



Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ciencias de la Computación

Estandarización del Proceso de Desarrollo y Puesta en Producción de Aplicaciones en las Gerencias TI del Banco Estado

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN TECNOLOGÍAS DE LA
INFORMACIÓN

ARIEL GUSTAVO CAMUSET AVILEZ

PROFESOR GUÍA:
SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:
DANIEL PEROVICH GEROSA
MARÍA CECILIA BASTARRICA PIÑEYRO
IVAN SIPIRAN MENDOZA
ALCIDES QUISPE SANCA

SANTIAGO DE CHILE
2023

Resumen

Este trabajo de tesis se enmarca en la organización Banco Estado de Chile, más conocido como Banco Estado. Esta organización posee un plan estratégico u hoja de ruta para los próximos tres años. Para ayudar a abordar esta hoja de ruta, el Banco Estado cuenta con dos gerencias de TI las cuales generaron un plan estratégico para el desarrollo de sistemas, y con eso poder alinearse con la hoja de ruta de la organización. Además, se iniciaron diversos proyectos que permitirán escalar las capacidades de desarrollo del Banco. Dentro de estos proyectos está la iniciativa llamada *“integrar DevOps al proceso de desarrollo”*, que tiene como principal objetivo crear un único proceso estandarizado para el ciclo de desarrollo de un requerimiento; esto, independiente de la tecnología que se utilice.

En una primera instancia se trabajó en identificar los problemas en el flujo asociado a la puesta en producción de los sistemas o piezas de software, para luego buscar dar solución a dichos problemas. Teniendo en cuenta los dolores e insatisfacciones recopilados a lo largo de este trabajo de tesis, se creó un proceso mejorado de desarrollo, y todo lo que comprende el uso de esta solución. Para orquestar todos los cambios requeridos por este proyecto, se utilizó la suite UrbanCode de IBM.

Dado que el objetivo de esta tesis es mejorar el proceso de desarrollo y puesta en producción de los productos de software desarrollados por las gerencias de TI de Banco Estado, se trabajó en su refinamiento, estandarización y automatización. La solución creada fue validada para determinar en qué medida los problemas identificados fueron resueltos con el nuevo proceso.

Los resultados de la evaluación indican que el proceso creado es mejor que lo que ya existía. Aunque los resultados aún son pocos, en términos de cantidad, como para sacar conclusiones definitivas, estos están alineados con los objetivos planteados en el trabajo de tesis. Esto hace prever que la organización está ahora en una mejor posición para enfrentar su plan estratégico de TI, así como su evolución interna de cara al futuro.

Tabla Contenido

Resumen	i
1. Introducción	1
1.1. Contexto Organizacional	1
1.2. Problema Abordado	2
1.3. Objetivos de la Tesis.....	4
1.4. Resumen de la Solución Desarrollada	5
1.5. Metodología.....	7
1.6. Estructura del documento	9
2. Marco Teórico.....	10
2.1. Introducción a DevOps.....	10
2.2. Experiencias similares con DevOps	11
3. Proceso de Desarrollo de las Aplicaciones del Banco Estado.....	13
3.1. Proceso de desarrollo base de un software en Banco Estado	13
3.1.1. Actividades del proceso desarrollo base	13
3.1.2. Diferencias de actividades respecto al proceso desarrollo base.....	16
Proceso base COBOL.....	16
Proceso base .net/java	17
Proceso base bus de servicio	18
3.2. Tecnologías de apoyo al proceso de desarrollo base	19
3.3. Metodología de apoyo para identificar los problemas del proceso desarrollo base	20
3.3.1. Protocolo de Identificación de problemas e insatisfacciones	21
3.3.2. Identificación de problemas e insatisfacciones	21
3.3.3. Ejecución del método KPT	24
Gerencia de Sistemas (GSIS).....	24
Gerencia de Tecnología (GTEC)	25
Gerencia de Operaciones (GOP).....	26
3.4. Análisis situación actual	26
Criterio I.....	26
Criterio II.....	27
4. Solución desarrollada	31

4.1.	Contexto, Objetivos y Principios	31
4.1.1.	Contexto	31
4.1.2.	Objetivos	31
4.1.3.	Principios técnicos y de negocio	32
4.2.	Proceso de Desarrollo Refinado	33
4.2.1.	Actividades y técnicas del proceso desarrollo refinado	33
4.2.2.	Roles, equipos que interactúan en el proceso desarrollo	37
4.3.	Plataforma Tecnológica	37
4.3.1.	Herramientas tecnológicas de apoyo	37
4.3.2.	Ambientes de Trabajo	39
4.4.	Análisis Cobertura de Objetivos	40
5.	Validación	42
5.1.	Validación Pilotos	42
5.2.	Validación de la solución	45
5.2.1.	Registro de Validación de la solución	45
5.2.2.	Ejecución de la entrevista	46
5.2.3.	Resultados de la encuesta	47
5.2.4.	Análisis	50
6.	Conclusiones y Trabajo a Futuro	52
7.	Bibliografías	54
8.	Anexos	56
	Anexo A: Actividades del proceso base, descrito por fase	56
	Anexo B: Roles de los participantes en el proceso	59
	B.1. Cliente	59
	B.2. Expertos	61
	B.3. Especialistas	63
	Anexo C: Descripción de los principales dolores detectados	77

1. Introducción

1.1. Contexto Organizacional

Banco Estado es el único banco comercial estatal de Chile. Fue creado en 1953 por el DFL 126, el cual dispuso la fusión de la Caja Nacional de Ahorros, la Caja de Crédito Hipotecario, la Caja de Crédito Agrario y el Instituto de Crédito Industrial. Éste presta servicios financieros a particulares y empresas, siendo sus principales giros de negocio los ahorros y los préstamos hipotecarios. Además, por medio de la Cuenta Única Fiscal, administrada por la Tesorería General de la República, es la institución que realiza todas las actividades financieras del Fisco de Chile.

Banco Estado cuenta con dos gerencias dentro de la Gerencia TI: la *gerencia de sistemas* y la *gerencia de tecnología*. La primera de ellas está a cargo de la planificación, diseño, instalación y administración de los sistemas de información administrativos. Por su parte, la gerencia de tecnología, cuya responsabilidad es dirigir y planificar las asesorías tecnológicas a las áreas de negocio respecto a requerimientos provenientes de las áreas de sistemas, comunicaciones, redes e infraestructura tecnológica.

En el año 2017, la gerencia de TI de Banco Estado generó un plan estratégico de sistemas, el cual fue presentado al comité de Procesos y Tecnologías de Banco Estado, y fue aprobado para dar comienzo ese mismo año. Este nuevo plan estratégico de sistemas se llama *Programa de Transformación*, y está basado en cuatro ámbitos: *estabilidad, arquitectura empresarial, tercerización y mejora del proceso de desarrollo*.

Con el objetivo de alinearse con este Programa de Transformación, la Gerencia de Sistemas con apoyo de la Gerencia de Tecnología, ha estado trabajando en el ámbito de *mejora del proceso de desarrollo*. Particularmente, creando un conjunto de proyectos que permitirán lograr el escalamiento de las capacidades de desarrollo, que es el principal objetivo a cubrir de este ámbito.

Los proyectos que apuntan a lograr ese escalamiento de las capacidades de desarrollo tienen diversos objetivos, por ejemplo:

- Agilizar el proceso de desarrollo y su calidad.
- Implementar un modelo para proyectar, aumentar y optimizar las capacidades de desarrollo.
- Definir un modelo de desarrollo ágil que integre metodologías, prácticas de desarrollo y operación.
- Incluir metodologías de desarrollo ágil en la Gerencia de Sistemas.
- Adecuar el gobierno de la Gerencia TI (gerencia de desarrollo y la gerencia de sistemas).
- Reducir brechas y aumentar entrenamientos y capacidades para asumir nuevos roles y funciones.

Entendiendo que estos son los proyectos que buscan cubrir todas las aristas del escalamiento de las capacidades de desarrollo, se crearon un conjunto de iniciativas para apoyarlos. Dentro de ellas nace la iniciativa llamada *Integrar DevOps al proceso de desarrollo*, la cual tiene como principal objetivo crear un único proceso estandarizado para el ciclo de desarrollo de un requerimiento; esto, independiente de la tecnología que se utilice.

Actualmente, para llevar a producción todas las mantenciones y nuevos desarrollos, la Gerencia de Sistemas trabaja en base a un modelo en cascada, que permite ordenar los tiempos y actividades proyectadas para poner en producción el software asociado a un requerimiento. Las fases de trabajo para llegar a producción son cuatro: *Desarrollo* (construcción del producto, chequeo de normativas por el equipo que desarrolla el producto), *Testing* (validación de normativas tanto propias del banco como de la tecnología utilizada, chequeo de código fuente y pruebas de funcionalidad), *QA* (análisis de vulnerabilidades, stress y performance) e *Implantación* (instalación del producto en producción). Sin embargo, las actividades concretas que se hacen en cada fase difieren significativamente, dependiendo de la tecnología que se usa para construir las aplicaciones. Actualmente, los tipos de tecnologías que se trabajan en la Gerencia de Sistemas son COBOL, Bus de Servicios (en datapower), Bus de Servicios (IIB9) y Plataforma Universal (PU) que comprende los desarrollos en .NET y Java.

1.2. Problema Abordado

Cuando la Gerencia de Sistemas creó la iniciativa *Integrar DevOps al proceso de desarrollo*, identificó un conjunto de problemas en el flujo de puesta en producción de los sistemas. Entonces, la Gerencia realizó un análisis cuantitativo acerca de cuán efectivos son los pasos a producción para cada una de las tecnologías. Como resultado del análisis se detectó que se realiza una gran cantidad intentos de pasos a producción, pero no todos llegan a finalizarse correctamente.

La Tabla 1 ilustra la cantidad de intentos de pasos a producción, por tecnología, durante el período de enero a noviembre de 2017. La Tabla 2 muestra los pasos a producción fallidos, ya sea porque lo implantado no cumple con el requerimiento del cliente (vuelta atrás) o porque falla en el ambiente de producción (incidencias). La Tabla 3 da a conocer los tiempos que se utilizan en las etapas de Test, QA e Implantación de los pasos a producción, lo que evidencia que se gasta demasiado tiempo por recurso en cada fase. Por último, la Tabla 4 muestra la cantidad de artefactos (documentos y/o informes) que se deben crear cada vez que se realiza un paso a producción; independiente del tipo de tecnología que se use.

El análisis detectó que la realización de pruebas y los pasos a producción conllevan un esfuerzo considerable de los equipos de trabajo. Ya sea porque estos procesos son manuales, o porque existe dependencia entre una acción y la siguiente acción. A esto se le suma también los ciclos repetitivos y/o innecesarios que existen entre etapas, y la regeneración de artefactos en distintas etapas.

Tecnologías	Total
Plataforma P.U. (java/.net)	140
Bus de servicio (datapower)	13
Bus de servicio (IIB9)	7
Cobol	233

Tabla 1. Cantidad de intentos de pasos a producción de aplicaciones, agrupados por tecnología.

Tipo de fallas	Total
Vueltas atrás	15
Incidencias	50

Tabla 2 Cantidad de vueltas atrás e incidencias de los pasos a producción.

Tiempos por fase	Total
Tiempo en test	5 a 10 días
Tiempo en QA (preproducción)	5 a 20 días
Tiempo de implantación	15 días

Tabla 3 Tiempos por fase, para cada paso a producción.

Cantidad de artefactos por etapas	Total (artefactos)
Dirección	7
Análisis	10
Diseño	10
Construcción	9
Pruebas (test)	7
Implementación (despliegue)	2
Artefactos complementarios	8
Calidad	3
Ciclo de la demanda	7
Ciclo de compra	5
Validar condiciones de uso	3

Tabla 4. Cantidad de artefactos metodología banco paso a producción.

Este análisis cuantitativo permite dar cuenta a las gerencias TI que la realidad actual sobre la cantidad de pasos a producción es poco eficiente y debe trabajarse en mejorarla. Esta necesidad motiva el mejorar el proceso de desarrollo de la organización, y entregar propuestas de valor mejorando los pasos a producción, lo cual implica

aumentar la cantidad de pasos a producción correctos, mejorar los tiempos de respuesta por incidencia, y así mejorar en la calidad del entregable final.

El haber transparentado lo poco eficiente de los pasos a producción, permitió entender que el proceso de desarrollo y puesta en producción del software asociado a un requerimiento es un problema para la organización, dado que:

- Existe un gran tiempo de demora debido a que son muchos los involucrados en llevar un producto a producción. Esta forma de trabajo genera gran descontento en todas las áreas que intervienen en el flujo de puesta en producción del requerimiento.
- Se producen muchos errores en las etapas de pruebas o etapas posteriores del paso a producción, que provocan altos costos de esfuerzo y tiempo de las personas para dar solución a los mismos, siendo que estos errores se podrían haber identificado en revisiones previas.
- Existen varias tecnologías por aplicativos en las gerencias TI, lo cual provoca que llevar a producción una pieza o producto, tenga estricta dependencia del tipo de tecnología que se utiliza y de las personas (roles) que hacen posible llevar a producción este producto.

1.3. Objetivos de la Tesis

El objetivo general de este trabajo de tesis es mejorar el proceso de desarrollo y puesta en producción de los productos de software desarrollados por las gerencias de TI del Banco Estado, refinándolo mediante la estandarización y automatización. Para lograr este objetivo general se plantean los siguientes objetivos específicos:

1. Estandarizar el proceso y el conjunto de herramientas para el desarrollo y puesta en producción.
2. Reducir en un 40% el tiempo que lleva el proceso general de paso a producción (respecto a la información señalada en las Tablas 1 a 4).
3. Automatizar un 60% de las pruebas realizadas en testing y QA, enfocado en reducir el esfuerzo y errores manuales (respecto a la información señalada en las Tablas 1 a 4).

La meta de mejora en un 40% y 60% fue fijada como resultado de reuniones de trabajo realizadas entre los participantes del proceso de desarrollo y puesta en producción, y también participaron las personas a cargo de las gerencias de TI.

Al lograr estos objetivos, el Banco Estado tendrá a su haber un proceso estandarizado que permitirá dejar disponible frecuentemente nuevas versiones del mismo, cada una de mayor calidad que la anterior para que así ayuden a entregar beneficios a sus clientes. De esta forma, podrá mantener una imagen corporativa potente que demuestre más eficiencia y calidad de cara al país, indicando que es un banco confiable y con altos estándares de calidad en sus procesos.

1.4. Resumen de la Solución Desarrollada

Para enfrentar el problema planteado, en este trabajo de tesis se define un nuevo proceso de desarrollo estandarizado (pipeline), el cual fue abordado a través de varios pasos. Primero, se busca llegar a la estandarización de herramientas utilizadas, las cuales deben estar vinculadas a las fases del ciclo de desarrollo. Para ello, se desechará las herramientas que actualmente no entregan gran valor al flujo, y que aumentan los tiempos y cantidad de errores para llevar un software a producción. A su vez, se incorporarán nuevas herramientas que apoyen el ciclo completo de desarrollo, permitiendo orquestar y automatizar las tareas de pruebas, y el paso a producción de un producto.

Segundo, se complementarán las herramientas que ya posee el Banco Estado, con la Suite UrbanCode recién adquirida por la institución. UrbanCode [5] es un producto de IBM que consiste en un conjunto de tecnologías que apoyan la adopción de las técnicas de DevOps. Este conjunto de herramientas permite enfrentar todas las fases del proceso de desarrollo de nuevos productos y la mantención de productos existentes.

Con el uso de la suite UrbanCode de IBM se busca agilizar y planificar los pasajes entre los distintos ambientes, automatizando los despliegues de los diferentes sistemas informáticos de la Gerencias de TI. Esta suite incluye herramientas para el versionamiento de todo lo construido a través de repositorios, automatización de la compilación para cada tecnología, también sobre procesos de planificación y seguimiento, desarrollo, creación y entrega de piezas terminadas. La suite también apoya la supervisión de las pruebas y el monitoreo en todas las etapas del ciclo de un paso a producción.

La figura 1 muestra el esquema general de las fases del nuevo proceso. Éste tiene un pipeline lo más estandarizado e integrado posible. En la fase de desarrollo se distingue al equipo de desarrollo que construye el requerimiento solicitado por los clientes internos. Luego, en la fase de testing, se trabaja en la instalación de componentes, configuración de ambientes y componentes, se realizan validaciones de normativas propias del Banco Estado y de la tecnología utilizada en el desarrollo; además, se trabaja en las pruebas funcionales.

En la fase de QA se trabaja en la instalación de componentes, configuración de ambientes y componentes, y en pruebas de performance y pruebas de seguridad. Por último, se trabaja en la fase Producción que se preocupa de la instalación del producto construido y validado por las fases anteriores. También se preocupa de la configuración del ambiente y de los componentes instalados, y se valida la implantación del producto construido.

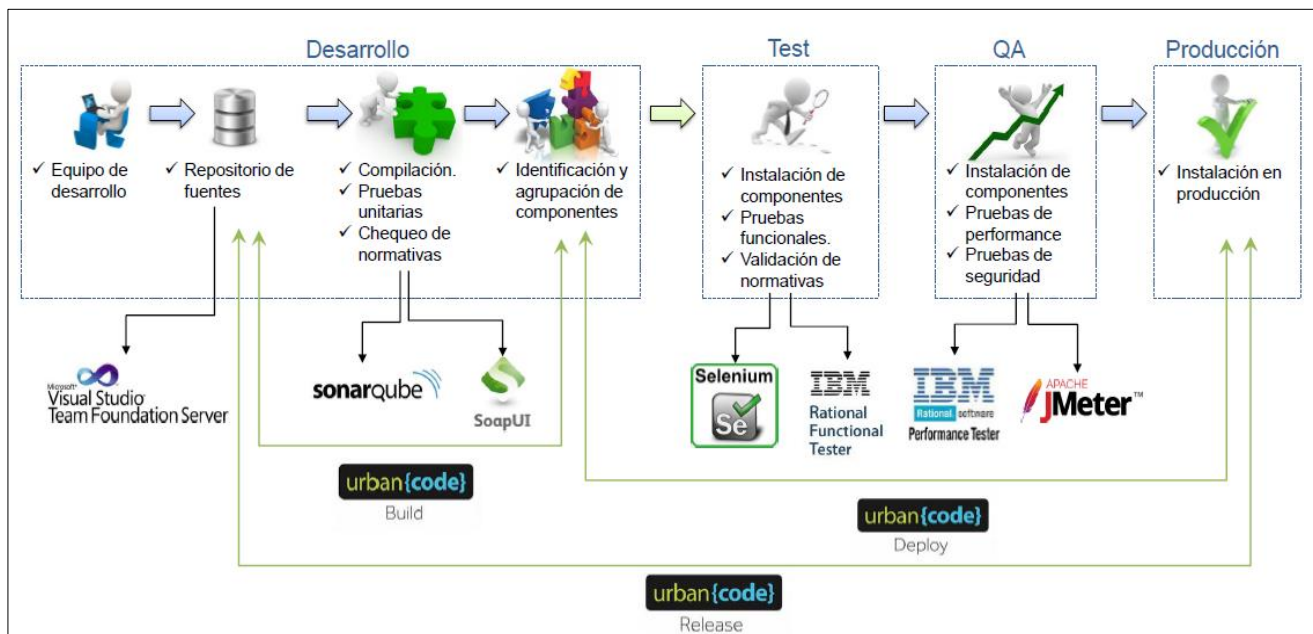


Figura 1. Estrategia de solución en el uso de herramientas de apoyo al proceso de desarrollo.

Los objetivos de este trabajo de tesis son parte de los proyectos estratégicos del Banco Estado para el período 2017-2020. Particularmente, la iniciativa *integrar DevOps al proceso de desarrollo* fue una de las más relevantes. Esta iniciativa fijó como meta la reducción del 40% del tiempo que lleva el paso a producción (Objetivo 2) y la automatización del 60% de las pruebas realizadas por Test y QA (Objetivo 3). Para decidir dicha meta, los equipos de trabajo analizaron varios casos de oportunidad de mejora, detectando así el potencial de mejora efectivo. Se discute a continuación dos de los casos revisados.

El primer caso hace mención a los altos tiempos que se gasta en la revisión de código fuente sobre un desarrollo creado para los sistemas computacionales del Banco Estado. Este tiempo va desde el momento que el desarrollador solicita la revisión de código fuente al área de testing, hasta que la persona designada de testing entrega su resolución por medio de un artefacto (o documento) que evidencie la revisión. Esto puede tomar 1 o 2 ciclos de revisión y entrega. Un ciclo es de aproximadamente 5 días hábiles, y termina con una aprobación a lo revisado; el segundo ciclo se genera sólo si hubo un rechazo a lo revisado en el primer ciclo.

Con la integración de las nuevas herramientas, la revisión de código tomará sólo unos minutos ya que ésta realiza la revisión de código considerando normativas de cada lenguaje de programación, más las normativas del Banco Estado. Por lo tanto, al finalizar su revisión automática le enviará un correo electrónico al desarrollador, entregándole los resultados de la revisión de código fuente, bajando así los tiempos a 1 día sobre los 5 días que se requerían antes. Aun así, siempre existirán casos de borde, donde sólo una persona puede saber cómo manejar esa situación y entregar su criterio.

Esta automatización del flujo de revisión de código fuente permitirá mejorar los tiempos en un 80% (si hablamos de un ciclo antes eran 5 días por persona, y ahora baja a sólo unos minutos en un día hábil).

El segundo caso hace mención a la generación o repetición innecesaria de creación de artefactos (o documentos) en las distintas etapas del ciclo de vida de un desarrollo. Por ejemplo, volver a crear un artefacto que fue creado en la etapa de análisis y diseño para definir la arquitectura tecnológica. Aunque este documento puede ayudar a guiar la iniciativa, es innecesario crearlo nuevamente en etapas futuras ya que la arquitectura debe venir correctamente definida desde un comienzo. Este documento tiene un máximo por persona de 10 días hábiles, pero dependiendo de la tecnología a utilizar puede llegar a 30 días hábiles. Se debe tener presente que la aprobación de un artefacto es necesario para toda iniciativa, por lo tanto, es primordial tener la aprobación al inicio, y es innecesario volver a solicitar una aprobación en etapas futuras.

Con la nueva forma de trabajar y/o metodología a utilizar que trae este proyecto, se busca eliminar tiempos/documentos repetidos o innecesarios, bajando sustancialmente los tiempos y el trabajo que no agrega valor a las iniciativas en curso.

1.5. Metodología

Para la construcción de la solución y el logro de los objetivos planteados, la metodología de trabajo fue organizada en dos ciclos. El primer ciclo buscó proveer y validar en la práctica una solución inicial, de forma de validar tempranamente y aprender de la experiencia para construir una solución refinada. El segundo ciclo buscó generar las validaciones y evidencias que muestren la satisfacción de los interesados a este nuevo modelo de trabajo. A continuación, se explica brevemente cada ciclo.

Ciclo 1

- 1.1. **Fase Preliminar.** *Su objetivo es analizar los procesos actuales de la Gerencia de Sistemas.* En esta actividad se realizará un levantamiento de los procesos y tecnologías actuales que son utilizados en los pasos a producción, definiendo también los requisitos del nuevo proceso y de las nuevas plataformas. De esta forma se busca llegar a contar con un conjunto de herramientas lo más estandarizadas independiente de las tecnologías existentes.
- 1.2. **Fase de Construcción.** *Su objetivo es definir el nuevo proceso a construir.* En esta actividad se definen e implantan todos los pasos y flujo que debe tener el nuevo proceso de desarrollo y paso a producción. Además, se define la arquitectura tecnológica que sostendrá al nuevo proceso de desarrollo y paso a producción.
- 1.3. **Fase de Validación.** *Su objetivo es validar el nuevo proceso.* En esta actividad se realizará un primer piloto con las nuevas tecnologías, de modo de lograr validar el nuevo proceso. Luego, se evaluará la satisfacción de los interesados por medio de una primera encuesta de satisfacción, que entregue información cuantitativa y cualitativa respecto al uso del nuevo modelo. Esta encuesta tiene preguntas cerradas, como por ejemplo, qué nivel de mejora percibe el interesado con el uso del nuevo modelo, y preguntas abiertas que permitan la retroalimentación para futuras mejoras.

Ciclo 2

Fase Preliminar. *Su objetivo es analizar los procesos actuales de la Gerencia de Sistemas.* En esta actividad concretamos y revisamos el levantamiento de los procesos y tecnologías ya generado en el primer ciclo de análisis de los pasos a producción, para lograr una mejora y bajar los tiempos de demora, bajar los errores y armar flujos únicos según tecnología trabajada

2.1. Fase de Construcción. *Su objetivo es definir el nuevo proceso a construir.* En esta actividad se vuelve a revisar lo definido e implantado sobre los pasos y el flujo que tiene este nuevo proceso de desarrollo y paso a producción; volviendo a validar lo definido sobre la arquitectura tecnológica que sostiene al nuevo proceso de desarrollo y paso a producción.

2.2. Fase de Validación. *Su objetivo es validar el nuevo proceso.* En esta actividad se vuelve a realizar un segundo y final piloto por tecnología existente, de modo de lograr validar el nuevo proceso construido. Luego se evalúa nuevamente la satisfacción de los interesados por medio de una primera encuesta de satisfacción y entrevistas.

1.6. Estructura del documento

Este trabajo de tesis está estructurado de la siguiente forma. En el capítulo 2 se da a conocer el marco teórico que explica los conceptos que utilizará este trabajo de tesis. En el capítulo 3 se presenta el proceso de desarrollo actual de desarrollo con sus dolores e insatisfacciones (actividades realizadas, lenguajes utilizados, roles existentes, tecnologías) que dieron pie a este trabajo de tesis. En el capítulo 4 se da a conocer la solución desarrollada que implica el desarrollo del nuevo proceso de desarrollo y todo lo que comprende esta solución. En el capítulo 5 se detalla el trabajo de validación por medio de pilotos con los cuales validar el proceso mejorado. En el capítulo 6 se presenta las conclusiones y trabajo futuro sobre el nuevo proceso mejorado y que da solución a los dolores existentes. Y finalmente el capítulo 7 da a conocer información sobre referencias de donde se obtuvo la información de apoyo a este trabajo de tesis.

2. Marco Teórico

La mirada central de este trabajo de tesis está puesta en la generación de un nuevo modelo del proceso de desarrollo, que abarca todo el ciclo de desarrollo hasta su puesta en producción usando DevOps. A continuación, se da a conocer los principales conceptos y definiciones utilizadas en esta tesis.

2.1. Introducción a DevOps

DevOps es un enfoque que promueve la colaboración entre las áreas de desarrollo y operaciones de TI [1]. Es una estrategia de desarrollo empresarial que habilita la entrega continua, el despliegue continuo y la supervisión continua de aplicaciones (Figura 2). Esta estrategia reduce el tiempo necesario para tratar la retroalimentación y puesta en producción de cara al cliente. Es importante aclarar que en el desarrollo de software tradicional, los proyectos se generaban en silos independientes unos de otros.

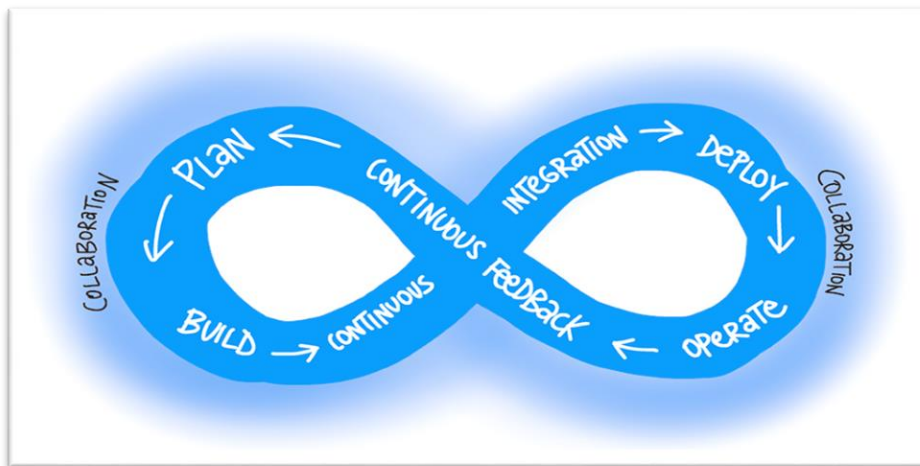


Figura 2. DevOps y su colaboración entre desarrollo y operaciones (obtenido de [1]).

Las metodologías ágiles buscan permitir la entrega rápida de software de alta calidad y un enfoque comercial que alinee el desarrollo, con las necesidades del cliente, y los objetivos de la empresa. En este paradigma, la inspección y adaptación frecuentes son necesarias. Además, el trabajo en equipo, la auto-organización y la responsabilidad son fundamentales para el éxito de todo proyecto. Esto elimina los silos (sistemas aislados), logrando la colaboración entre todos los participantes de un proyecto [4].

La Integración Continua (CI) es el proceso de automatización de la compilación y prueba del código cada vez que un miembro del equipo confirma cambios en el sistema de control de versiones [2]. CI alienta a los desarrolladores a compartir sus pruebas de código y unidad, fusionando sus cambios en un repositorio de control de versiones compartido después de cada tarea pequeña. El código de confirmación activa un sistema de compilación automatizado para tomar el último código del repositorio compartido y para compilar, probar y validar la rama maestra completa.

La integración continua sirve a los desarrolladores, dado que a menudo trabajan de forma aislada, y luego necesitan integrar sus cambios con el resto de la base de código del equipo. Se establece que trabajar apoyándose de metodologías ágiles es una buena forma de ayudar en el ciclo de desarrollo y pruebas de un paso a producción.

La Entrega Continua (Continuous Delivery) es el proceso para compilar, probar, configurar y desplegar desde una compilación a un entorno de producción [2]. La entrega continua no significa que todos los cambios se implanten en el ambiente de producción lo antes posible. Significa que cada cambio ha demostrado ser desplegable en cualquier momento, y que se prioriza que la versión esté en un estado en el que puede ser puesta en producción.

La Implantación Continua (Continuous Deployment) es el siguiente paso de la entrega continua. Aquí cada cambio que pasa las pruebas automatizadas, se implanta automáticamente en el ambiente de producción. La implantación continua debe ser el objetivo de la mayoría de las empresas que no están limitadas por los requisitos reglamentarios u otros; siempre que el responsable dé su aprobación al paso a producción [3].

El Monitoreo Continuo (Continuous Monitoring) proporciona retroalimentación de la producción. La supervisión proporciona información sobre el rendimiento y los patrones de uso de una aplicación. Uno de los objetivos del monitoreo es lograr una alta disponibilidad, minimizando el tiempo de detección y de mitigación de incidencias. En otras palabras, tan pronto como surgen los rendimientos y/o problemas, los datos de diagnóstico enriquecidos sobre estos se envían a los equipos de desarrollo a través de los mecanismos de supervisión automática [2].

2.2. Experiencias similares con DevOps

El Banco República de Uruguay trabajó en un Proyecto de Cambio de Aplicaciones Centrales (CORE Bancario) donde buscaba contar con una herramienta que permita agilizar y planificar los pasajes entre ambientes de testing, automatizando los despliegues de los diferentes sistemas informáticos utilizados en el Banco [6]. Con el uso de la herramienta DevOps llamada IBM UrbanCode lograron cumplir estos objetivos.

Otra experiencia similar se llevó a cabo en ITPAMI [7], que es una organización social estatal que brinda cobertura a jubilados y pensionados en todo el territorio nacional argentino. ITPAMI da a conocer como la institución logró mejoras en su flujo de trabajo incluyendo ambientes de testing, aseguramiento de calidad y preproducción bajo el uso de docker con metodologías y cultura DevOps.

Respecto a otras tesis del Magíster en TI, que realizaron también trabajos con DevOps, me gustaría mencionar a la tesis de Eduardo Díaz [10], que realizó un trabajo de investigación e implantación de DevOps en la empresa Previred S.A., empresa que presta servicios de seguridad social. En ese trabajo se mostró que la organización, previo al proceso de implantar DevOps, tenía una tasa de falla bastante alta sobre procesos críticos. Para lograr mejorarlos, se buscó implantar un proceso de integración y entrega continúa automatizado incorporando procesos y herramientas DevOps dentro de la organización. Particularmente esto se hizo para un servicio

productivo de Previred, con el fin de reducir la tasa de fallos críticos debido al proceso actual (manual o mal realizado). Este trabajo de tesis me permitió evidenciar cuantitativamente y cualitativamente formas de medir mi trabajo de investigación, y a ver otras experiencias reales en Chile en el uso de DevOps en las organizaciones.

3. Proceso de Desarrollo de las Aplicaciones del Banco Estado

El proceso de desarrollo base (o inicial), tanto para nuevos desarrollos como mejoras a piezas ya existentes, hace uso de una planificación y flujo muy rígidos. Estos artefactos casi no permiten mejoras sobre lo “ya desarrollado y/o planificado”, haciendo que varios proyectos no terminen bien de cara al cliente Operaciones. También se debe mencionar que el proceso base al poseer tiempos muy rígidos es casi imposible tener nuevos tiempos para hacer algún cambio o una vuelta atrás en cualquiera de sus fases; aunque existen las personas, las tecnologías y los procesos ya identificados y operando. El llevar un producto hasta la línea productiva no garantiza que haya sido llevado de la mejor forma. O sea, no garantiza tiempos adecuados, flujo correcto, entrega de un buen producto de calidad, etc.

Las gerencias TI poseen 2 formas de llevar a producción una iniciativa y sus desarrollos a producción. Particularmente, existen Iniciativas de corto alcance que implican pasar a producción un desarrollo dentro de un periodo de 30 días hábiles. También existen las iniciativas de largo alcance, que implican poder pasar un producto a producción en periodos más extensos en el tiempo que 30 días.

En este capítulo se describe el proceso de desarrollo base utilizado en el Banco Estado. En la Sección 3.1 se describe el proceso con sus actividades y ambientes utilizados. En la Sección 3.2 se detallan las tecnologías utilizadas para apoyar al proceso. La Sección 3.3 documenta el estudio cualitativo que se realizó en el Banco para identificar los principales problemas que presenta el proceso, y que representan un obstáculo para la organización. La Sección 3.4 concluye con el análisis cuantitativo y cualitativo en la búsqueda de una solución a los dolores antes identificados.

3.1. Proceso de desarrollo base de un software en Banco Estado

El proceso de desarrollo base busca identificar todas las actividades que se realizan en el proceso inicial de un paso a producción y a la vez evidenciar el logro de los objetivos a cumplir que se espera al ir pasando de fase en fase (ver Tabla 5). Dependiendo de la tecnología del aplicativo, se define un pipeline para avanzar y obtener las aprobaciones necesarias para pasar a la siguiente fase, por ejemplo, el pipeline para validar el código construido en lenguaje COBOL es muy distinto a cuando se usa lenguaje .NET o ASP.

El proceso actual posee un conjunto de fases y actividades donde también participan diversos roles que permiten llevar y/o concretar el paso a producción de un producto.

3.1.1. Actividades del proceso desarrollo base

El proceso base (o inicial) de paso a producción, contiene las siguientes actividades independientes del tipo de lenguaje de programación con el cual se trabaje (cobol, .NET, Java, Bus de Servicio).

Desarrollo	Testing	QA/Preproducción	Producción
Desarrollar/construir	Asignar recurso del banco, para atender solicitud	Asignar recurso del banco, para atender solicitud	Asignar recurso del banco, para atender solicitud

Analizar el código estático	Descargar fuentes/instalar	Descargar fuentes e instalar	Habilitar/disponibilizar ambientes productivos
Compilar	Compilar	Ejecutar carga de datos para las pruebas	Instalar en producción
Realizar pruebas unitarias	Chequear cumplimiento de normativas	Implementar las mesas técnicas	Ejecutar la vuelta atrás
Reportar de resultados	Revisar el listado de piezas del desarrollo	Implementar las mesas de administración del cambio	Realizar el piloto o marcha blanca
Generar casos de pruebas y datos para pruebas	Ejecutar la carga de datos para pruebas.	Aprobar/rechazar la solicitud de paso a producción	Generar artefactos
Generar solicitud de paso a producción	Realizar las pruebas integración		
Generar artefactos	Evaluar el rendimiento (performance/carga/stress)		
	Aprobar/rechazar la solicitud de paso a producción		

Tabla 5. Proceso base de un paso a producción.

Como muestra la tabla 5, las fases por las cuales cual deben pasar los actuales desarrollos son las siguientes:

1. **Desarrollo:** Fase de elaboración del software donde el equipo encargado creará y certificará su construcción. Una vez listo el desarrollo y hechas las pruebas del equipo, se entrega a la fase de Testing. En esta fase se entregan los artefactos creados por los desarrolladores. Los *roles* que interactúan dependiendo de la actividad son los siguientes:

- El programador y el jefe proyecto participan en el desarrollo/construcción, análisis de código estático, pruebas unitarias, y reporte de resultados.
- El programador participa en compilar y subir fuentes al repositorio.
- El programador, el jefe proyecto y el usuario certificador participan en generación de casos y datos de prueba.
- El programador y el líder de desarrollo participan en generar solicitudes paso a producción.
- El programador, el jefe proyecto y el líder de desarrollo participan en generar artefactos.

2. **Testing:** Fase de prueba del equipo de testing, donde se certifica que lo entregado por el equipo de desarrollo está correcto, y cumple tanto en las normas de programación, así como con la normativa del banco. En esta fase se entregan los artefactos creados por los recursos de testing, y los roles que participan, dependiendo de la actividad, son los siguientes:

- El especialista de testing realiza la función de descargar y compilar el código fuente en su ambiente, luego chequea que el código fuente este de acuerdo a la normativa banco que exige para cada lenguaje programación, para finalmente aprobar o rechazar esta revisión/certificación

en el documento de solicitud de paso a producción; de este modo pueda continuar avanzando al área de producción.

- El especialista testing, jefe proyecto, programador ejecutan en un laboratorio las pruebas de carga de datos para comprobar que funcione de acuerdo a lo indicado (pruebas unitarias, pruebas integración)
- El especialista testing, jefe proyecto y programador también realizan en un laboratorio las pruebas de rendimiento (performance/carga/stress) para ver que no afecte la arquitectura banco, o servidores; y no impacte a otros sistemas, y de este modo no se vea afectado el funcionar de los sistemas computacionales banco

3. QA (Quality Assurance, aseguramiento de la calidad en español) o Preproducción: Fase en la que se vuelve a certificar el desarrollo entregado por el equipo de desarrollo, pero con la diferencia que QA es un ambiente igual a lo que es el ambiente de producción; con lo cual se espera que estas pruebas sean lo más semejantes al ambiente y realidad lado del cliente y operaciones. En esta fase se entregan los artefactos creados por los recursos de QA. En esta fase, los roles que interactúan dependiendo de la actividad son los siguientes:

- El especialista QA participa en conjunto con el especialista testing, descargando el código fuente entregado por el líder del proyecto, luego lo instala en el laboratorio que se creó, y da a conocer este trabajo creando un artefacto (documento) para que se hagan las pruebas.
- El especialista QA, jefe proyecto, programador ven la evidencia realizada en la etapa de testing y aprueban o rechazan lo realizado antes en las pruebas unitarias testing.
- El especialista de riesgo, especialista seguridad, jefe proyecto van a validar el trabajo realizado en una mesa llamada mesa técnica seguridad; para certificar que no daña la seguridad de la arquitectura sistémica banco.
- El especialista de administración de cambios, jefe proyecto participa en una mesa llamada mesa administración de cambios para ver todos los pasos a realizar en el paso a producción, y tener todo controlado el día del paso a producción.
- El especialista QA, especialista de administración de cambio participan en la aprobación/rechazo de la solicitud de paso a producción; que se fue certificando en los puntos anteriores de QA.

4. Producción: Fase en que el software se pasa a producción y se deja disponible para ser usado por los usuarios finales. En esta fase se entregan los artefactos creados tanto por los usuarios finales como también por los ingenieros de sistemas. En esta fase los roles que interactúan dependiendo de la actividad son los siguientes:

- El especialista en administración del cambio (producción) participan en asignar el recurso banco (la persona idónea de esta área de producción) para la instalación y todos los pasos que se deban contemplar el día del paso a producción.

- El especialista de seguridad, especialista middleware, especialista telecomunicaciones participan en habilitación/disponibilizar ambientes que permitan hacer funcionar el desarrollo o piezas que se están pasando en cada paso a producción.
- El especialista de administración de cambios, y jefe proyecto participan en la mesa administración de cambios, y vuelta atrás. Esto sucede siempre y cuando en el momento del paso a producción existan problemas; se analiza los pasos a realizar; los cuales ya están documentado en el artefacto de solicitud de paso a producción (anexado).
- El especialista de administración de cambio, y usuario certificador participan en llevar a cabo las pruebas que certifican que el paso a producen quedo bien instalado en producción, y se da por aprobado/rechazado la solicitud de paso a producción.
 - Posterior al paso a producción también se lleva una marcha blanca, y el piloteo de lo puesto en producción; realizando una aprobación/rechazo documento de aprobación del piloteo de las piezas/desarrollo que ya existe en producción.

3.1.2. Diferencias de actividades respecto al proceso desarrollo base

Dependiendo del lenguaje de programación que se utilice en el proceso de desarrollo, se deja en evidencia las actividades nuevas, o eliminación de las mismas a través de un código de colores. Este código determina la acción de agregar o eliminar tareas al proceso (Tabla 6). Luego se describe cómo cambia cada uno de los procesos base de la organización.

Proceso de Desarrollo	Actividades Nuevas	Actividades Eliminadas
Proceso COBOL		
Proceso .NET/JAVA		
Proceso bus de servicio		

Tabla 6. Actividades nuevas y eliminadas según el proceso.

Proceso base COBOL

Desarrollo	Testing/QA	Producción
Desarrollar/construir	Asignar recurso del banco, para atender solicitud	Asignar recurso banco, para atender solicitud
Realizar análisis estático de código	Descargar fuentes/instalar	Habilitar/disponibilizar ambientes
Compilar	Instalar parmlib	Instalar parmlib
Realizar pruebas unitarias	Compilar los archivos fuentes	Instalar en producción
Generar reporte de resultados	Chequear normativas	Ejecutar la vuelta atrás

Generar de casos de pruebas y de datos para pruebas	Revisar listada de piezas del desarrollo	Realizar piloto o marcha blanca
Subir fuentes al repositorio (versionamiento)	Ejecutar la carga de datos para pruebas.	Generar artefactos
Crear CCF	Realizar pruebas unitarias	
Generar solicitud de paso a producción	Realizar pruebas integración	
Generar artefactos	Realizar pruebas de rendimiento (performance / carga / stress)	
	Aprobar/rechazar solicitud paso producción	

Tabla 7. Proceso base de un paso a producción cobol.

Proceso base .net/java

Desarrollo	Testing	QA/Preproducción	Producción
Desarrollar/construir	Asignar recurso banco, para atender Solicitud	Asignar recurso banco, para atender solicitud	Asignar recurso banco, para atender solicitud
Analizar el código estático	Descargar las fuentes e instalar	Descargar las fuentes e instalar	Habilitar/disponibilizar ambientes productivos
Compilar los archivos fuentes	Crear el perfilamiento	Ejecutar la carga de datos para pruebas	Instalar en producción
Realizar las pruebas unitarias	Compilar los archivos fuentes	Implementar las mesas técnicas	Ejecutar la vuelta atrás
Revisar el reporte de resultados	Chequear el cumplimiento de normativas	Realizar las pruebas ATS (código estático y dinámico)	Realizar piloto o marcha blanca
Generar los casos de prueba y datos para las pruebas	Revisar el listado de piezas del desarrollo	Habilitación/disponibilizar ambientes (no formal)	Generar artefactos
Subir fuentes al repositorio (versionamiento)	Ejecutar la carga de datos para pruebas.	Realizar las pruebas de rendimiento (performance / carga/stress)	Realizar el monitoreo de usuarios externos
Generar solicitud de paso a producción	Realizar las pruebas unitarias	Implementar las mesas de administración del cambio	
Generar artefactos	Realizar las pruebas integración	Aprobar/rechazo la solicitud de paso a producción	

	Realizar las pruebas de rendimiento (performance/carga/stress)	Generar artefactos	
	Aprobar/rechazar la solicitud de paso a producción		

Tabla 8. Proceso base de un paso a producción .Net/Java.

Proceso base bus de servicio

Desarrollo	Testing	QA/Preproducción	Producción
Desarrollar/construir	Asignar recurso del banco, para atender solicitud	Asignar recurso del banco, para atender solicitud	Asignar recurso del banco, para atender solicitud
Analizar el código estático	Descargar fuentes e instalar	Descargar fuentes e instalar	Habilitar/disponibilizar ambientes productivos
Compilar los archivos fuente	Compilar los fuentes	Ejecutar la carga de datos para pruebas	Instalar el software en ambiente de producción
Realizar las pruebas unitarias	Chequear el cumplimiento de normativas	Implementar mesas técnicas	Realizar la vuelta atrás
Revisar el reporte de resultados	Revisar el listado de piezas del desarrollo	Realizar las pruebas ATS (código estático y dinámico)	Realizar el piloto o marcha blanca
Generar casos de pruebas y datos para las pruebas	Ejecutar la carga de datos para las pruebas.	Habilitar/disponibilizar ambientes (no formal)	Generar artefactos
Subir fuentes al repositorio (vers.)	Realizar las pruebas unitarias	Realizar las pruebas de ethical hacking (a pedido)	
Generar solicitud de paso a producción	Realizar las pruebas integración	Implementar las mesas de administración del cambio	
Generar artefactos	Realizar las pruebas de rendimiento (performance/carga/stress)	Aprobar o rechazo la solicitud de paso a producción	
	Aprobar/rechazar la solicitud de paso a producción	Generar artefactos	

Tabla 9. Proceso base de un paso a producción Bus de Servicio.

Los roles que interactúan en las tareas que se agregan, dependiendo del lenguaje de programación, son los siguientes:

- El programador cobol participa en la creación de los controles de cambio (CCF), e instalación de piezas para dejar ambientado sus desarrollos, piezas creadas, y bibliotecas (instalación parmlib y piezas).
- El especialista testing participa en crear perfiles si lo amerita el desarrollo (en los ambientes de pruebas).
- El arquitecto seguridad, jefe proyecto, programador participan en conjunto para realizar las Pruebas ATS y/o de código estático; de este modo, determinar si deben realizarse mejoras a lo desarrollado.
- El especialista de seguridad, especialista middleware, especialista telecomunicaciones participan en la habilitación/disponibilizar los ambientes de producción para que funcione de acuerdo a lo esperado en producción (generación y pruebas de ambientación).
- El especialista QA, jefe proyecto y programador participan en las pruebas de rendimiento (performance/carga/stress).
- El rol de la empresa externa es participar en pruebas ethical hacking para certificar que los desarrollos cumplan con las normativas existentes.

3.2. Tecnologías de apoyo al proceso de desarrollo base

En el proceso de desarrollo base (o inicial) que se utiliza para llevar a producción un producto banco, intervienen las siguientes herramientas tecnológicas, según el entorno:

- Entornos de gestión (herramientas que el Banco Estado ha creado para gestionar las iniciativas):
 - *FAS (Flujo de Artefactos de Sistemas)*: Sitio web que se utiliza para toda gestión de crear, registrar y promocionar un label llevándolo hasta producción. label es el empaquetado o release que se genera con la herramienta TFS.
 - *GDI (Gestión pasos a producción de iniciativas)*: Sitio web donde cada líder de desarrollo va registrando sus avances en el proceso de paso a producción de una iniciativa.
 - *Página ADC*: Sitio web de la Gerencia de Tecnología, que permite al líder del desarrollo crear las solicitudes de paso a producción de una iniciativa e ir viendo el avance de la misma, ya que en esta solicitud se puede ir viendo el estado de avance y que recurso banco la tiene en su poder.
- Entornos de programación: Visual Studio, Eclipse, Roscoe, Putty, WinSCP.
- Entornos de almacenamiento o repositorios: TFS, Librarian.
- Entornos de base dato y diagramación: Erwin.
- Entornos de pruebas: Rational Funtional Tester, Rational Performance Tester.

Tecnología vs Fases	Herramientas por Fases			
	Desarrollo	Testing	QA/Preproducción	Producción
JAVA/.Net	Visual Studio (.NET)		Visual Studio	
	TFS	TFS	TFS	TFS
	FAS	FAS	FAS	FAS

	GDI / página ADC	GDI / página ADC	GDI / página ADC	GDI / página ADC
	Erwin	Erwin	Erwin	
	Eclipse (Java)		Rational Funtional Tester	
Bus de Servicios	TFS	TFS	TFS	TFS
	FAS	FAS	FAS	FAS
	GDI / página ADC	GDI / Página ADC	GDI / página ADC	GDI / página ADC
	Putty	Putty	Putty	Putty
	WinSCP	WinSCP	WinSCP	WinSCP
	Eclipse		Rational Perfom. Tester	
			Pruebas de seguridad ext.	
COBOL	Roscoe	Roscoe	Roscoe	Roscoe
	Librarian	Librarian	Librarian	Librarian
	GDI / Página ADC	GDI / Página ADC	GDI / Página ADC	GDI / Página ADC
	Erwin	Erwin	Erwin	

Tabla 10. Herramientas por tecnología.

3.3. Metodología de apoyo para identificar los problemas del proceso desarrollo base

Las gerencias TI, al alinearse con el Programa de Transformación 2020 que posee el Banco Estado, han establecido que se debe mejorar el proceso de desarrollo actual, particularmente la eficiencia de los pasos a producción de las 120 aplicaciones departamentales que son administradas por estas gerencias TI. Con el fin de alcanzar la eficiencia esperada, se trabajó en un diagnóstico cualitativo donde se pidió apoyo a los principales interesados dentro del banco. Este diagnóstico consiste en obtener resultados a través del uso de 2 métodos.

- i. Una **entrevista** a diferentes actores de la organización, cubriendo todos los roles principales de los procesos descritos en la Sección 3.1. Esto permite transparentar y caracterizar los problemas del proceso desarrollo actual.
- ii. El uso del **método KPT** (del acrónimo en inglés: Keep, Problem, Try), proveniente de la práctica ágil llamada retrospectiva (Figura 3).

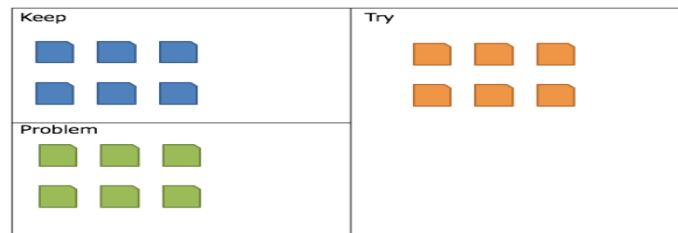


Figura 3. Método retrospectivo KPT

La utilización de este método implica una gestión visual creando la revisión y generación de un listado con ideas y acciones que deben ser mejoradas, y otras que deben mantenerse. Es decir, se usará *Keep* para anotar las cosas buenas que se deben mantener como parte del proceso o cultura del equipo. Se usará *Problem* para identificar los desafíos que se presentaron en el proceso y las mejoras que se deben realizar. Finalmente, se utilizará *Try* para determinar las acciones correctivas para resolver los problemas o mejoras, y así obtener mejores resultados en la siguiente iteración del proceso.

3.3.1. Protocolo de Identificación de problemas e insatisfacciones

El registro formal de problemas (protocolo) tiene por objetivo identificar las insatisfacciones o dolores que tiene cada funcionario en su rol dentro del proceso desarrollo actual. Esto permite distinguir el área de trabajo y origen de la insatisfacción o dolor.

- Para determinar la insatisfacción, se entrevista a un conjunto de actores que son parte del proceso, y se registra y levanta sus principales dolores. El poder entender las acciones que no cumplen con una correcta forma de trabajar, permite tomar acción para lograr mayor precisión y fluidez en el proceso. Se usará una escala de 1 a 7 para calificar cada dolor, donde 1 será la actividad que considera con menos dolor; y 7 será para indicar que esa actividad es la que le causa mayor dolor.
- Al tener identificados los dolores, también se materializó y se pudo identificar de dónde vienen estos. Así se puede tomar acciones para mejorar el trabajo en esas áreas. Para poder saber de dónde provienen esos dolores se usó el método KPT, que nos permitió segmentar por gerencia estas insatisfacciones.

3.3.2. Identificación de problemas e insatisfacciones

Durante un periodo de aproximadamente un mes se realizaron *entrevistas* a un grupo de personas que representan los siguientes *roles* dentro de la organización:

1. Gerentes y Subgerentes que son los principales interesados en el proceso.
2. Jefaturas asociadas al proceso.
3. Ejecutores del proceso (responsables de testing, ingenieros de QA, operaciones, arquitectos, e ingenieros de sistemas que apoyan y mantienen cada sistema computacional existente en banco).
4. Otros roles: Usuarios o Clientes

Por cada rol se entrevistaron a los siguientes *funcionarios y/o actores*:

- i. Gerente de Sistemas, Gerente Tecnología, Gerente de Operaciones, Subgerente de Sistemas y productos de apoyo.
- ii. Jefa del área de QA, jefe de metodología, jefe área de administración de cambio, y jefe del área operaciones.
- iii. Un arquitecto de integración, un tester, un QA, dos ingenieros de sistemas, y dos desarrolladores.
- iv. Otros: dos usuarios de operaciones.

Para llevar a cabo las entrevistas se tomaron las siguientes acciones:

- Se presenta a los entrevistados el actual proceso de desarrollo, dándoles a conocer las actuales fases y actividades que posee el proceso (tabla 5).
- Como acción siguiente se les pide que den su diagnóstico a la situación actual, y que identifiquen sus principales preocupaciones, molestias o dolores.
- Posteriormente se les pide que prioricen de 1 a 7 como se indica en la sección 3.3.1.

A partir de la información recolectada en las entrevistas, se identificaron un total de 30 dolores o insatisfacciones que logran evidenciar lo que creen y piensan los actores del proceso de desarrollo, y las causas de éstas. En el Anexo C se detallan los ítems identificados.

Una vez identificados los 30 dolores del proceso de desarrollo actual, se ordenaron los dolores por peso, según lo establecido por los participantes. La siguiente tabla muestra estos ítems ordenados por peso.

N°	Dolores/insatisfacciones detectadas	Gerente Sistema	Gerente Tecnología	Gerente Operaciones	Subgerente Sistemas Productos	Subgerente Tecnología Arquitecturas e Integración	Jefe de área QA	Jefe de área Metodología	Jefe de área Administración de Cambios	Arquitecto Integración	Tester	QA	Ingeniero Sistemas (GSIS)	Programadores	Operaciones	Sumatoria Votos
6	Proceso de compra muy extenso en tiempos y acciones a seguir	7	7	7	7	1	1	1	1	1	1	1	7	1	7	50
24	Abastecimiento de los sistemas core, desde varias fuentes de datos	7	7	7	7	7	1	1	1	1	1	1	7	1	1	50
5	Errores colaterales post-paso a producción a todo ámbito (desarrollos y/ infraestructuras)	5	5	5	5	5	1	1	1	1	1	1	7	1	7	46
1	Falta clarificar el alcance de los cambios	6	1	6	6	1	1	6	1	1	1	1	7	7	1	46
28	Se asumen nuevos estándares de calidad en seguridad informática, sin saber mucho lo que esto implica.	1	7	1	1	7	7	1	1	5	5	1	7	1	1	46
25	Repositorio de información de los aplicativos poco eficiente, o casi inexistentes.	5	5	5	5	5	1	5	1	1	1	1	7	1	1	44
27	planificaciones muy rígidas.	5	1	5	5	1	1	6	1	1	1	1	7	1	7	43

30	La actual metodología del banco es un cuello de botella para establecer fechas de entrega de cara al cliente.	6	1	1	6	1	1	7	1	1	1	1	7	1	7	42
22	Muchas mesas de trabajo para un paso a producción.	5	5	5	5	1	1	1	5	1	1	1	7	1	1	40
8	Ambientes no homologados.	6	6		6	1	7	1	1	1	5	1	1	1	1	38
2	Artefactos (documentos) de poco valor para certificar y evidenciar pruebas entre ambientes.	1	1	1	1	5	6	6	1	5	1	1	7	1	1	38
23	Muchos silos de trabajo entre ambientes	7	7	7	1	1	1	1	1	1	1	1	7	1	1	38
29	Creación de nuevas áreas que provoca nuevos flujos al ciclo actual	5	5	1	5	5	1	1	1	1	1	1	7	1	1	36
14	Tiempos excesivos para certificar por ambiente	1	1	1	1	1	4	1	1	1	7	1	7	7	1	35
4	Entregas poco prolijas por parte del equipo de desarrolladores	1	1	1	1	1	7	1	1	1	7	1	4	7	1	35
12	Data center lentos en aplicar los cambios	4	5	1	4	1	1	1	5	1	1	1	7	1	1	34
13	Diagnósticos pobres ante fallas de infraestructura / arquitectura banco	5	5	1	5	1	5	1	1	5	1	1	1	1	1	34
18	Mesa de ayuda informática poco eficiente	4	4	6	4	1	1	1	1	1	1	1	7	1	1	34
19	Poco dominio de la arquitectura banco que soporta los aplicativos	4	4	1	1	6	1	1	1	5	1	1	5	1	1	33
15	Infraestructura obsoleta para los tiempos actuales	4	5	1	4	1	4	1	1	6	1	1	1	1	1	32
26	Sobre dependencia de plataformas departamentales, estilo frameware	5	1	1	5	5	1	1	1	1	1	1	7	1	1	32
16	No contar con herramientas de control y calidad para el testeo y monitoreo	1	1	1	1	1	6	1	1	1	7	7	1	1	1	31
20	Tiempos excesivos al aplicar nuevos cambios de infraestructura	4	4	1	4	6	1	1	1	1	1	1	1	1	1	28
21	eficiencia de los desarrollos entregados	1	1	1	1	1	5	1	1	1	5	1	1	7	1	28
12	Sites de contingencia inestables	5	5	1	5	1	1	1	1	1	1	1	1	1	1	26
7	Facilidad para cambiar prioridades de entregables y/o proyectos	1	1	1	1	1	1	1	1	1	1	1	7	1	7	26
3	Excesivo papeleo a registrar	1	1	1	1	1	1	6	1	1	1	1	7	1	1	25
10	Competencias de los equipos de bancos	1	1	1	1	1	4	1	1	1	1	1	5	1	1	21
9	Cambios en los ambientes sin aviso	1	1	1	1	1	1	1	1	1	1	1	1	1	1	14
17	Monitoreo / marcha blanca poco eficientes en ambientes productivos	1	1	1	1	1	1	1	1	1	1	1	1	1	1	14

Tabla 12. Dolores e insatisfacciones detectadas del proceso desarrollo actual, post entrevista.

3.3.3. Ejecución del método KPT

Una vez que se generó y entrevistó a los actores seleccionados, se ejecuta el método KPT para entender ahora; *de dónde provienen los dolores de este proceso de desarrollo inicial*, identificando y segmentando las áreas con mayores insatisfacciones; así saber dónde atacar más fuerza las insatisfacciones.

Gerencia de Sistemas (GSIS)

Las personas que conforman la GSIS evidenciaron en la tabla 13 sus dolores y como ellos creen que es posible dar solución a los problemas.

<p>Keep</p> <ul style="list-style-type: none"> - Se mantiene la programación en lenguajes COBOL CIC, .NET, JAVA y Uso de bus de servicio y componentes (datapower, broker e IIB9). - Se mantiene la necesidad de la aprobación Inicial del líder del Aplicativo para comenzar el ciclo entre ambientes para llegar a producción. 	<p>Try</p> <ul style="list-style-type: none"> - Capacitar y/o generar equipos idóneos para tener el alcance de cada iniciativa (integración funcional). - Capacitar y/o generar equipo de trabajo para identificar artefactos con valor, y eliminar el resto. - Mejorar la revisión por parte de los Líderes y a la vez Banco debe entregar mejores herramientas para revisar. - Mejorar el flujo proceso de compra (desde gestión de compra, incluyendo cliente y líder aplicativo). - Entregar de mejores herramientas de tester y monitoreo; en conjunto con estandarizar y/o actualizar normativas banco. - Capacitar para entregar mejores respuestas asociadas a ámbitos sistémicos. - Capacitar para entregar mejores respuestas asociadas a ámbitos de infraestructura. - Generar equipos de trabajo que identifiquen la real necesidad de las mesas de trabajo, y ver cual entrega real valor a un paso a producción. - Se está comenzando a trabajar en el CORE Banco. - Generar y posee herramientas de repositorio unificada (tener herramientas y establecer dónde se guarda toda información necesaria por aplicativo). - Para planificaciones rígidas, abrirse a uso de metodologías ágiles y DevOps según corresponda. - Capacitar respecto a nuevas normas y estandarizar a la brevedad las normativas de seguridad en Banco. - Mejorar metodología banco actual.
<p>Problem</p> <ul style="list-style-type: none"> - Falta clarificar alcance de los cambios. - Artefactos (documentos) de poco valor para certificar y evidenciar pruebas entre ambientes - Excesivo papeleo a registrar. - Entregas poco prolijas por el equipo de desarrolladores informáticos. - Errores colaterales post paso producción a todo ámbito (desarrollos y/ infraestructuras). - Proceso compra muy extenso en tiempos y acciones a seguir. - Facilidad para cambiar prioridades de entregables y/o proyectos. - Competencias de los equipos de bancos. - Tiempos excesivos para certificar por ambiente - Mesa de ayuda informática poco eficiente - Poco dominio de la arquitectura banco que soporta los aplicativos. - Muchas mesas de trabajo para un paso a producción. - Abastecimiento de data proveniente de varias fuentes en los sistemas CORE del Banco - Repositorio de información de los aplicativos poco eficiente, o casi inexistentes. - Planificaciones muy rígidas. - Nuevos estándares de calidad en seguridad informática, sin saber mucho lo que implica. - Creación de nuevas áreas o subgerencias lo que provoca nuevos flujos al ciclo actual. - La actual metodología es cuello de botella para establecer fechas de entrega de cara al cliente. 	

Tabla 13. Método KPT a la gerencia sistema

Gerencia de Tecnología (GTEC)

Las personas que conforman la GTEC evidenciaron en la tabla 14 sus dolores y como ellos creen que es posible dar solución a los problemas.

<p>Keep</p> <ul style="list-style-type: none"> - Mantener contrato con UrbanCode para seguir utilizándolo - Mantener y Actualizar Contrato con Herramientas TFS y SonarQube 	<p>Try</p> <ul style="list-style-type: none"> - Mejorar revisión por parte de los líderes, y a la vez el Banco debe entregar mejores herramientas para revisar. - Entregar de mejores herramientas de testing y monitoreo; en conjunto con estandarizar y/o actualizar normativas banco. - Se está comenzando a trabajar en el CORE Banco. <p>El Banco Estado está actualmente especializándose y capacitando a sus áreas de seguridad para entregar mejores respuestas asociadas a infraestructura de seguridad, ataques informáticos, etc.</p>
<p>Problem</p> <ul style="list-style-type: none"> - Ambientes no homologados. - Cambios en los ambientes sin aviso. - Competencias de los equipos de Bancos. - Data Center lentos en aplicar los cambios. - Diagnósticos poco eficientes ante fallas de infraestructura/arquitectura Banco. - Infraestructura obsoleta para los tiempos actuales. - No contar con Herramientas de control y calidad para el testeo y monitoreo. - Poco dominio de la arquitectura Banco que soporta los aplicativos. - Tiempos excesivos cuando se debe aplicar nuevos cambios de infraestructura. - muchos silos de trabajo entre ambientes - Nuevos estándares de calidad en Seguridad informática, sin saber mucho lo que implica. - Creación de nuevas áreas o subgerencias lo que provoca nuevos flujos al ciclo actual 	

Tabla 14. Método KPT a la Gerencia Tecnología

Gerencia de Operaciones (GOP)

Las personas que conforman la GOP evidenciaron en la tabla 15 sus dolores y como ellos creen que es posible dar solución a los problemas.

<p>Keep</p> <ul style="list-style-type: none"> - Seguirán entregando los requerimientos o necesidades a cumplir por las Gerencias de TI 	<p>Try</p> <ul style="list-style-type: none"> - Entregas más prolijas a poner en producción
<p>Problem</p> <ul style="list-style-type: none"> - Monitoreo inadecuado o marcha blanca poco eficiente en los ambientes productivos - Eficiencia de los desarrollos entregados 	

Tabla 15 Método KPT a la gerencia operaciones

3.4. Análisis situación actual

En base a los resultados obtenidos tras la generación del protocolo de identificación de problemas, y el determinar de dónde son estos dolores, se realizó un análisis cuantitativo que permite reflejar la priorización de los dolores más votados, en conjunto con poder determinar la existencia de una acción real de solución a estos dolores a través de este trabajo de tesis (UrbanCode).

3.4.1. Identificar dolores con máxima prioridad

Para determinar la búsqueda de una solución a los dolores antes mencionados, se adoptaron los siguientes criterios:

Criterio I

Del universo total de los 30 dolores reflejados y seleccionados tras la entrevista realizada en conjunto con la utilización del método KPT por gerencia que se ve influenciada por este trabajo de tesis, se trabajó en determinar los dolores más votados V/S ver si estos dolores pueden ser atendidos según las opiniones de los mismo encuestados. Esto se puede entender revisando la sección 3.3 de este trabajo de tesis.

- a) **Priorizar por sumatoria de votos por dolor (tabla 12).**
 - i. *Alta prioridad (A):* El 50% de los dolores con mayor suma en su votación corresponden a los de alta prioridad, es decir, los 15 dolores más votados. Esto se traduce en los dolores que obtuvieron en sus sumatorias de votos entre 50 y 35 votos.
 - ii. *Media prioridad (M):* Del 50% restante que queda al sacar los de alta prioridad, se vuelve a usar el mismo criterio de utilizar el 50% de los más votados, es decir, 7.5 pero lo dejo en 7 dolores. Esto se traduce en los dolores que obtuvieron en sus sumatorias de votos entre 34 y 31 votos.
 - iii. *Baja prioridad (B):* Los dolores restantes, es decir, los 7 dolores menos votados equivalen a los de baja prioridad. Esto se traduce en los dolores que obtuvieron en sus sumatorias de votos entre 28 y 14 votos.

b) Priorizar por dolores, determinando si existe una acción para dar solución a estos.

Se puede observar que al aplicar el método KPT por gerencia esta herramienta ágil nos permitió desglosar los dolores por gerencia y a la vez poder determinar las que sí se consideran que realmente pueden ser atacadas y buscar una solución a ella. Esto se puede apreciar viendo la sección 3.3.2.b que muestra el desglose del método KPT donde podemos ver la generación de un listado con ideas y acciones que se pueden mejorar en las gerencias TI del Banco Estado, focalizándonos en la división TRY.

Conclusión: Tomando los hitos **a) y b)** de este **criterio I** que se está trabajando se obtuvo un cuadro que permite visualizar.

Prioridad del dolor	Alta	12	3
	Media	5	2
	Baja	4	4
		Con Acción	Sin Acción
		Acción al dolor	

Tabla 16. Priorizar por dolores.

Podemos concluir que, de los 30 dolores observados, hay una gran cantidad de dolores (21) sobre los que sí se puede realizar acciones que puedan entregar una solución al(los) dolores; visto desde los entrevistados. Este primer criterio permite ahora preguntar y así confeccionar un segundo criterio y final para ver la utilidad de DevOps en esta tesis.

Criterio II

Este criterio busca determinar cuáles de estos dolores podrían ser mitigados con técnicas DevOps. Para identificar los dolores que son más susceptibles a ser apoyados por técnicas DevOps, se tomó el universo generado en el criterio I, asociado a los dolores de alta prioridad. Ahí se ve que existen acciones que pueden dar solución a estos dolores.

Prioridad del dolor	Alta	12
	Media	5
	Baja	4
		Con Acción
		Acción al dolor

Tabla 17. Identificar dolores a trabajar con DevOps.

Conclusión: Tomando en cuenta el cuadro recién visualizado, vemos que existen 12 dolores con alta prioridad, 5 con mediana prioridad y 4 dolores con baja prioridad. Esto permite enfocar las energías de este trabajo de tesis a los dolores que tienen mayor prioridad para los interesados; y a la vez introducir en determinar cuáles serán los que pueden ser trabajados con técnicas DevOps.

A continuación, se muestra el cuadro que permite explicar los dolores que DevOps puede trabajar (independiente de la prioridad con la cual serán trabajados):

Identificar técnicas DevOps que ayudan a solucionar los dolores detectados		Entrevista	Método KPT (Con/Sin Acción)	Agilidad en el proceso	Repositorio Fuentes (Código fuente)	Repositorio Artefactos	Gestor de versiones	Test Unitario	Test Funcional	Test de Performance	Test de Aceptación Usuario	Monitoreo en Fase Preproducción	Virtualización de Servicios	Administración de la Nube Híbrida	Administración de la Infraestructura	Administración de Laboratorio virtual	Instalación Automatizada	Orquestar Automatizado	Monitoreo Automatizado en Producción	Trasmitir conocimientos	
#	Dolores/Insatisfacciones detectadas	votos	TRY	Técnicas y/o Principios DevOps																	
6	Proceso compra muy extenso en tiempos y acciones a seguir	50	SIN																		
24	Abastecimiento de data proveniente de varias fuentes en los sistemas core banco	50	SIN																		
5	Errores colaterales post Paso producción a todo ámbito (desarrollos y/ infraestructuras)	46	CON																	X	
1	Falta clarificar alcance de los cambios	46	CON																		
28	Nuevos estándares de calidad en seguridad informática, sin saber mucho lo que implica	46	CON																		
25	Repositorio de información de los aplicativos poco eficiente, o casi inexistente	44	CON		X	X												X			X
27	planificaciones muy rígidas	43	CON	X														X			
30	La actual metodología banco es cuello de botella para establecer fechas de entrega de cara al cliente	42	CON	X									X			X	X	X			
22	Muchas mesas de trabajo para un paso a producción	40	CON	X			X											X			

8	Ambientes no homologados	38	SIN																	
2	Artefactos (documentos) de poco valor para certificar y evidenciar pruebas entre ambientes	38	CON		X	X														X
23	Muchos silos de trabajo entre ambientes	38	CON														X			
29	Creación de nuevas áreas o subgerencias lo que provoca nuevos flujos al ciclo actual	36	SIN																	
14	Tiempos excesivos para certificar por ambiente	35	CON	X				X	X	X	X	X		X		X				
4	Entregas poco prolijas por el equipo de desarrolladores informáticos	35	CON	X			X	X	X		X									
12	Data center lentos en aplicar los cambios	34	SIN																	
13	Diagnósticos poco eficientes ante fallas de infraestructura/arquitectura banco	34	CON											X						
18	Mesa de ayuda informática poco eficiente	34	CON																	
19	Poco dominio de la arquitectura Banco que soporta los aplicativos	33	CON																	
15	Infraestructura obsoleta para los tiempos actuales	32	SIN																	
26	Sobre dependencia de plataformas departamentales banco estilo frameware	32	CON																	
16	No contar con Herramientas de control y calidad para el testeo y monitoreo	31	CON									X								X
20	Tiempos excesivos cuando se debe aplicar nuevos cambios de infraestructura	28	SIN																	
21	Eficiencia de los desarrollos entregados	28	CON	X														X		
12	Site de contingencia Inestables	26	SIN																	
7	Facilidad para cambiar prioridades de entregables y/o proyectos	26	SIN																	

3	Excesivo papeleo a registrar	25	CON			X													
10	Competencias de los equipos de bancos	21	CON	X															
9	Cambios en los ambientes sin aviso	14	SIN																
17	Monitoreo inadecuado o marcha blanca poco eficiente en los ambientes productivos	14	CON								X							X	

Tabla 18. Dolores y técnicas DevOps para apoyar.

4. Solución desarrollada

Ya que se conocen los dolores e insatisfacciones recopilados y reconocidos en el capítulo 3, es necesario ahora dar solución a estos problemas. El capítulo 4 da a conocer el nuevo proceso de desarrollo y todo lo que comprende esta solución.

En la sección 4.1 se puede ver el contexto, objetivos y principios que transparentan el origen de donde nace esta iniciativa de búsquedas de mejoras formas de refinar el proceso de desarrollo actual. La sección 4.2 da a conocer la forma como se enfrentó y se creó la solución refinada. La sección 4.3 muestra la plataforma tecnológica que soporta a todo el proceso refinado de desarrollo en conjunto con las herramientas que acompañan en la automatización y mejoras al nuevo proceso. En la sección 4.4 cierra el capítulo con un análisis de la cobertura del proceso y la plataforma, con respecto a los objetivos y motivadores que se definen en la próxima Sección.

4.1. Contexto, Objetivos y Principios

4.1.1. Contexto

Las gerencias de TI del Banco Estado en su búsqueda de alinearse al plan estratégico del Banco Estado, han creado nuevas iniciativas para apoyar y causar un impacto favorable sobre el negocio que se desenvuelve y pertenece al Banco Estado. Entre estas iniciativas se encuentra una que busca mejorar al proceso de desarrollo actual a través de la estandarización y automatización del proceso con apoyo de herramientas que posee actualmente el mercado.

4.1.2. Objetivos

Cuando las gerencias de TI crearon la iniciativa *“integrar DevOps al proceso de desarrollo”*, se identificaron un conjunto de problemas en el flujo de puesta en producción de los sistemas. Estos problemas son independientes de la tecnología que se utilice para implementar los sistemas. Entonces, la gerencia de sistemas y la gerencia de tecnología detectaron que pueden apoyar y dar solución a un conjunto de objetivos de negocios. Sobre estos objetivos de negocio se espera generar el impacto cuantitativo y cualitativo que Banco Estado y el proceso de desarrollo actual esperan de estas gerencias

Este análisis de detección de problemas, detectó un esfuerzo considerable de los equipos de trabajo, ya sea por ser manuales o porque existe dependencia entre una acción y la siguiente acción, sumando también los ciclos repetitivos y/o innecesarios que existen entre etapas; como así también la regeneración de artefactos en distintas etapas. Ante lo ya mencionado, los objetivos de negocio que se establecieron fueron los siguientes:

- G1: Aumentar la estabilidad de los sistemas. Esto implica una continua optimización y renovación de los sistemas y de esta forma trabajar en asegurar su funcionamiento 7x24.
- G2: Asegurar la integración de los sistemas actuales con las nuevas capacidades digitales que trae el nuevo proceso de desarrollo.

- G3: Reducir el tiempo que lleva el proceso general de paso a producción.
- G4: Automatizar las pruebas y monitoreos realizados en Testing y QA, enfocado en reducir el esfuerzo y errores manuales
- G5: Liberar tiempo de las personas que integran el área de tecnología
- que integre metodologías y prácticas de desarrollo y operación agilizando el proceso actual de desarrollo y su calidad
- G7: Reducir brechas y aumentar entrenamientos y capacidades para asumir nuevos roles y funciones.
- G8: Escalamiento Capacidades de Desarrollo. Esto implica agilizar el proceso de desarrollo y su calidad, mejorar el proceso de gestión de la demanda, e incorporarse en la gestión preproductiva.
- G9: Aumentar el uso del ecosistema de terceros. Esto implica potenciar la adopción de soluciones de mercado y los partners tecnológicos que nos acompañan en este nuevo proceso.

4.1.3. Principios técnicos y de negocio

Con el fin de cumplir con los objetivos de negocio planteados, se crean los siguientes principios de negocio:

- B1: La solución parte con mejoras sobre los procesos asociados a sus tecnologías de desarrollo de aplicaciones existentes (.NET, COBOL CICS, Java, Bus de Servicios). Una vez terminadas estas se verá el impacto para crear nuevos pipelines de otras tecnologías utilizadas en las Gerencias TI.
- B2: El proceso debe ser transparente para todos aquellos interesados que son parte del proceso de desarrollo y que se ven involucrados en un paso a producción.
- B3: La solución debe permitir que todo evento o tarea ya sea humana o automática, debe entregar un aviso o alerta inmediata ante cualquier falla en el flujo del proceso; de este modo tomar las medidas que sean necesarias por parte de los interesados.
- B4: La solución entregada debe poder automatizar toda acción humana y actividad repetitiva que fue definida en el análisis de este proyecto.
- B5: No debe haber interrupción apreciable de los servicios que están operando ante la aplicación de cambios en los mismos.

Estos principios implican garantizar principios tecnológicos tales como:

- T1: Todo desarrollo o piezas deben ser generadas con las tecnologías y/o herramientas que posee o se trabaja en el Banco Estado y sus gerencias de TI.
- T2: Esta solución no puede cambiar lo existente respecto a la infraestructura básica que sostiene a las gerencias de operación del Banco Estado; aun cuando si se puede realizar cambios respecto al equipamiento que utiliza cada funcionario banco de la gerencia de sistemas dependiendo del trabajo que realice; de esta forma poder crear desarrollos acordes a las necesidades de banco y sus gerencias
- T3: Banco Estado actualmente está trabajando en mejorar sus arquitecturas, plataformas y aplicaciones con mejoras normas y herramientas de seguridad informática; estas no pueden verse afectadas con esta nueva solución.
- T4: En cualquier etapa del nuevo proceso de desarrollo; debe existir la posibilidad de vuelta atrás, ante

el NO logro de lo comprometido o desarrollo mal realizado. De este modo garantizar la fluidez de la operación del Banco Estado.

4.2. Proceso de Desarrollo Refinado

En la búsqueda de lograr los objetivos antes mencionados y de resolver los dolores detectados por las gerencias TI, es necesario en primera instancia refinar el proceso de desarrollo base que utilizan actualmente las gerencias de TI. En la sección 3.1, y específicamente en la tabla 5 de esta sección, se puede ver detallado el proceso de desarrollo base y las actividades que lo componen, y que corresponde al proceso previo a la adopción de las nuevas técnicas aplicadas en el Banco Estado y este trabajo de memoria. En la figura 4 se puede ver el proceso de desarrollo refinado trabajado en esta tesis, donde se puede visualizar las técnicas DevOps y las que el Banco Estado espera utilizar para lograr un proceso automatizado y estandarizado.

4.2.1. Actividades y técnicas del proceso desarrollo refinado

Con el fin de explicar las nuevas tareas y técnicas DevOps utilizadas en el proceso de desarrollo trabajado y refinado, se puede ver en la figura 4 las nuevas actividades y técnicas DevOps utilizadas.

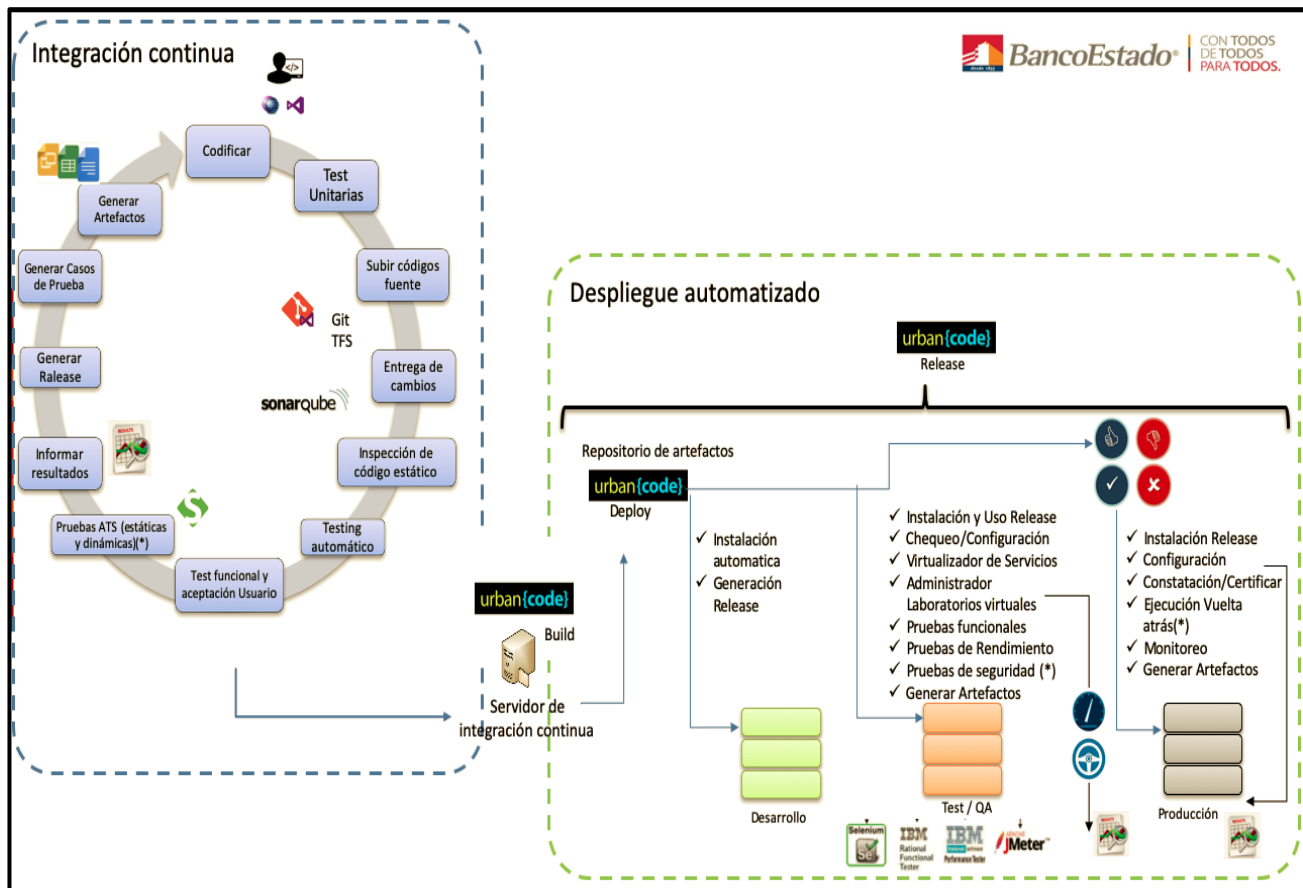


Figura 4. Proceso de desarrollo refinado.

Actividades y técnicas DevOps utilizadas en este proceso refinado.

El proceso refinado que plantea este trabajo de tesis, posee un conjunto de técnicas DevOps más técnicas utilizadas normalmente por el Banco Estado, que a su vez pasan por distintos niveles de madurez que permitan lograr el paso a producción deseado.

i. El primer nivel de madurez que utiliza este proceso refinado del Banco Estado es el llamado **integración continua**, y nos presenta un conjunto de actividades tanto manuales y automáticas; tales como:

• Actividades manuales:

- *Codificar*: Desarrollar, construir los códigos fuentes suficientes para crear la solución pedida por el cliente o necesidad a cumplir.
- *Subir códigos fuentes*: Almacenar y organizar todas las creaciones/mejoras de código que son creadas por el(los) programador(es) a diario. Se hace mediante la herramienta de Visual Studio Team Foundation Server (TFS).
- *Tests unitarios*: Pruebas sobre el desarrollo, así comprobar si lo realizado está bien construido
- *Tests de aceptación/tests funcionales de usuario*: Es una etapa previa a la liberación de versiones (nuevas o mejoradas) a fin de determinar si cumplen con las necesidades y/o requerimientos del cliente banco que solicitó este desarrollo. Las pruebas funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones que especificó el cliente.
- *Generación de casos de pruebas y datos*: Creación de datos y escenarios de pruebas, tanto a través de un script para carga de datos, como de un documento (artefacto) que deben generar para la siguiente etapa o rol.
- *Generar artefactos*: Es la creación de documentos que posee la organización, y que permiten dar a conocer tanto acciones a seguir como también las aprobaciones que hacen los distintos roles para poder seguir avanzando en el paso a producción.
- *Generar Release*: no es un release totalmente manual, pero el programador también puede planificar dónde y cuándo desea él ejecutar y generar el release. La idea de un release es generar un paquete con los desarrollos o mejoras que el programador construyó. En este release va una lista de fases a realizar y por ejemplo también plan de despliegue a ejecutar para instalar lo desarrollado.

• Actividades automatizadas:

- *Entrega de cambios*: Llevar y generar un registro de cambios sobre los desarrollos creados, de esta forma crear un control de versiones de los desarrollos. Esto se hace con la herramienta Git.
- *Inspección/análisis del código estático*: Es la actividad de evaluar el código fuente construido por el programador, que por medio de la herramienta de análisis estático de código fuente llamado SonarQube se inspecciona el código fuente desarrollado y generar métricas y avisos que pueden ayudar a mejorar la calidad del código de un programa.

- *Testing automático*: Método usado en banco como medio de ejecución de pruebas sin el protagonismo de la intervención humana, creados con lenguajes scripting para pruebas. Estos casos de prueba son previamente programados, como por ejemplo:
 - Probar una parte del software específica en la que se está trabajando, en distintas fases del mismo.
 - Evitar errores manuales de las pruebas humanas.
 - Una prueba automática que falla nos da más información que una prueba manual que detecta un fallo.
 - *Test ATS (análisis estático y dinámico de código)*: Son pruebas de seguridad que se realizan para saber si en temas de seguridad sobre el software desarrollado, cumple con las normativas de seguridad internacional que se debe cumplir. En banco se utiliza la herramienta IBM Rational AppScan.
 - *Generar reportes de resultados*: Una vez ejecutados las pruebas y revisiones sobre lo construido, se generan como paso siguiente en forma automática reportes e informes que dan aviso de la aprobación o rechazo; o de información que ayudará a las gestiones posteriores del desarrollo realizado.
 - *Generar release*: Generar un paquete con los desarrollos o mejoras que el programador construyó en este release va una lista de fases a realizar y por ejemplo también plan de despliegue a ejecutar para instalar lo desarrollado.
- ii. A continuación, se identifica el nivel de madurez llamado **despliegue automatizado** donde se identifican un conjunto de actividades, tales como:
- **Actividades manuales:**
 - *Instalación*: instalación manual del paquete generado en el release del nivel anterior. Esto lo hace el especialista y lo puede instalar en su PC.
 - *Chequeo de normativas*: Se revisan las normativas propias del Banco Estado que se exigen y se deben plasmar en todo tipo de desarrollo de las aplicaciones del Banco Estado.
 - *Revisión de listado de piezas*: En ciertos casos específicos las piezas deben ser revisadas por el especialista. Esto fue declarado en la especificación funcional de esta iniciativa.
 - *Pruebas funcionales*: Las pruebas funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones que especificó el cliente (son pruebas específicas, concretas y exhaustivas para probar y validar que el software hace lo que debe y lo especificado).
 - *Generar artefactos*: Es la creación de documentos que posee el Banco Estado y que permiten dar a conocer tanto acciones a seguir como también las aprobaciones que hacen los distintos roles para poder seguir avanzando en el paso a producción.
 - **Actividades automatizadas:**

- *Ejecución del release*: Utilización del release instalado en el servidor de integración continua; de este modo usar el paquete instalado y ya chequeado, pero que no le es necesario ni la revisión de chequeo de normativas del Banco Estado, ni de piezas específicas de lo desarrollado.
- *Virtualizador de servicios*: Tecnología existente en banco y que se mantiene bajo este nuevo modelo; y que está basada en un software que posibilita la ejecución de varios sistemas operativos diferentes entre sí, como invitados dentro de un único host del servidor físico. Son las llamadas *máquinas virtuales* (VMs) que ejecutan en una imitación virtual del hardware del servidor.
- *Administrador laboratorios virtuales*: Esta actividad no existía en Banco, y permite armar laboratorios virtuales que permitan al especialista de testing certifique pruebas muy específicas; que requieren de un ambiente específico bajo configuraciones específicas.
- *Pruebas funcionales*: Las pruebas funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones que especificó el cliente (son pruebas específicas, concretas y exhaustivas para probar y validar que el software hace lo que debe y lo especificado).
- *Pruebas de rendimiento*: Test para diagnosticar pueden medir que partes del sistema o de carga de trabajo provocan que el conjunto rinda mal. Se hacen pruebas de carga, de estrés, performance.
- *Pruebas de seguridad (*)*: Es el uso de pruebas de vulnerabilidad y ethical hacking tanto con empresas externas y con el nuevo equipo de seguridad para estos temas.

Finalmente, para llevar el desarrollo a producción, las actividades son las siguientes:

- *Instalar release*: Es el mismo release aprobado en la fase anterior, y que está instalado en los servidores de producción a espera de ser aprobado para ser finalmente instalado y usado en producción.
- *Configuración/ambientar*: Es toda configuración requerida en los ambientes productivos para un correcto funcionamiento del nuevo desarrollo puesto estos ambientes.
- *Constatación/Certificar*: asociada al cliente, si todo está ok se firma; en caso contrario vuelta atrás.
- *Ejecución vuelta atrás (*)*: si algo falla o no le gusta al cliente se hace una vuelta atrás de lo instalado en producción, ya sea porque al cliente que solicitó este desarrollo, considera que eso es lo que se necesitaba o llegó muy tarde; o porque técnicamente no cumple con los estándares que exige banco. Tener presente que si al cliente no le satisfago este desarrollo y por tanto el desarrollo debe ser mejorado; estos temas se alejan a los límites de este trabajo de tesis.
- *Monitoreo*: Estando ya en producción, se monitorea lo construido. Este tipo de casos se realiza en escenarios de prestar apoyos a aplicativos que se comunican con aplicativos de otras instituciones de fuera del Banco Estado.
- *Generar artefactos*: Es la creación de documentos que posee el Banco Estado y que permiten dar a conocer tanto acciones a seguir como también las aprobaciones que hacen los distintos roles para poder seguir avanzando en el paso a producción.

4.2.2. Roles, equipos que interactúan en el proceso desarrollo

Con el fin de explicar todas las partes del nuevo proceso es necesario también se debe explicar los distintos actores y las responsabilidades al participar en este proceso. Es importante mencionar cada rol ya fue definido en las secciones 3.1.1 y 3.1.2; ahora sólo basta dar a conocer los nuevos roles, estos son los siguientes:

- *Líder de Integración continua*: Rol encargado de dar la aprobación del release inicial generado por el equipo de desarrollo; y de esta forma aprobando el desarrollo recién podrá avanzar a la fase posterior.
- *Administrador de release (release managers)*: se encargan de la gestión y coordinación del producto desde el desarrollo hasta la producción con el apoyo del líder de integración continua, Los release managers supervisan la coordinación, la integración y el flujo de desarrollo, testing e implementación para apoyar la entrega continua del servicio. Están enfocados no solo en crear, sino también en mantener la cadena de aplicaciones de principio a fin,
- *Ingeniero DevOps*: Rol encargado de realizar funciones como la de codificación o scripting, como también la reingeniería de procesos y la comunicación con el resto de las áreas de la empresa.
- *Arquitecto de seguridad DevOps*: que acompaña en las actividades de asegurar las actividades y pruebas de seguridad del Banco Estado.

4.3. Plataforma Tecnológica

En la búsqueda de generar el nuevo proceso de desarrollo, también fue necesario definir la plataforma tecnológica que de soporte y entregue un ecosistema que permita crear la estandarización y automatización esperada por el proyecto. Dicho lo anterior, el nuevo proceso de desarrollo y las herramientas que lo apoyan tendrán la misión de orquestar y lograr la automatización y estandarización del proceso.

En esta sección se documenta la plataforma tecnológica definitiva para este trabajo en el Banco Estado y que en la sección 4.3.1. define los tipos de herramientas que apoyan el nuevo proceso, y que luego en la sección 4.3.2. da a conocer los diferentes ambientes de ejecución utilizados en el nuevo proceso.

4.3.1. Herramientas tecnológicas de apoyo

En esta sección se dan a conocer todas las herramientas que poseerá el nuevo proceso, vistos por fases y se describe el objetivo o acción que cumplen en el proceso.

- *UrbanCode* [5]. Este es un producto de IBM que consiste en un conjunto de tecnologías que apoyan la adopción de las técnicas de DevOps. Este conjunto de herramientas permite enfrentar todas las fases del proceso de desarrollo de nuevos productos y la mantención de productos existentes. Con el uso de la suite UrbanCode de IBM se busca agilizar y planificar los pasajes entre los distintos ambientes, automatizando los despliegues de los diferentes sistemas informáticos de la Gerencias de TI, donde se incluye herramientas para el versionamiento de todo lo construido a través de repositorios, automatizar la compilación para cada tecnología, también sobre procesos de planificación y seguimiento, desarrollo,

creación y entrega de piezas terminadas, como también sobre la supervisión de las pruebas y el monitoreo en todas las etapas del ciclo de un paso a producción de un desarrollo.

- *Build*: Es una herramienta de gestión de compilación distribuida, multiplataforma y banco la utiliza para sistema basado en plantillas para configurar y ejecutar compilaciones. Usada ahora en el nivel de integración continua que estableció banco.
- *Deploy*[9]: Es la solución automatizada que permite el despliegue automatizado de lo desarrollado.
 - Release: es el que orquesta todo, el que gobierna todas las acciones que realizan para llegar a producción.
- *Eclipse*: Entornos de desarrollo Integrado que permite a los programadores tener una interfaz para construir sus códigos fuentes para cumplir las necesidades requeridas por los clientes internos de banco.
- *Visual Studio*: permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros.
- *Team Foundation Server (comúnmente abreviado como TFS)*: es un producto de Microsoft que proporciona administración de código fuente (ya sea a través de Team Foundation Versión Control o Git), informes, administración de requisitos, gestión de proyectos (para equipos de desarrollo de software ágil y en cascada), compilaciones automatizadas Capacidades de gestión de laboratorio, pruebas y gestión de lanzamientos. Cubre todo el ciclo de vida de la aplicación. TFS se puede usar como un back-end para numerosos entornos de desarrollo integrado, pero está diseñado para Microsoft Visual Studio y Eclipse.
- *Git*: Es un software de control de versiones pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos
- *SonarQube*: Es una plataforma para evaluar código fuente. Es software libre y usa diversas herramientas de análisis estático de código y permite obtener métricas que pueden ayudar a mejorar la calidad del código de un programa.
- *Selenium*: Es un entorno de pruebas de software para aplicaciones basadas en la web. Selenium provee una herramienta de grabar/reproducir para crear pruebas sin usar un lenguaje de scripting para pruebas (Selenium IDE). Incluye también un lenguaje específico de dominio para pruebas (Selenese) para escribir pruebas con los lenguajes de programación que usa banco, tales como Java, y próximamente Python. Las pruebas pueden ejecutarse entonces usando la mayoría de los navegadores web modernos en diferentes sistemas operativos como Windows, Linux y OSX.
- *SoapUI*: SoapUI es una herramienta desarrollada en java, para la realización de pruebas a aplicaciones con arquitectura orientada a servicio (SOA) y transferencia de estado representacional (REST). Soporta múltiples protocolos como SOAP, REST, HTTP, JMS, AMF y JDBC. Posee una versión de código abierto.

- *Jmeter*: Utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web. Es usado en Banco como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas. JMeter puede también ser configurado como un monitor, aunque es comúnmente considerado una solución ad-hoc respecto de soluciones avanzadas de monitoreo.
- *PuTTY*: Es un cliente SSH y Telnet con el que podemos conectarnos a servidores remotos iniciando una sesión en ellos que nos permite ejecutar comandos. El ejemplo más claro es cuando empleamos PuTTY para ejecutar comandos en un servidor VPS y así poder instalar algún programa o configurar alguna parte del servidor. La parte de cliente Telnet es más desconocida para mí, pero el ejemplo más claro de uso es conectarse a nuestro router doméstico para configurarlo a través de Telnet y abrir puertos, etc. Resumiendo, con PuTTY conseguimos abrir una sesión de línea de comandos en el servidor remoto para administrarlo.
- *Rational Function Tester*: Es una herramienta que el banco la utiliza para realizar pruebas automáticas, pruebas de regresión, pruebas de interfaces de usuario (si se requieren) y pruebas controladas por datos (si se requieren). Proporciona soporte de integración para Rational Functional Tester. Con este soporte puede realizar las tareas siguientes: Grabar y exportar scripts de prueba de Rational Functional Tester a ITCAM for Transactions, reproducir scripts de prueba de Rational Functional Tester en el agente Tiempo de respuesta de robótica y ver medidas de tiempo de respuesta, sucesos y mensajes generados durante la reproducción de scripts de Rational Functional Tester, en los espacios de trabajo y las vistas de Tivoli Enterprise Portal. Es una herramienta para pruebas automatizadas de aplicaciones que permite a los usuarios crear pruebas que imitan las acciones y evaluaciones de un probador humano
- *Rational Performance Tester*: Es un entorno libre de creación de scripts para la carga de automatización y las pruebas de escalabilidad de web y aplicaciones de software basadas en servidor. En banco se usa para la generación de informes que ayudan a identificar la presencia y causa de cuellos de botella del sistema. Captura el tráfico de red que se puede emular en varios usuarios virtuales para reproducir la prueba. Con este entorno en banco también se puede hacer pruebas a aplicaciones que utilicen los protocolos que usa banco (como por ejemplo http(s) y servicios (SOA)), y puede registrar ciertas partes de la aplicación que desee probar. Rational Performance Tester genera una prueba de la grabación, y puede editar las pruebas añadiendo datos dinámicos y puntos de verificación. Además, ejecuta la prueba individualmente e inspecciona los resultados para asegurarse de que la prueba esté llevando a cabo lo esperado, y puede emular una carga de trabajo añadiendo grupos de usuarios al editor de planificación.

4.3.2. Ambientes de Trabajo

Los ambientes utilizados en este proyecto son aquellos en los cuales se instalan las aplicaciones desarrolladas. La separación de los ambientes constituye una de las prácticas fundamentales de la disciplina de entrega continua de DevOps, y sirven como parte del apoyo al proceso de desarrollo refinado, que busca el Banco Estado. Estos ambientes son los siguientes:

1. *Estaciones de trabajo de los desarrolladores*: son las máquinas locales propias de cada desarrollador. En ellas se creará el código fuente de las aplicaciones. Los desarrolladores en este ambiente también trabajan en la compilación del código fuente, sus pruebas internas y pruebas de empaquetado de los componentes.
2. *Ambiente de desarrollo*: corresponde a un ambiente de integración para cada proyecto. En este ambiente los desarrolladores podrán hacer pruebas funcionales menores. La instalación en este ambiente es realizada y gestionada por los desarrolladores.
3. *Ambiente de test/QA*: corresponde a un ambiente con visibilidad hacia los usuarios (laboratorios virtuales). En estos laboratorios se instalarán los desarrollos generados por el equipo del desarrollo de los aplicativos para que sean probadas por los usuarios. La instalación en este ambiente es realizada y gestionada por los especialistas de testing e ingenieros DevOps y Administrador Release. Este ambiente funciona como:
 - *Ambiente de integración de aplicaciones*: una vez finalizado el desarrollo, la aplicación pasa a un ambiente de integración en la que ejecutará junto a las demás aplicaciones. De esta forma se podrán detectar errores técnicos resultantes de la interoperabilidad entre las herramientas.
 - *Ambiente de aceptación*: es de similares características que el ambiente de integración, es decir, la aplicación desarrollada está funcionando junto a las demás aplicaciones. La diferencia es que en este ambiente se realizan pruebas de aceptación de usuario.
4. *Ambiente de producción*: Es el ambiente donde se dejan los desarrollos que ya están aprobados y listos para ser usados por el cliente interno banco; o en algunos casos para llegar hasta el cliente final banco.

4.4. Análisis Cobertura de Objetivos

Las mejoras sobre el proceso de desarrollo generaron un mejorado proceso de desarrollo. Esto implicó que los desarrolladores debieron adoptar herramientas que permiten el build automatizado de las componentes de software, además de elaborar scripts que luego son ejecutados en la actividad de integración continua. Estas decisiones están alineadas con los objetivos G8: Escalamiento Capacidades de Desarrollo, G5: Liberar tiempo de las personas que integran el área de tecnología, y G3: Reducir el tiempo que lleva el proceso general de paso a producción. También esto nos permite garantizar la adherencia al principio B3, que nos exige detectar manejo de falla en forma oportuna.

Además, el nuevo proceso agrega formalmente la actividad de integración continua, para reflejar el hecho de que, ante cada integración del código fuente que queda disponible en el repositorio, se realiza la actividad de compilar (build), validar la adhesión a estándares de codificación y buenas prácticas (análisis de código) y verificar la correctitud de los sistemas (ejecución de pruebas). Esta actividad se realiza en forma automatizada, ya que es una tarea repetitiva, cumpliendo con los principios G4 (Automatizar las pruebas y monitoreos realizados en Testing y QA, enfocado en reducir el esfuerzo y errores manuales) y B4 (La solución entregada debe poder automatizar toda acción humana y actividad repetitiva que fue definida en el análisis de este proyecto).

En el proceso de desarrollo, la actividad de puesta en producción era una actividad manual llamada Paso a producción, donde esta actividad requería especificar una Solicitud de Cambio, consistente en un instructivo para los operadores de las áreas de producción; que indicaba todas las modificaciones que debían realizarse en la plataforma productiva y la vuelta atrás es solo manual. En el nuevo proceso se mejoró y automatizó la actividad de puesta en producción de los sistemas, sin necesidad de intervención manual y se logra el principio a la vez el B2 (el proceso debe ser transparente para todos aquellos interesados que son parte del proceso de desarrollo y que se ven involucrados en un paso a producción) y B5 (No debe haber disrupción apreciable de los servicios que están operando ante la aplicación de cambios en los mismos). Además, se alinea con los objetivos G1 (aumentar la estabilidad de los sistemas; esto implica una continua optimización y renovación de los sistemas y de esta forma trabajar en asegurar su funcionamiento 7x24), y G3 (reducir el tiempo que lleva el proceso general de paso a producción).

El uso de esta nueva forma de llevar a producción los desarrollos, generando nuevas plataformas de trabajo y/o ambientes y arquitecturas en desarrollo; permiten cumplir el principio tecnológico T3 (Banco Estado actualmente está trabajando en mejoras sus arquitecturas, plataformas y aplicaciones), y en particular el principio propuesto. En particular, el principio T2 y T4. T2 indica que la solución no puede cambiar lo existente respecto a la infraestructura básica que sostiene a las gerencias de operación del Banco Estado, pero si pudiese cambiar piezas o PCs que usan los creadores de nuevos desarrollos o sistemas ya existentes. T4 establece que en cualquier etapa del nuevo proceso de desarrollo debe existir la posibilidad de vuelta atrás, ante el “no logro” de lo comprometido o desarrollo mal realizado. De este modo, se puede garantizar la fluidez de la operación del Banco Estado.

Por último, con estos cambios nos hacemos cargo de gran parte de los dolores de la sección 3.3.2:

- Ambientes poco prolijos al aplicar cambios.
- Errores colaterales, al incorporar pruebas automatizadas a lo largo del proceso de construcción.
- Tiempo empleado en corregir una falla, al automatizar las tareas de building, testing e implantación de las soluciones.
- Excesivo papeleo, al generar documentos de avisos automáticos.
- Mejoras en los tiempos de duración de las etapas del paso a producción, pues al automatizar y al aplicar técnicas DevOps de despliegue y entrega continua, se pueden hacer entregas más frecuentes y de menor tamaño.
- Esfuerzo necesario para validar cambios, gracias a la automatización y técnicas y herramientas DevOps.

5. Validación

La validación de la solución ha sido trabajada utilizando 2 formas de evaluación, de este forma entender si los dolores del proceso base fueron realmente resueltos.

En la sección 5.1 se detalla el trabajo de validación por medio de pilotos, y de este modo validar el proceso mejorado. En la sección 5.2 se busca identificar el grado de cumplimiento con el cual evaluó cada actor del proceso refinado de desarrollo (validar la solución planteada).

5.1. Validación Pilotos

La idea de pilotear este trabajo de tesis busca poder ejecutar el nuevo proceso asociado a cada una de las tecnologías utilizadas (.net, java, cobol, Bus de servicio) y buscar el aprendizaje en ellos, de esta forma no volver a cometer errores en ciclos o situaciones futuras. Se estableció trabajar con 2 pilotos para visualizar las validaciones y evaluar la factibilidad del uso de este nuevo proceso de desarrollo.

Piloto 1:

Se realizó un primer piloto ejecutando el nuevo proceso bajo una aplicación construida en el lenguaje de programación cobol. En este primer piloto se detectaron tanto aciertos como fallas del nuevo proceso, los cuales permitieron generar una primera evaluación del nuevo proceso.

Este piloto se realizó con los actores que nos apoyaron en las entrevistas de la sección 3.3.2. y se les solicitó su apoyo en esta etapa de validación

Para este piloteo se solicitó el apoyo al equipo del aplicativo CRM desarrollado en lenguaje de programación cobol, y teniendo presente que ellos también fueron parte del equipo que fue encuestado como parte de la sección 3.3.2. Con ellos se realizó una sesión de retrospectiva utilizando la técnica Keep Problem Try (KPT) de desarrollo ágil para poder validar el nuevo proceso; y de esta forma ellos nos puedan indicar qué les gustaría preservar (keep), qué cosas nuevas les gustaría incluir (try), y qué les gustaría modificar (Problem), deben indicar también cuáles son los riesgos que perciben del proceso y la plataforma.

Keep	Try
K1- Uso de creación de registros de Cambios con Git, un mejor versionador que hacía falta.	T1- Se debe eliminar pruebas repetitivas que existen en la fase de Preproducción asociada a pruebas ATS estática, y dejar sola en Fase Desarrollo.

<p>K2- Uso de las nuevas herramientas tecnológicas incluidas en este nuevo proceso</p> <p>K3- Generación de Ambiente de prueba Usuario.</p> <p>K4- Se quitó la sobre dependencia de clientes en la aprobación final, ya que ahora el cliente da su OK cuando se define el alcance (trabajo de células).</p> <p>K5- Nuevos roles.</p>	<p>T2- Se debe crear nuevas pruebas de rendimiento sobre desarrollos de tecnologías web y bus de servicio; de este modo mejorar los estándares y performance sobre los desarrollos.</p> <p>T3- Dependiendo del tipo de desarrollo, se debe mejorar criterios de instalación de herramientas, ya que se debe evaluar si instalar en PC o Máquinas virtuales</p> <p>T4- Creación de Gestión de conocimiento --> extraer métricas y estadísticas, así como un componente visual, tipología de errores del estado del servicio en cada momento</p> <p>T5- incluir/tipificar nuevas normativas de excepción del Banco Estado para Tecnología COBOL no consideradas en esta 1er piloto (creaciones antiguas q pasen por excepción)</p> <p>T6- incorporar nuevas buenas prácticas de la Pruebas de regresión, asociadas que cuando se localiza y corrige un bug, se grave una prueba que exponga el bug y se vuelvan a probar regularmente después de los cambios subsiguientes que experimente el programa</p>
<p>Problem</p> <p>P1- tratar de eliminar los silos burocráticos que quedan respecto a la forma de trabajar.</p> <p>P2- Seguir mejorando estándares de calidad en Seguridad informática, sin saber mucho lo que implica</p>	

Tabla 19. Método KPT Piloto 1.

Análisis de los resultados Piloto 1

Los resultados obtenidos post sesión de retrospectiva apoyan la decisión tomada para incluir técnicas DevOps como parte del nuevo proceso de desarrollo en estas aplicaciones desarrolladas en lenguaje COBOL.

- K1 permite tener un control de registros de versión que antes no existía para tecnologías como COBOL.
- K2 resuelven un conjunto de problemas que presentaba el antiguo proceso.
- K3 permite generar ambientes idóneos para pruebas con usuarios, que simplemente antes no existían y se debían excepcionar o certificar con pruebas mitad manual y mitad automatizada.
- K4 existía una sobre dependencia del cliente para que entregase su OK al proyecto, cosa que ahora ya no existe porque el cliente es parte interactiva del proyecto.
- K5 nacen nuevos roles que permiten que este nuevo proceso fluya y se logre las mejoras esperadas de tiempos, y nuevas pruebas automatizadas.
- P1 siguen los silos burocráticos, pero se adoptarán nuevas acciones para mejorar este proceso y se logre el tan ansiado proceso mejorado.
- P2 Tenemos seguridad muy mala la cual debe mejorar, pero con este proceso se busca lograr esta mejora
- T1 con el nuevo proceso, se eliminan muchas pruebas que son repetitivas y que dependen de personas; ahora serán automáticas.
- T2 nuevas pruebas automatizadas que mejoran la forma de certificar los desarrollos.
- T3 se trabaja en la búsqueda de nuevos ambientes de pruebas y laboratorios de ser necesarios, según las pruebas de usuario que se requieran.
- T5 se está evaluando la incorporación de nuevas normas a UrbanCode, de este modo se pueda automatizar otras revisiones y sus validaciones.

Piloto 2:

Se realizó un segundo piloto ejecutando todo el nuevo proceso sobre los desarrollos que son construidos con lenguajes de programación .NET y JAVA COBOL y tecnología Bus de Servicio, ya que del primer piloteo se identificó la falta de poseer una arquitectura idónea en las fases de Preproducción; y que ahora sí están correctamente identificadas y mejoradas gracias a este segundo piloto.

En este segundo piloto se logró ver muchas mejoras respecto al anterior, permitiendo terminar correctamente el proceso para las tecnologías COBOL, .NET y JAVA. En este segundo piloto, utilizando el método KPT mencionado en la sección 3.3, se puede identificar que los dolores a trabajar son:

Keep	Try
K1- Pruebas ATS estáticas sólo en Fase Desarrollo. K2- Nueva habilitación de arquitectura en Fase Preproducción; lo que permite generar pruebas de consumo de piezas vía internet por usuarios externos a 'Banco Estado	T1- creación de nuevo rol de apoyo para monitoreo continuo y gestión del conocimiento; para generar una especie de paso a producción continuo, aun cuando existe freezing. T2- es necesario generar un nuevo pipeline SQL que acompañe a las tecnologías actuales; de este modo llegar ambos de buena forma y en fecha al paso a producción T3-incorporar nuevas buenas prácticas de la Pruebas de regresión, asociadas que cuando se localiza y corrige un bug, se grabe una prueba que exponga el bug y se vuelvan a probar regularmente después de los cambios subsiguientes que experimente el programa T4- Mejorar Pruebas de Vulnerabilidad, de este modo generar un único Criterio sobre el desarrollo construido V/S lo que indica el Ethical Hacking o prueba vulnerabilidad
Problem	
P1- Seguir mejorando estándares de calidad en Seguridad informática, sin saber mucho lo que implica	

Tabla 20. Método KPT Piloto 2.

Análisis de los resultados Piloto 2

Los resultados obtenidos post sesión de retrospectiva apoyan la decisión tomada para incluir técnicas DevOps como parte del nuevo proceso de desarrollo en estas aplicaciones desarrolladas en lenguaje COBOL, .NET, Java y Bus de Servicio.

- K1 Sacamos del flujo pruebas ATS innecesarias que se hacían en etapas posteriores.
- K2 mantener los nuevos ambientes de pruebas necesarios y que antes no existían.
- P1 no tenemos ambientes idóneos para temas de seguridad, pero se sigue mejorando en estos temas de seguridad.
- T1 La necesidad de nuevos roles potenciarán el proceso de desarrollo; mejorando tiempos y generando actividades no repetitivas y que implicaban nuevas tareas no necesarias.
- T3 y T4 Se está trabajando en establecer únicos criterios de seguridad, para las pruebas que si se están haciendo; de esta forma Banco potencie su forma de crear seguridad.
- T2 como actividades futuras se debe generar este trabajo de SQL.

Con este segundo piloteo ya podemos dar a conocer si el nuevo proceso resuelve o no los dolores presentados del proceso actual.

5.2. Validación de la solución

Para validar la solución propuesta, se utilizó como medio de evaluación la misma encuesta que se dio a conocer en la sección 3.3.1 que permite identificar los dolores de aquellos funcionarios encuestados, pero la forma de evaluación sobre las preguntas cambia. La encuesta busca poder responder si se logró satisfacer los dolores antes mencionados, y a su vez ver resueltos los problemas/dolores encontrados post piloteo de la sección 5.1.

5.2.1. Registro de Validación de la solución

Objetivo: Identificar el grado de cumplimiento que evaluó cada actor del nuevo proceso de desarrollo.

Medición: La forma de medir la satisfacción y dolor de los actores, fue solicitarles que volviesen a responder una encuesta y nos dieran a conocer su percepción de acuerdo al uso de esta nueva forma de trabajar, es decir, evidenciar si con este nuevo proceso se está mejor o peor que antes; en relación a los dolores que se presentaron en la encuesta inicial.

Formato de la encuesta

La encuesta realizada da a conocer:

Nota: De acuerdo a su percepción, evalúe la situación de cada dolor bajo el nuevo proceso desarrollo.					
Cargo:		Gerencia:		Fecha:	

Evaluación del impacto del nuevo proceso desarrollo con DevOps							
N°	Dolores/insatisfacciones detectadas	Mucho peor	Peor	Igual	Mejor	Mucho mejor	No sé
6	Proceso compra muy extenso en tiempos y acciones a seguir						
24	Abastecimiento de data proveniente de varias fuentes en los sistemas core banco						
5	Errores colaterales post paso a producción a todo ámbito (desarrollos y/ infraestructuras)						
1	Falta clarificar alcance de los cambios						
28	Nuevos estándares de calidad en seguridad informática, sin saber mucho lo que implica						
25	Repositorio de información de los aplicativos poco eficiente, o casi inexistentes						
27	Planificaciones muy rígidas						

30	La actual metodología banco es cuello de botella para establecer fechas de entrega de cara al cliente						
22	Muchas mesas de trabajo para un paso a producción						
8	Ambientes no homologados						
2	Artefactos (documentos) de poco valor para certificar y evidenciar pruebas entre ambientes						
23	Muchos silos de trabajo entre ambientes						
29	Creación de nuevas áreas o subgerencias lo que provoca nuevos flujos al ciclo actual						
14	Tiempos excesivos para certificar por ambiente						
4	Entregas poco prolijas por el equipo de desarrolladores informáticos						
12	Data center lentos en aplicar los cambios						
13	Diagnósticos poco eficientes ante fallas de infraestructura/arquitectura Banco						
18	Mesa de ayuda informática poco eficiente						
19	Poco dominio de la arquitectura banco que soporta los aplicativos						
15	Infraestructura obsoleta para los tiempos actuales						
26	Sobre dependencia de plataformas departamentales banco estilo						
16	No contar con Herramientas de control y calidad para el testeado y monitoreo						
20	Tiempos excesivos cuando se debe aplicar nuevos cambios de infraestructura						
21	Poca eficiencia de los desarrollos entregados						
12	Site de contingencia inestables						
7	Facilidad para cambiar prioridades de entregables y/o proyectos						
3	Excesivo papeleo a registrar						
10	Competencias de los equipos de Bancos						
9	Cambios en los ambientes sin aviso						
17	Monitoreo inadecuado o marcha blanca poco eficiente en los ambientes productivos						

Tabla 21. Encuesta de evaluación cualitativa de la solución.

5.2.2. Ejecución de la entrevista

Recordar que los entrevistados fueron los siguientes funcionarios y/o actores:

- Gerente de Sistemas, Gerente Tecnología, Gerente de Operaciones, y el Subgerente de Sistemas y productos de apoyo.

- Jefe del área de QA, jefe de metodología, jefe área de Administración de Cambio, jefe Área Operaciones.
- 1 arquitecto de Integración, 1 tester, 1 QA, y 2 Ingeniero de Sistemas, 2 desarrolladores.
- A diferencia de la primera encuesta; en esta ocasión no se utilizó los 2 Usuarios de operaciones; ya que el Gerente Operaciones recibió el feedback de sus empleados de operaciones.

5.2.3. Resultados de la encuesta

Se obtuvieron 450 respuestas a la encuesta expuesta en la Tabla 21, y llevada a la Tabla 22 que muestra la evaluación cualitativa de la solución, y se muestra el detalle de las respuestas obtenidas.

Evaluación del impacto del nuevo proceso desarrollo con DevOps						
N°	Dolores/insatisfacciones detectadas	No sé	Mucho peor	Igual	Mejor	Mucho mejor
6	Proceso compra muy extenso en tiempos y acciones a seguir	7			8	
24	Abastecimiento de data proveniente de varias fuentes en los sistemas core banco	11		4		
5	Errores colaterales post paso a producción a todo ámbito (desarrollos y/ infraestructuras)	3			9	3
1	Falta clarificar alcance de los cambios	1			9	5
28	Nuevos estándares de calidad en seguridad informática, sin saber mucho lo que implica	4			11	
25	Repositorio de información de los aplicativos poco eficiente, o casi inexistentes	4			7	4
27	Planificaciones muy rígidas	5			6	4
30	La actual metodología banco es cuello de botella para establecer fechas de entrega de cara al cliente	1			13	1
22	Muchas mesas de trabajo para un paso a producción			1	4	10
8	Ambientes no homologados	4			10	1
2	Artefactos (documentos) de poco valor para certificar y evidenciar pruebas entre ambientes	3			9	3
23	Muchos silos de trabajo entre ambientes	4			7	4
29	Creación de nuevas áreas o subgerencias lo que provoca nuevos flujos al ciclo actual	8			6	1
14	Tiempos excesivos para certificar por ambiente	2			3	10
4	Entregas poco prolijas por el equipo de desarrolladores informáticos	1			9	5
12	Data center lentos en aplicar los cambios	9		6		
13	Diagnósticos poco eficientes ante fallas de infraestructura/arquitectura banco	4		2	9	
18	Mesa de ayuda informática poco eficiente	8		7		
19	Poco dominio de la arquitectura banco que soporta los aplicativos	7		2	6	
15	Infraestructura obsoleta para los tiempos actuales	4			10	1

26	Sobre dependencia de plataformas departamentales banco estilo frameware	9		2	4	
16	No contar con Herramientas de control y calidad para el testeo y monitoreo	3			4	8
20	Tiempos excesivos cuando se debe aplicar nuevos cambios de infraestructura	6			9	
21	Eficiencia de los desarrollos entregados	1			9	5
12	Site de contingencia inestables	8		3	2	2
7	Facilidad para cambiar prioridades de entregables y/o proyectos	8		3	4	
3	Excesivo papeleo a registrar	5			6	4
10	Competencias de los equipos de Bancos	5			7	3
9	Cambios en los ambientes sin aviso	9		2	4	
17	Monitoreo inadecuado o marcha blanca poco eficiente en los ambientes productivos	5			5	5

Tabla 22. Respuestas obtenidas.

El gráfico siguiente permite visualizar en resultado final de este trabajo de tesis, donde se puede ver lo que piensan los 15 actores visualizados en porcentaje por voto vs. la percepción sobre si mejoró o no sus dolores.

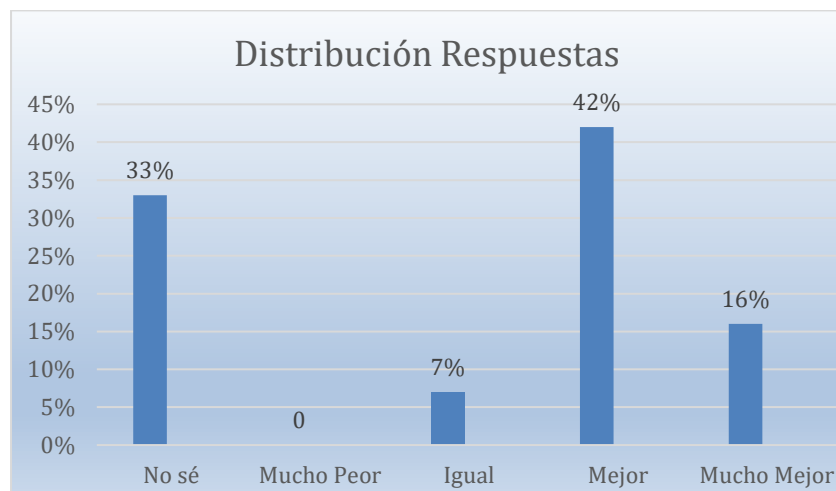


Figura 5. Distribución de las respuestas en la evaluación de la solución.

La figura 5 muestra la distribución de las respuestas en esta evaluación. El 42% de los entrevistados considera que la solución **mejora** la situación actual, basados en los 30 dolores evaluados, mientras que el 16% considera que sí se logra solucionar los dolores existentes y es **mucho mejor** que la situación actual. El 7% considera que el proceso y su forma de trabajo se mantienen **igual** ya que no ve cambios. Se debe también dar a conocer que un 33% de los entrevistados **no sabe** si este nuevo proceso le significa un cambio atractivo en su trabajo (a estas personas se les consulto a estas personas y se logró determinar que les costó trabajar/entender éste nuevo proceso y no siguieron usándolo). Y, por último, no hubo nadie que haya contestado que el nuevo proceso refinado trae consigo un trabajo **peor** al que ya existía.

La siguiente Tabla 23 muestra los porcentajes obtenidos por las respuestas a cada pregunta de la encuesta.

N°	Dolores/insatisfacciones detectadas	no sé (%)	mucho peor (%)	igual (%)	mejor (%)	mucho mejor (%)
6	Proceso compra muy extenso en tiempos y acciones a seguir	47	0	0	53	0
24	Abastecimiento de data proveniente de varias fuentes en los sistemas core banco	73	0	27	0	0
5	Errores colaterales post paso a producción a todo ámbito (desarrollos y/ infraestructuras)	20	0	0	60	20
1	Falta clarificar alcance de los cambios	7	0	0	60	33
28	Nuevos estándares de calidad en seguridad informática, sin saber mucho lo que implica	27	0	0	73	0
25	Repositorio de información de los aplicativos poco eficiente, o casi inexistentes	27	0	0	47	27
27	Planificaciones muy rígidas	33	0	0	40	27
30	La actual metodología banco es cuello de botella para establecer fechas de entrega de cara al cliente	7	0	0	87	7
22	Muchas mesas de trabajo para un paso a producción	0	0	7	27	67
8	Ambientes no homologados	27	0	0	67	7
2	Artefactos (documentos) de poco valor para certificar y evidenciar pruebas entre ambientes	20	0	0	60	20
23	Muchos silos de trabajo entre ambientes	27	0	0	47	27
29	Creación de nuevas áreas o subgerencias lo que provoca nuevos flujos al ciclo actual	53	0	0	40	7
14	Tiempos excesivos para certificar por ambiente	13	0	0	20	67
4	entregas poco Prolijas por el equipo de desarrolladores informáticos	7	0	0	60	33
12	Data center lentos en aplicar los cambios	60	0	40	0	0
13	Diagnósticos poco eficientes ante Fallas de infraestructura/arquitectura banco	27	0	13	60	0
18	Mesa de ayuda informática poco eficiente	53	0	47	0	0
19	Poco dominio de la arquitectura banco que soporta los aplicativos	47	0	13	40	0
15	Infraestructura obsoleta para los tiempos actuales	27	0	0	67	7
26	Sobre dependencia de plataformas departamentales banco estilo frameware	60	0	13	27	0
16	No contar con herramientas de control y calidad para el testeo y monitoreo	20	0	0	27	53
20	Tiempos excesivos cuando se debe aplicar nuevos cambios de infraestructura	40	0	0	60	0
21	Eficiencia de los desarrollos entregados	7	0	0	60	33
12	Site de contingencia inestables	53	0	20	13	13
7	Facilidad para cambiar prioridades de entregables y/o proyectos	53	0	20	27	0
3	Excesivo papeleo a registrar	33	0	0	40	27

10	Competencias de los equipos de bancos	33	0	0	47	20
9	Cambios en los ambientes sin aviso	60	0	13	27	0
17	monitoreo inadecuado o marcha blanca poco eficiente en los ambientes productivos	33	0	0	33	33

Tabla 23. Distribución de las respuestas.

5.2.4. Análisis

Una vez implementado el proceso mejorado, para lograr la automatización y estandarización buscada con apoyo de herramientas DevOps, y una vez respondida la encuesta realizada a los actores que apoyaron en este trabajo, se puede realizar el siguiente análisis respecto al manejo de los dolores con el nuevo proceso. Los siguientes valores muestran que sí se logró mejorar con el nuevo proceso. Particularmente, tenemos 4 dolores/insatisfacciones que fueron resueltos, ya que tenemos actores encuestados que indicaron que sí se trabaja **mucho mejor** en ese ámbito. Estos dolores son los siguientes:

Nº	Dolor/insatisfacción
22	Muchas mesas de trabajo para un paso a producción
14	Tiempos excesivos para certificar por ambiente
16	No contar con herramientas de control y calidad para el testeo y monitoreo
17	Monitoreo inadecuado o marcha blanca poco eficiente en los ambientes productivos

Tabla 24. Dolores resueltos (respuesta mucho mejor).

Además, tenemos 17 dolores/insatisfacciones que fueron resueltos, ya que tenemos actores encuestados que indicaron que sí se trabaja **mejor** ahora en estos ámbitos. Estos dolores son los siguientes:

Nº	Dolor
6	proceso de compra muy extenso en tiempos y acciones a seguir
5	errores colaterales post paso a producción a todo ámbito (desarrollos y/ infraestructuras)
1	falta clarificar alcance de los cambios
28	nuevos estándares de calidad en seguridad informática, sin saber mucho lo que implica
25	repositorio de información de los aplicativos poco eficiente, o casi inexistentes
27	planificaciones muy rígidas
30	la actual metodología banco es cuello de botella para establecer fechas de entrega de cara al cliente
8	ambientes no homologados
2	artefactos (documentos) de poco valor para certificar y evidenciar pruebas entre ambientes
23	muchos silos de trabajo entre ambientes
4	entregas poco prolijas por el equipo de desarrolladores informáticos
13	Diagnósticos poco eficientes ante Fallas de infraestructura/arquitectura banco
15	Infraestructura obsoleta para los tiempos actuales
20	Tiempos excesivos cuando se debe aplicar nuevos cambios de infraestructura
21	Eficiencia de los desarrollos entregados
3	Excesivo papeleo a registrar
10	Competencias de los equipos de bancos

Tabla 25. Dolores resueltos (respuesta mejor).

- Tenemos 8 dolores/insatisfacciones que no fueron resueltos ni tampoco están peor que antes del proceso refinado, ya que son parte de los dolores que cayeron en la categoría de dolores que no corresponden a las funciones que realizan estos actores, es decir, **no saben** de esos dolores. Estos dolores son:

Nº	Dolor/insatisfacción
24	Abastecimiento de data proveniente de varias fuentes en los sistemas core banco
29	Creación de nuevas áreas o subgerencias lo que provoca nuevos flujos al ciclo actual
12	Data center lentos en aplicar los cambios
19	Poco dominio de la arquitectura banco que soporta los aplicativos
26	Sobre dependencia de plataformas departamentales banco estilo frameware
12	Site de contingencia Inestables
7	Facilidad para cambiar prioridades de entregables y/o proyectos
9	Cambios en los ambientes sin aviso

Tabla 26. Dolores resueltos (respuesta no saben).

- Quedando un último dolor que **no mejoro, ni peor**, pero se mantiene igual ya que este proceso mejorado no aportó a su mejora ni empeoramiento. Este dolor es el siguiente:

Nº	Dolor/insatisfacción
18	Mesa de ayuda informática poco eficiente

Tabla 27. Dolores resueltos (respuesta: ni mejor ni peor).

6. Conclusiones y Trabajo a Futuro

El trabajo de tesis se ejecutó en tres fases: 1) identificar dolores, 2) construir la solución requerida y 3) validar la solución desarrollada. En la primera fase se invirtió gran parte del esfuerzo en tener claridad sobre cuáles eran los dolores de la organización con respecto al proceso de desarrollo, y su puesta en producción del software construido, hasta llegar a la entrega al cliente de lo construido. En esta etapa se tuvo la participación de diversos actores que son parte del flujo del proceso de desarrollo, y que permitieron identificar un conjunto los dolores actuales de este proceso.

En la fase de construir la solución requerida, se definieron los objetivos y principios que conformaron el marco de referencia para el diseño de una arquitectura que solucionase los dolores e insatisfacciones detectadas en la fase de identificar los problemas. Previo a generar y dar a conocer la solución tecnológica, se enfocó el trabajo en generar un análisis de cobertura que identifique las actividades, técnicas y herramientas que se verían impactadas por la arquitectura. Es decir, se trabajó en alinear la arquitectura del proceso mejorado con los principios, objetivos tecnológicos y de negocio, y de este modo lograr los cambios esperado con el proceso refinado trabajado en esta tesis. Durante este período de tiempo también se trabajó en establecer las mejores prácticas DevOps, que fueran más adecuadas para la organización Banco Estado.

Finalmente, durante la fase de validación, se realizaron pruebas de piloto que permitieron certificar, verificar la factibilidad técnica y económica de la solución refinada. Además, se realizó una evaluación cualitativa que permitió validar que el proceso refinado cumple en dar solución con los dolores que tenían los actores que apoyaron en este trabajo.

El haber implantado técnicas DevOps al proceso de desarrollo base que existe aún en el Banco Estado, demuestra que no es nada fácil realizar un trabajo colaborativo y técnico en la organización, y por ello, siempre se requerirá de un trabajo colaborativo de todas las gerencias y su gente.

Un aspecto importante para poder lograr este alineamiento, fue descubrir los dolores y socializarlos entre los líderes y los ejecutores de cada etapa del proceso. Con este alineamiento de las distintas partes se logra abordar dolores concretos, y trabajar en la obtención de resultados positivos

6.2 Evaluación general de la propuesta

La última evaluación de satisfacción a este proceso de desarrollo refinado muestra la distribución de las respuestas en esta evaluación. El 42% de los entrevistados considera que la solución mejora la situación actual, basados en los 30 dolores evaluados; mientras que el 16% de ellos considera que sí se logra solucionar los dolores existentes y es mucho mejor que la situación actual. El 7% considera que el proceso y su forma de trabajo se mantienen igual ya que no ve cambios. Por último, nadie contestó que el proceso refinado trae consigo un trabajo peor al que ya existía. Los resultados obtenidos muestran que se valora positivamente la solución propuesta, lo que se considera satisfactorio para el cumplimiento de los objetivos y alcance de esta memoria.

Logros

El objetivo general de este trabajo de tesis es mejorar el proceso de desarrollo y puesta en producción de los productos de software desarrollados por las gerencias de TI de Banco Estado, refinándolo mediante la estandarización y automatización. Este proceso refinado comprobó factibilidad de uso en las gerencias de TI del Banco Estado. Particularmente, la evaluación cuantitativa mostró que la propuesta es técnicamente factible de ser aplicada en las áreas de tecnología del Banco Estado, como también se comprobó que este proceso mejorado es satisfactorio para los actores involucrados.

Con respecto al logro de los objetivos específicos, podemos destacar que:

- Fue posible implantar el proceso refinado a través de estandarizar el proceso, así como el conjunto de herramientas para el desarrollo y puesta en producción. Esto requirió modificar el proceso de desarrollo, entre otras cosas.
- Con respecto al objetivo de reducir el tiempo que insume el proceso general de paso a producción, sí se logró mejorar, ya que varios de los encuestados y actores del nuevo proceso dan a conocer que sí se mejoró con respecto a ciertos dolores que se tenían en ese ámbito.
- Respecto al objetivo de automatizar las pruebas realizadas en testing y QA, lo cual estaba enfocado a reducir el esfuerzo y errores manuales, también se puede mencionar que sí se logró. Ahora el trabajo realizado en las áreas de QA/Testing se ve mejorado, gracias a procesos automatizados que se generaron en el nuevo proceso. El logro de este objetivo también se puede ver en la validación cualitativa de este trabajo de tesis.

Al lograr estos objetivos, el Banco Estado tendrá a su haber un proceso estandarizado, que permitirá dejar disponible frecuentemente nuevas versiones de mayor calidad, que ayuden a entregar beneficios a sus clientes. De esta forma, podrá mantener una imagen corporativa potente que demuestre más eficiencia y calidad de cara al país, indicando que es un banco confiable y con altos estándares de calidad en sus procesos.

Impacto Potencial

Debido al éxito reflejado en el capítulo de validación, y a la buena recepción de los actores y funcionarios que ahora están haciendo uso de este nuevo proceso de desarrollo, las Gerencias de TI del Banco Estado han puesto en práctica el uso de este nuevo proceso, y eso está llevando a que todas las áreas de TI lo utilicen. Este trabajo de tesis va alineado a la hoja de ruta definida por la organización.

El mayor impacto potencial percibido en el proceso de adaptación, ha sido la generación y proyección de los próximos pipelines a construir. Este nuevo proceso ha sido evaluado positivamente por parte de los equipos de desarrollo y testing. También se percibe mayor confianza en la entrega y algunas métricas de calidad han mejorado. Todo esto facilitará la adopción del nuevo proceso cuando podamos contar con una plataforma integral para todos los proyectos de la organización.

Lecciones aprendidas

El principal aprendizaje obtenido en este trabajo de tesis, aplicado en Banco Estado, es la relevancia de la participación de los stakeholders, en la definición y validación de los procesos que se usan en las Gerencias TI del banco. La participación de los stakeholders permitió lograr un trabajo sistematizado y ordenado, para obtener así un proceso refinado que era lo que se buscaba con este trabajo de tesis. Si bien en esta tesis no se logró dar solución a todos los dolores dados a conocer, este trabajo permite tomar un camino que le garantice a la Gerencia TI y al Banco Estado, cómo trabajar en mejoras futuras al nuevo proceso.

La experiencia de este trabajo de tesis ha sido muy grato y enriquecedor, ya que permitió trabajar coordinando y liderando un equipo multidisciplinario que antes nunca realice a esta dimensión de proyecto. Por lo tanto, pensando en trabajos futuros ya está la capacidad de armar equipos en forma más ágil, y sacar proyectos en tiempos más adecuados que es lo que espera el mercado y el Banco Estado como organización.

Trabajo a Futuro

Como trabajo futuro, en el corto plazo se propone realizar un piloto de mayor extensión, incorporando nuevas plataformas de seguridad que están en proceso de compra y licitación en el Banco Estado. Con este piloto podrá detectarse qué problemas poseen nuestros ambientes y fases, qué cosas deben mejorarse en el nuevo proceso de desarrollo, y de esta forma, saber cómo enfrentarlas en los ambientes de contingencia no productivas y de producción.

Otra línea de trabajo consiste en agregar mayor monitoreo de los servicios a la gestión de incidencias, y a la planificación de las mismas. De esta forma, la Gerencia de TI actuará en forma más activa ante las exigencias de los usuarios internos y finales del Banco Estado.

Por último, se propone ir generando equipos en las distintas áreas para tener un trabajo de gestión del conocimiento. De este modo, se busca no perder todo este trabajo de investigación ganado en este tiempo de tesis, sin perder de vista las mejoras que implique estar alineado con el plan estratégico u hoja de ruta del Banco Estado, buscando siempre reducir los dolores organizacionales.

7. Bibliografías

- [1] Sanjeev Sharma, Bernie Coyne. *DevOps For Dummies*, 3rd IBM Limited Edition. John Wiley & Sons, Inc. ISBN 978-1-119-41588-6. 2017.
- [2] Sam Guckenheimer. *What is Monitoring?* 2017. <https://docs.microsoft.com/en-us/azure/devops/learn/what-is-monitoring> (último acceso en diciembre 2021).
- [3] Martin Jackson. *Deliver better software, faster*. 2018. <https://puppet.com/blog/continuous-delivery-vs-continuous-deployment-what-s-diff> (último acceso en diciembre 2021).

- [4] Doug Erickson, Kathryn EE. *Agile Glossary*. <https://docs.microsoft.com/en-us/azure/devops/boards/work-items/agile-glossary?view=vsts> (último acceso en diciembre 2021).
- [5] DISQUS. *Descripción general de IBM UrbanCode Deploy*. 2014. <https://www.ibm.com/docs/es/urbancode-deploy/7.2.1> (último acceso en diciembre 2021).
- [6] Departamento de Abastecimiento Banco República. *Servicio y licencias IBM para herramienta de pasaje entre ambientes UrbanCode*. 2015. http://comprasestatales.gub.uy/Resoluciones/acta_493133.pdf (último acceso en diciembre 2021).
- [7] Rodrigo Garcés. *Docker sumergido en una cultura DevOps*. 2014. <https://es.slideshare.net/ITPAMI/presentacin-docker> (último acceso en diciembre 2021).
- [8] D. C. y. B. Cabtree, *Qualitative Research Guidelines Project*, <http://www.qualres.org/HomeInte-3595.html> [Último acceso en diciembre 2021]
- [9] Descripción general de IBM UrbanCode Deploy. 2014. https://www.ibm.com/support/knowledgecenter/es/SS8NMD_6.1.2/com.ibm.ucbuild.doc/topics/intro_ch.html (último acceso en diciembre 2021).
- [10] Eduardo Díaz. *Mejora del Proceso de Desarrollo de una Empresa de Servicios de Información mediante la incorporación DevOps. Tesis de Magíster en TI. DCC, Universidad de Chile. Mayo 2019.*

8. Anexos

Anexo A: Actividades del proceso base, descrito por fase

A.1. Actividades de la Fase de Desarrollo

- *Desarrollar/construir*: Desarrollar el software o mejoras solicitadas por el cliente, tomando en cuenta la tecnología en la cual está creado o se definió el aplicativo, y considerando también si se debe hacer cambios en la arquitectura de datos de ese desarrollo.
- *Analizar el código estático*: Revisión visual del programador sobre el código fuente creado por el programador, donde evalúa bajo criterio de buenas prácticas de programación y de Normativas Banco; si lo desarrollado cumple con ellas y así poder llevar este desarrollo a producción.
- *Compilar*: Revisar si posee fallas y luego convertirlo en ejecutable siempre y cuando no posea errores.
- Realizar las pruebas unitarias: Comprobar, validar que lo desarrollado está acorde a normativas de programación según el lenguaje de programación; como también de acuerdo al negocio.
- *Revisar el reporte de resultados*: El jefe proyecto junto al líder del equipo de desarrollo da su aprobación o rechazo a lo construido por el(los) programador(es). Si existe algún problema se devuelve al programador,
- *Generar los casos de pruebas y los datos*: Creación de casos de prueba con ejemplos correctos e incorrectos de acuerdo al negocio que se necesita probar en las siguientes fases. También esta actividad es la que debe garantizar la entrega de los datos que servirán para realizar las pruebas en las siguientes fases.
- *Subir fuentes al repositorio*: Respalda y subir todo lo construido a los repositorios que tiene estipulado el Banco Estado para todo tipo de desarrollo de software y mejoras.
- *Generar solicitud de paso a producción*: Es la acción e instrumento a la vez, por el cual se da a conocer a las siguientes fases que un equipo de desarrollo que forma parte de una iniciativa, necesita realizar un paso a producción del desarrollo construido.
- *Generar artefactos*: Creación de documentación que permita entender lo realizado por los programadores y entender pruebas realizadas por ellos.

Actividades de la Fase de Testing

- *Asignar recurso del banco, para atender solicitud paso a producción*: El especialista de testing (funcionario del área de testing) le asignan una solicitud de paso a producción para que él después evalúe y certifique si cumple con los requerimientos instaurados por reglamentación del Banco Estado, si es apto para seguir a la siguiente fase.
- *Descargar fuentes/instalar*: Descargar mediante la herramienta TFS la versión del desarrollo, que el equipo de programadores creó y que debe ser revisado por el especialista de Testing.

- *Compilar*: Revisar si posee fallas y luego convertirlo en ejecutable siempre y cuando no posea errores.
- *Chequear normativas*: Revisar y validar que el desarrollo que se está revisando, cumpla con la normativa tanto de programación y/o de Banco Estado, ya que muchas veces se debe excepcionar algunos desarrollos dependiendo de la antigüedad de lo construido u otros hechos a considerar; que el Banco Estado solicita sean revisados.
- *Revisar el listado de piezas del desarrollo*: Revisar que todas las piezas que sean llevadas a producción, correspondan a las indicadas por el equipo de desarrollo; ya que en ocasiones se pasan piezas que no estaban contempladas en la solicitud de paso a producción.
- *Ejecutar la carga de datos para pruebas*: Ejecutar el script o carga de datos, de este modo a posterior si se pueda realizar las pruebas funcionales para certificar lo construido por el equipo que desarrolló las piezas o software.
- *Realizar las pruebas unitarias*: Comprobar, validar que lo desarrollado está acorde al negocio que se busca resolver por medio de la iniciativa.
- *Realizar las pruebas integración*: Comprobar, validar que lo desarrollado por más de un equipo de desarrollo o aplicativo, pero que son parte de la misma iniciativa, están acorde al negocio que se busca resolver en conjunto (aplicativos).
- *Realizar las pruebas de rendimiento (performance/carga/stress)*: Se usan para mitigar riesgos relacionados con la continuidad del negocio, potenciar o asegurar la buena imagen corporativa, que podría verse afectada ante el mal funcionamiento de sus aplicaciones. Y en última instancia para controlar los costos derivados de la gestión correcta.
 - *Realizar las pruebas de rendimiento (performance testing)*: Validan la escalabilidad, estabilidad y velocidad (tiempos de respuesta) del sistema bajo evaluación. Niveles de uso de los recursos que cumplan con los objetivos de desempeño para el sistema.
 - *Realizar las pruebas de carga (load testing)*: Esta subcategoría de pruebas de rendimiento se centra en determinar o validar las características de rendimiento del sistema o la aplicación cuando es sometido a cargas de trabajo y los volúmenes de carga prevista para la operación normal en producción.
 - *Realizar las pruebas de estrés (stress testing)*: Pruebas centradas en validar características de rendimiento del sistema bajo condiciones superiores a lo previsto en las operaciones de producción (grandes volúmenes de carga).
- *Aprobar/rechazar la solicitud de paso a producción*: El especialista de testing (funcionario del área de testing) al cual le asignaron la revisión de la solicitud de paso a producción que fue generada por el líder de desarrollo en la fase de desarrollo, debe certificar que las piezas que contiene la solicitud de paso a producción cumple tanto en normativa del Banco Estado como también cumple con las regulaciones que exige cada lenguaje de programación, es decir, revisa, certifica y da una primera aprobación para avanzar a las fases siguientes. Si la solicitud de paso a producción no es aprobada por el especialista de

Testing, es devuelta al líder de desarrollo o es rechazada y el líder del desarrollo deberá crear una nueva solicitud de paso a producción.

- *Generar artefactos:* Creación de documentación realizada por las personas que trabajan en testing, y permite visualizar las pruebas realizadas, así como las piezas que se espera que lleguen a producción.

En Fase de QA/Preproducción:

- *Asignar recurso banco, para atender solicitud paso a producción:* Al especialista de preproducción (funcionario del área de preproducción) le asignan una solicitud de paso a producción para que él después evalúe y certifique si cumple con los requerimientos instaurados por reglamentación del Banco Estado si es apto para llegar a producción. En esta fase interactúa los roles: QA, jefe proyecto y Programador.
- *Descargar fuentes/instalar:* Descargar mediante la herramienta TFS la versión del desarrollo, que el equipo de programadores creó y que debe ser revisada por el especialista de testing. En esta fase interactúa los roles: QA.
- *Ejecución de carga de datos para prueba:* Realizar la carga de datos en los ambientes pre-productivos, de cara a ejecutar pruebas funcionales o casos de uso del desarrollo que está pasando a producción. En esta fase interactúa los roles: QA, y con apoyo del Programador de ser necesario.
- *Implementar las mesas técnicas:* En estas mesas se ve la factibilidad de pasar a producción el desarrollo; revisando que no tenga problemas de Seguridad y/o de riesgos operacional (mesa de seguridad), también si es necesario pasar a producción con alguna excepción o la iniciativa viene con problemas, pero debe pasar a producción (mesa de gerencia). En esta fase interactúa los roles: Especialista de Riesgo, Especialista Seguridad, jefe proyecto.
- *Implementar las mesas de administración de cambios:* Reunión para velar por la integridad del ambiente productivo, es decir, ver la factibilidad técnica del paso a producción; revisando que no se vea impactado el ambiente productivo con este nuevo paso a producción y no se dañe lo existente en producción. En esta fase interactúa los roles: especialista Administrador de Cambio, jefe proyecto.
- *Aprobar/rechazar:* El especialista de Preproducción o QA (funcionario del área de preproducción) al cual le asignaron la revisión de la solicitud de paso a producción que fue generada por el líder de desarrollo en la fase de desarrollo, debe certificar que las piezas que serán certificadas en el ambiente en un ambiente pre-productivo cumple con las regulaciones que se exigen para estar en la fase final de producción (existir permisos, cumplir con el rendimiento exigido, normas de seguridad, etc.). Si la solicitud de paso a producción no es aprobada por el especialista de preproducción, es devuelta al líder de desarrollo o es rechazada y el líder del desarrollo deberá crear una nueva solicitud de paso a producción. En esta fase interactúan los roles: especialista, administrador de cambio, ingeniero QA, y jefe proyecto.

- *Generar artefactos*: Creación de documentación realizada por las personas que trabajan en QA, y permite visualizar las pruebas realizadas, así como las piezas que se espera que lleguen a producción. En esta fase interactúa los roles: QA, jefe proyecto.

En Fase de Producción:

- *Asignar recurso del banco, para atender solicitudes de paso a producción*: Al especialista de producción (funcionario del área de producción) le asignan una solicitud de paso a producción para que él evalúe y certifique si cumple con los requerimientos exigidos en producción. En esta fase interactúa los roles: especialista Administrador de Cambio, jefe proyecto y Programador.
- *Habilitar/disponibilizar ambientes*: Para los casos de paso a producción para tecnologías .NET y Bus de Servicios, se habilitan piezas o se configuran entornos necesarios para cumplir con las exigencias de este tipo de tecnologías; por ejemplo: datapower, telecomunicaciones, etc. En esta fase interactúa los roles: Especialistas de seguridad, especialista Telecomunicaciones, especialista Middleware.
- *Instalar en producción*: Instalar en producción el desarrollo construido que permitirá resolver el requerimiento del usuario o cliente. En esta fase interactúa los roles: especialista Administrador de Cambio.
- *Ejecutar vuelta atrás*: En caso que el paso a producción no cumple con lo solicitado por el usuario o cliente, se debe volver atrás lo construido para que sea reparado y/o mejorado tal como solicita el cliente. En esta fase interactúa los roles: especialista Administrador de Cambio, jefe Proyecto y Programador.
- *Pilotear o marcha blanca*: Verificar y validar que lo construido es lo que se pidió, y que en ambiente de Producción funciona correctamente. En esta fase interactúa los roles: Usuario Certificador.
- *Generar artefactos*: Creación de documentación realizada por las personas que trabajan en el área de operaciones o es el mismo cliente, y permite visualizar las pruebas realizadas y que evidencia también que todo está de acuerdo a lo solicitado inicialmente en la iniciativa. En esta fase interactúan los roles: Especialistas de seguridad, especialista Telecomunicaciones, especialista Middleware, usuario certificador, y jefe proyecto.

Anexo B: Roles de los participantes en el proceso

A continuación, se describen los roles de los participantes en el proceso definido.

B.1. Cliente

La figura 6 muestra los distintos tipos de cliente que pueden participar del proceso. Luego se describen las acciones que puede realizar cada uno.

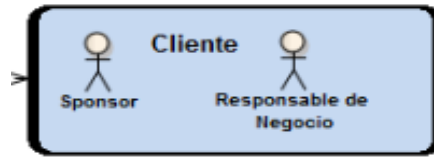


Figura 6. Roles del cliente.

Sponsor. El sponsor del proyecto es quien patrocina y apoya a la Iniciativa (cualquiera que sea su tipo: Proyectos Estratégicos, Proyectos, Mantenimientos, etc.) por parte de la organización y mantiene alineado al negocio con el cambio que se producirá con el nuevo proyecto. Sus responsabilidades y tareas son:

- Este rol también podrá ser ejercido por la misma persona que ejerza el rol de Ejecutivo Responsable de un Programa Estratégico siempre y cuando ésta no ejerza a su vez como responsable de Proceso dentro de él.
- Proveer un patrocinio real a la iniciativa y que ésta sea de conocimiento en la organización.
- Articular las metas de la iniciativa, exponerla y formalizarla, explicitando su aporte a la estrategia.
- Lograr el compromiso a nivel ejecutivo y de la organización con los objetivos Estratégicos que tenga el tipo de iniciativa.
- Proveer dirección y asegurar la captura del valor de los resultados de la Iniciativa en la organización que está patrocinando.
- Comunicar su visión y mantener alineados a todos los grupos de interés de la organización.
- Definir y evaluar al responsable de Iniciativa, aportar su visión en la constitución del equipo y brindar soporte para el cambio.
- Participar en la marcha de la iniciativa para entregar su aprobación para iniciar o continuar actividades claves o para resolver problemas críticos que requieran su involucramiento. Es el responsable final por los cambios de alcance de la iniciativa.
- Controlar el alineamiento estratégico del equipo, es decir, asegurar la consistencia entre la estrategia y la provisión de las capacidades aportadas por la iniciativa, así como la forma y la secuencia del despliegue de tales capacidades.
- Da cuenta de la iniciativa y su estado de avance donde esté previsto.

Responsable de negocio. Es el responsable de los entregables del Modelo Comercial, realiza la formulación del proyecto, alcances, objetivos, beneficios cualitativos, cuantitativos, Requerimiento Comercial, caso de negocio, estrategia de piloto y despliegue e indicadores de satisfacción y checklist de marketing. Para ello debe:

- Generar acuerdos con los responsables de Procesos y AFAs que son intervenidos por la iniciativa con el objetivo de asegurar la posterior implementación.
- Entregar al proyecto de forma oportuna todas las definiciones de negocio y validar con riesgo operacional, fiscalía y contraloría en caso de ser necesario.
- Unificar criterios de negocio en caso de que la iniciativa involucra a más de una Banca.

- Participar activamente de las instancias de seguimiento y control de avance del proyecto, asegurando el cumplimiento de los compromisos relacionados con el negocio.
- Escalar el JP y/o LE riesgos, problemas o puntos de atención relacionados con el proyecto, proponer planes de mitigación desde la perspectiva del negocio y obtener los consensos necesarios.
- Participar en la definición y actualización de requerimientos comerciales y aprobación de los documentos de especificación funcionales y definición de la Interfaz de Usuario.
- Para la etapa de despliegue, coordinar las acciones necesarias con el área de comunicaciones y área de Marketing, controlando los hitos asociados a estas acciones.

B.2. Expertos

La figura 7 muestra los distintos tipos de expertos que pueden participar del proceso. Luego se describen las acciones que puede realizar cada uno.



Figura 7. Tipos de expertos.

Responsable del Proceso. Administrar y controlar el proceso a cargo, con la finalidad de mantener el proceso estable, garantizando el cumplimiento de los niveles de servicio y generando las acciones correctivas pertinentes para lograr los objetivos de calidad, eficiencia y riesgo operacional.

- Asesora en la identificación de requerimientos y gestiona las mejoras del proceso en las etapas pertenecientes a su ámbito, para lograr el objetivo: Calidad- eficiencia – riesgo operacional, durante toda su participación en el desarrollo de la iniciativa y solución de la misma
- Propone los indicadores y los acuerdos de niveles de servicios y entregables de las etapas del proceso a su cargo con la coordinación de los otros responsables involucrados.
- Monitorea y controla el cumplimiento de los acuerdos de sus etapas del proceso.
- Administra permanentemente y en detalle las etapas del proceso a cargo, incluyendo su plataforma tecnológica en los aspectos funcionales.
- Gestiona la mejora continua del proceso.
- Define y lidera el diseño de las etapas de los procesos a su cargo, estableciendo directrices.
- Asesora en la formulación de requerimientos asociados a la oferta de valor.
- Aprueba y formaliza los cambios a los procesos y/o etapas de su responsabilidad. Asegura la actualización de la documentación de la etapa del proceso.
- Evalúa con la mirada del proceso del cual es responsable, la iniciativa antes de ingresar al ciclo de demanda.
- Actúa de experto en relación al conocimiento del proceso de negocio con los jefes de proyecto.

- Al final del proyecto de desarrollo e implementación recibe el nuevo proceso para su puesta en marcha y entrada en operación, participando además, en la certificación y liderando el paso a producción de la iniciativa que lo soporta.

En el caso de mantenimientos, el responsable del proceso podría desempeñar además el rol del responsable de la Iniciativa, el cual debe ser provisto por el Cliente. En el caso de un Proyecto Estratégico, el rol de responsable de Proceso podrá ser ejercido también por el “Ejecutivo Responsable”.

Administrador funcional de aplicación (AFA). Tiene como rol la administración de la(s) funcionalidades de la(s) Aplicación(es) a su cargo y su entorno directo, con la finalidad de asegurar su correcto funcionamiento y su relación con otras aplicaciones, y la preservación del conocimiento acerca de las reglas de negocio implementadas en las funcionalidades del sistema. Se entenderá por aplicación a un conjunto de funcionalidades que son ofrecidas por un sistema computacional, que permite la obtención de los objetivos del proceso al que presta servicios, sistematizando actividades. Ejemplos de aplicaciones son las siguientes: Sendmail (SDM); PU Transversal (PUT); Crédito Moneda Nacional (CRE); etc. Las funciones asociadas al Administrador Funcional de Aplicación son las siguientes:

1. Entregar asesorías funcionales
2. Integrar demanda sobre aplicativo
3. Proponer perfiles y accesos
4. Coordinar la ejecución de pruebas funcionales
5. Generar y ejecutar pruebas funcionales
6. Aprobar y certificar
7. Planificar y ejecutar acciones asociadas a la Contingencia
8. Realizar mejora continua
9. Informar los contenidos en las funcionalidades
10. Documentar
11. Ejecutar acciones asociadas a órdenes de proceso

Responsable de la aplicación. Este rol es el responsable de gestionar, coordinar y asegurar la resolución en mantenimientos e incidencias sobre las aplicaciones del banco. Además debe:

- Determinar el enfoque y alcance de las incidencias.
- Gestionar el pool de incidencias que deben resolverse.
- Realizar y/o revisar las estimaciones y premisas de estimación para el plan de incidencia, esfuerzo y costos.
- Medir y monitorear la incidencia para asegurar el cumplimiento del cronograma, presupuesto y las expectativas. Balancear el alcance, plan de incidencia, presupuesto, calidad y riesgos.
- Desarrollar, implementar y mantener el plan de incidencia, incluyendo los riesgos, problemas y comunicación.

- Asegurar e informar el cierre de las incidencias.
- Administrar las relaciones y coordinar el trabajo entre los diferentes equipos en los distintos lugares, con los proveedores y subcontratados.
- Validar que el desarrollo resuelve la incidencia.
- Comunicar al Gestor Funcional y al Administrador Funcional de Aplicaciones (AFA) correspondiente el estado y avance de la incidencia en curso, sean tanto Mantenciones como proyectos.
- Asegurar junto al equipo que la incidencia solucionada cumple con los objetivos de calidad, rendimiento y eficacia establecida, es decir, sea puesta en producción en el menor tiempo posible pero que cumpla con la calidad requerida.
- Asegurarse que el equipo de incidencia y los Ejecutores comprendan completamente la urgencia, diseños y requerimientos de la incidencia.
- Trabajar con los gestores, especialistas funcionales y Administradores Funcionales de Aplicaciones (AFAs) para asegurar que el diseño y la realización de las pruebas sean las necesarias para la incidencia.
- Asegurar la entrega de información oportuna al AFA antes de la puesta en producción de cambios en el aplicativo respectivo, respecto a:
 - Acciones de respaldo y recuperación de información de las aplicaciones, especificando los elementos a respaldar, periodicidad de respaldo y mecanismos de recuperación de la información.
 - Cambios en el perfilamiento y funcionalidades de la aplicación.
 - Fechas del paso a producción y de toma de control, acordadas con el Líder de Desarrollo de la iniciativa que está impactando el aplicativo.
- Participar en actividades acordadas con el Líder de Desarrollo para generar Documento de Toma de Control.
- Asistir a reunión de Presentación del Documento de Toma de Control, definir acuerdos y condiciones que deberán quedar establecidos en el mencionado documento antes de dar su aprobación para dar por cerrado los preparativos de la Toma de Control.
- Participar en la aprobación de la Arquitectura Aplicativa del aplicativo impactado por la solución de la Iniciativa, en fase de Diseño.

B.3. Especialistas

La figura 8 muestra los distintos tipos especialistas entre ellos los arquitectos que pueden participar del proceso. Luego se describen las acciones que puede realizar cada uno.

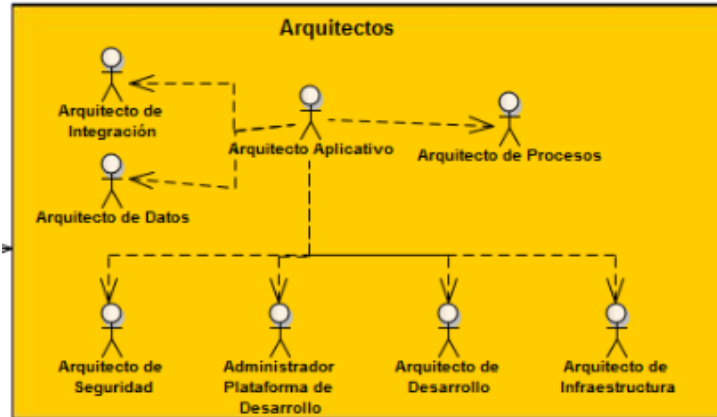


Figura 8. Tipos de arquitectos

Arquitecto de aplicativo. Es el responsable de asegurar que los distintos módulos de la aplicación sean diseñados de acuerdo a los lineamientos de la arquitectura aplicativa para el desarrollo de la solución. Además, debe:

- Validar los componentes de Software de la aplicación a ser reutilizados por cada uno de los módulos que componen la aplicación completa.
- Asegurar el uso de los lineamientos en la definición de la arquitectura aplicativa en los distintos módulos que componen la nueva aplicación.
- Ejecutar la validación sobre la solución punta a punta construida en la Gestión del Ciclo del Desarrollo.
- Administrar el mapa aplicativo de las Aplicaciones.
- Alinear la Arquitectura Aplicativa con la de Procesos, con el apoyo del Arquitecto de Procesos.
- Apoyar en la definición de los requerimientos funcionales y no funcionales.
- Implementar las definiciones existentes en el marco de seguridad aplicativa que regulan el diseño de las aplicaciones intervenidas por las iniciativas.
- Definir y construir las piezas de Arquitectura de Sistemas.
- Definir las normas y estándares de codificación de Sistemas.

Arquitecto de datos. Este es el responsable de asegurar que el modelo de datos, definido en cada iniciativa, esté alineado al modelo corporativo de datos del Banco.

- Mantener el mapa de datos del Banco, los diccionarios de datos y modelos de bases de datos.
- Asegurar el uso de los estándares en la definición de nuevos modelos de datos.
- Ejecutar la validación de los modelos de datos construidos en Gestión de Desarrollo.
- Asegurar que las aplicaciones sean usuarias de la información y no dueñas.
- Dar soporte en la identificación de los datos, donde y como obtenerlos.
- Apoyar y dar soporte al ciclo de desarrollo en las especificaciones de diseño de modelos y esquemas de datos, diagramas de flujo de datos y controles de calidad de los datos.
- Revisar los desarrollos que impacten a la base de datos corporativa y asegurar la calidad de los desarrollos y el uso de los estándares.

- Validar que la documentación de los modelos de datos conceptuales, lógicos y físicos sean modificados cuando corresponda.
- Definir la forma de acceder a los datos del repositorio.
- Participar en el diseño de las interfaces de usuario que se integran con el repositorio de datos.
- Responsable por la creación y actualización de documentos y principales guías de la Arquitectura Empresarial de Información.
- Velar y difundir las definiciones existentes en el marco de seguridad que regulan los Modelos de Datos de las aplicaciones intervenidas por las iniciativas.
- Arquitecto de Desarrollo
- Participar y contribuir en el desarrollo y mejora de estándares y normas, y métodos de controles enfocados a la construcción de software.
- Generar, fortalecer y optimizar los ciclos de desarrollo.
- Preparar y capacitar a las iniciativas de desarrollo en el uso correcto de: estándares y normas, y ciclos de desarrollo, enfocados a la construcción de software.
- Velar que las iniciativas usen estándares y normas a lo largo del ciclo del desarrollo.
- Declarar y reportar la No Aplicación de estándares y normas de desarrollo encontrados en los sistemas de software a la organización.
- Canalizar los requerimientos técnicos de la GTEC factibles de ser gestionados con Arquitectura de Sistemas e Integración Funcional para el fortalecimiento continuo de: estándares y normas de desarrollo, y ciclos de desarrollo.

Arquitecto de integración. Es el responsable de asegurar el uso de los estándares para la integración de las aplicaciones en el Banco. Además, debe:

- Diseñar y/o validar la integración entre las aplicaciones.
- Mantener el mapa de integración del Banco.
- Administrar y asegurar el uso del catálogo de los servicios de integración disponibles. (responsabilidad del equipo Servicios Multicanal)
- Asegurar el uso de los estándares de integración.
- Ejecutar la validación de los estándares de integración para las aplicaciones construidas.
- Apoyar en la definición de las piezas de arquitectura y su integración.
- Implementar las definiciones existentes en el marco de seguridad aplicativa que regulan las integraciones entre las aplicaciones intervenidas por las iniciativas.
- Mantención de las piezas de arquitectura de Sistemas.
- Realizar definiciones de Software Básico.
- Apoyar en la definición de los requerimientos funcionales y no funcionales

Arquitecto de procesos. Es el responsable de asegurar los lineamientos en la definición de los procesos necesarios para soportar al negocio.

- Asegurar el uso de la metodología en la definición de nuevos procesos y en el mantenimiento de estos.

- Asegurar y apoyar en el uso de las herramientas de monitoreo, desarrollo y operación de los procesos
- Ejecutar la validación de los procesos construidos o modificados en la Gestión del Ciclo de Desarrollo.
- Responsable por la creación y actualización de documentos y principales guías de la Arquitectura Empresarial de Procesos.
- Velar y difundir las definiciones existentes en el marco de seguridad que regulan los procesos intervenidos o creados por las iniciativas.

Arquitecto de seguridad. Es el encargado de entregar los requerimientos de seguridad necesarios para que las Iniciativas de Desarrollo estén en cumplimiento con la política, normativas y estándares de seguridad de la Corporación. Además, debe:

- Proponer recomendaciones de seguridad, alineados a estándares de seguridad de la industria y buenas prácticas
- Especificar los componentes tecnológicos de seguridad que permitan implementar los requerimientos y recomendaciones de seguridad
- Velar por que las tecnologías que se implementan cumplan con los estándares de seguridad del banco
- Prestar asesorías de arquitectura de seguridad durante el ciclo de desarrollo (Atender dudas con respecto a la aplicabilidad de los estándares de Seguridad, apoyar el levantamiento de puntos críticos de seguridad de software y servicios, declarar amenazas y vulnerabilidades que pueden poner en riesgo la seguridad de la información, entre otros)

Administrador de plataforma de desarrollo

- Disponer, controlar, administrar, gestionar y soportar servicios de plataforma para Desarrollo, según definiciones y estándares.
- Apoyar la implementación de software en ambientes de desarrollo para desarrollar las bases de instalación de ambientes y aplicaciones.
- Velar por el cumplimiento de los estándares y definiciones de arquitectura en la implementación de software sobre la plataforma de Desarrollo.
- Canalizar solicitudes de las áreas de desarrollo respecto a información de configuración o elementos existentes en ambiente de Producción.

Arquitecto de infraestructura. Es el responsable de definir la arquitectura de infraestructura tecnológica para el Banco Estado. Además, debe:

- Analizar, definir y cuantificar las necesidades asociadas a requerimientos sobre la infraestructura tecnológica de Banco Estado, proponiendo y generando alternativas viables de implementar desde el punto de vista de la tecnología y los estándares definidos.
- Estandarizar, desarrollar y definir políticas o normas asociadas a la Infraestructura Tecnológica.
- Apoyar desde la mirada de Arquitectura de Infraestructura los nuevos diseños de soluciones aplicativos para el negocio.

- Desarrollar los diseños de la Arquitectura Tecnológica para los ambientes requeridos.
- Desarrollar las evaluaciones, estimaciones y análisis de arquitectura de infraestructura tecnológica del Banco.
- Apoyar en la construcción de casos de negocios en los aspectos tecnológicos desde la mirada de la infraestructura.
- Desarrollar y mantener el mapa de la infraestructura tecnológica, analizando el TOBE y el Roadmap de obsolescencia del hardware y software base.
- Velar por el cumplimiento permanentemente del Plan de Procesos, Sistemas y Tecnología en lo que a definiciones de hardware y software se refiere.
- Investigar y analizar las nuevas Tecnologías disponibles en el Mercado para el Banco con el fin de proponer y definir nuevas alternativas viables de solución.
- Apoyar en la definición de los requerimientos funcionales y no funcionales.

El especialista de iniciativa en la figura 9 muestra los distintos tipos de especialistas de iniciativa que pueden participar del proceso. Luego se describen las acciones que puede realizar cada uno.



Figura 9. Tipos de especialista iniciativa

Especialista de aplicación. Es el responsable de realizar los diseños y construcción de las diferentes piezas de software que constituyen la solución del requerimiento. Además, debe:

- Transformar y evaluar los requerimientos funcionales en desarrollo de soluciones operativas.
- Apoyarse en el Arquitecto de aplicación para entender los lineamientos, estándares a utilizar en el desarrollo.
- Asistir en la definición y revisión de requerimientos y casos de uso para el desarrollo de una solución.
- Realizar diseños funcionales de la solución.
- Informar los resultados al Especialista de Procesos y Aplicación (éste último reporta al arquitecto) de posibles problemas que pueden afectar a otras áreas del desarrollo de la iniciativa.
- Aplicar el control de versiones de la iniciativa
- Participar en la transición de los entregables de diseños a los Ejecutores, y garantizar una comprensión clara y completa de éstos.

- Completar todas las documentaciones relacionadas a la aplicación (manuales, entregables) requeridas por la metodología en las etapas que participe.
- Analizar, diseñar y apoyar en la construcción de los componentes de la solución.
- Coordinar el desarrollo de las iniciativas junto a los proveedores cuando corresponda.
- Tomar contexto de cada requerimiento de alto nivel para realizar la identificación de los requerimientos funcionales y no funcionales que se desprendan, como también de cualquier otro requerimiento que surja en el análisis y diseño de la solución, con el fin de documentarlos y clasificarlos
- Confeccionar el artefacto “especificación de requerimiento funcional” asociada a una funcionalidad, dentro de un proyecto y de realizar las actualizaciones de este documento cuando una mantención lo requiera,
- Validar artefacto “especificación de requerimientos”, con todos los involucrados (cliente, proyecto, AFAs, RA, entre otros).

Especialista de procesos. Es el responsable de realizar los diseños y construcción de los diferentes procesos que involucren la solución al requerimiento.

- Conocer el mapa de procesos del Banco.
- Realizar los diseños y construcción de los diferentes procesos del proyecto o mantenimiento.
- Apoyarse en el arquitecto de proceso para entender los lineamientos, estándares a utilizar en el desarrollo.
- Realizar el trabajo de acuerdo a la metodología y los procedimientos establecidos por el Banco.
- Diseñar el flujo de interacción para los procesos que requieren procedimientos manuales.
- Diseñar la implementación de cada proceso a ser automatizado mediante BPM (Business Process Management).
- Dar soporte al desarrollo de los procesos.
- Detectar problemas en el ámbito de procesos, comunicarlos y escalarlos al Líder de Proceso, cuando sea necesario y buscar las soluciones necesarias.
- Actualizar el mapa de procesos del Banco, bajo la supervisión del Arquitecto de Procesos y Líder de Proceso.
- Completar todas las documentaciones relacionadas con temas de procesos (manuales, entregables) pertinentes requeridas por la metodología, y demás equipos que brindarán el soporte a la solución.

Desarrollador. Es el responsable de diseñar, construir, desarrollar y probar los componentes de la solución realizando y ejecutando pruebas unitarias. Además, debe realizar diversas actividades de mantenimiento:

- Realizar el diseño detallado de los componentes de la solución cuando corresponda.
- Construir y desarrollar los componentes de la solución.
- Realizar las pruebas unitarias de los componentes de la solución.
- Preparar junto al Especialista de Pruebas los datos de prueba a utilizar en la ejecución de la misma.
- Realizar y ejecutar las pruebas de los componentes de la solución.

- Preparar y ejecutar las pruebas no funcionales (ensamble, producto, conversión/migración, performance y procesos).
- Participar en la construcción y revisión de código.
- Informar y levantar a los especialistas los riesgos que detecten.
- Participar en el traspaso de la solución o de los componentes en las pruebas.
- Corregir cualquier defecto o problema descubierto en los scripts de pruebas.
- Realizar actividades de mantenimiento.
- Gestionar el esfuerzo del diseño de interfaz de usuario, incluyendo el diseño visual.
- Diseñar la distribución de las páginas y la navegación.
- Asegurar que los diseños de interfaz de usuario cumplan con los requerimientos y la arquitectura que los soporta.
- Depurar los medios utilizados en el prototipo para ser integrados en la solución.
- Ayudar en el desarrollo de materiales de baja y alta fidelidad, tales como mapas de sitio, pruebas del sistema, facilidad de uso.
- Elaborar informes de usabilidad.
- Preparar el Plan de Vuelta Atrás con el apoyo de especialistas en Integración, arquitectura, implementación y aplicación.

Especialista de pruebas funcionales. Es el responsable de liderar, organizar y coordinar las pruebas. Además, es el responsable de generar y monitorear el Plan de Pruebas, junto con la generación y ejecución de los scripts de pruebas.

- Validar y aprobar los informes/reportes generados para la Etapa de Pruebas para el desarrollo de cada proyecto, mantenimiento e incidencia.
- Definir en conjunto con el Líder de Desarrollo la estrategia de Pruebas a utilizar en todas las etapas de Pruebas definidas para el desarrollo de cada proyecto, mantenimiento e incidencia.
- Trabajar con otros miembros del equipo para planificar todas las pruebas (de producto, conversión/migración, ensamble, performance, procesos y aceptación de usuario).
- Definir los planes de prueba y criterios de aceptación para todos los tipos de pruebas a realizar (Pruebas de producto, conversión/migración, ensamble, performance, procesos y aceptación de usuario).
- Propone métricas para gestionar y controlar el desarrollo de cada ciclo de prueba (Cantidad de defectos detectados, defectos corregidos por día, cantidad de casos ejecutados por día).
- Aprobar criterios de entrada/salida a ser utilizados en las pruebas (Datos de prueba y resultados esperados).
- Liderar y/o coordinar todos los esfuerzos relacionados con las pruebas.
- Actuar como primer punto de contacto para los encargados de ejecutar y de validar las pruebas.
- Administrar y dirigir el trabajo del equipo de pruebas equilibrando la capacidad del equipo y el cronograma de pruebas.
- Asegurar el seguimiento de los estándares definidos, guías y metodologías establecidos para las pruebas.

- Monitorear el trabajo del equipo durante el desarrollo del plan y la ejecución de la prueba a fin de asegurar que las pruebas se realicen en tiempo y forma, según lo planificado.
- Validar los resultados de las pruebas y asegurar que se satisfagan los criterios de entrada y salida para caso de prueba.
- Asignar recursos para la ejecución de los ciclos de prueba, según se requiera.
- Coordinar y validar que los ciclos de pruebas aseguren el seguimiento de los defectos detectados (identificación, solución, re-testeo y migración).
- Asegurar que todos los resultados de las pruebas estén disponibles para su revisión (como por ejemplo: repositorio centralizado de la evidencia obtenida, directorio de red).
- Desarrollar documento de cierre de prueba, que contenga todos los resultados obtenidos por cada ciclo de prueba realizado.
- Apoyar a los Ejecutores y a los equipos de prueba en la creación de los datos de prueba.
- Definir la estrategia de las pruebas.
- Definir estándares de los entregables utilizados en la fase de pruebas.
- Crear el plan de pruebas.
- Asistir a los especialistas en la estimación del impacto de las pruebas.
- Desarrollar los scripts y las condiciones de prueba, así como los resultados esperados para el(los) plan(es) de prueba, incluyendo las pruebas de producto, conversión/migración, ensamble, performance, procesos y aceptación de usuario.
- Desarrollar y mantener procedimientos de prueba.
- Ejecutar y validar los scripts de pruebas.
- Actualizar los datos que se utilizan en la ejecución de las pruebas.
- Monitorear los resultados de la ejecución de las pruebas.
- Registrar los problemas surgidos durante las pruebas conforme a los planes de gestión de problemas y algunos problemas menores que tenga el equipo.
- Informar al Líder de Desarrollo sobre cualquier problema que pueda afectar el cronograma, el presupuesto o calidad del producto o el proceso de prueba.
- Preparar junto al ejecutor los datos de prueba a utilizar en la ejecución de la misma.
- Apoyar en la definición de métricas de la ejecución de las pruebas.
- Determinar todas las herramientas y requerimientos de ambiente para realizar las pruebas.
- Preparar y ejecutar las pruebas no funcionales (ensamble, producto, conversión/migración, performance y procesos).
- Generar los informes de resultados de ejecución de las pruebas.
- Completar todas las documentaciones relacionadas a temas de pruebas (manuales, entregables) pertinentes requeridas por la metodología.

Especialista de Gestión del Cambio. Este es el responsable de liderar, organizar y coordinar el frente de Gestión del Cambio, generando, diseñando (materiales), validando y monitoreando los planes de Gestión del Cambio definidos para cada iniciativa y elaborados sobre la base de los impactos, en los formatos definidos. Para ello debe:

- Elaborar, en conjunto con el equipo de proyecto, el análisis de impacto y plan (es) de mitigación en la organización, en sus distintos ámbitos (organizacionales, estructura, funciones, etc.).
- Generar las estrategias de Capacitación y Comunicación en conjunto con el equipo de proyecto velando por su ejecución.
- Construir los planes de capacitación y comunicación, con sus respectivos cronogramas de trabajo, con asignación de responsables y de plazos.
- Definir y validar, en conjunto con el equipo de proyecto y usuario final, los materiales de capacitación, comunicación, y soporte.
- Generar, en conjunto con el equipo de proyecto, los materiales de Capacitación y Comunicación.
- Trabajar con los especialistas del proyecto para evaluar la capacitación y comunicación y determinar mejoras.
- Velar por la calidad de los entregables relativos a la capacitación, comunicación, soporte y diseño organizacional.
- Diseñar una estrategia de piloto en conjunto del equipo de proyecto y especialistas para evaluar efectividad de capacitación y comunicación.
- Monitorear y dar seguimiento al desarrollo de la capacitación y comunicación.
- Ejecutar, en conjunto con el equipo de proyecto, las actividades definidas en el plan de comunicación y capacitación.
- Ejecutar planes de mitigación al interior de la organización, en caso de ocurrir o requerir cambios en la estructura interna y/o funciones.

El especialista GSIS en la figura 10 muestra los distintos tipos de especialistas GSIS que pueden participar del proceso. Luego se describen las acciones que puede realizar cada uno.

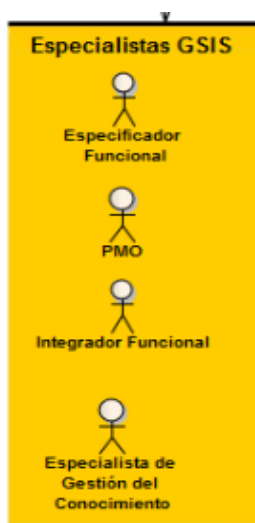


Figura 10. Tipos de especialistas GSIS

Especialista funcional. Este es el responsable de participar del desarrollo de la iniciativa, dando soporte y definiciones para superar las distintas etapas del proceso.

- Participar en la definición de requerimientos funcionales y casos de uso.
- Participar de la aprobación de los Documentos de Requerimientos Funcionales y definición de la Interfaz de Usuario.
- Obtener la aprobación de los usuarios de la definición de la Interfaz de Usuario.
- Trabajar activamente en la etapa de pruebas validando la solución.
- Participar en la creación de scripts de prueba para la PAU.
- Recopilar los comentarios de los usuarios durante la PAU.
- Reportar los resultados obtenidos en las pruebas de usuarios.
- Levantar puntos de atención o riesgos que puedan aparecer en la revisión de la solución al Gestor Funcional.
- Realizar las pruebas de certificación y en caso de existir errores, reportarlos al Líder de Desarrollo, caso contrario, informar al mismo que se gestione el Paso a Producción.
- Participar en la determinación de realizar la Vuelta Atrás, en caso de ser necesario.
- Aprobar la comunicación de la puesta en producción de la iniciativa.
- Completar todas las documentaciones relacionadas con temas funcionales (manuales, entregables) pertinentes requeridas por la metodología.

PMO. Es la responsable de administrar, organizar y coordinar las planificaciones de proyectos y recursos en el Project Server. Para ello debe:

- Implementar solicitudes a cartas gantts, por ejemplo, ingresos de gantts nuevas, control de cambio clientes, congelar gantts y proceso de cierre de gantts finalizadas con ficha de cierre, entre otros.
- Implementar solicitudes a recursos, dentro de las cuales se encuentran Creación/eliminación/actualización de recursos, habilitación de perfiles de acceso, reserva y mantención de HH para iniciativas de programas, entre otros.
- Generar informes y alertas.
- Entregar información de apoyo a otras áreas.
- Generar indicadores del Ciclo de Desarrollo para Tableros de Gestión y Seguimiento y Tablero de Incentivos.
- Realizar gestión de capacidades, como por ejemplo incorporación de especialidad de los RR por sistemas, incorporación de segmento que origina la demanda en las gantts, generación semanal del mapa de capacidades, entre otros.

Integrador funcional. Este es el responsable de realizar la evaluación del requerimiento (SUR y/o Requerimiento Comercial), para ello debe:

- Validar que el SUR y/o requerimiento comercial contenga toda la información necesaria.
- Generar Evaluación Funcional.
- Identificar funcionalidades y componentes afectados, tales como: canales, sistemas y procesos impactados, integraciones, reportaría u otras.

- Relacionar los requerimientos funcionales con los diferentes requerimientos que se estén desarrollando y/o que se vayan a desarrollar, con la finalidad de identificar sinergias/conflictos entre diferentes iniciativas.
- Coordinar las iniciativas para capitalizar las sinergias y/o resolver los conflictos.
- Especialista Gestión del Conocimiento
- Es el responsable de resguardar el conocimiento de los aplicativos, visto desde una mirada de aprendizaje y resguardar, organizar y coordinar el conocimiento en banco.

El Especialista GTEC (Gerencia de Tecnología) en la figura 11 muestra los distintos tipos de especialistas GTEC que pueden participar del proceso. Luego se describen las acciones que puede realizar cada uno.

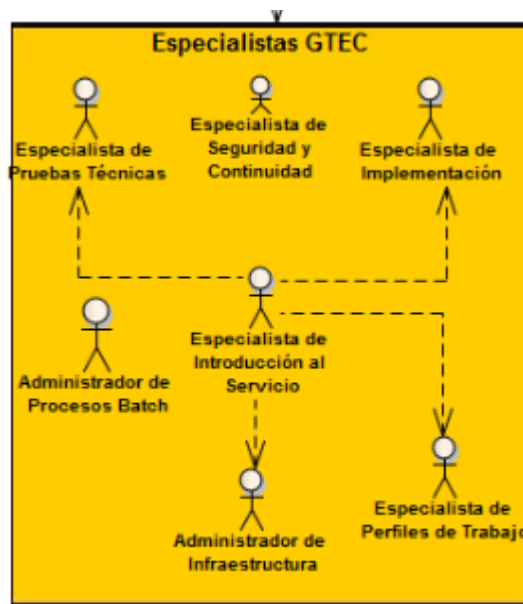


Figura 11. Tipos de especialistas GTEC.

Especialista pruebas técnicas. Este es el responsable de participar de todas las pruebas que se dan en la fase de Testing.

- Participar en la definición de requerimientos técnicos de cada prueba y sus casos de uso.
- Participar de la aprobación de los Documentos de Requerimientos Funcionales y definición de la Interfaz de Usuario.
- Entregar la aprobación de Testing sobre lo revisado de la iniciativa.
- Trabajar activamente en la etapa de pruebas validando la solución.
- Participar en la ejecución de scripts de prueba si es que existen.
- Recopilar los comentarios de los distintos actores de Testing.
- Reportar los resultados obtenidos en las pruebas técnicas.
- Levantar puntos de atención o riesgos que puedan aparecer en la revisión.

- Realizar las pruebas de certificación y en caso de existir errores, reportarlos al Líder de Desarrollo, caso contrario, informar para que continúe a las siguientes fases.
- Participar en la determinación de realizar la Vuelta Atrás, en caso de ser necesario.
- Aprobar la comunicación de la puesta en producción de la iniciativa.
- Completar todas las documentaciones relacionadas con temas funcionales (manuales, entregables) pertinentes requeridas por la metodología.

Especialista de seguridad y continuidad. Es responsable de velar por la integridad, disponibilidad y confiabilidad de la información, gestionar la implementación de estándares de seguridad en base a distintos procedimientos, asegurando que las distintas plataformas se adhieran a las normativas vigentes. Como puntos específicos del rol se considera lo siguiente:

- Generar y actualizar procedimientos en base a políticas del banco y estándares de seguridad.
- Informar resultados de pruebas de Vulnerabilidad.
- Aprobar acciones correctivas de Vulnerabilidades.
- Gestionar Consultas con respecto a la mitigación de observaciones.
- Administrar y Gestionar las plataformas de Seguridad.
- Control de Seguridad Externa.
- Control de Seguridad red Interna.
- Control de Seguridad Datos Corporativos.

Especialista de implementación del testing. Este es el responsable de coordinar las actividades relativas a la GTEC. Para ello debe:

- Gestionar todas las actividades del proceso de paso a producción por ambientes testing hacia producción.
- Responsable de gestionar el cumplimiento del Plan de Pruebas (testing).
- Gestionar y participar del flujo de control de ejecución de las Pruebas técnicas (testing).

Especialista de introducción al servicio. Este es el responsable de coordinar las actividades relativas a la GTEC. Para ello debe:

- Gestionar todas las actividades del proceso de paso a producción por ambientes previos.
- Responsable de gestionar el cumplimiento del Plan de Pruebas, constituido por los tipos de pruebas de ensamble, performance y seguridad. También se responsabiliza de la creación del Informe de Pruebas de Performance.
- Presentar, asesorar, y coordinar con las iniciativas, la elaboración, revisión y aprobación del Documento de Condiciones de Operación.
- Asegurar el cumplimiento y la aprobación de las condiciones de operación antes del paso a producción. Esto es dar conformidad a los requerimientos y capacidades necesarias para operar la solución.

- Desarrollar un Plan de Introducción al Servicio y la Declaración de Impacto de Soporte.
- Orientar o direccionar al área correspondiente las dudas de las iniciativas acerca de los estándares y normas vigentes en la GTEC que están asociadas a las Aplicaciones y/o Tecnologías involucradas en la solución a desplegar
- Gestionar y participar del flujo de control de ejecución de las Pruebas de Seguridad.
- Dar Capacitación a las iniciativas respecto de la herramienta GDI (Gestión de iniciativas).

Especialista de perfiles de trabajo. Es responsable por la aplicación de las normas, estándares y reglas de Asignación de Perfiles a los Usuarios de toda aplicación, nueva o en mantención. En el transcurso del desarrollo de la Iniciativa, trabaja para asegurar el alineamiento de estándares y normas de los perfiles de trabajo. Una vez que la Iniciativa está en Certificación para pasar a Producción, inicia las validaciones finales de las Cuentas de Acceso. Su participación temprana en el desarrollo de las Iniciativas, permite asegurar la detección de posibles mejoras de asignación de perfiles de trabajo antes de que la aplicación se libere a Producción. Sus responsabilidades y tareas son las siguientes:

- Asegurar la definición de estándares y normas de perfilamiento en las iniciativas a desarrollar.
- Participar en la revisión de la definición de requerimientos No Funcionales, subclasificados como de Perfilamiento.
- Participar en la revisión del análisis del nuevo proceso (etapa de análisis), con respecto a la asignación de Roles.
- Participar en la revisión de la definición de Perfiles de trabajo en la etapa de diseño.
- Supervisar y validar las soluciones de perfilamiento de las aplicaciones instaladas en ambiente de Producción.
- Asegurar la correcta asignación de perfiles de trabajo a los usuarios identificados en la aplicación en producción, acorde con su rol organizacional, para garantizar la seguridad de acceso y manipulación de la información.

Administrador de procesos batch. Este es el responsable de controlar, gestionar, monitorear y brindar soporte a la Producción de los Procesos batch. Además, debe:

- Brindar soporte a la Producción necesaria para la correcta continuidad de los procesos batch del Banco Estado.
- Asistir a los especialistas en evaluar el impacto de las nuevas implementaciones de mallas de procesos batch en la organización.
- Garantizar el monitoreo y disponibilidad de todos los procesos batch puestos en Producción.
- Autorizar y validar nuevos pasos de procesos batch a producción evaluando posibles impactos que puedan ocasionar problemas en el Banco.
- Explotar los ambientes de producción que contengan o aporten al procesamiento batch.

Administrador de Infraestructura. Es el responsable de controlar, gestionar y brindar soporte al ambiente que le corresponda: prueba o producción.

- Brindar soporte de infraestructura necesaria para el desarrollo de iniciativas.
- Dar lineamientos y estándares referentes al soporte para el hardware, software y disponibilidad para el desarrollo de aplicaciones.
- Asistir a los especialistas en evaluar el impacto de las nuevas soluciones en la organización.
- Ayudar en la implementación de nuevas soluciones para la organización y para la puesta en producción de la solución.
- Garantizar el soporte para el hardware, software y disponibilidad de acuerdo a las normativas y especificaciones existentes en el Banco.
- Asegurar y gestionar la instalación y mantenimiento de servidores, software estándar y aplicativo aprobado por los arquitectos.
- Gestionar que los entornos (producción y pruebas) se encuentren listos en tiempo y forma para cumplir los objetivos del proyecto. Realizar mantenciones a los entornos (producción y pruebas).

a) Especialista funcional

Especialista funcional. Es el responsable de participar del desarrollo de la iniciativa, dando soporte y definiciones para superar las distintas etapas del proceso. Además, debe:

- Participar en la definición de requerimientos funcionales y casos de uso.
- Participar de la aprobación de los Documentos de Requerimientos Funcionales y definición de la Interfaz de Usuario.
- Obtener la aprobación de los usuarios de la definición de la Interfaz de Usuario.
- Trabajar activamente en la etapa de pruebas validando la solución.
- Participar en la creación de scripts de prueba para la PAU.
- Recopilar los comentarios de los usuarios durante la PAU.
- Reportar los resultados obtenidos en las pruebas de usuarios.
- Levantar puntos de atención o riesgos que puedan aparecer en la revisión de la solución al Gestor Funcional.
- Realizar las pruebas de certificación y en caso de existir errores, reportarlos al Líder de Desarrollo, caso contrario, informar al mismo que se gestione el Paso a Producción.
- Participar en la determinación de realizar la Vuelta Atrás, en caso de ser necesario.
- Aprobar la comunicación de la puesta en producción de la iniciativa.
- Completar todas las documentaciones relacionadas con temas funcionales (manuales, entregables) pertinentes requeridas por la metodología.

Anexo C: Descripción de los principales dolores detectados

El listado de los 30 dolores más importantes es el siguiente:

N°	dolores/insatisfacciones	Descripción
1	falta clarificar alcance de los cambios	los usuarios de operaciones que entregan o definen el requerimiento a pedir, no tienen la total claridad al generar el alcance de la solicitud hacia la gerencia de sistemas
2	artefactos (documentos) de poco valor para certificar y evidenciar pruebas entre ambientes	la metodología de trabajo del Banco Estado pide documentar los pasos que se utilizaron para certificar y/o probar, pero en muchos de los documentos existentes no se entrega una descripción real de lo realizado, haciendo que sean documentos sin valor a la realidad actual de la tecnología que certifican.
3	excesivo papeleo a registrar	se debe generar mucha documentación para poder pasar entre ambientes, y dejar claridad de lo realizado
4	entregas poco prolijas por el equipo de desarrolladores informáticos	los desarrolladores no siempre cumplen con pruebas unitarias y de borde de gran valor; lo que implica entregas no correctas y afecta los SLA y provoca aumento en los ciclos de prueba.
5	errores colaterales post paso producción	existen muchos errores colaterales, es decir, fallos en partes de los sistemas que no deberían verse afectadas por el cambio a realizar.
6	proceso compra muy extenso en tiempos y acciones a seguir	el proceso de compras no es eficiente en costos, y el tiempo de entrega de la inversión es muy largo.
7	facilidad para cambiar prioridades de entregables y/o proyectos	muchos desarrollos se realizan para cumplir las necesidades del cliente, pero sin previo aviso entran nuevos proyectos con máxima prioridad, lo cual afecta los tiempos comprometidos de entrega al cliente.
8	ambientes no homologados	los ambientes de desarrollo, testing, qa y producción no presentan las mismas características, produciendo diferencias que generan errores o retrasos en las pruebas y futura instalación.
9	cambios en los ambientes sin aviso	se ha detectado que hay cambios en los ambientes no informados a ninguna de las partes, los que provocan errores o demora en el diagnóstico de los problemas.
10	competencias de los equipos de bancos	se ha detectado falta de conocimiento o habilidades por parte de los equipos e instancia que se está trabajando para lograr un correcto paso a producción, lo que no ayuda a diagnosticar en muchas ocasiones tiempos y adecuadamente de solución a los problemas /sobre dependencias de las fábricas de desarrollo, testeo, arquitecturas, etc.)

12	data center lentos en aplicar los cambios	a pesar de que se programa con anticipación la ejecución de un paso a producción, el cambio definitivo no se ejecuta inmediatamente y pueden pasar varios días; dejando el data center de contingencia asumiendo el cambio
12	site de contingencia inestables	cuando ocurren problemas e Banco, el site de contingencia no proporciona la real continuidad esperada; retrasando los tiempos de los clientes finales
13	diagnósticos poco eficientes ante Fallas de infraestructura/arquitectura banco	los diagnósticos post caídas o problemas en la red Banco, son poco eficientes y no siempre corresponden a la causa real, o no contienen los detalles necesarios para que el equipo de desarrollo pueda abordarlos.
14	tiempos excesivos para certificar por ambiente	los ambientes que certifican los desarrollos (test y QA) poseen SLA muy elevados y los actores que participan en ellos son poco empáticos en sus certificaciones, ya que no siempre entregan su revisión antes del cumplimiento del SLA; pudiendo haber ganado tiempos en el caso de mejoras pequeñas
15	infraestructura obsoleta para los tiempos actuales	no se dispone de suficientes recursos de infraestructura para construir ambientes de desarrollo o testing que satisfagan las necesidades del negocio.
16	no contar con herramientas de control y calidad para el testeo y monitoreo	los testeos y controles poseen mucha manualidad, y se debiese contar con tecnologías más acorde a los nuevos tiempos y tecnologías existentes.
17	monitoreo inadecuado o marcha blanca poco eficiente en los ambientes productivos	muchas fallas se detectan ante reclamos de clientes, en vez de ser detectadas en etapas tempranas por herramientas de monitoreo, para permitir tomar medidas oportunas.
18	mesa de ayuda informática poco eficiente	en caso de problemas en los ambientes de trabajo de los desarrolladores, las mesas de ayuda informática no cumplen en definir alcance del problema y los tiempos de solución
19	poco dominio de la arquitectura Banco que soporta los aplicativos	se percibe que tanto el equipo de sistemas como de tecnología no dominan sus ámbitos; y esto provoca asesorías no 100% correctas y no cumplen los tiempos esperados para los clientes
20	tiempos excesivos cuando se debe aplicar nuevos cambios de infraestructura	obtener recursos de infraestructura puede tomar varias semanas o meses
21	eficiencia de los desarrollos entregados	se espera que los cambios entregados sean relevantes para las necesidades que se pidieron cumplir en forma eficiente, y en los tiempos correctos de entrega.
22	muchas mesas de trabajo para un paso a producción	para llegar a producción se debe pasar por muchas mesas de trabajo y/o recursos banco, lo que hace poco eficiente el tiempo de solución para lo construido para el cliente final

23	muchos silos de trabajo entre ambientes	existen muchas islas de trabajo, provocando pérdida importante del ciclo de vida y fluidez del mismo para poner una pieza o desarrollo en producción.
24	abastecimiento de data proveniente de varias fuentes en los sistemas core del banco	el uso de la información de los sistemas core del banco, provienen de muchas fuentes de datos (BDs).
25	repositorio de información de los aplicativos poco eficiente, o casi inexistentes	no existe un repositorio o una única fuente confiable para recurrir a buscar historia de problemas, logs y estado de cada aplicación
26	sobre dependencia de plataformas departamentales banco estilo frameware	si hay problemas en la plataforma web principal banco, la gran mayoría de los aplicativos webs colapsan
27	planificaciones muy rígidas	planificaciones muy rígidas, sin posibilidad de cambios en etapas posteriores al análisis
28	nuevos estándares de calidad en seguridad informática, sin saber mucho lo que implica	se implementan reglas de seguridad que implican tiempos muy extensos de solución, por el poco manejo de ellas
29	creación de nuevas áreas o subgerencias lo que provoca nuevos flujos al ciclo actual	nuevas áreas de calidad o tecnologías, provocando nuevos tiempos de entrega; lo cual no empatiza con los tiempos ya planificados
30	la actual metodología banco es cuello de botella para establecer fechas de entrega de cara al cliente	la actual metodología entrega establece periodo muy cortos o muy largos de programación, lo que implica regirse por plazos ya existentes sin valor (siempre se dice que va a cambiar y no cambian la metodología)

Tabla 11. dolores e insatisfacciones detectadas del proceso desarrollo actual.