

## Tabla de contenido

<b>Definición del Proceso</b> .....	5
Entender el proceso : .....	5
Involucrados y áreas afectadas:.....	5
<b>Narrativa del proceso</b> .....	7
<b>Análisis de Brecha (Gap Analysis)</b> .....	8
<b>Fase de diagnostico</b> .....	9
Problemática .....	9
Proceso actual .....	9
<b>Estrategias recomendadas</b> .....	12
Proceso de Integración Continua.....	13
Proceso de Entrega continua .....	14
<b>Conclusión</b> .....	17

## Introducción

La tecnología actualmente es uno de los pilares que sostiene la economía mundial, a tal punto de incidir en la calidad o competencia que una empresa le puede brindar a sus clientes. ¿Crearía usted, por ejemplo, en un banco que no posea una banca electrónica? Seguramente su respuesta se incline hacia un “No” rotundo.

Curiosamente, la innovación tecnológica libra una poderosa batalla contra la necesidad de cambiar o no un aplicativo. Instituciones como bancos, aerolíneas, o incluso entidades gubernamentales, se escudan mucho de su estabilidad para promover confianza a sus clientes, lo que tiende a frenar la innovación tecnológica que acarrea un riesgo inherente a hacer las cosas de forma diferente. La estabilidad, en muchas ocasiones, lleva consigo muchísima burocracia, procesos tediosos y mucha documentación, sin mencionar otros elementos que retrasan el proceso procurando mantener esa estabilidad buscada por los clientes.

El fruto de todo lo anterior en una institución donde los cambios abruptos y la innovación no es tan bien vista son procesos rigurosos, largos y con múltiples revisiones para garantizar que no se cometan errores que afecten la imagen o credibilidad de esta institución. Lo que en organizaciones o en empresas chicas pudiera significar cambios de una o dos semanas, se traduce en uno o dos meses de trabajo.

El reto del profesional de ingeniería de software con ansias de innovar es luchar en muchas ocasiones con la célebre frase de estas instituciones que reza “si funciona, no lo toques, ni para mejorarlo.”

Para el presente proyecto final, hemos tomado una reconocida institución bancaria de índole nacional que, por su gran trayectoria, estabilidad y confianza, propician los elementos y el contexto necesario para una evaluación y mejora de los procesos de ingeniería de software. Hemos llamado esta organización:

Banco particular   
sus buenos pasieros



Banco particular cuenta con alrededor de 5500 colaboradores (clientes internos) y alrededor de 400 mil clientes externos de los servicios como su banca virtual móvil y de escritorio.

Por la estabilidad que brinda a sus clientes y la confianza, dedica muchos recursos a soporte, solución de incidentes de core bancario, y banca en línea; sin embargo, pocos recursos para innovación. La tarea de sacar un cambio a producción es un tanto tediosa, tanto que muchas veces la corrección a nivel de código se realiza en

menos de 20 minutos y el cambio puede demorar en salir hasta dos (2) días después, entre documentaciones, aprobaciones y pruebas de calidad.

Es precisamente por lo antes planteado que nace la motivación en los autores para llevar el alcance del estudio a la mejora en el proceso de resolución de incidentes para clientes internos y externos que utilizan los sistemas informáticos de la institución bancaria.

## Definición del Proceso

### Entender el proceso :

El proceso a evaluar y mejorar estará enfocado en la liberación de cambios a producción luego de que se presenta alguna incidencia en el aplicativo web de banca electrónica. Este proceso se puede dar de forma proactiva (a través de pruebas de calidad que detecten la falla) o reactiva (identificando la falla en un ambiente productivo).

Para entendimiento del lector, centraremos el análisis de este proceso desde el punto de vista del equipo de desarrollo, quienes son los encargados de identificar, corregir o crear nuevos componentes de software para la aplicación mencionada anteriormente.

El proceso estará dentro del contexto de una aplicación que todavía se encuentra en fase de estabilización, lo que conlleva un volumen de incidentes bastante elevado y un equipo que día a día se encuentra tratando de cerrarlos.

Se ha establecido una categorización de los incidentes para definir la prioridad de atención; sin embargo, por alcance de este proyecto, no ahondaremos a profundidad en esta categorización.

### Involucrados y áreas afectadas:

De forma resumida, cuando el equipo de desarrollo y mantenimiento del sistema de banca electrónica desea liberar un cambio para corregir alguna incidencia detectada en el ambiente productivo, las personas que se ven involucradas en este proceso son las siguientes:

Las áreas que se ven afectadas por esto, son:

- **Equipo de desarrollo:** Equipo encargado de la actualización, mejora y corrección del aplicativo.
  - Trabajan sobre un servidor de desarrollo que tiene una réplica de la aplicación que se encuentra en producción.
  - Gestionan su trabajo con una aplicación de registro y gestión de incidentes.
  - Son liderados por un jefe de desarrollo e interactúan con el área de calidad (QA), liberaciones y liberaciones en producción para la atención de incidentes.
- **Jefe de desarrollo:** Encargado de gestionar al equipo de desarrollo y de priorizar sus esfuerzos.
  - Tiene la potestad de aprobar cambios del equipo de desarrollo que son registrados a través de una herramienta de manejo de incidentes.
  - Interactúa con el equipo de liberaciones enviándole la aprobación de los cambios realizados por el equipo de desarrollo.

- **Equipo de liberaciones:** Equipo encargado de liberar las actualizaciones del equipo de desarrollo en el ambiente de calidad; realizan esta tarea luego de recibir la aprobación del jefe de área.
  - Revisan que la solicitud (ticket de liberación) cuenta con la información requerida y velan por la integridad de las liberaciones, asegurando el estado final del ambiente en el cual se liberan.
  - Envían la solicitud de liberación a producción de componentes que ya han pasado por la verificación de calidad necesaria y obtuvieron resultados exitosos.
  - Interactúan con el equipo de desarrollo, el equipo de QA y el equipo de liberaciones en producción.
- **Equipo de calidad (QA):** Son los encargados de velar por la integridad de la aplicación que se encuentra en producción.
  - Se aseguran de que los cambios o mejoras al aplicativo cumplan con las expectativas del negocio, las especificaciones de software y los criterios de calidad que deben representar a una institución de alto renombre, salvaguardando su imagen y los activos del cliente.
  - Interactúan con el área de liberaciones para liberar nuevos componentes en el ambiente de calidad.
  - Gestionan el estado de los incidentes para que el resto del equipo pueda llevar trazabilidad del estado de los cambios.
  - Pueden devolver algún cambio al equipo de desarrollo en caso de que no se cumplan los criterios de aceptación esperados.
- **Equipo de liberaciones en producción:** Pasan los cambios ya aprobados y verificados por el equipo de calidad y el jefe de liberaciones en producción al ambiente productivo.
  - Velan por la estabilidad del ambiente productivo y revisan que cada cambio a liberarse tenga los controles y documentación necesaria para poder pasar a ser utilizada por los clientes.
- **Jefe de liberaciones en producción:** Responsable del equipo de liberaciones en producción.
  - Responsable de la aprobación a cambios de carácter sensitivo en el ambiente productivo.
  - Vela por la estabilidad e integridad del aplicativo en producción.

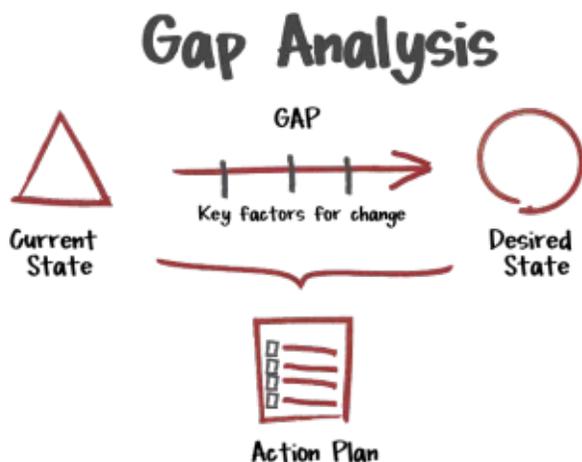
## **Narrativa del proceso**

Tal como se encuentra hoy en día el proceso, este se puede explicar de la siguiente forma:

- El equipo de desarrollo desea liberar un cambio en producción.
- Cuando la solución a liberar está lista y verificada en el ambiente de desarrollo, el programador crea un ticket en el cual solicita al equipo de liberaciones que pase un componente al ambiente de calidad.
  - En este ticket se indica que se desea pasar un control de cambio por incidencia a su ambiente para garantizar que el cambio se haya hecho de la forma correcta.
  - Dentro de esta solicitud debe ir la documentación que explica el cambio y, de ser necesario, un plan de reversa por si falla la liberación.
- El programador envía una solicitud de aprobación al jefe de desarrollo para que apruebe el cambio a realizar.
- El equipo de liberaciones recibe la aprobación del cambio y la notificación del ticket generado por el equipo de desarrollo.
  - El equipo de liberaciones debe revisar si toda la documentación necesaria está colocada en el ticket abierto previamente.
  - Si se cumple con la aprobación y los criterios de aceptación del equipo de liberaciones, se procede a pasar al ambiente de calidad (QA).
  - Si no se cumple con la aprobación y/o los criterios de aceptación anteriormente mencionados, se devuelve el ticket y se solicita al equipo de desarrollo que haga las correcciones pertinentes.
- El equipo de calidad (QA) recibe una notificación de que nuevos componentes están disponibles para ser liberados en su ambiente.
- El equipo de QA debe abrir un ticket solicitando a un recurso del equipo de liberaciones para que compile y despliegue los cambios en el ambiente de calidad (QA).
- El equipo de calidad (QA) empieza a validar la funcionalidad o el componente de software para verificar que todo esté funcionando correctamente.
  - De funcionar adecuadamente, el equipo de calidad (QA) abre un ticket indicando al equipo de liberaciones que la corrección ha sido validada exitosamente y se desean liberar los cambios en producción.
  - De lo contrario, se devuelve el ticket al equipo de desarrollo para que hagan las correcciones necesarias.
- El equipo de liberaciones entonces abre otro ticket solicitando al equipo de liberaciones en producción la liberación de las correcciones.
- Cuando el equipo de liberaciones en producción recibe la solicitud de liberación a producción, y el jefe de área de la autorización para que este pase, proceden a liberar los cambios.
- El cliente final, el usuario y el equipo del proyecto verifican que todo esté en orden en producción.
- Finaliza el proceso de liberación.

## Análisis de Brecha (Gap Analysis)

Usando como guía en un alto nivel de abstracción el modelo IDEAL.



Partiendo desde un punto avanzado ya que el departamento de tecnología ha solicitado la mejora y ya se tiene el patrocinio aprobado y el conocimiento del negocio de la necesidad de esta mejora. ✓

**¿Dónde estamos?** El proceso se encuentra en un estado de inactividad, el mismo tiene más de 7 años de madures, funciona así y la frase que se utiliza a menudo es, “si funciona no lo toques”. Este proceso cuenta con áreas las cuales están formadas por tan solo 2 o 3 personas, dichas áreas se fueron creadas a medida que el proceso tenía inconvenientes, solamente para satisfacer una necesidad en dicho momento y se quedado así por años, por el simple hecho de no re plantear el proceso para mejorarlo.



**¿Hacia dónde queremos ir?** Se desea mejorar los tiempos de liberación de cambios a producción, automatizando tareas que hoy en día se hacen de forma manual y que según los trabajadores son tediosas, les quitan tiempo para resolver más incidentes y ser más productivos. Logrando esto se podría dar una respuesta rápida y acertiva a los clientes internos y externos, usuarios de los sistemas informáticos del [Banco Particular](#).

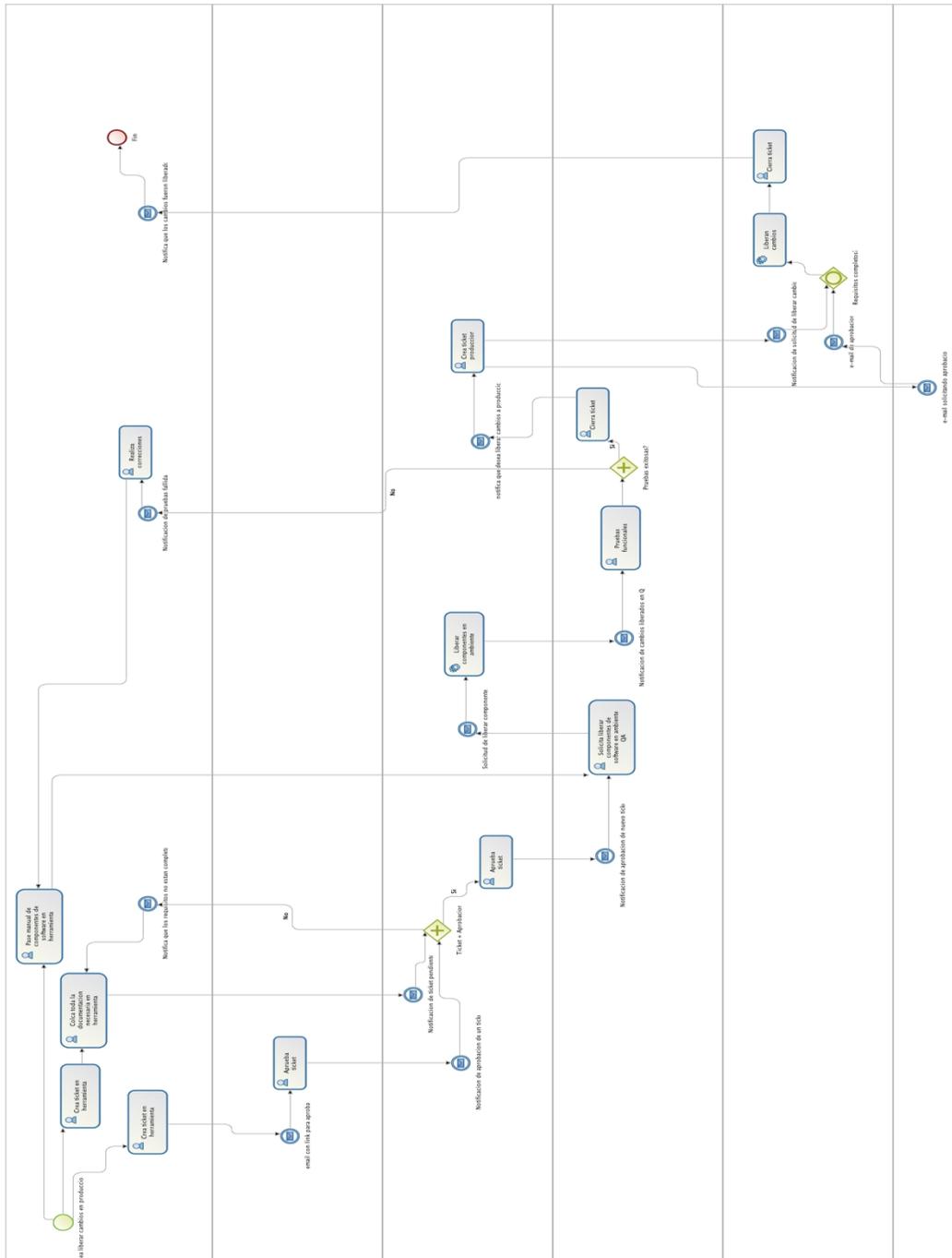
# Fase de diagnostico

## Problemática

Tiempos de espera altos para pasar correcciones de incidencias a producción que se deslinda en una lenta respuesta hacia los clientes internos o externos.

## Proceso actual

(GAP “Donde estamos”)



**GAP** cuadro de evaluación (Evaluación real completada por uno de los autores el cual interactúa día a día con el proceso).

Proceso	Calificaciones del proceso				Mejores prácticas	Cumple?
	Deficiente	Regular	Bueno	Excelente		
Liberaciones de desarrollo a pruebas		✓			Automatizadas y controladas por una herramienta moderna.	✗
Liberaciones de pruebas a producción	✓				Automatizadas y controladas por una herramienta moderna.	✗
Crear meta data de la liberación a producción		✓			La necesaria y que siga el flujo desde desarrollo hasta producción como parte de los adjuntos al código.	✗
Comunicación entre las áreas implicadas para la liberación		✓			Activa y mediante medios modernos de integración continua, disminuir el uso de emails.	✗
Controles para evitar errores en producción	✓				Revisiones de código por personal al que el programador reporta, que queden entregables y pruebas de revisiones.	✗
Pruebas en ambiente QA		✓			Flujos críticos de los servicios que brinda la empresa automatizados y manualmente	✗

					realizar los flujos específicos que la liberación requiera en ambientes lo más parecido a producción.	
--	--	--	--	--	---	--

GAP Cuadro de problemática detallada

Procesos identificados	
Proceso actual 😞	Proceso deseado 😎
El programador abre dos tickets uno para liberar en pruebas y otro para pedir la aprobación del jefe del área.	El programador solo debe abrir un ticket, con la documentación estrictamente necesaria y no perder tiempo en tareas tediosas para dedicar
El jefe del área debe estar pendiente a su correo porque es allí donde llega la notificación de un link diciendo que tiene algo que aprobar	Debe haber una herramienta la cual tenga todo centralizado y tenga acceso desde el móvil y desde su estación de trabajo para aprobar con un solo clic y reducir la cantidad de correos innecesarios y usar solo el serve mail para cosas realmente necesarias.
Si la documentación esta correcta el área de liberaciones cierra el ticket para que el equipo de QA, abra otro ticket solicitando la liberación de componentes en su ambiente.	Esto no debe suceder, debe haber un proceso de liberación automático en el ambiente en el momento que se aprueba el ticket de desarrollo, no se deben abrir ticket para cada tarea.
Las personas del área de liberaciones deben liberar manualmente los componentes en el ambiente de QA y luego cerrar el ticket.	La liberación debe ser automática no manual, es pérdida de tiempo productivo.
En el momento que las pruebas fueron culminadas, el encargado de pruebas debe redactar un correo para enviarlo al personal de liberaciones para confirmar que las pruebas terminaron.	En el momento que el equipo de QA termina de realizar las pruebas solo debe bastar con cambiar el estado del ticket, no redactar más correos.
El personal de liberaciones debe crear un ticket más solicitando la liberación a producción de los componentes.	Esto no debería ser, se debe tener un único flujo de un ticket que va desde desarrollo a pruebas y no varios relacionados entre sí que al momento de buscar el flujo que siguió es tedioso.

Se debe solicitar la aprobación por correo a un vicepresidente para liberar en producción todos los días.	Debería poder aprobar la liberación en producción la persona que está encargada de administrar el ambiente productivo que está más cerca de lo que sucede todos los días, los riesgos, etc. No una persona que tiene un rol administrativo alto, el cual debe invertir tiempo en tareas gerenciales, no aprobar cambios técnicos.
---	---

## **Estrategias recomendadas**

Para agilizar los procesos de entrega de productos hacia producción o a la mano del usuario final, se recomienda utilizar una colección de procesos de desarrollo de software que garanticen la entrega rápida, sostenida y segura de cambios (ya sea de nuevas funcionalidades, de configuración, arreglo de fallas o experimentos en producción. Esta propuesta se enfoca en dos grandes grupos de implementación y se dividirán de la siguiente manera:

- Integración continua
- Entrega continua

Por el momento no estaremos indagando en el proceso de despliegue continuo ya que este proceso no necesita ser mejorado. El área de liberación a producción todos los días procede con liberaciones durante su flujo actual. Consideramos que no es necesario idear un plan de acción para agilizar sus procesos hasta que estos sean el cuello de botella de un engranaje agilizado realmente. Por el momento no vislumbramos mayores cambios en esta parte del proceso.

Durante la fase de integración continua, se propone que todo aquel desarrollo nuevo que se desee implementar esté registrado con una herramienta de control de versiones. En el caso del Banco particular, ya se cuenta con una herramienta denominada Gitlab, por lo que procederemos a apoyarnos, en las estrategias de integración continua, en las bondades que ofrece esta herramienta.

Para mayor control de lo que ocurre en los departamentos de desarrollo y pruebas, se estará utilizando la herramienta Jira. Jira es una herramienta de software cuyo propósito es tener una visión integral del tiempo que toma darle solución a un problema; también puede ser utilizada para proponer nuevas funcionalidades a través de historias de usuario y dimensionar los esfuerzos del equipo técnico que desarrolla una solución de software.

El desarrollador antes de iniciar su proceso de resolución del problema, debe generar un ticket en esta aplicación para justificar el tiempo que estará invirtiendo en el mismo.

## Proceso de Integración Continua

Definiremos integración continua como la práctica de implementar cambios en la rama de código productivo, probar los cambios lo más pronto posible y finalmente integrar el código nuevo con el código productivo todos los días.

Para salvaguardar la calidad haciendo uso de este proceso se requiere el uso de dos prácticas de la integración continua que son las siguientes:

- Construcción continua
- Automatización de las pruebas

Para la **construcción continua** es necesario la construcción del artefacto final, definimos artefacto final como producto tangible que resulta del proceso de desarrollo de software, el mismo puede ser código, resultado de compilación del código o cambio en configuración que será puesto producción tan pronto como el cambio sea realizado. Es decir, una vez un desarrollador termine su cambio y proceda a registrarlo en el control de versiones. Este cambio propuesto es construido y se encuentra listo para ser probado manualmente o automáticamente.

Para la **automatización de las pruebas**, es importante que cada desarrollador inicie con sus pruebas unitarias apoyándose en aplicaciones como [Cucumber.io](https://cucumber.io/). Que facilitan la creación de pruebas integrales, pruebas unitarias y la ejecución de las mismas, con una sintaxis que hasta un usuario final puede leer y de esta manera, ver que su criterio de aceptación de la solución está conforme a lo que solicitó. Este tipo de aplicación es ideal para la creación de pruebas funcionales que serán leídas posteriormente por el personal de Control de Calidad, lo que ayuda a reducir tiempos de análisis pruebas y asegurar la calidad del entregable.

La ejecución del proceso de integración continua tiene dos grandes fases, la primera de ella es **asegurarse que el código compila**, y la segunda fase es **asegurarse que el código funciona tal cual como fue diseñado**.

El proceso de integración continua se apoya tanto en construcción continua con pruebas automatizadas para garantizar que cada construcción de código tome en cuenta la calidad del mismo y el equipo de desarrollo pueda verificar código roto prematuramente, reduciendo considerablemente los tiempos entre la liberación del código al ambiente de calidad (QA) y la llegada de este componente de software a un ambiente productivo.

Para el caso del Banco particular, una vez el desarrollador tenga su cambio listo para ser llevado al área de pruebas, el desarrollador cambiara el estado de la tarea en Jira, de “En Proceso” a “Listo para desplegar”. Al hacer este cambio, automáticamente la herramienta le enviara una notificación al Superior para adjuntar su aprobación. Una vez el superior de su aprobación, la tarea cambiará de estado de “Listo para desplegar” a “En pruebas” donde el personal de Control de Calidad asumirá el control.

Con esta parte de la estrategia y la implementación de las herramientas sugeridas, se busca:

- Automatizar la notificación a los líderes que aprueban cambios
- Reducir tiempos de prueba garantizando un menor tiempo en el mercado.
- Mejorar la calidad de los componentes a liberar incluyendo pruebas automatizadas y prematuras de los componentes de software.
- Dar mayor trazabilidad a las mejoras y al proyecto en general.

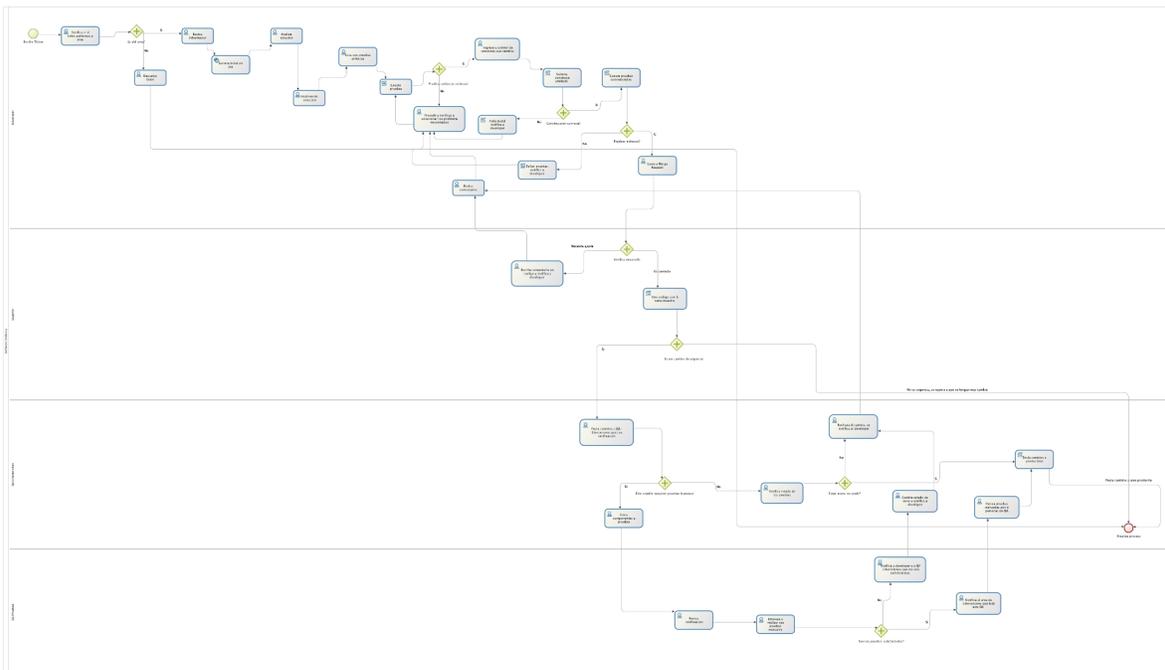


Diagrama de proceso una vez culmine la fase 1 del proyecto

## Proceso de Entrega continua

Definiremos entrega continua como la representación filosófica y compromiso del código proporcionado por el desarrollador se encuentra compilando y de acuerdo a las necesidades del usuario final. Lo anterior se logra a través del trabajo realizado en la fase 1 de implementación que indica que los desarrolladores deben estar generando sus pruebas unitarias lo más automatizado posible.

Adicional a este punto, el Banco Particular tiene una necesidad y es que la documentación que tiene actualmente no se pierda y requiere que se siga generando aún con los nuevos cambios. Por lo que se estará agregando un proceso extra y un control al proceso actual que tienen los desarrolladores.

Este nuevo proceso se trata de adjuntar la documentación del cambio al “Merge Request”. Definimos “Merge Request” como el mecanismo que permite compartir los cambios realizados bajo el archivo o los archivos que desean colocar en un entorno productivo. Durante este flujo, se requiere la creación de un nuevo proceso automatizado que verifique que el documento este adjunto, si esta adjunto la construcción del artefacto se dará de la forma correcta, si no está adjunta entonces fallará.

Para evitar que los desarrolladores suban cualquier tipo de documento, el departamento de control de calidad, será el que salvaguarda la calidad del mismo. Una vez reciben un nuevo cambio su proceso deberá incluir la revisión de la documentación generada por el desarrollador.

Si el personal de control de calidad encuentra que la documentación no sigue los estándares establecidos por el Banco Particular, entonces procederá a notificar al desarrollador que no cumplió con los estándares y debe generar nuevamente la documentación. Si la documentación se encuentra y cumple con los estándares propuestos por el Banco Particular demanda entonces se continuará con el flujo de atención.

El flujo de atención conjunto con la fase 1 de implementación y la fase 2 de implementación quedará de la siguiente manera:

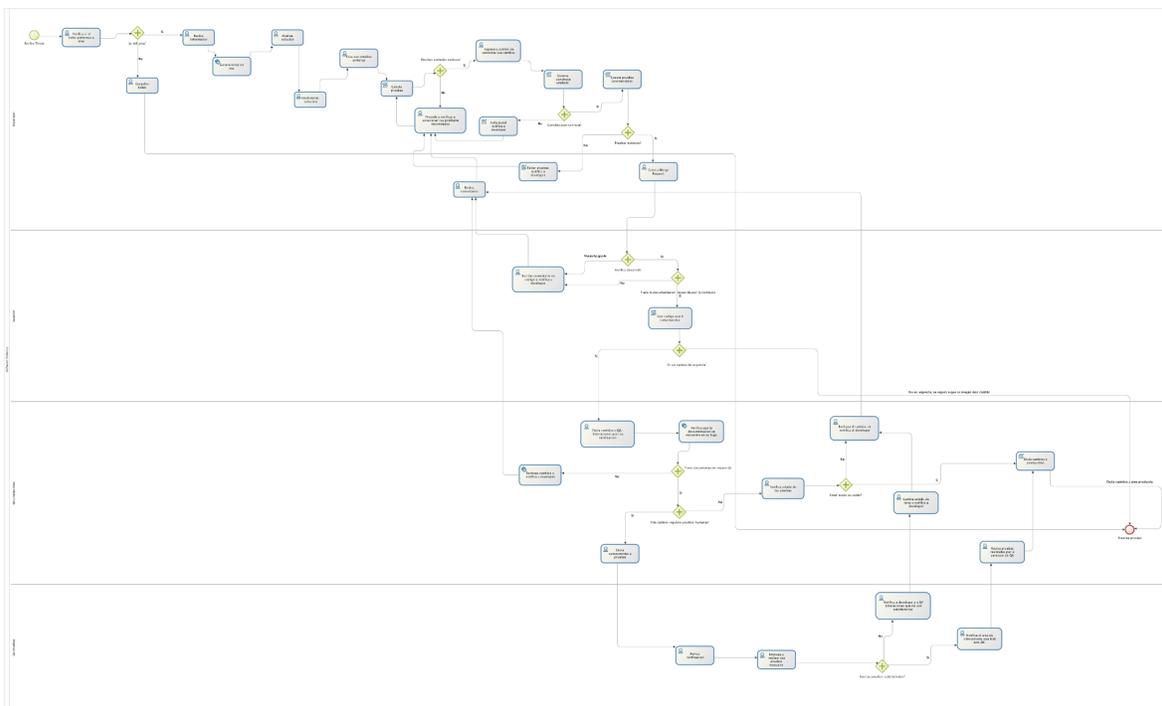


Diagrama de proceso completo que incluye los aspectos necesarios por la entidad.

Si el Banco Particular, logra adaptar estas dos propuestas de cambio en su proceso estamos seguros que su tiempo de entrega de cambios al ambiente productivo se mejoraran hasta un 50% del tiempo actual. Se reducirá el tiempo que es necesario

para la ejecución de las pruebas que pueden tardar hasta 2 días en la actualidad a 1 día, ya que si el desarrollador realiza sus pruebas la cantidad de bugs encontrados en el ambiente de pruebas se reducirá hasta un 60%. Y finalmente lograremos nuestro objetivo de agilizar la puesta en producción de los cambios generados por los desarrolladores.