitos, líneas de código, puntos de historia. Definir historias de usuario y los criterios de aceptación de las funcionalidades necesarias para el registro del proyecto y la estimación. Priorizar la lista de historias de usuario según la importancia del negocio. Desarrollar la plataforma a la medida de la organización que permita el registro del proyecto, de las características y sus condiciones, de los riesgos y excepciones, del equipo para generar una estimación. Almacenar en históricos la información de cada proyecto registrado. Realizar respaldos diarios de la información generada por la plataforma. 													
Acercos de la Iniciativa Impacto: Medio Esfuerzo: Medio		Roadmap FASE: 2023 AREA: Negocio													
Costos referenciales 12k		Plazo de Implementación <input type="checkbox"/> < 2 meses <input type="checkbox"/> < 6 meses <input type="checkbox"/> < 12 meses <input checked="" type="checkbox"/> > 12 meses													
Actividades Macro		Participantes <table border="1"> <tr><td>CEO</td><td>A</td></tr> <tr><td>PMO</td><td>R</td></tr> <tr><td>Financiero</td><td>R</td></tr> <tr><td>Operaciones</td><td>R</td></tr> <tr><td>Comercial</td><td></td></tr> <tr><td>Gerente de TI</td><td>R</td></tr> </table>		CEO	A	PMO	R	Financiero	R	Operaciones	R	Comercial		Gerente de TI	R
CEO	A														
PMO	R														
Financiero	R														
Operaciones	R														
Comercial															
Gerente de TI	R														

Figura 36. Conceptualización de la iniciativa INEG1.

6.1.2. Iniciativa: Fortalecer el proceso de estimación mediante el registro de riesgos

Para fortalecer el proceso de estimación es necesario identificar factores que afectaron a otros proyectos en la etapa de construcción en términos de tiempos, calidad, recursos y costos adicionales, por tanto, es importante desarrollar la interfase entre el sistema de gestión de riesgos y el sistema centralizado de estimación.

6.1.2.1. Conceptualización de la implementación

Para fortalecer el proceso de estimación y su relación con el procedimiento de estimación es necesario considerar los aspectos de la figura 36.

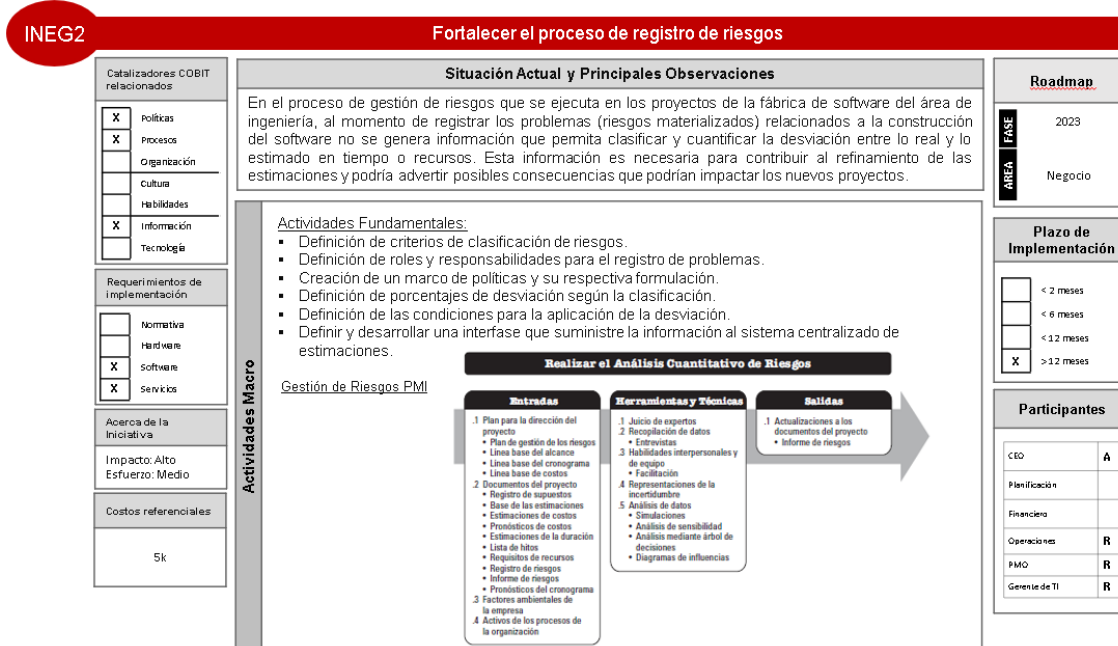


Figura 37. Conceptualización de la iniciativa INEG2.

6.1.3. Iniciativa: Fortalecer el proceso de aseguramiento de la calidad del software

La fábrica de software utiliza el marco de trabajo Ágil, que se complementa con las áreas de capacidad de CMMI.

Para el aseguramiento de la calidad se considera el área ENQ- Asegurando la calidad que contempla cuatro áreas prácticas (CMMI-2018):

- RDM-Desarrollo y gestión de requisitos
- PQA-Proceso de aseguramiento de calidad
- VV-Verificación y validación
- PR-Revisiones hechas por colegas

“La objetividad en las evaluaciones de aseguramiento de la calidad del proceso es fundamental para el éxito del proyecto” (CMMI-2018). Eso implica que los evaluadores no deben evaluar su propio trabajo por tanto es primordial que se establezcan revisiones por pares que son

comprobaciones de los procesos realizados o productos construidos cumpliendo criterios de aceptación.

El aseguramiento de la calidad permite verificar el cumplimiento y satisfacción de los requisitos y que el producto funciona como se esperaba en un ambiente determinado.

6.1.3.1. Conceptualización de la implementación

Como paso inicial se plantea la necesidad de verificar la madurez del proceso en la fábrica de software mediante una nueva ejecución que identifique brechas.

A continuación, se detallan los aspectos principales a considerar.



Figura 38. Conceptualización de la iniciativa INEG3.

6.1.4. Iniciativa: Fortalecer el proceso de valoración del desempeño de los colaboradores

La evaluación del desempeño es una herramienta que proporciona grandes beneficios para los evaluadores y evaluados que repercuten en el éxito empresarial.

Es primordial para la fábrica de software contar con recursos calificados en la tecnología, en las herramientas de desarrollo, pero adicional, que tengan conocimiento del negocio y que identifiquen las necesidades del cliente.

6.1.4.1. Conceptualización de la implementación

Los objetivos de la evaluación son:

- motivar a un alto nivel de calidad y cantidad de trabajo, esto es posible cuando se han establecido objetivos a cumplir y se comprenden claramente.
- fomentar el desarrollo personal al detectar necesidades de los empleados y que permiten ofrecer iniciativas que ayuden en su crecimiento personal.
- proporcionar un plan de mejora para garantizar que los colaboradores comprenden los requisitos e imposiciones de su puesto de trabajo, mismas identificadas a partir del estándar SFIA.

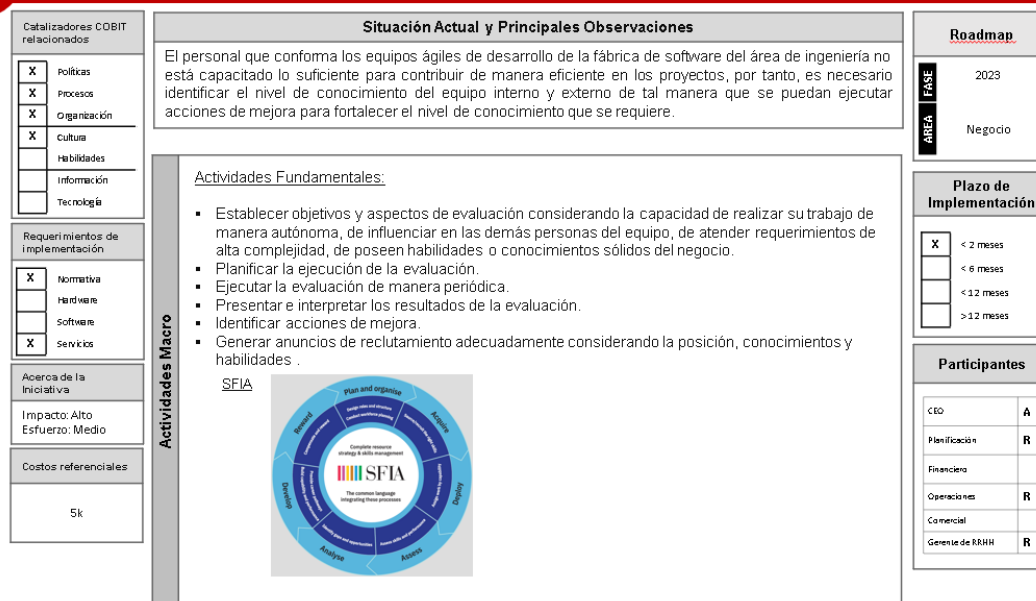


Figura 39. Conceptualización de la iniciativa INEG4.

6.2. Arquitectura de Aplicaciones y Datos

6.2.1. Iniciativa: Implementar JUnit como herramienta de pruebas automatizadas

La implementación de la herramienta tiene una participación relevante en la fábrica de software y busca lograr mayor confiabilidad en los productos construidos.

JUnit	
Tipo	Herramienta para prueba unitaria
Usuarios	Equipo Scrum
Casos de Uso	Desarrollo ágil de software Gestión de Pruebas Supervisión de errores Generación de Indicadores de confiabilidad y completitud
Integraciones	Java
Alojamiento	Local
Licencia	Common Public License
Costo	Sin Costo

Figura 40. Detalle de la iniciativa IAPL1.

6.2.1.1. Conceptualización de la implementación

La siguiente figura muestra los principales aspectos de la implementación.

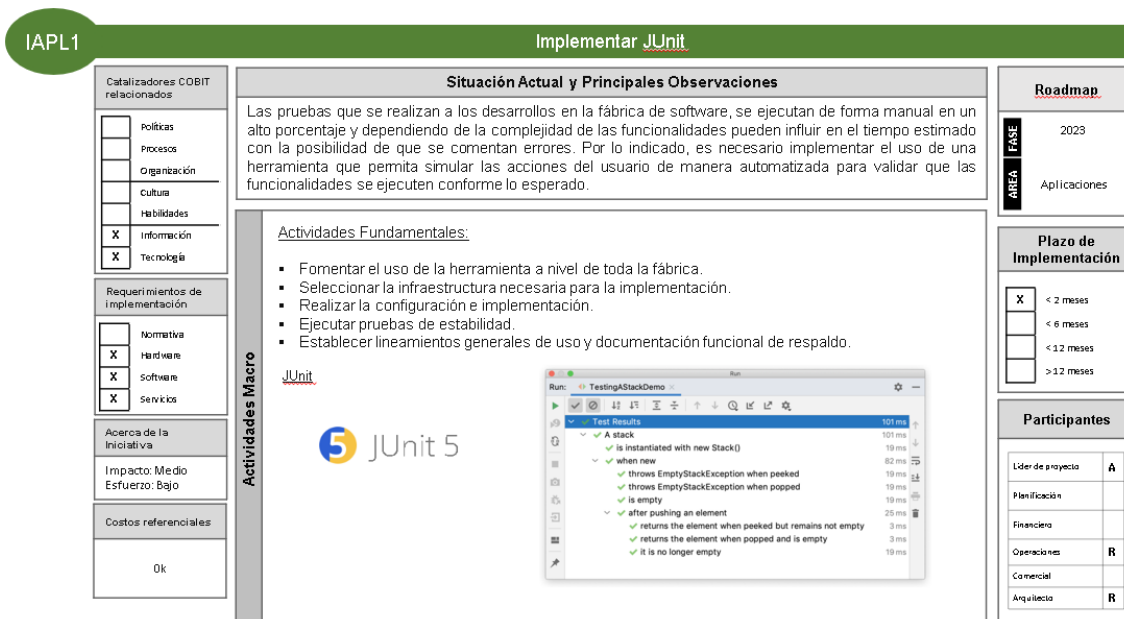


Figura 41. Conceptualización de la iniciativa IAPL1.

6.2.2. Implementar JMeter como herramienta para pruebas de carga y rendimiento

Para garantizar la calidad del software que se construye en la fábrica de ingeniería, es necesario demostrar que cumple con los estándares de seguridad, funcionalidad y rendimiento.

El rendimiento en el software se define como la capacidad de utilizar eficientemente los recursos de hardware donde fue desplegado.

6.2.2.1. Conceptualización de la implementación

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

IAPL2
Implementar JMeter

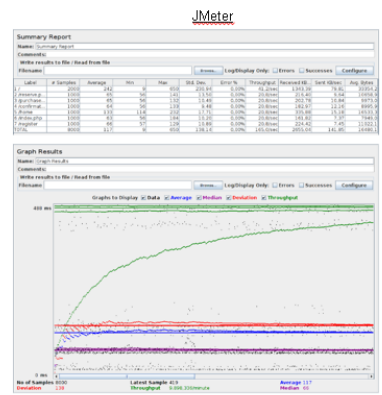
<p>Catalizadores COBIT relacionados</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td><input type="checkbox"/></td><td>Políticas</td></tr> <tr><td><input type="checkbox"/></td><td>Procesos</td></tr> <tr><td><input type="checkbox"/></td><td>Organización</td></tr> <tr><td><input type="checkbox"/></td><td>Cultura</td></tr> <tr><td><input type="checkbox"/></td><td>Habilidades</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Información</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Tecnología</td></tr> </table> <p>Requerimientos de implementación</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td><input type="checkbox"/></td><td>Normativa</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Hardware</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Software</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>Servicios</td></tr> </table> <p>Acercos de la Iniciativa</p> <p>Impacto: Medio Esfuerzo: Bajo</p> <p>Costos referenciales</p> <p style="text-align: center;">Ok</p>	<input type="checkbox"/>	Políticas	<input type="checkbox"/>	Procesos	<input type="checkbox"/>	Organización	<input type="checkbox"/>	Cultura	<input type="checkbox"/>	Habilidades	<input checked="" type="checkbox"/>	Información	<input checked="" type="checkbox"/>	Tecnología	<input type="checkbox"/>	Normativa	<input checked="" type="checkbox"/>	Hardware	<input checked="" type="checkbox"/>	Software	<input checked="" type="checkbox"/>	Servicios	<p style="text-align: center;">Situación Actual y Principales Observaciones</p> <p>Las pruebas que se realizan en la fábrica de software se ejecutan de forma manual utilizando un conjunto de datos establecidos como muestra y por un grupo acotado de especialistas que se centran en garantizar la funcionalidad, pero, en entornos productivos, la realidad cambia y se requiere garantizar que los desarrollos mantengan el mejor rendimiento con muchos usuarios más conectados al tiempo, por tal razón es necesario el uso de una herramienta que permita ejecutar pruebas de carga y garantizar el rendimiento.</p> <p>Actividades Fundamentales:</p> <ul style="list-style-type: none"> Calcular la cantidad de datos que sirven como muestra que se utilizarán en la simulación. Seleccionar o preparar los datos más idóneos conforme a la necesidad reflejada en los casos de prueba. Programar el escenario de pruebas para los casos identificados. Ejecutar pruebas de carga, de estrés, de estabilidad, de pico para establecer la estabilidad de la funcionalidad. Identificar y ajustar el desarrollo para resolver las novedades encontradas. <div style="text-align: center;">  </div>	<p style="text-align: center;">Roadmap</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="background-color: #f0f0f0;">FASE</td><td style="text-align: center;">2023</td></tr> <tr><td style="background-color: #f0f0f0;">AREA</td><td style="text-align: center;">Aplicaciones</td></tr> </table> <p style="text-align: center;">Plazo de Implementación</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td><input checked="" type="checkbox"/></td><td>< 2 meses</td></tr> <tr><td><input type="checkbox"/></td><td>< 6 meses</td></tr> <tr><td><input type="checkbox"/></td><td>< 12 meses</td></tr> <tr><td><input type="checkbox"/></td><td>> 12 meses</td></tr> </table> <p style="text-align: center;">Participantes</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Lider de proyecto</td><td style="text-align: center;">A</td></tr> <tr><td>Planificación</td><td></td></tr> <tr><td>Financiero</td><td></td></tr> <tr><td>Operaciones</td><td style="text-align: center;">R</td></tr> <tr><td>Comercial</td><td></td></tr> <tr><td>Arquitecto</td><td style="text-align: center;">R</td></tr> </table>	FASE	2023	AREA	Aplicaciones	<input checked="" type="checkbox"/>	< 2 meses	<input type="checkbox"/>	< 6 meses	<input type="checkbox"/>	< 12 meses	<input type="checkbox"/>	> 12 meses	Lider de proyecto	A	Planificación		Financiero		Operaciones	R	Comercial		Arquitecto	R
<input type="checkbox"/>	Políticas																																															
<input type="checkbox"/>	Procesos																																															
<input type="checkbox"/>	Organización																																															
<input type="checkbox"/>	Cultura																																															
<input type="checkbox"/>	Habilidades																																															
<input checked="" type="checkbox"/>	Información																																															
<input checked="" type="checkbox"/>	Tecnología																																															
<input type="checkbox"/>	Normativa																																															
<input checked="" type="checkbox"/>	Hardware																																															
<input checked="" type="checkbox"/>	Software																																															
<input checked="" type="checkbox"/>	Servicios																																															
FASE	2023																																															
AREA	Aplicaciones																																															
<input checked="" type="checkbox"/>	< 2 meses																																															
<input type="checkbox"/>	< 6 meses																																															
<input type="checkbox"/>	< 12 meses																																															
<input type="checkbox"/>	> 12 meses																																															
Lider de proyecto	A																																															
Planificación																																																
Financiero																																																
Operaciones	R																																															
Comercial																																																
Arquitecto	R																																															

Figura 42. Conceptualización de la iniciativa IAPL2.

7. PLANIFICAR LA MIGRACIÓN

7.1. Priorización

7.1.1. Análisis de impacto

A continuación, se establecen la escala que se utilizará para el análisis de impacto de las iniciativas y los objetivos del ejercicio.

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

		Objetivos						Impacto	
		20%	15%	15%	20%	15%	15%		
		Habilitando o en marcha	Desarrollar software con calidad en tiempo y forma evitando reprocesos y reutilizaciones	Evitar subestimar los esfuerzos y presupuestos de los proyectos	Evaluar una nueva valoración de madurez del proceso de desarrollo de software	Implementar aplicaciones que actúen al dimensionamiento de los proyectos	Ejecutar procesos de valoración de desempeño del personal técnico	Establecer un modelo de estimación que contribuya a un mejor dimensionamiento	Valoración cualitativa
Domino	Id	Iniciativa							
INNOVATION	INEG1	Crear un repositorio histórico que permita almacenar los datos de las estimaciones realizadas en cada proyecto.							1,00
INNOVATION	INEG2	Fortalecer el proceso de estimación mediante el registro de riesgos inherentes a la construcción de software para mantener un histórico.							1,00
IT EXCELLENCE	INEG3	Fortalecer la actividad de validación y verificación para garantizar que el producto de software construido funcione correctamente.							0,70
BUSINESS SUPPORT	INEG4	Fortalecer el proceso de valoración del desempeño para incorporar los aspectos de autonomía, influencia, complejidad, habilidad para negocios.							0,70
IT EXCELLENCE	IAPL1	Implementar JUnit como herramienta para pruebas automatizadas.							0,40
IT EXCELLENCE	IAPL2	Implementar JMeter como herramienta para pruebas de carga y rendimiento.							0,40
BUSINESS SUPPORT	IDAT1	Fortalecer el procedimiento para clasificar adecuadamente la información en función de su valor, sensibilidad para su debido tratamiento.							0,60
BUSINESS SUPPORT	IDAT2	Fortalecer los controles de acceso a equipos, servidores, al almacenamiento en sistemas locales o en nube y a la red para evitar acceso no autorizado.							0,60
BUSINESS SUPPORT	IDAT3	Fortalecer campañas de formación y sensibilización sobre la importancia de la confidencialidad y protección de la información.							0,70
BUSINESS SUPPORT	IDAT4	Mantener los acuerdos de confidencialidad formalizados con los colaboradores que acceden y manejan información confidencial.							0,55

Figura 43. Análisis de impacto.

7.1.2. Análisis de Esfuerzo

Para establecer el esfuerzo que significan implementar las iniciativas en términos de recursos económicos, complejidad y capacidad de TI.

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

Análisis de Esfuerzo

No	Área	Id	Iniciativa	Habilitante o en marcha	Criterios Esfuerzo			Suma ponderada	Esfuerzo
					40%	30%	30%		
1	INNOVATION	INEG1	Crear un repositorio histórico que permita almacenar los datos de las estimaciones realizadas en cada proyecto.	▲	▲	●	1,70	Bajo	
2	INNOVATION	INEG2	Fortalecer el proceso de estimación mediante el registro de riesgos inherentes a la construcción de software para mantener un histórico.	▲	▲	●	1,60	Bajo	
3	IT EXCELLENCE	INEG3	Fortalecer la actividad de validación y verificación para garantizar que el producto de software construido funcione correctamente.	●	▲	●	1,30	Bajo	
4	BUSINESS SUPPORT	INEG4	Fortalecer el proceso de valoración del desempeño para incorporar los aspectos de autonomía, influencia, complejidad, habilidad para negocios.	●	●	●	1,00	Bajo	
5	IT EXCELLENCE	IAPL1	Implementar JUnit como herramienta para pruebas automatizadas.	●	●	●	1,00	Bajo	
6	IT EXCELLENCE	IAPL2	Implementar JMeter como herramienta para pruebas de carga y rendimiento.	●	●	●	1,00	Bajo	
7	BUSINESS SUPPORT	IDAT1	Fortalecer el procedimiento para clasificar adecuadamente la información en función de su valor, sensibilidad para su debido tratamiento.	▲	●	●	1,40	Bajo	
8	BUSINESS SUPPORT	IDAT2	Fortalecer los controles de acceso a equipos, servidores, al almacenamiento en sistemas locales o en nube y a la red para evitar acceso no autorizado.	▲	●	●	1,40	Bajo	
9	BUSINESS SUPPORT	IDAT3	Fortalecer campañas de formación y sensibilización sobre la importancia de la confidencialidad y protección de la información.	●	●	●	1,00	Bajo	
10	BUSINESS SUPPORT	IDAT4	Mantener los acuerdos de confidencialidad formalizados con los colaboradores que acceden y manejan información confidencial.	●	●	●	1,00	Bajo	

Figura 44. Análisis de esfuerzo.

7.1.3. Fases

Para establecer las fases, se ha identificado el impacto y el esfuerzo de cada iniciativa. Adicional, se considera la capacidad del equipo para no afectar las actividades ya comprometidas.

Dominio		Id		Iniciativa		Habilitante o en marcha	Impacto	Esfuerzo	Prioridad	Fase
INNOVATION	INEG1	Crear un repositorio histórico que permita almacenar los datos de las estimaciones realizadas en cada proyecto.	0	Medio	Bajo	Habilitante	1			
INNOVATION	INEG2	Fortalecer el proceso de estimación mediante el registro de riesgos inherentes a la construcción de software para mantener un histórico.	Medio	Bajo	Habilitante	1				
IT EXCELLENCE	INEG3	Fortalecer la actividad de validación y verificación para garantizar que el producto de software construido funcione correctamente.	Bajo	Bajo	Alta	1				
BUSINESS SUPPORT	INEG4	Fortalecer el proceso de valoración del desempeño para incorporar los aspectos de autonomía, influencia, complejidad, habilidad para negocios.	Bajo	Bajo	Habilitante	2				
IT EXCELLENCE	IAPL1	Implementar JUnit como herramienta para pruebas automatizadas.	Bajo	Bajo	Media Alta	1				
IT EXCELLENCE	IAPL2	Implementar JMeter como herramienta para pruebas de carga y rendimiento.	Bajo	Bajo	Media Alta	1				
BUSINESS SUPPORT	IDAT1	Fortalecer el procedimiento para clasificar adecuadamente la información en función de su valor, sensibilidad para su debido tratamiento.	Bajo	Bajo	Media	2				
BUSINESS SUPPORT	IDAT2	Fortalecer los controles de acceso a equipos, servidores, al almacenamiento en sistemas locales o en nube y a la red para evitar acceso no autorizado.	Bajo	Bajo	Media Alta	1				
BUSINESS SUPPORT	IDAT3	Fortalecer campañas de formación y sensibilización sobre la importancia de la confidencialidad y protección de la información.	Bajo	Bajo	Media	1				
BUSINESS SUPPORT	IDAT4	Mantener los acuerdos de confidencialidad formalizados con los colaboradores que acceden y manejan información confidencial.	Bajo	Bajo	Media	1				

Figura 45. Proyectos y fases.

7.2. Análisis de dependencias

Para garantizar la implementación del fortalecimiento de la valoración de desempeño se ha establecido que forme parte de la fase dos, de manera habilitante.

7.3. Mapa de ruta

Para lograr la implementación de las iniciativas se plantea el siguiente mapa de ruta. El mismo que ha sido priorizado conforme el valor que aporta al negocio y su importancia.



Figura 46. Mapa de ruta.

8. CONCLUSIONES Y RECOMENDACIONES

8.1. Conclusiones

- La necesidad del cliente normalmente se refleja en las órdenes de trabajo y su entendimiento es la actividad más importante, y garantiza la efectividad del proceso de estimación en la fábrica de software, toda información histórica que no sea registrada responsablemente podría perjudicar las estimaciones resultando en un sobre dimensionamiento que afectaría la credibilidad de la herramienta e incluso obligando su desuso.
- La aplicación integral del método ADM a la resolución del caso asegura resultados, por consiguiente, es importante socializar con los interesados el objetivo de identificar el escenario de solución más adecuado en la fase de visionamiento, para que las iniciativas que se determinen en las siguientes fases, se encaminen con certeza a cumplir sus expectativas.
- La calidad de un producto de software no puede determinarse solamente por el cumplimiento funcional que se otorga al cliente, solamente se consolida cuando se garantiza su escalabilidad, fiabilidad y el buen uso de los recursos y que deben contemplarse dentro de los criterios de “terminado” de las historias de usuario.
- Para que la fábrica de software soporte al negocio de una manera efectiva se deben garantizar un conjunto de capacidades como el conocimiento del negocio, la gestión adecuada de personal, conocimiento técnico amplio en los colaboradores, procesos ágiles que soporten la operación y herramientas de vanguardia que automaticen tareas.

8.2. Recomendaciones

- Considerando que el proceso de estimación consolidará información producto de la experiencia en otros proyectos para generar una base histórica de datos importantes, se recomienda la aplicación de analítica e inteligencia de negocios para establecer patrones, tendencias y correlaciones que permitan a futuro enriquecer el afinamiento y lograr que cada vez sea más acertado.
- Las evaluaciones de desempeño permiten identificar las debilidades y fortalezas del equipo que forma parte de la fábrica de software con el fin de buscar estrategias para afinar la eficacia en los proyectos, pero adicional se recomienda que la organización gestione eficazmente los procesos de selección considerando los perfiles necesarios para la fábrica y como complemento, una estrategia de fidelización para disminuir la rotación del personal, factor que también influye en el cumplimiento de los objetivos.
- Debido a la importancia del giro de negocio de la organización que es brindar servicios y productos de software a entidades financieras, se recomienda aplicar procesos de revisión de fuentes (manual o automática) para verificar que el código generado cumple con los estándares de seguridad establecidos por la empresa y generar un reporte de conformidad para garantizar la fiabilidad del producto instalado.
- Como complemento a las campañas de sensibilización sobre la importancia de la confidencialidad y protección de la información, se sugiere que se incluyan contenidos relacionados con ciberseguridad, con especial énfasis en los tipos de ataques de ingeniería social para que la comunidad pueda identificarlos y responda de manera adecuada.

9. Referencias

Cobiscorp (Nosotros-Clientes,2022)

<https://www.cobiscorp.com/nosotros/>

Previde, G. E. (2004). Evolución de la Oficina de Proyectos "De un inventario a una PMO": Evolution of the Project Office " From an inventory to a PMO" Paper presented at PMI® Global Congress 2004—Latin America, Buenos Aires, Argentina. Newtown Square, PA: Project Management Institute.

Project Management Institute, Inc Guía del PMBOK, (Sexta Edición,2017) Chicago, IL 60610 EE.UU, Independent Publishers Group.

Scaled Agile, Inc. (s. f.). Business Owners. Business Owners Scaled Agile. Recuperado 4 de abril de 2022, de <https://www.scaledagileframework.com/business-owners/>

Marco de competencias para la era de la información. (2018). The SFIA Foundation Ltd.

Mitre Hernández, H. A., Ortega Martínez, E., & Lemus Olalde, C. (2015). Estimación y control de costos en métodos ágiles para desarrollo de software: un caso de estudio. Ingeniería Investigación Y Tecnología, 15(3). Recuperado a partir de <https://www.revistas.unam.mx/index.php/ingenieria/article/view/45871>

Owais, M., & Ramakishore, R. (2017). Effort, duration and cost estimation in agile software development. Paper presented at the 2016 9th International Conference on Contemporary Computing, IC3 2016, doi:10.1109/IC3.2016.7880216 Recuperado de www.scopus.com

Morales.O. (2022) A software-based cost estimation technique in scrum using a developer's expertise, *Advances in Engineering Software*. Elsevier. Recuperado de <https://www.sciencedirect.com/science/article/abs/pii/S0965997822000709?via%3Dihub#preview-section-cited-by>

ANEXOS

ANEXO I ABREVIACIONES

PR	Proceso
PMO	Oficina de Administración de Proyectos
SAFe	Scaled Agile Framework
PM	Product Manager
BO	Business Owner
PO	Product Owner
SA	System Architect
RTE	Release Train Engineer
DM	Developer Manager
SM	Scrum Master
ST	System Team
WS	Working Software
SI	Incremento de Sistema
RTE	Release Train Engineer
ART	Tren de Liberación Ágil (Agile Release Train)
ON PREMISE	Componentes residentes en las instalaciones de la persona u organización.

