Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Escuela Profesional de Ingeniería de Sistemas

Uso de herramientas de integración continua para reducir el tiempo de despliegue de procesos de Big Data en una entidad financiera

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniera de Sistemas

AUTOR

Ivette Pamela CHÁVEZ MARTINEZ

ASESOR

Norberto Ulises ROMÁN CONCHA

Lima, Perú

2021

Referencia bibliográfica

Chávez, I. (2021). *Uso de herramientas de integración continua para reducir el tiempo de despliegue de procesos de Big Data en una entidad financiera*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	IVETTE PAMELA CHÁVEZ MARTINEZ
Tipo de documento de identidad	DNI
Número de documento de identidad	47473167
URL de ORCID	https://orcid.org/0000-0003-0182-0935
Datos de asesor	
Nombres y apellidos	Norberto Ulises Román Concha
Tipo de documento de identidad	DNI
Número de documento de identidad	08510560
URL de ORCID	https://orcid.org/0000-0002-3302-7539
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Luzmila Elisa Pró Concepción
Tipo de documento	DNI
Número de documento de identidad	08862360
Miembro del jurado 1	
Nombres y apellidos	Pablo Jesús Romero Naupari
Tipo de documento	DNI
Número de documento de identidad	06182185
Datos de investigación	
Línea de investigación	Data Science
Grupo de investigación	ITDATA
Agencia de financiamiento	Financiamiento Propio
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima

	Jr. Carlos Amezaga No. 375 Universidad Nacional Mayor de San Marcos Latitud: -12.0564232 Longitud: -77.0843327
Año o rango de años en que se realizó la investigación	2021
URL de disciplinas OCDE	2.02.04 -- Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Sistemas

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 20:50 horas del día 13 de diciembre del año 2021, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por la Dra. Pró Concepción Luzmila Elisa (Presidente), Lic. Romero Naupari Pablo Jesús (Miembro) y el Lic. Román Concha Norberto Ulises (Miembro Asesor), usando la plataforma Meet (https://meet.google.com/gfv-qdyi-szt), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: ***"USO DE HERRAMIENTAS DE INTEGRACIÓN CONTINUA PARA REDUCIR EL TIEMPO DE DESPLIEGUE DE PROCESOS DE BIG DATA EN UNA ENTIDAD FINANCIERA",*** *por la Bachiller* ***Chávez Martinez Ivette Pamela;*** *para obtener el Título Profesional de Ingeniera de Sistemas.*

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, la Presidente invitó a la Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

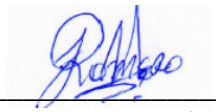
La Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, la Bachiller obtuvo la nota de ***18*** *(dieciocho)*

A continuación la Presidente de Jurados la Dra. Pró Concepción Luzmila Elisa, declara al Bachiller ***Ingeniera de Sistemas****.*

Siendo las 21:48 horas, se levantó la sesión.

_____
Presidente
Dra. Pró Concepción Luzmila Elisa



_____
Miembro
Lic. Romero Naupari Pablo Jesús

_____
Miembro Asesor
Lic. Román Concha Norberto Ulises

DEDICATORIA

A mis padres por todo el apoyo que me brindaron durante mi formación académica.

AGRADECIMIENTOS

A la entidad bancaria que me permitió adquirir conocimientos y desarrollarme profesionalmente.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

USO DE HERRAMIENTAS DE INTEGRACIÓN CONTINUA PARA REDUCIR EL TIEMPO DE DESPLIEGUE DE PROCESOS DE BIG DATA EN UNA ENTIDAD FINANCIERA

Autor: Chávez Martinez, Ivette Pamela
Asesor: Román Concha, Ulises

Título: Trabajo de Suficiencia Profesional
Fecha: Diciembre, 2021

RESUMEN

El presente Trabajo de Suficiencia Profesional (TSP) describe el uso de herramientas de integración continua en los despliegues de procesos de big data en una entidad financiera bajo el enfoque de la metodología Kanban. Debido a que la entidad financiera tiene una alta demanda de pases a producción se requiere que el flujo de despliegues sea adecuado y tome menos tiempo.

Ante esta problemática, la entidad bancaria propone que se usen herramientas de integración continua y un flujo basado en la metodología Kanban. Dentro de las herramientas propuestas están el uso de Bitbucket como repositorio, Git para el control de versiones y Jenkins para la ejecución de pipelines automatizados, así como el uso de un tablero Jira que cuente con las etapas del desarrollo sugeridas por la metodología Kanban.

Como resultado del uso de las herramientas mencionadas, se van a realizar despliegues en un menor tiempo permitiendo eficiencia operativa. Además, gracias al uso de pipelines automatizados se pueden reducir los errores en los pases a producción. También cabe mencionar que debido a que este nuevo flujo es sencillo de usar, el tiempo requerido para capacitar a nuevos integrantes es menor.

Palabras claves: Integración continua, Big Data, Entidad Financiera, Despliegue de Procesos, Kanban

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF SYSTEMS ENGINEERING AND INFORMATIC
PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING

USE OF CONTINUOUS INTEGRATION TOOLS TO REDUCE THE DEPLOYMENT TIME OF BIG DATA PROCESSES IN A FINANCIAL INSTITUTION

Author: Chávez Martinez, Ivette Pamela
Adviser: Román Concha, Ulises
Title: Professional sufficiency report
Date: December, 2021

ABSTRACT

This Professional Sufficiency Work (TSP) describes the use of continuous integration tools in the deployments of big data processes in a financial entity under the Kanban methodology approach. Due to the fact that the financial institution has a high demand of deployments, the flow of deployments is required to be adequate and take less time.

Faced with this problem, the bank proposes that continuous integration tools and a flow based on the Kanban methodology be used. Among the proposed tools are the use of Bitbucket as a repository, Git for version control and Jenkins for the execution of automated pipelines, as well as the use of a Jira board that has the development stages suggested by the Kanban methodology.

As a result of the use of the aforementioned tools, deployments will be made in a shorter time allowing operational efficiency. In addition, thanks to the use of automated pipelines, errors in production passes can be reduced. It is also worth mentioning that because this new flow is simple to use, the time required to train new members is less.

Keywords: Continuous integration, Big Data, Financial Institution, Process Deployment, Kanban

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

En el presente informe profesional se va a describir el uso de herramientas de integración continua para el despliegue de procesos big data apoyándose en la metodología Kanban con el fin de disminuir el tiempo de puesta en producción de los procesos.

La necesidad de utilizar un nuevo flujo de integración continua surge debido a la alta demanda de despliegues que tiene la entidad financiera por lo que el uso de nuevas tecnologías favorece la eficiencia del uso de recursos lo que le permite seguir siendo una entidad competitiva en el mercado.

El informe se desarrolló con la siguiente estructura:

En el CAPITULO I se detalla cronológicamente la formación profesional de la autora, sus funciones, logros, experiencia y principales aprendizajes obtenidos en el mundo laboral; también se mencionan los conocimientos complementarios adquiridos por la autora.

En el CAPITULO II se resume parte de la experiencia desempeñada por la autora en la entidad financiera, la visión, misión y la estructura de la organización.

En el CAPITULO III se detalla el principal problema que atravesó la organización y se indica la solución para la misma, en donde se describe las etapas, metodologías utilizadas y la evaluación económica.

En el CAPITULO IV se indica la reflexión crítica de la experiencia realizada luego de implementar esta solución en la organización.

En el CAPITULO V se detallan las conclusiones de informe y las recomendaciones para una siguiente versión de la solución.

CAPITULO I:
TRAYECTORIA PROFESIONAL

La autora del presente trabajo, actualmente tiene el grado de Bachiller en la carrera de Ingeniería de Sistemas de la Universidad Nacional Mayor de San Marcos. Tiene experiencia en gestión de información, inteligencia de negocios, análisis de indicadores y big data.

También posee habilidades blandas que le permiten desarrollar liderazgo y trabajo en equipo, así como buenas relaciones interpersonales que le ayudan a resolver problemas, comunicar sus ideas con claridad e influir en las personas.

Adicionalmente, la autora ha tomado cursos y especializaciones en diversas entidades educativas reconocidas.

Actualmente, la autora busca seguir consolidando sus habilidades en el mundo de big data, así como contribuir al logro de los objetivos de su equipo de trabajo.

A continuación, se detalla la experiencia profesional de la autora:

Tabla 1

Experiencia Profesional - Empresa 1

Banco de Crédito del Perú	
Febrero 2020 – Actualidad	
Cargo:	Data Engineer Advanced
Funciones:	Migración de información de los diferentes modelos del Sandbox de Planeamiento Banca Minorista a Data lake en sus diferentes capas (RDV, UDV, DDV) mediante el uso de Cloudera, Hadoop, PySpark, HQL, Datastage, HDFS, Herramientas de integración contínua (Jenkins, Bitbucket, Git) y Jira.

Nota: Elaboración propia

Tabla 2

Experiencia Profesional - Empresa 2

Banco de Crédito del Perú	
Febrero 2019 – Enero 2020	
Cargo:	Analista Senior de Información y Control de Gestión - Compensaciones
Funciones:	• Cálculo de incentivos trimestrales en base a indicadores del negocio para realizar el pago de la remuneración mensual a los colaboradores del banco. • Mantenimiento del sistema de cálculo de incentivos trimestrales (modificación de indicadores y/o reglas del negocio) • Automatización del proceso de envío de información de pagos al área de Nómina. • Ejecución presupuestal mensual (Seguimiento y justificación del gasto mensual real sobre el valor presupuestado) • Mejora y automatización de procesos y reportes internos del área.

Nota: Elaboración propia

Tabla 3

Experiencia Profesional - Empresa 3

Banco de Crédito del Perú	
Febrero 2018 – Enero 2019	
Cargo:	Analista de Información y Control de Gestión - Compensaciones
Funciones:	• Cálculo de comisiones en base a la producción y reglas del negocio para realizar el pago de la remuneración mensual a los colaboradores del banco. • Mantenimiento del sistema de cálculo de comisiones mensuales (modificación de indicadores y/o reglas del negocio) • Implementación de alertas y controles para validar los pagos mensuales.

Nota: Elaboración propia

Tabla 4

Experiencia Profesional - Empresa 4

Interbank	
Marzo 2017 – Enero 2018	
Cargo:	Analista Jr. de Control de Campañas
Funciones:	• Seguimiento y control de Campañas de Adquisición dirigidas a canales tradicionales (red de tiendas y televentas) mediante el uso de indicadores como efectividad, desembolsos y contribución. • Análisis del feedback de envíos de comunicaciones (email/mensajes de texto) para la medición de campañas de consumo. • Elaboración de nuevos reportes y análisis basados en los resultados de las iniciativas comerciales del banco. • Mejora y automatización de procesos actuales.

Nota: Elaboración propia

A continuación, se detalla la formación académica profesional de la autora:

Tabla 5

Formación Académica Profesional

Formación:	**Grado Académico de Bachiller en Ingeniería de Sistemas.**
Institución:	Universidad Nacional Mayor de San Marcos Facultad de Ingeniería de Sistemas e Informática Escuela Académica Profesional de Ingeniería de Sistemas
Periodo:	2012 -2016

Nota: Elaboración propia

A continuación, se detalla la formación académica complementaria de la autora:

Tabla 6

Formación Académica Complementaria / Cursos

Especialización en Business Intelligence	
Institución:	DMC Perú
Periodo:	2018/06 - 2018/11
R for Business Analytics	
Institución:	DMC Perú
Periodo:	2019/03 - 2019/06
Big Data Analysis: Hive, Spark SQL, DataFrames and GraphFrames	
Institución:	Coursera
Periodo:	2020/06
Introduction to SQL on Databricks	
Institución:	Databricks
Periodo:	2020/09
Fundamentals of Unified Data Analytics with Databricks	
Institución:	Databricks
Periodo:	2020/09
Fundamentals of Big Data	
Institución:	Databricks
Periodo:	2020/09
Applications of SQL on Databricks	
Institución:	Databricks
Periodo:	2020/09
Programming for Everybody (Getting Started with Python)	
Institución:	Coursera
Periodo:	2021/02
Python Data Structures	
Institución:	Coursera
Periodo:	2021/03
Programa BigData Specialist	
Institución:	Datahack
Periodo:	2021/01 - 2021/03
Using Python to Access Web Data	
Institución:	Coursera
Periodo:	2021/05
Using Databases with Python	
Institución:	Coursera
Periodo:	2021/06
Capstone: Retrieving, Processing, and Visualizing Data with Python	
Institución:	Coursera
Periodo:	2021/06

Programa Especializado - Python for Everybody	
Institución:	Coursera
Periodo:	2021/06

Nota: Elaboración propia

A continuación, se detallan los idiomas que la autora conoce:

Tabla 7

Idiomas

Idioma:	**Inglés (Avanzado)**
Institución:	Británico
Periodo:	2014

Nota: Elaboración propia

CAPITULO II:
CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA

2.1. EMPRESA – ACTIVIDAD QUE REALIZA

La entidad bancaria es uno de los bancos más grandes del Perú, cuenta con una variedad de productos y servicios financieros bastante amplia, esto le permite atender las necesidades de los diversos segmentos de consumo del país.

La entidad bancaria tiene 130 años realizando operaciones en el país lo que le ha permitido consolidarse muy bien en el mercado. Para ofrecer sus servicios de manera adecuada cuenta con dos divisiones: Banca Mayorista y Minorista; siendo esta última la que ofrece productos como tarjetas de crédito, cuentas de ahorro, préstamos, etc.

Cabe resaltar también que la entidad tiene presencia en todo el territorio peruano y asegura la atención de sus más de 13 millones de clientes al contar con más de 8340 puntos de atención a lo largo del país.

2.2. VISIÓN

- Ser la empresa peruana que brinda la mejor experiencia a los clientes. Simple, cercana y oportuna.
- Ser la comunidad laboral de preferencia en el Perú, que inspira, potencia y dinamiza a los mejores profesionales.
- Ser referentes regionales en gestión empresarial potenciando nuestro liderazgo histórico y transformador de la industria financiera en el Perú.

2.3. MISIÓN

- Ayudar a los peruanos a transformar sus planes en realidad construyendo su historia de desarrollo y superación.

2.4. ORGANIZACIÓN DE LA EMPRESA

A continuación, se presenta el organigrama de la entidad bancaria:

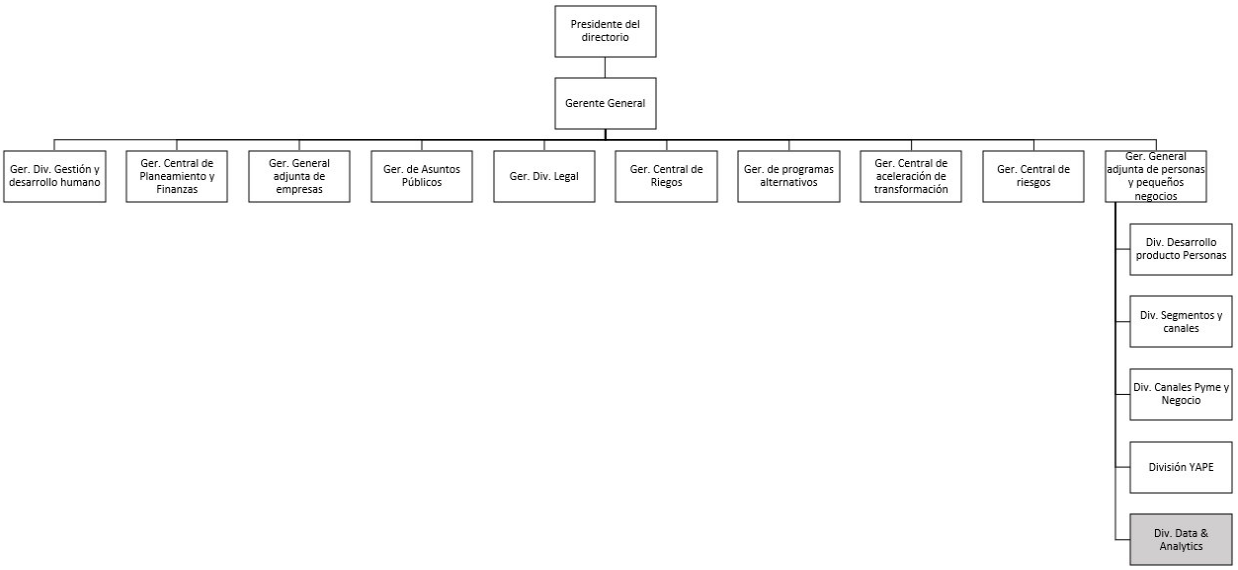


Figura 1. Organigrama de la empresa
Nota: Elaboración propia

Como se observa en la Figura 1, la entidad bancaria cuenta con diversas divisiones y gerencias centrales; de manera específica se procede a describir la División de Data & Analytics.

La división de Data & Analytics es la división del banco encargada de obtener, gestionar, generar soluciones y brindar propuestas de valor en base a la información; de manera que pueda usarse a nivel transversal en la entidad.

El squad donde se desarrolló la experiencia se encuentra en esta división y está compuesto por diversos roles como son: 5 data engineers, 1 data governance y 1 data modeler.

De manera interna, la división está compuesta por Chapters. Cada chapter representa una especialidad, entre ellas tenemos:

- Chapter de Ingeniería: Es la especialidad que desarrolla la línea de carrera de los ingenieros de datos; quienes son los principales responsables de los desarrollos de soluciones.

- Chapter de Arquitectura: Es la especialidad que desarrolla la línea de carrera de los arquitectos de datos; quienes son los principales

responsables de la generación de lineamientos de buenas prácticas en la división, así como retar procesos y arquitectura actual.

- Chapter de Modelamiento: Es la especialidad que desarrolla la línea de carrera de los modeladores de datos; estos son roles Cross que participan en las implementaciones de todas las soluciones.
- Chapter de Gobierno y Calidad: Es la especialidad que desarrolla la línea de carrera de los custodios de información; al igual que el modelador, también es un rol Cross que apoya en la implementación de los lineamientos de calidad y gobierno a los despliegues de todas las soluciones.

2.5. ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS

La autora de este trabajo de experiencia profesional actualmente se desempeña como Data Engineer Advanced en un squad de la División de Data & Analytics.

Las funciones que desempeña la autora son:

- Migración de información de los diferentes modelos del Sandbox de Planeamiento Banca Minorista a Data lake on Premise haciendo uso de las tecnologías de almacenamiento y procesamiento de grandes volúmenes de información como son Haddop, HDFS, Pyspark, Hive, etc.
- Análisis de estrategias de migración de información priorizando la necesidad del cliente. Como parte de este análisis se revisan la viabilidad de las propuestas de migración, así como la evaluación del tiempo y esfuerzo necesario con la finalidad de planificar objetivos de manera realista.
- Desarrollo y despliegue de nuevos componentes / procesos usando metodologías ágiles y herramientas de integración continua como Bitbucket, Git y Jenkins asegurando la correcta puesta en producción de los procesos del equipo.

2.6. EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN

La autora del presente trabajo de experiencia profesional participó como team member de un squad de la División de Data & Analytics obteniendo las siguientes experiencias:

- Participó en el mapeo de fuentes sandbox habilitadas para su migración al data lake identificando las prioridades con las que debían ser migradas estas fuentes. Para este mapeo se revisaron los procesos sandbox (scripts sql) para identificar los procesos involucrados.
- Participó en la estimación de tiempos y recursos (capacity del equipo) para el cumplimiento de los OKRs de la división. Llevando a cabo ceremonias de planificación con todo el equipo involucrado a fin de sincerar actividades y tiempos de entrega.
- Participó en la implementación de un componente de generación de backups de tablas de formato parquet usando pySpark. Realizando las pruebas, validaciones y comparando tiempos de ejecución sin el componenente. Además, participó en la reunión de comunicación al Chapter de Ingenieria sobre el uso aprobado de este componente.
- Participó en el despliegue de pipelines usando Jenkins, logrando que el squad lo en el 100% de los desarrollos soportados por la herramienta. Para ello la autora recibió la capacitación del equipo de arquitectura y posteriormente capacitó al resto de team members del squad, logrando que el equipo sea uno de los primeros en adoptar el uso de la integración continua al 100%.

CAPITULO III
ACTIVIDADES DESARROLLADAS

3.1. SITUACIÓN PROBLEMÁTICA

Las entidades financieras en la actualidad vienen cambiando sus paradigmas, esto se ve reflejado tanto en sus servicios y como en el uso de las tecnologías para reducir tiempos y optimizar sus procesos vinculados a la gestión de sus datos debido a que en el contexto actual existen tiempos de espera largos en los procesos, islas de datos, información no integrada, entre otros que no permite a las entidades sacar el máximo provecho de la información que poseen.

La entidad bancaria, a la que hacemos referencia en este trabajo de suficiencia profesional, al ser una de las más grandes del Perú, también cuenta con numerables sistemas y desarrollos que pasan a producción los 365 días del año y al ser la empresa muy grande, la demanda de pases a producción es muy elevada, lo que requiere exista algún flujo que permita hacer el seguimiento ordenado de todas estas actividades para que se lleven a cabo de manera adecuada.

La entidad bancaria no tenía un flujo de despliegues que funcione de manera ágil e integrada que permita atender a tiempo la cantidad de despliegues que se generan a diario, lo que genera demoras y retrasos en los pases a producción.

3.1.1. DEFINICIÓN DEL PROBLEMA

3.1.1.1. PROBLEMA PRINCIPAL

No se tiene un flujo de despliegue de procesos big data que use herramientas de integración continua en un squad de una entidad financiera.

3.1.1.2. PROBLEMAS SECUNDARIOS

a. Los pases a producción de los procesos de big data toman demasiado tiempo.

b. Existen errores al pasar a producción los procesos debido a que no se usan repositorios integrados.

c. Los tiempos de desarrollo y pruebas son muy extensos, debido a que no se usan pipelines automatizados.

d. Tiempos de capacitación muy extensos en la aplicación del despliegue de los procesos de big data.

3.2. SOLUCIÓN

La adopción de un flujo de integración continúa apoyada en la metodología Kanban ayudaría a reducir el tiempo de despliegue de los procesos big data de la entidad financiera mejorando la eficiencia y reduciendo errores.

3.2.1. OBJETIVOS

3.2.1.1. GENERAL

Usar herramientas de integración continua para reducir el tiempo de despliegue de procesos de big data en una entidad financiera bajo el enfoque de la metodología Kanban.

3.2.1.2. ESPECÍFICOS

1. Reducir el tiempo de pases a producción de procesos big data de un squad de una entidad financiera.
2. Disminuir los errores al pasar a producción los procesos de big data mediante el uso de repositorios integrados.
3. Reducir los tiempos de desarrollo y pruebas de procesos de big data mediante el uso de pipelines automatizados.
4. Disminuir el tiempo de capacitación para el despliegue de un pase a producción en un squad de una entidad financiera.

3.2.2. ALCANCE

3.2.2.1. ALCANCE FUNCIONAL

A nivel funcional, el nuevo flujo de pases a producción estará apoyado en un tablero Jira que reflejará las etapas de desarrollo de la metodología Kanban, haciendo que el seguimiento e interacción entre el desarrollador y otros equipos sea mucho más sencillo.

3.2.2.2. ALCANCE ORGANIZACIONAL

A nivel organizacional, el nuevo flujo de pases a producción será una herramienta que va a permitir mejorar la integración entre las diferentes áreas de la entidad financiera involucradas en los pases a producción, lo que traerá como consecuencia la disminución de tiempos. Las áreas que participan son:

- Squad de desarrollo: Es el equipo de desarrolla el proceso que se llevará a producción.
- Área de seguridad: Es el equipo que valida que se cumplan los lineamientos de seguridad de la entidad en el desarrollo.
- Área de gobierno: Es el área que da la conformidad del pase a producción del proceso validando que cumpla los lineamientos de gobierno y no impacte a otros procesos.
- Área de operaciones: Es el área que ejecuta el pase a producción.

3.2.2.3. ALCANCE TECNOLÓGICO

A nivel tecnológico, el nuevo flujo de pases a producción contará con el uso de las herramientas: amarra con la organizacional

- Bitbucket: Servirá como un repositorio centralizado donde los data engineers colocarán los entregables que pasarán a producción.

- Git: Servirá como un controlador de versiones para los data engineers.
- Jenkins: Servirá para la automatización con pipelines, podrá ser usado por los data engineers para pruebas en desarrollo, así como para el equipo de operaciones en los despliegues en producción.
- Jira: En este tablero se registrará el pase a producción y permitirá visualizar en que etapa se encuentra y que rol lo está atendiendo.
- Data Lake: En este ambiente se desplegarán los pases a producción.

3.2.3. ETAPAS Y METODOLOGIA

Se consideraron 2 etapas principales en el desarrollo del proyecto; estas fueron la habilitación del squad y en segundo lugar el uso de la metodología Kanban para desplegar el flujo de integración continua. A continuación, se describen ambas etapas:

3.2.3.1. Habilitación del squad:

Para que el squad de la entidad financiera pueda utilizar las herramientas de integración continua, el Chapter de Arquitectura de la división de Data & Analytics solicitó que se sigan los siguientes pasos para habilitar al squad antes de los pases a producción, en la Tabla 8 se indica las etapas:

Tabla 8

Etapas para la habilitación del squad

Etapa	Descripción
Identificación de roles	Para el desarrollo del flujo se requiere que participen diferentes actores como son desarrolladores, un líder técnico, un analista de calidad y un product owner (las funciones

	que desempeñan estos roles están detalladas más adelante), por ello fue necesario asignar a los team member del squad en los diferentes roles.
Solicitud de accesos	Solicitud al equipo de accesos de la entidad bancaria a los grupos de red para el logeo en jira, bitbucket y Jenkins, indicando el rol para los accesos correspondientes.
Capacitación de Roles	El Chapter de Arquitectura brindó una capacitación, manuales y guías de uso de herramientas a los líderes técnicos y producto owners de cada squad sobre su rol y funciones habilitadas en el flujo.
Capacitación de Squad	El líder técnico realiza una capacitación a los demás team members del squad en el uso de herramientas de integración continua y uso del tablero Jira de acuerdo al rol que puede ser de Analista de calidad o desarrollador.

Nota. Elaboración propia

A continuación, se detallan los roles que participan en el flujo de integración continua:

a. **Desarrollador**:

Es el team member del squad que ha desarrollado la solución que será llevada al ambiente de producción. Será el encargado de hacer las pruebas en desarrollo y registrar el pase a producción en el tablero Jira.

b. **Líder técnico**:

Es el team member del squad que valida el correcto registro del pase a producción en Jira y da la aprobación para que el ticket inicie su flujo en el carril de seguridad. Adicionalmente, el líder técnico puede ejecutar el despliegue en ambiente de certificación.

c. **Analista de seguridad**:

Analista de la gerencia de TI encargado de validar los accesos, permisos, entre otros puntos que se estén solicitando en el pase a producción que puedan vulnerar la seguridad de los ambientes.

d. **Analista de Calidad**:

Es el team member del squad encargado de hacer las validaciones de la solución durante los estadíos de certificación y ratificación del pase a producción.

e. **Product Owner**:

Es el encargado de traer las iniciativas de desarrollo al squad así como aprobar las mismas cuando han sido validadas en congelamiento para su calendarización y programación de pase a producción.

f. **Agile operator (Agile ops)**:

Es el encargado de ejecutar todas las tareas que componen el despliegue en el ambiente productivo. La calendarización de pases a producción depende de la disponibilidad del equipo Agile ops.

3.2.3.2. Uso de la metodología KANBAN

La metodología Kanban es una metodología ágil ampliamente usada en el mundo de desarrollo de software. Esta metodología fue seleccionada por la división de Data & Analytics como guía para las etapas de desarrollo de procesos de big data. Para plasmar las fases de esta metodología se utiliza un tablero Jira que tiene etapas o carriles diferenciados según el estadío del proceso de big

data. En la Figura 2 se muestran los carriles por los que debe pasar todo proceso para llegar a un ambiente productivo.

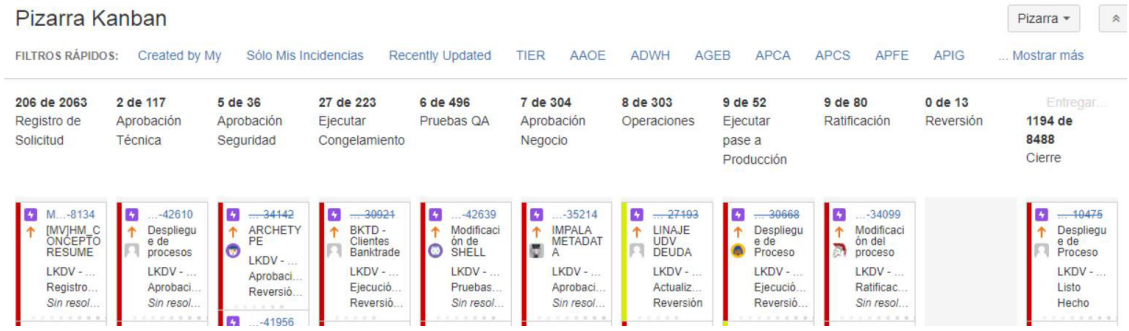


Figura 2. Flujo de integración continua
Nota: Tomado de fuente interna de la entidad financiera

A continuación, se detallan las actividades que se realizan en cada uno de los carriles, así como el rol que las realiza:

a. Registro de solicitud:

En esta fase realiza el registro de un proceso en el tablero Jira, esta tarea debe ser realizada por el rol desarrollador y genera un ticket en la pizarra Jira con su ID de proceso con el que se podrá hacer el seguimiento en el resto de etapas. En esta etapa el desarrollador debe registrar todas las instrucciones que se van a desarrollar en la etapa de certificación y pase a producción, adicionalmente se deben indicar instrucciones de reversión en caso hubiera algún problema en la certificación o en el pase a producción. Una vez culminado el registro, el desarrollador le envía el ticket al líder técnico del squad para iniciar la siguiente etapa.

b. Aprobación técnica:

En esta etapa el líder técnico recibe el ticket y debe realizar la validación de los entregables que pasarán a producción, revisando que los entregables se encuentren bajo los estándares y lineamientos definidos por la entidad bancaria. Una vez culminada esta etapa, el líder técnico le envía el ticket al analista de seguridad para iniciar la siguiente etapa.

c. Validación de seguridad:

El analista de seguridad recibe el ticket y debe realizar las validaciones correspondientes a accesos, permisos y otros temas que puedan vulnerar los lineamientos de seguridad establecidos por la entidad bancaria. Al culminar, el analista de seguridad moverá el ticket al carril de congelamiento.

d. Congelamiento:

En esta etapa de congelamiento o certificación intervienen dos roles: el líder técnico y el Agile OPS. El líder técnico inicia las actividades de este carril generando una subtarea en Jira y asignándola al Agile OPS para su atención. El agile OPS atiende la subtarea y adjunta las evidencias para la posterior validación. En estas subtareas se encuentran detalladas las ejecuciones de los pipelines de Jenkins que apuntan al repositorio Bitbucket que el desarrollador completó. Una vez culminadas las subtareas, el operador deriva el ticket al QA del squad.

e. Pruebas QA:

En esta etapa, el QA del squad realiza las validaciones necesarias en el ambiente de certificación para garantizar que se realizó el despliegue de manera correcta. Una vez culminada esta validación el ticket se deriva con el Product Owner (PO) del squad.

f. Aprobación de negocio:

En esta etapa el PO del squad da su conformidad para que los cambios validados en certificación sean desplegados en producción. Una vez se brinda este conforme el ticket es derivado a la etapa de operaciones donde el líder técnico del squad continuará con el flujo.

g. Operaciones:

En esta etapa el líder técnico programa en el calendario del equipo de operaciones la fecha y hora exacta en que se realizará el pase a producción, posteriormente solicitará una conformidad al equipo de operaciones y luego el ticket será derivado al carril de pase a producción.

h. Pase a producción:

En este carril únicamente interviene el agile OPS ya que se encargará de desplegar el proceso en el ambiente de producción. Este despliegue contará de la ejecución del pipeline de Jenkins en ambiente productivo. Así mismo debe dejar evidencias de los cambios realizados que puedan servir para la posterior validación. Una vez culminado el pase a producción el ticket es derivado al carril de ratificación.

i. Ratificación:

En este carril el QA del squad realiza la validación del proceso desplegado en el ambiente productivo basándose en las evidencias que se generaron en el carril anterior. En caso no se genere la conformidad se debe proceder a solicitar una reversión, caso contrario se procederá al cierre del ticket.

j. Reversión:

Se llega a este carril solo en caso que el pase a producción no haya sido exitoso. El agile OPS será el encargado de revertir los cambios realizados en producción a fin de no interferir en otros procesos.

k. Cierre:

Si el pase a producción resulta exitoso, el QA del squad deriva el ticket a este carril donde se cierra la atención considerando cambios correctos en producción.

A continuación, se muestra un gráfico con el flujo que debe seguir todo proceso de big data para pasar a producción y los roles que participa.

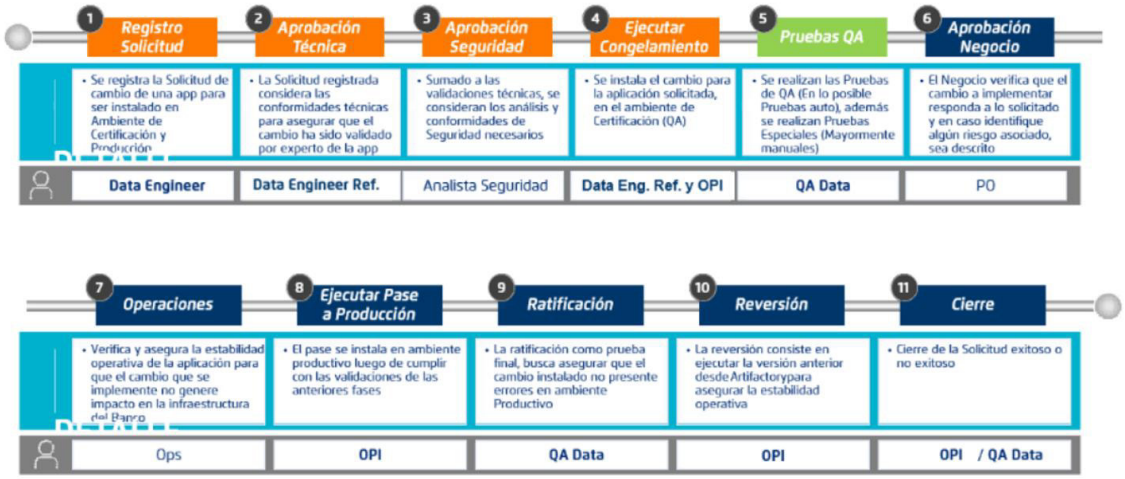


Figura 3.Flujo que siguen los procesos big data y roles participantes
Nota: Tomado de fuente interna de la entidad financiera

3.2.4. FUNDAMENTOS UTILIZADOS

3.2.4.1. Metodología Kanban

Kanban es una metodología originada en el sistema de producción de Toyota y fue adaptada a la ingeniería y desarrollo de software a finales de los años 2000. Kanban se centra en limitar los trabajos de desarrollo, así como en visualizar la cadena de valor del desarrollo lo que permite reducir el Cycle time (tiempo que demora en pasar a producción un proceso). (Hofmann, Lauber, Haefner, & Lanza, 2018)

Para usar esta metodología, en la entidad bancaría se hace uso de un tablero Kanban plasmado en la herramienta Jira, que será detallado más adelante. (Hofmann, Lauber, Haefner, & Lanza) nos dicen sobre el tablero Kanban que este no está restringido a un equipo y una iteración, sino que visualiza todo el flujo de trabajo y permite la colaboración de múltiples equipos e individuos de múltiples dominios para construir un producto complejo.

Según (Turner, Ingold, Lane, Ray, & Andersond), Kanban se centra en más en limitar el trabajo en curso según la capacidad disponible del team; es

decir, no se pueden iniciar nuevas tareas o actividades sin que se cuente con un recurso disponible que pueda desarrollarlas. El proceso se gestiona mediante el uso de WIP, por sus siglas en ingles Work in progress, que permite controlar el máximo de actividades que se pueden desarrollar en paralelo con los recursos del equipo.

3.2.4.2. Jira

Jira es un software potente diseñado para ayudar a equipos de todo tipo a gestionar el trabajo. En sus inicios Jira fue utilizado solo para la gestión de incidencias y errores, pero con el tiempo ha adquirido mayor relvancia y funcionalidades tanto así que en la actualidad puede ser usado para cualquier caso de uso entre ellos el desarrollo de software ágil. (Atlassian, 2021)

De acuerdo con el sitio web oficial de (Atlassian), el tablero Kanban de Jira es mucho más que un tablero para gestionar tareas ya que permite a los equipos realizar lo siguiente:

- Promover la transparencia: Jira sirve como una fuente de información para conocer el estado de las historias y permite la comunicación efectiva.
- Optimizar flujos de trabajo: Debido a la flexibilidad de Jira se pueden configurar los flujos que el equipo requiera representar de forma visual.
- Detectar los cuellos de botella fácilmente: El trabajo en curso (WIP) es la cantidad de historias que hay por cada carril, si se establecen estos límites correctamente se podrán prevenir cuellos de botella ya que solo se asignará lo que el capacity del equipo pueda completar.
- Mejora continua: Jira brinda reportes de métricas visuales que permiten monitorear la duración del ciclo y detectar impedimentos.

A continuación, se muestra un tablero Kanban general que permite identificar las actividades por carril, así como los cuellos de botella de manera rápida y visual.

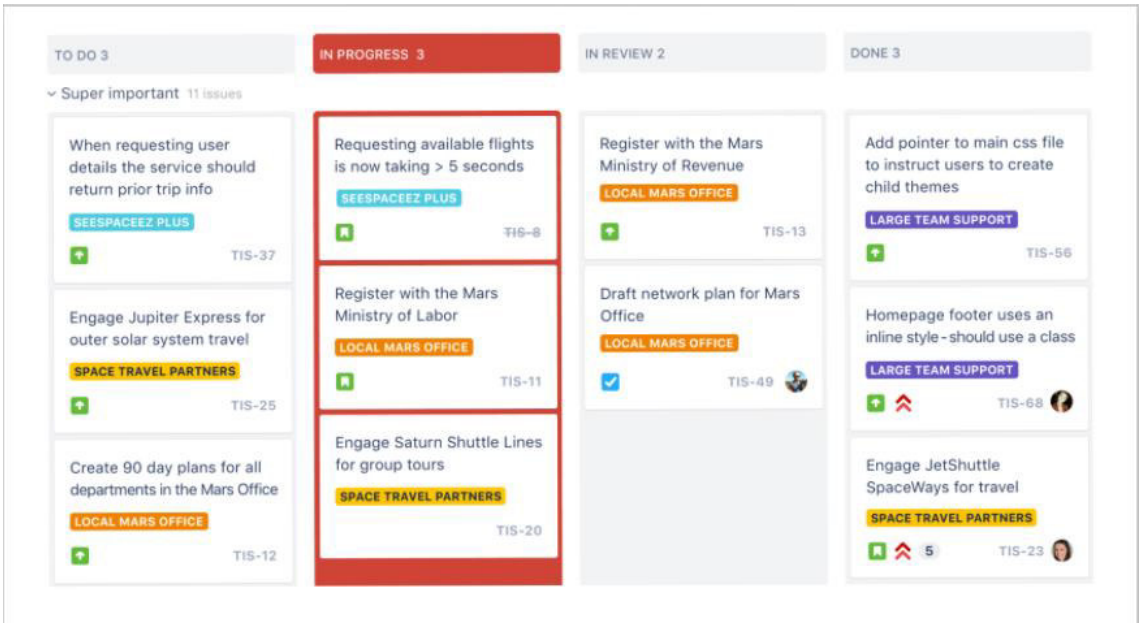


Figura 4. Tablero Jira con enfoque Kanban
Nota: Tomado de la web de (Atlassian)

3.2.4.3. Integración continua

La integración continua es una práctica utilizada en el desarrollo de software, en la cual los desarrolladores utilizan un repositorio central para actualizar cambios en el código generando versiones que pueden ser ejecutadas y probadas. El objetivo principal de la integración continua es detectar y corregir errores de manera temprana, permitiendo mejorar la calidad del software y reduciendo el tiempo de validación. (AWS, 2021)

La integración continua favorece el trabajo colaborativo ya que varios desarrolladores pueden utilizar un repositorio centralizado y compartido utilizando software de control de versiones, a la vez que pueden ejecutar pruebas de manera local para identificar de manera inmediata cualquier error. (AWS, 2021)

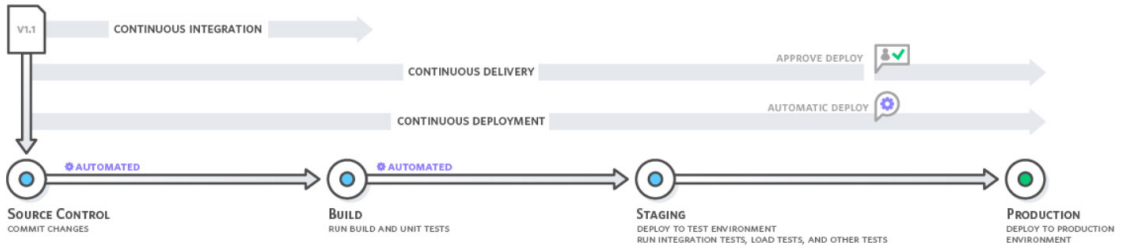


Figura 5. Diagrama de Integración continua
Nota: Tomado de la web de (AWS)

3.2.4.4. Bitbucket

Bitbucket es un repositorio para colaborar, probar y desplegar código ya que se puede integrar con otras herramientas para mejorar la calidad del código y desplegar pruebas de manera automatizada. Permite mantener control de versiones y control mediante flujos de aprobaciones previamente designadas, así mismo permite implementar opciones de seguridad y control sobre las ramas, así como comprobación de fusiones. (Atlassian, Bitbucket, 2021)

3.2.4.5. Jenkins

Jenkins es un servidor open source que sirve para la integración continua. Esta herramienta permite compilar y probar proyectos de software de manera continua, lo que favorece la detección rápida de errores y mejoras. Gracias al uso de Jenkins los equipos pueden acelerar su desarrollo y entrega de software debido a la automatización ya que cada cambio que realice un desarrollador puede ser probado rápidamente. (Sentrio, 2021)

Podemos resumir el uso y ventajas que ofrece Jenkins en los siguiente:

- Cada desarrollador maneja su Commit y se centra en el mismo para realizar sus pruebas.
- Se pueden conocer los resultados de las pruebas de manera rápida.
- El despliegue y pruebas puede ser automatizado.
- El ciclo de desarrollo es más rápido.

3.2.4.6. Git

Git es un sistema de control de versiones distribuido de código abierto y gratuito, está diseñado para manejar todo tipo de proyectos con velocidad y eficiencia. (Git, 2021) Utilizando Git se puede descargar el contenido de Bitbucket de forma local para trabajar los cambios y al culminar nuevamente subirlos al repositorio.

3.2.4.7. Big Data

(Miloslavskaya & Tolstoy) nos dicen que nos referimos a big data cuando tenemos dataset de un tamaño muy grande y con una estructura que excede la capacidad de procesamiento de las herramientas tradicionales usadas en la programación como son las bases de datos. Los datos pueden ser de diferentes tipos como son estructurados, semi estructurados y no estructurados lo que hace imposible su recolección, almacenamiento y procesamiento con herramientas tradicionales. Además, podemos mencionar 3 criterios que nos sirven para identificar big data: Volumen alto de datos, velocidad de transferencia de datos alta, variedad de tipos y estructura de datos.

3.2.4.8. Data Lake

Un data lake o lago de datos se refiere a un repositorio de almacenamiento escalable y que puede contener una gran cantidad de datos de forma cruda o nativa hasta que sea necesario procesarlos sin comprometer su estructura. Generalmente los lagos de datos se construyen con el fin de manejar alta volumetría de data no estructurada. Las estrategias para almacenar la información pueden contener enfoques de datos SQL y NoSQL así como también se puede contar con análisis y procesamiento en línea. (Miloslavskaya & Tolstoy, 2016)

Los autores (Miloslavskaya & Tolstoy), también indican que los data lakes se pueden dividir en 3 capas que componen de manera sencilla su arquitectura: uno para la adquisición de data cruda, un segundo nivel para data transformada con algún procesamiento, y una tercera capa para publicar o compartir información a terceros. A continuación, se muestra un gráfico que esquematiza las 3 capas de la arquitectura del data lake:

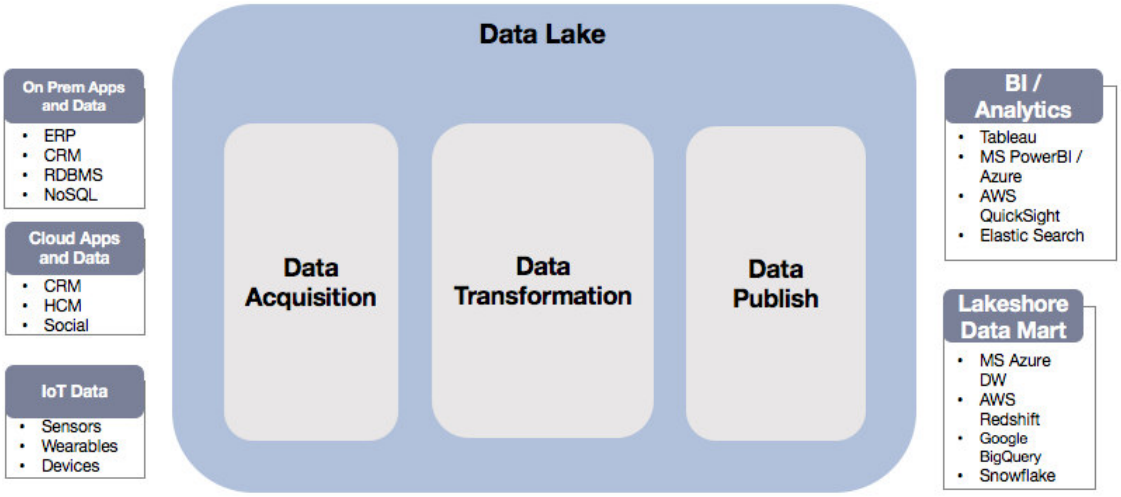


Figura 6. Arquitectura del data lake
Nota. Imagen tomada de la web de (Snaplogic, 2021)

Los autores (Miloslavskaya & Tolstoy), nos comentan que algunas características con las que debe contar el data lake son:

- Su arquitectura debe ser escalable.
- Debe existir un gobierno de información.
- Se debe almacenar el linaje de la información, es decir conocer la trazabilidad y origen de los datos.
- El almacén debe ser único y accesible.
- Se debe contar con calidad del servicio.

3.2.4.9. Cadena de valor de Big data

De acuerdo a los autores (Kumar Bhadani & Jothimani) la cadena de valor del big data se refiere a las actividades que realiza una entidad para agregar valor en cada entrega de un producto o servicio a sus clientes, en este caso generar valor a partir del uso de los datos. Esta cadena de valor se puede dividir en siete etapas que se muestran en el siguiente gráfico:

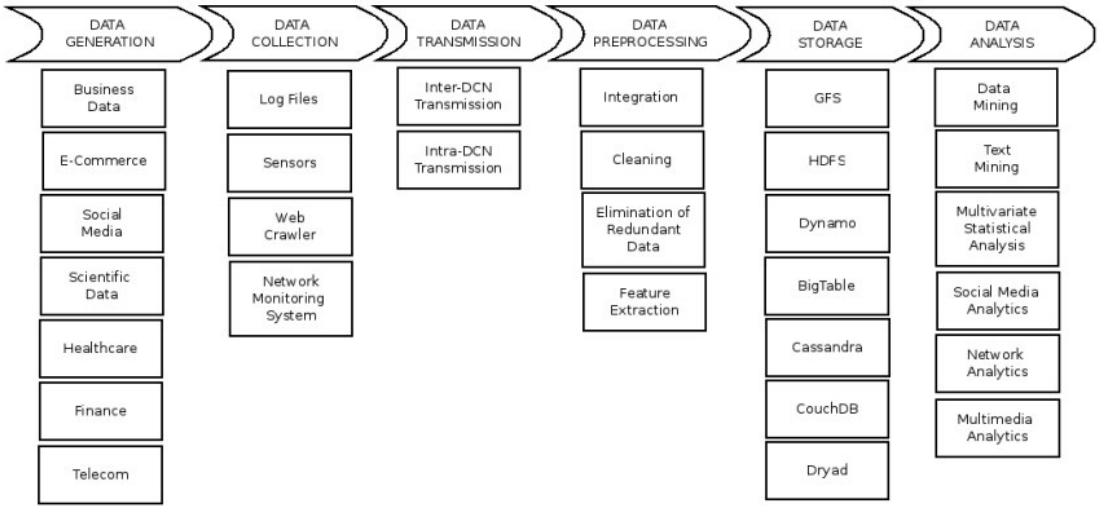


Figura 7. Cadena de valor del data lake

Nota. Imagen tomada de (Kumar Bhadani & Jothimani, 2017)

A continuación, se describen las etapas de la cadena de valor que nos describen (Kumar Bhadani & Jothimani):

- **Generación de datos:** Es el paso que da inicio a la cadena de valor y es el origen de los datos; estos pueden provenir de diversas fuentes que pueden incluir llamadas, redes sociales, entre otros.
- **Recolección de datos:** En este paso se trata de obtener todos los datos posibles de las fuentes disponibles, esta recolección puede ser a través de archivos, sensores, entre otros.
- **Transmisión de datos:** Una vez que los datos fueron recopilados es necesario transferirlos a una infraestructura de almacenamiento que permita su posterior tratamiento.
- **Procesamiento previo de datos:** Esta fase consiste en mejorar la calidad de los datos eliminando datos inconsistentes o redundantes, con esto se busca mejorar las fases posteriores de análisis. Para lograr este preprocesamiento se deben seguir los siguientes pasos:
    - **Integración:** Al existir una amplia variedad de fuentes de datos es necesario contar con una vista unificada; esto se puede lograr al momento de realizar la carga a la infraestructura de almacenamiento agregando varias fuentes de datos a la misma.

    - **Limpieza:** En este paso se valida la integridad y consistencia de los datos, se pueden incluir controles en los procedimientos de entrada para evitar errores futuros.
    - **Eliminación de datos redundantes:** Con el fin de no afectar la calidad se pueden realizar algunas técnicas de filtrado para retirar datos excedentes o redundantes.
- **Almacenamiento de datos:** El almacenamiento en big data es un punto clave ya que debe brindar un almacenamiento confiable y un acceso potente a los datos. Los sistemas de almacenamiento para big data son distribuidos y deben cumplir con algunas características como la consistencia, la alta disponibilidad y la tolerancia al particionado.
- **Análisis de datos:** Luego del almacenamiento de datos se debe realizar un análisis y para ello se deben seguir estos pasos:
    - **Definir métricas:** En función al problema a tratar se deben definir un conjunto de métricas a considerar.
    - **Seleccionar la arquitectura:** En función a algunas características del análisis a realizar se selecciona una arquitectura adecuada, por ejemplo, en el caso de un flujo de datos que cambian constantemente se requiere realizar un análisis en tiempo real, mientras que en otros casos se puede realizar un análisis offline. La arquitectura de uso más general es la plataforma Hadoop.
    - **Selección de herramientas:** De acuerdo al tipo de análisis y métrica que se quiera conseguir se puede optar por el uso de diferentes herramientas que pueden software de código abierto o comercial, esto en función de la técnica que se quiera utilizar como pueden ser algoritmos de minería, análisis de regresión entre otros.
- **Visualización de datos:** Esta etapa consta del uso de una interfaz gráfica para mostrar los análisis obtenidos. Algunas opciones para generar la visualización son Tableau, QlikView, entre otros.
- **Toma de decisiones:** Esta es la etapa final y se puede lograr en base a los análisis realizados y la visualización de los resultados, con esto se puede decidir los siguientes pasos y corregir causas de problemas.

3.2.5. IMPLEMENTACIÓN DE LAS ÁREAS, PROCESOS, SISTEMAS Y BUENAS PRÁCTICAS

El uso de herramientas de integración continua en la entidad financiera se inició con un piloto en la división de Data & Analytics, las fases que se siguieron para conseguir realizar un despliegue de proceso de big data a través del uso de estas herramientas fueron la habilitación del squad, la selección de historia de usuario y el uso del flujo basado en la metodología Kanban. Estas fases de describen a continuación:

3.2.5.1. Habilitación del squad:

A. Roles participantes:

Para habilitar a un squad en el uso de herramientas de integración continua en un primer momento se requiere definir quién de los team member tomará el rol de líder técnico y quien el de QA, quedando el resto de team members con el rol de desarrollador. Adicionalmente en el flujo se mantiene el rol de Product Owner y algunos roles externos al squad que se detallaron en la Tabla 8.

A continuación, se muestra cual fue la distribución de roles dentro del squad para realizar este piloto:

Tabla 9

Distribución de roles en el squad

Rol	Cantidad de miembros del Squad
Líder Técnico (LT)	1
Analista de Calidad (QA)	1
Desarrollador	3
Product Owner (PO)	1

Nota. Elaboración propia

B. Solicitud de accesos

Debido a las políticas de seguridad de la entidad bancaria, el acceso a las herramientas de integración continua se realiza a través de grupos de red, para ello seguridad disponibilizó 2 grupos de red que se debían solicitar según el rol según se muestra en la siguiente tabla:

Tabla 10

Grupos de red según rol

Tipo de Grupo de Red	Roles
Grupo de red developer	Desarrolladores, QA
Grupo de red admin	LT, PO

Nota. Elaboración propia

Adicionalmente en la herramienta Jira los accesos que se brindaron fueron de desarrollador por defecto por lo que también se debió de indicar al equipo de DevSecOps de la entidad bancaria la lista de roles y personas asignadas para que se den permisos especiales por rol.

C. Capacitación de roles

Una vez que los integrantes del equipo contaban con sus accesos a los grupos de red antes mencionados, el Chapter de Arquitectura convocó a los Product Owners y Líderes técnicos de cada squad a unas sesiones de capacitación ya que estos son los roles que tienen algunos accesos adicionales. En estas sesiones, los arquitectos de información indicaron la forma en la que se daría el uso de Bitbucket, Jenkins, git y jira siguiendo la metodología Kanban, así como el nivel de acceso a las herramientas con los que contará cada rol.

D. Capacitación de squad

Luego de que los Líderes técnicos y Product owners recibieron la capacitación, esta debía ser replicada a los demás team members (Analistas de calidad y desarrolladores) indicando como sería el flujo y cuál sería la interacción entre todos los roles.

3.2.5.2. Selección de Historia de Usuario:

Actualmente el squad se encuentra realizando la migración de la información de diversos aplicativos core de la entidad financiera, así como algunas fuentes sandbox, a continuación, se muestra una imagen donde se puede notar la arquitectura del datalake y flujo de migración que sigue el squad:

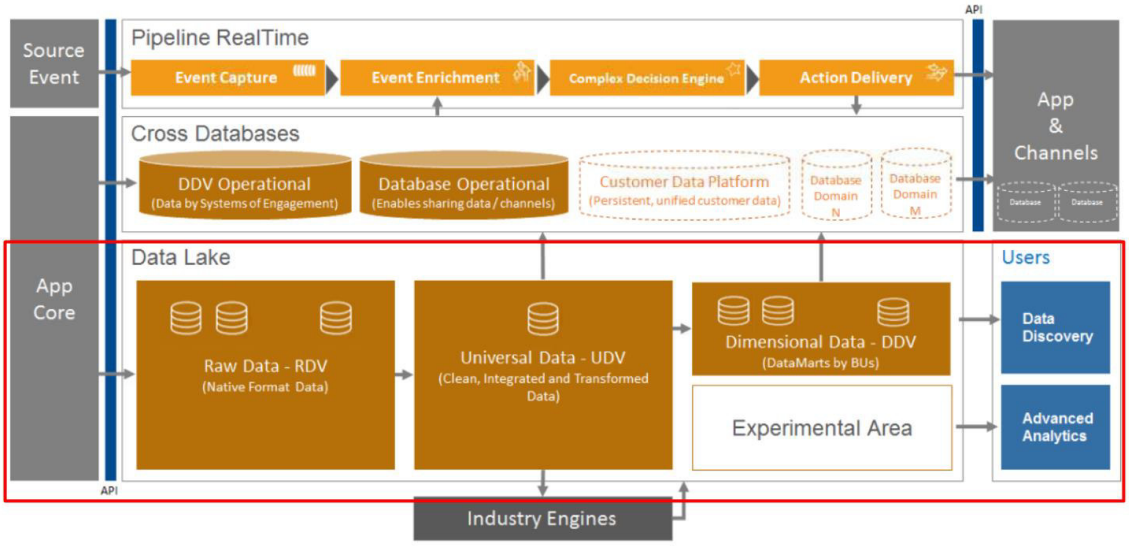


Figura 8. Arquitectura del data lake on premise de la entidad financiera
Nota: Tomado de fuente interna de la entidad financiera

Como se observa en la imagen el data lake cuenta con 3 capas que detallaremos brevemente:

- **RDV (Raw data vault):** Esta es la capa de ingesta de data cruda, es decir sin ninguna transformación.
- **UDV (Universal data vault):** Esta capa ya cuenta con data procesada y con validaciones de calidad, la información se puede encontrar agrupada por dominios.
- **DDV (Dimensional data vault):** Esta capa contiene la información lista para compartir a los usuarios finales, puede contener data agrupada de diversos dominios.

Conociendo un poco más la arquitectura, podemos notar que de manera tradicional la información que dejan los aplicativos (broads) era cargada a un datawarehouse y luego procesada y consumida en un sandbox, mientras que la

solución que esta disponibilizando el squad consiste en llevar las broads al data lake y realizar todo consumo desde la capa DDV del mismo.

Basándonos en lo descrito previamente, podemos comentar que los procesos que tiene el squad son de diversos tipos como: preparación de ambiente (creación de rutas y esquemas en Linux y HDFS), ingesta de datos a las capas RDV, UDV y DDV y Linaje de información para gobierno. Con fines prácticos, en este trabajo se tomará como ejemplo el proceso de ingesta de información del aplicativo Web Orgánico a la capa DDV.

3.2.5.3. Uso de flujo basado en la Metodología Kanban:

A continuación, se detallan las etapas por las que pasa todo proceso de big data para llegar a producción basándonos en el tablero Jira configurado bajo la metodología Kanban.

A. Registro de solicitud

La solicitud surge cuando el PO del squad trae un requerimiento y este es asignado a un data engineer, en este caso particular el requerimiento es la ingesta de información del aplicativo web Orgánico a la capa DDV. En ese momento el data engineer registra el pedido, que culminará en un pase a producción, en el tablero Jira.

Este registro lo realizará un data engineer con el rol de desarrollador y con ello se genera un código de despliegue único. Al registrar la solicitud el desarrollador debe indicar los datos necesarios para hacer identificable su proceso; estos datos son el nombre del proceso, la capa del data lake sobre la que se estará trabajando, una breve descripción del cambio, así como colocar los nombres de los roles involucrados, ya que el despliegue será derivado con alguno de ellos de acuerdo a la etapa.

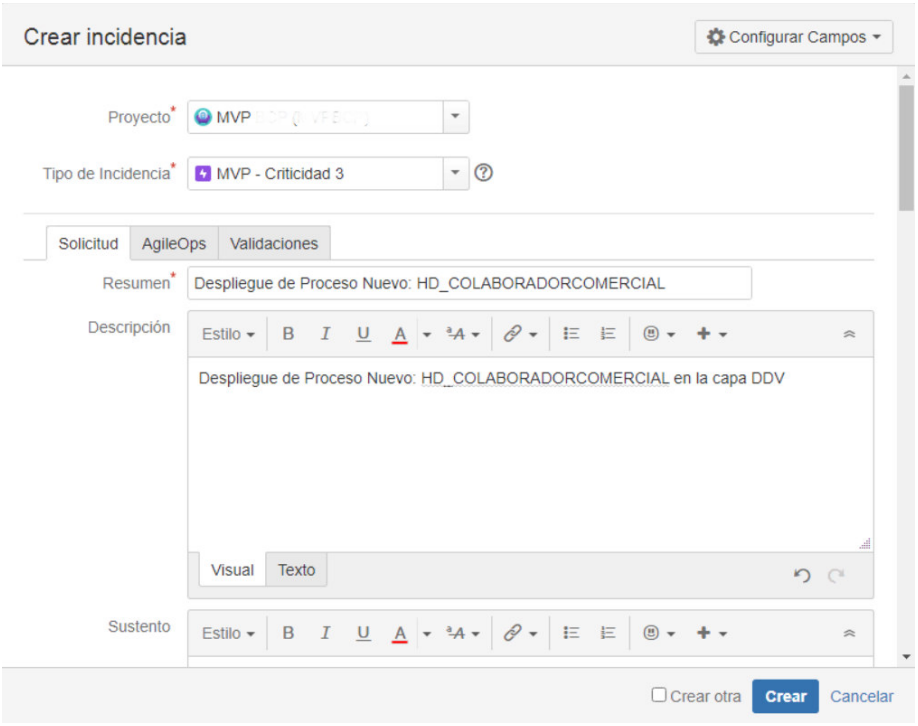


Figura 9. Creación de solicitud en Jira
Nota: Tomado de fuente interna de la entidad financiera

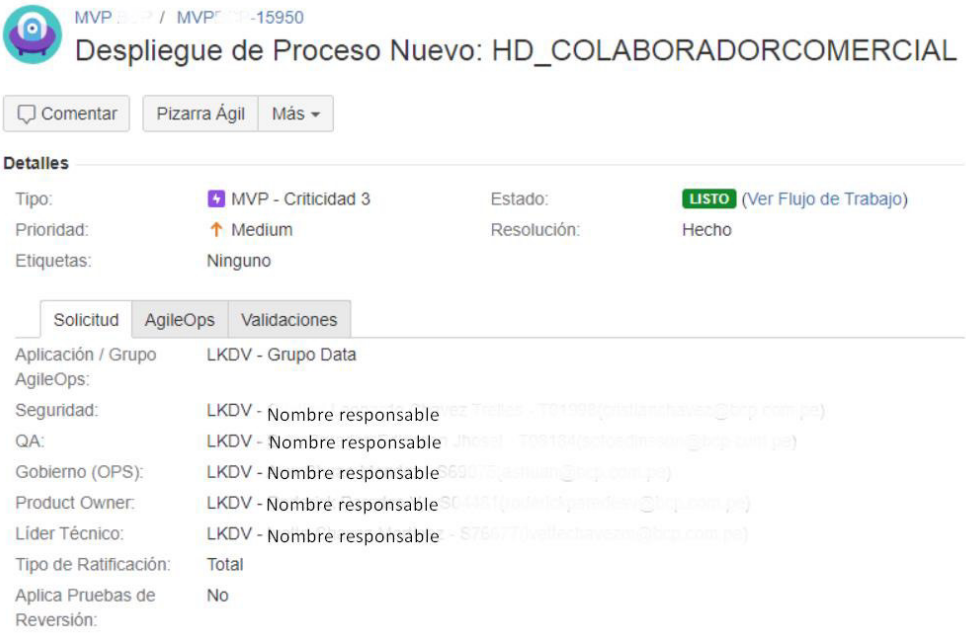


Figura 10. Solicitud registrada en Jira con actores involucrados
Nota: Tomado de fuente interna de la entidad financiera

Una vez realizado el registro en Jira, se inicia con el desarrollo del proceso propiamente dicho.

- **Desarrollo del proceso big data:**

Para el desarrollo propiamente dicho, el desarrollador debe solicitar al Líder técnico la creación de un repositorio en Bitbucket, luego debe completar los entregables y crear un job o pipeline de Jenkins para realizar las pruebas.

- **Creación de repositorio Bitbucket:**

El líder técnico del squad tiene habilitada la opción para crear repositorios en la herramienta, por lo que responde a la demanda del desarrollador. En el momento en el que se creó el repositorio por defecto se generó una rama “Master”, sin embargo, era necesario que el Líder técnico realice la creación de 2 ramas adicionales: la rama “construcción” cuya ejecución y pruebas se realizó en desarrollo y la rama “develop” que sería tomada como base para el congelamiento.

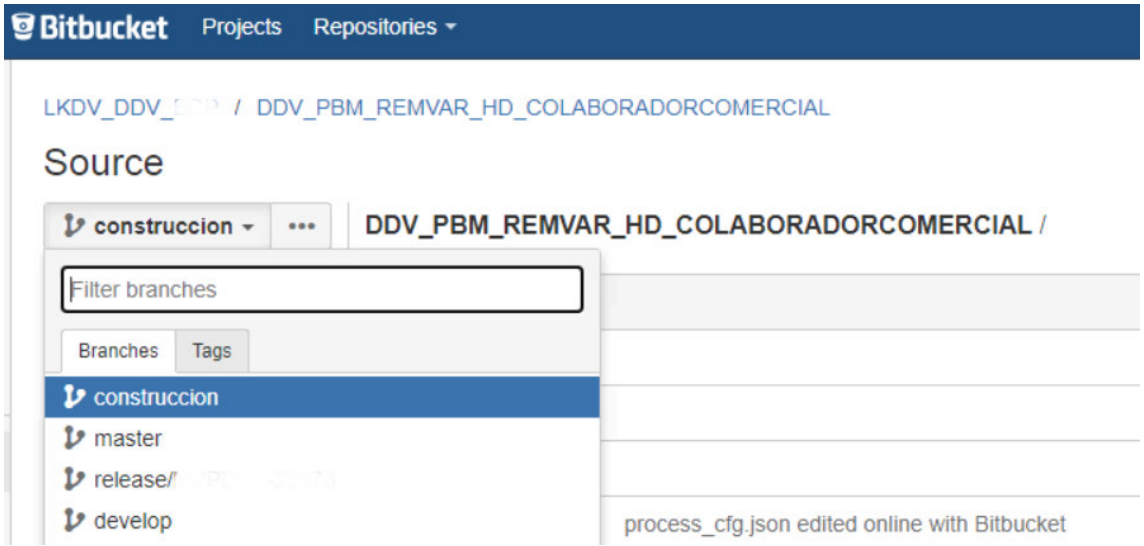


Figura 11. Repositorio Bitbucket y sus ramas
Nota: Tomado de fuente interna de la entidad financiera

- **Creación de Job Jenkins:**

El desarrollador tiene la potestad para crear sus propios pipelines en Jenkins. De acuerdo a los lineamientos de la entidad bancaria, el desarrollador creó 3 jobs: uno para desarrollo, el segundo para certificación y el tercero para producción los cuales van de la mano a las ramas de Bitbucket-.

Como ejemplo, en la siguiente imagen se muestra el Job Jenkins de desarrollo para el proceso DDV:

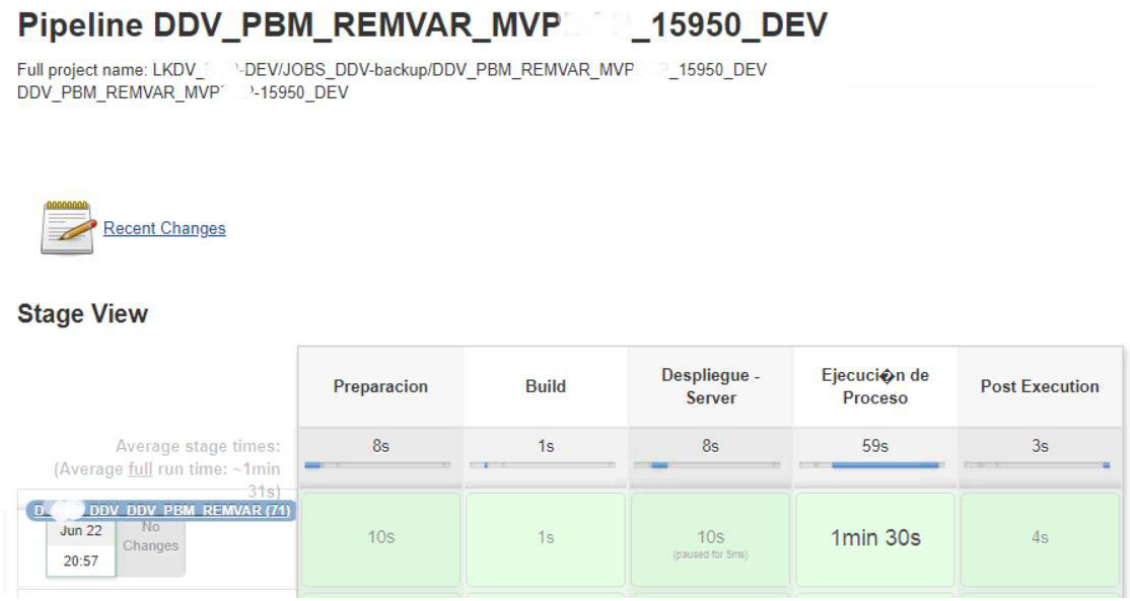


Figura 12. Job Jenkins ejecutado en ambiente de desarrollo
Nota: Tomado de fuente interna de la entidad financiera

Para garantizar el correcto funcionamiento del Job Jenkins, en la parte de la configuración interna se enlazó este con el repositorio Bitbucket a través de la URL del repositorio indicando también la rama que se ejecutará. A continuación, se muestra un gráfico con la configuración interna del pipeline del proceso DDV:

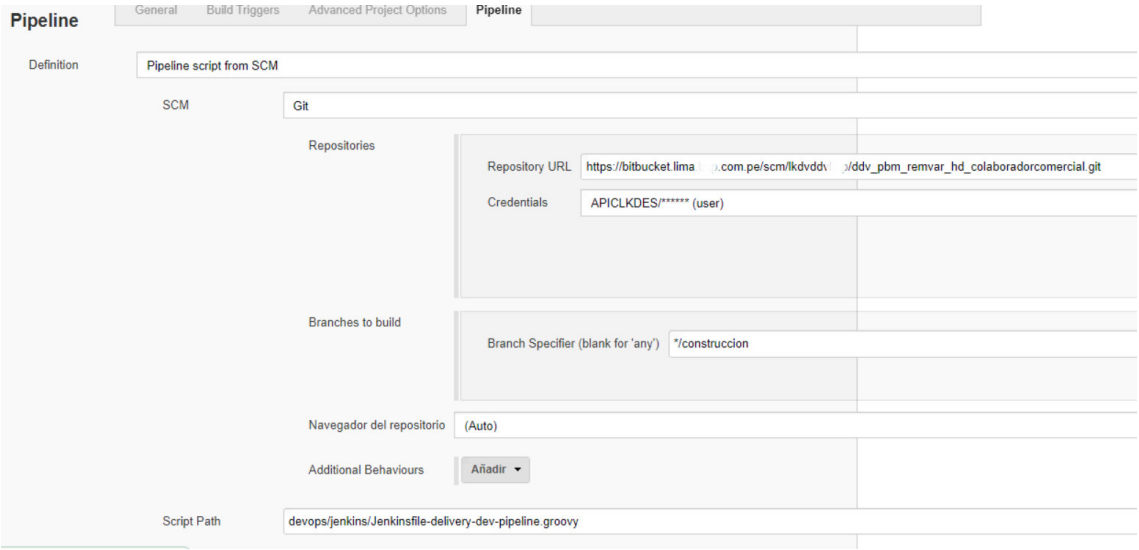


Figura 13. Sección de configuración básica de Jenkins en ambiente de desarrollo

Nota: Tomado de fuente interna de la entidad financiera

- **Entregables:**

Una vez que el desarrollador tuvo enlazado Bitbucket y Jenkins procedió a la construcción de sus entregables entre ellos tenemos:

- DDL para creación de tabla en la capa DDV que contempla la creación de una tabla principal, una tabla de registros rejectados y vistas de las mismas, este entregable lo generó el desarrollador tomando como base una estructura preliminar que compartió el data Modeler.
- Script de parametría para schedulado de proceso con frecuencia diaria y tunning para el motor spark.
- Script pySpark con proceso de ingesta a DDV y adicionalmente se requirió el uso de la librería utilspark.py que nos sirvió para la lectura de recursos necesarios para spark.

Con las primeras versiones de los entregables, el desarrollador procedió a subir los entregables al repositorio, esto se hizo con la ayuda de Git. Los comandos que fueron necesarios para realizar la subida de archivos al repositorio se detallan en el Anexo 1.

Luego de la subida de entregables se actualizaron los repositorios y se observaron los archivos en la rama construcción de la siguiente manera:

Figura 14. Distribución de carpetas en el repositorio Bitbucket
Nota: Tomado de fuente interna de la entidad financiera

Figura 15. Contenido de la carpeta py: Script pySpark y librería utilSpark
Nota: Tomado de fuente interna de la entidad financiera

Figura 16. Contenido de la carpeta scripts: DDL's hive y parametría de schedulado
Nota: Tomado de fuente interna de la entidad financiera

```
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.master', 'yarn');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.app.name', 'HD_COLABORADORCOMERCIAL');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.driver.memory', '10g');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.driver.cores', '1');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.executor.cores', '4');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.executor.memory', '15g');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.dynamicAllocation.enabled', 'true');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.shuffle.service.enabled', 'true');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.dynamicAllocation.maxExecutors', '100');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.dynamicAllocation.minExecutors', '60');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.sql.shuffle.partitions', '200');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.default.parallelism', '80');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.ui.enabled', 'true');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.driver.memoryOverhead', '4g');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.executor.memoryOverhead', '4g');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.debug.maxToStringFields', '100');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.sql.autoBroadcastJoinThreshold', '-1');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.sql.join.preferSortMergeJoin', 'true');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.yarn.queue', 'default');
INSERT INTO ADMIN.CT_PARAMETROPROCESOCARGA VALUES ('CARGA_OCBM_ORG_BCAMIN_DDV','CDDV',1, 'spark.sql.crossJoin.enabled', 'true');
```

Figura 17. Parametría de tunning para motor spark

Nota: Tomado de fuente interna de la entidad financiera

```
CREATE TABLE ${hiveconf:PRM_HIVE_SCH_DDV}_DDV_PBM_REMVAR.HD_COLABORADORCOMERCIAL
(
    CODCLAVEPARTYCOLABORADOR VARCHAR(128) ,
    TIPROLCOLABORADOR CHAR(20) ,
    CODMATRICULA VARCHAR(30) ,
    CODCLAVEPARTYAGE VARCHAR(128) ,
    TIPROLAGE CHAR(20) ,
    CODAGE VARCHAR(30) ,
    NBRAGE VARCHAR(120) ,
    CODCLAVEPARTYAREAUNIDADORGANIZATIVA VARCHAR(128) ,
    TIPROLAREAUNIDADORGANIZATIVA CHAR(20) ,
    CODAREAUNIDADORGANIZATIVA VARCHAR(30) ,
    NBRAREAUNIDADORGANIZATIVA VARCHAR(120) ,
    CODCLAVEPARTYUNIDADORGANIZATIVAREGIONCOMERCIAL VARCHAR(128) ,
    TIPROLUNIDADORGANIZATIVAREGIONCOMERCIAL CHAR(20) ,
    CODUNIDADORGANIZATIVAREGIONCOMERCIAL VARCHAR(30) ,
    NBRUNIDADORGANIZATIVAREGIONCOMERCIAL VARCHAR(120) ,
    CODCLAVEPARTYUNIDADORGANIZATIVAEQUIPOCOMERCIAL VARCHAR(128) ,
    TIPROLUNIDADORGANIZATIVAEQUIPOCOMERCIAL CHAR(20) ,
    CODUNIDADORGANIZATIVAEQUIPOCOMERCIAL VARCHAR(30) ,
    NBRUNIDADORGANIZATIVAEQUIPOCOMERCIAL VARCHAR(120) ,
    CODSECTORISTADUPLA VARCHAR(30) ,
    CODSECTOR VARCHAR(30) ,
    CODCLAVEPARTYCOLABORADORSUPERIOR VARCHAR(128) ,
    TIPROLCOLABORADORSUPERIOR CHAR(20) ,
    CODMATRICULASUPERIOR VARCHAR(30) ,
    CODCANALCOMERCIAL VARCHAR(30) ,
    DESCANALCOMERCIAL VARCHAR(256) ,
    CODGRUPOCANALCOMERCIAL VARCHAR(30) ,
    DESGRUPOCANALCOMERCIAL VARCHAR(256) ,
    TIPNIVELJERARQUIACOMERCIAL CHAR(20) ,
    DESTIPNIVELJERARQUIACOMERCIAL VARCHAR(256) ,
    FLGCOLABORADORTRAINEE CHAR(1) ,
    CODMES INT ,
    FECRUTINA TIMESTAMP ,
    FECACTUALIZACIONREGISTRO TIMESTAMP ,
    TIPFRECUENCIAREGISTRO CHAR(1)
)
PARTITIONED BY (FECDIA STRING)
STORED AS PARQUET
LOCATION '/${hiveconf:PRM_HIVE_AMBIENTE}    /ddv/pbm/remvar/data/HD_COLABORADORCOMERCIAL'
TBLPROPERTIES("parquet.compress"="SNAPPY");
```

Figura 18. DDL Hive parametrizado

Nota: Tomado de fuente interna de la entidad financiera

```
spark.sql("""
    INSERT OVERWRITE TABLE {var:esquemaDdv}.{var:tabla_orgDDV} PARTITION (FECDIA)
    SELECT
        codclavepartycolaborador,
        tiprolcolaborador,
        codmatricula,
        codclavepartyage,
        tiprolage,
        codage,
        nbrage,
        codclavepartyareaunidadorganizativa,
        tiprolareaunidadorganizativa,
        codareaunidadorganizativa,
        nbrareaunidadorganizativa,
        codclavepartyunidadorganizativaregioncomercial,
        tiprolunidadorganizativaregioncomercial,
        codunidadorganizativaregioncomercial,
        nbrunidadorganizativaregioncomercial,
        codclavepartyunidadorganizativaequipocomercial,
        tiprolunidadorganizativaequipocomercial,
        codunidadorganizativaequipocomercial,
        nbrunidadorganizativaequipocomercial,
        codsectoristadupla,
        codsector,
        codclavepartycolaboradorsuperior,
        tiprolcolaboradorsuperior,
        codmatriculasuperior,
        codcanalcomercial,
        descanalcomercial,
        codgrupocanalcomercial,
        NULL AS desgrupocanalcomercial,
        tipniveljerarquiacomercial,
        NULL AS destipniveljerarquiacomercial,
        flgcolaboradortrainee,
        codmes,
        fecrutina,
        fecactualizacionregistro,
        tipfrecuenciaregistro,
        fecdia
    FROM dfOrganicoDDVTmp09""".\
    replace("{var:esquemaDdv}",PRM_SPARK_ESQUEMA_DDV).\
    replace("{var:tabla_orgDDV}",CONS_CARPETA_PROCESO)\
    )
```

Figura 19. Sección de código spark sql parametrizado
Nota: Tomado de fuente interna de la entidad financiera

- **Pruebas:**

Luego de que el desarrollador subió sus entregables a Bitbucket se configuró el archivo “Process_cfg.json”, en este archivó se indicaron los entregables que serían ejecutados y que son los que se mencionaron en el anterior punto. A continuación, se muestra una parte del contenido del archivo de configuración de proceso:

```
"estrategia" :
[
   {"ORACLE" : "INSERT_DML_DDV.sql"},
   {"HIVE" : "HD_COLABORADORCOMERCIAL_DDL.hql"}
],

"PY" :
[
   {"PY" : "HD_COLABORADORCOMERCIAL.py"},
   {"PY" : "utilspark.py"}
]
```

Figura 20. Sección de archivo Process_cdf.json con entregables a ejecutar

Nota: Tomado de fuente interna de la entidad financiera

Además cabe mencionar que los valores que se configuran en este archivo son leídos por otro archivo llamado “jenkinsfile.groovy” el cual contiene las funciones y secuencia de ejecución de entregables, es decir el pipeline como tal. A continuación, se muestra una sección de este archivo donde se observan algunas etapas como son el despliegue server, el despliegue del proceso pyspark propiamente dicho y la notificación de post ejecución:

```
    stage('Despliegue - Server') {

        switch("${TIPO_PASE}") {
            case "NUEVO":
                echo "****************Tipo de PASE : nuevo proyecto ********"
                //utils.variablesDDV("${GLOB_AMBT_COD}")
                utils.kinitSetup()
                utils.executeScriptsDDLaDMLDDV("${GLOB_AMBT_COD}", "pbm/remvar/tmp")
                //utils.copyHQLFilesDDV("${GLOB_AMBT_COD}","pbm/remvar/hql")
                utils.copyPYFilesDDV("${GLOB_AMBT_COD}","pbm/remvar/jar")
                break;
            default:
                break
        }

    }

    stage('Ejecución de Proceso') {
        utils.executeprocessUDV("${GLOB_AMBT_COD}")
    }

    stage('Post Execution') {
         utils.executePostExecutionTasks()
         utils.notifyByMail('SUCCESS',recipients)
    }
}
```

Figura 21. Sección del archivo groovy con funciones propias del pipeline
Nota: Tomado de fuente interna de la entidad financiera

Luego de que el desarrollador realizó todas estas configuraciones, se procedió a ejecutar el Job Jenkins de desarrollo, y de manera natural se encontraron diversas observaciones, es aquí donde se empezó a notar la utilidad de la integración continua ya que los cambios que se necesitaron se realizaron y fueron sincronizados con bitbucket rápidamente. Esto permitió poder realizar ejecuciones con Jenkins de manera continua pudiendo levantar y corregir observaciones de manera eficiente. A continuación, se muestra una imagen de las diversas ejecuciones que se realizaron en Jenkins previas a iniciar la etapa de certificación:

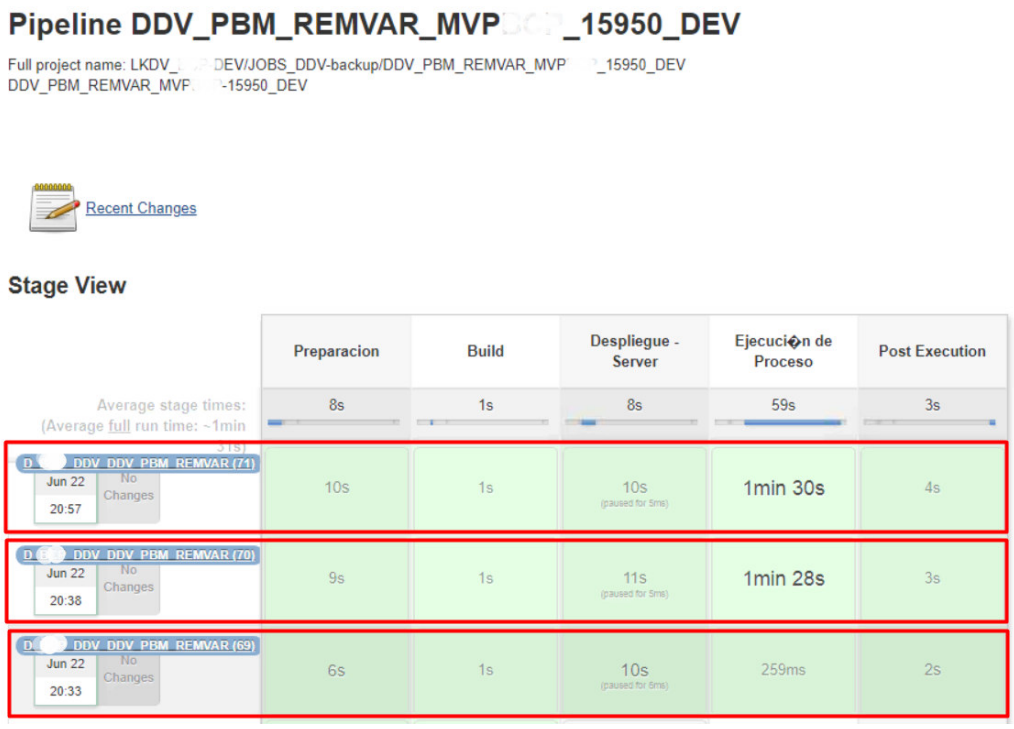


Figura 22. Job jenkins ejecutado a modo de prueba en ambiente desarrollo

Nota: Tomado de fuente interna de la entidad financiera

Jenkins además de mostrarnos esta interfaz gráfica en color verde como proceso exitoso también nos brindó un log mucho más detallado como se muestra en la siguiente imagen:

```
2.1.3.- ( 2021-06-22 20:59:25 ) -> REGISTRO/REPORTE EJECUCIÓN.
       -> OK - Se registro la ejecucion en la tabla

       ------------------------------------------------
       -> PROCESO          : CARGA_OCBM_ORG_BCAMIN_DDV
       -> MODULO           : CDDV
       -> SECUENCIA        : 1
       -> RUTA             : </desa/  /ddv/pbm/remvar/jar/>
       -> JAR              : HD_COLABORADORCOMERCIAL.py
       -> FECHA_RUTINA     : 2021-06-20
       -> INICIO EJECUCION : 2021-06-22 20:58:03
       -> FIN EJECUCION    : 2021-06-22 20:59:25
       -> ESTADO EJECUCION : (Finished OK)
       ------------------------------------------------


------------------------------ FIN DE EJECUCION ---------------------------------

3.- ( 2021-06-22 20:59:26 ) -> GENERANDO LOG DE EJECUCIÓN.
       -> Log de ejecucion OK
       -> Ruta: /desa/  /ctrl/log/CARGA_OCBM_ORG_BCAMIN_DDV_CDDV_2021-06-20_20210622_205759.log
```

Figura 23. Log detallado generado por Jenkins

Nota: Tomado de fuente interna de la entidad financiera

Con estas ejecuciones el desarrollador validó la funcionalidad y calidad de sus entregables.

- **Revisión de pares:**

Cuando el desarrollador dio por concluidas sus pruebas debe solicitar la revisión de todos sus entregables a un team member diferente. Este segundo desarrollador revisó que todos los scripts cumplan con los lineamientos de la entidad bancaria, además validó que el proceso tenga la última ejecución de Jenkins exitosa. También validó el llenado de la solicitud en Jira, así como las instrucciones de despliegue y reversión. Como el segundo desarrollador no encontró observaciones el proceso continuó el flujo Jira pasando a la siguiente etapa: Aprobación técnica.

B. Aprobación técnica

En esta etapa el responsable es el líder técnico del squad, este realizó una validación de los datos colocados en Jira como son: la asignación de roles responsables, los nombres de repositorio Bitbucket y Jenkins así como sus versiones, además validó que cada paso tanto de certificación como de producción cuente con su respectivo paso de reversión. Como todo se encontró conforme, el LT del squad pudo adelantar de carril la solicitud pasando a "Aprobación de seguridad"; en caso se hubiera encontrado alguna observación el LT hubiera tenido que devolver la solicitud al primer carril "Registro de solicitud" donde el desarrollador podría realizar las correcciones necesarias y volver a iniciar el flujo.

C. Aprobación de seguridad

En esta etapa interviene el Analista de Seguridad asignado al despliegue, su función principal fue validar que no se vulneren los lineamientos de seguridad de la entidad financiera. En el caso de los procesos big data las principales validaciones se realizan sobre la creación de usuarios, creación de grupos de red y asignaciones de permisos. Como se cumplieron los lineamientos de seguridad, la solicitud estuvo lista para su congelamiento o certificación; si se hubieran encontrado observaciones el analista de seguridad hubiera devuelto la solicitud al primer carril para que el desarrollador pueda realizar las correcciones necesarias.

D. Congelamiento

En esta etapa intervienen dos actores: el líder técnico del squad y el Agile Operator. Cuando el analista de seguridad promovió el ticket a este carril, por defecto quedó asignado el líder técnico, quien creó la primera subtarea y la asignó al Agile Operator. Una vez que se asignó al Agile OPS este tuvo que seguir todas las indicaciones descritas en la sección “Instrucciones para certificación” del ticket de Jira y mientras fue ejecutándolas guardó las evidencias correspondientes y las subió al tablero jira para su posterior validación, la principal actividad a ejecutar es el pipeline automatizado de Jenkins en ambiente de certificación. En caso hubieran existido más actividades el agile OPS hubiera sido el encargado de crear el resto de subtareas y asignarlas al responsable correspondiente. Como todas las actividades culminaron exitosamente, el agile OPS promovió la solicitud al siguiente carril que es “Pruebas QA”. Si se hubiera encontrado alguna observación en las subtareas de certificación el agile OPS hubiera tenido que ejecutar las instrucciones de reversión en certificación y luego enviar el ticket al primer carril de registro de solicitud.

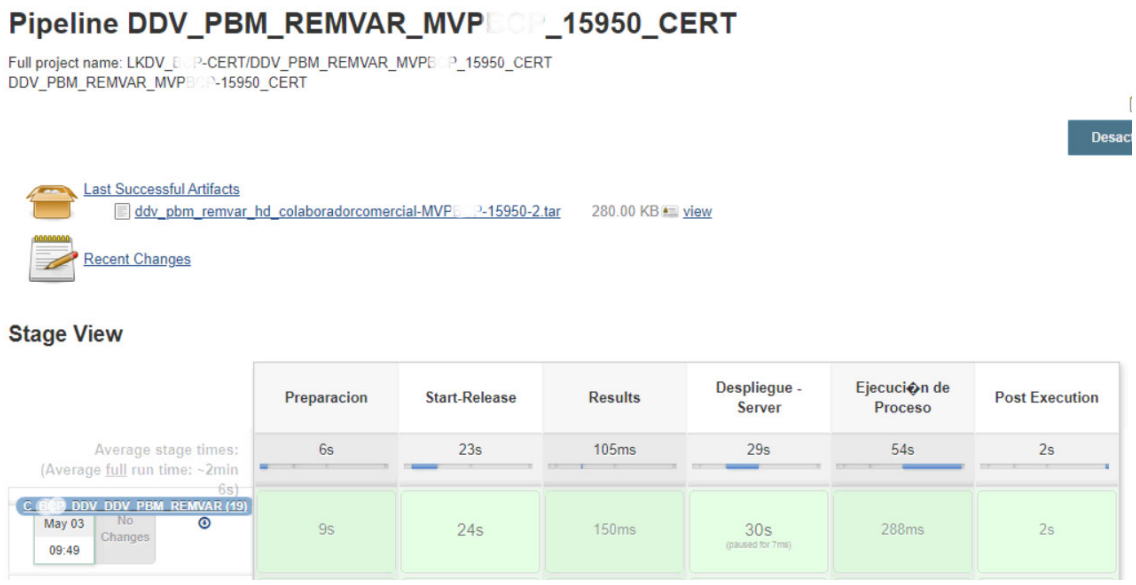


Figura 24.Ejecución de Job Jenkins en ambiente de certificación
Nota: Tomado de fuente interna de la entidad financiera

E. Pruebas QA

En este carril el actor es el Analista de Calidad, el debió revisar todas las evidencias adjuntas en Jira así como el log que fue generado al ejecutar el pipeline de Jenkins. Como todas las evidencias fueron correctas de acuerdo a lo esperado el Analista de Calidad del Squad ratificó exitosamente el congelamiento y promovió el despliegue al carril “Operaciones”, si se hubiera encontrado alguna observación se hubiera tenido que revertir el congelamiento y devolver la solicitud al carril inicial.

Figura 25. Log detallado de Jenkins en ambiente de certificación
Nota: Tomado de fuente interna de la entidad financiera

F. Operaciones

En este carril se solicita el calendario. El actor asignado es el líder técnico y el fue el encargado de solicitar el calendario para la ejecución del pase a producción. Una vez que el líder técnico encontró una fecha disponible solicitó la aprobación al equipo de Operaciones mediante un correo para la validación de la disponibilidad y capacity del equipo. Con el conforme de operaciones, el líder técnico promovió el despliegue al carril de “Pase a producción” donde se mantendrá hasta que llegue la fecha seleccionada en el calendario.

G. Pase a producción

Una vez que llegó el día y la hora establecida para el pase a producción el Agile Operator ejecutó todas las instrucciones que fueron detalladas en la sección “Instrucciones para producción” del tablero Jira. Para realizar estas ejecuciones el Agile Operator creó subtareas y las fue asignando a los responsables correspondientes y cada en cada una de estas subtareas el agile ops guardó las evidencias y los logs generados para su posterior revisión. La actividad más importante en esta etapa es la ejecución del pipeline de Jenkins en producción. Como el pase a producción resultó exitoso, el agile OPS promovió el despliegue al carril de Ratificación, en caso hubiera surgido algún inconveniente hubiera tenido que ser promovido al carril de Reversión.

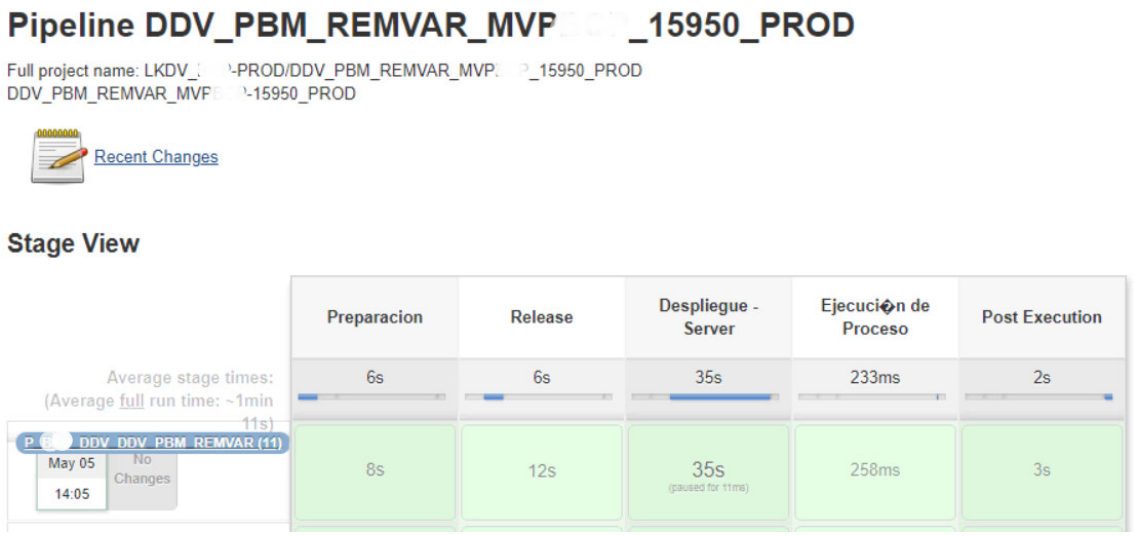


Figura 26. Ejecución de Job jenkins en ambiente productivo
Nota: Tomado de fuente interna de la entidad financiera

H. Ratificación

En esta etapa intervino el analista de calidad y este realizó las validaciones de los cambios desplegados en producción en base a las evidencias que se adjuntaron en el paso previo. Como las validaciones fueron exitosas el analista de calidad promovió el despliegue al carril de cierre. Si se hubiera encontrado alguna observación se hubiera tenido que realizar la reversión del pase y luego enviarlo al carril inicial para las correcciones necesarias.

I. Reversión

Esta etapa no fue necesaria en el despliegue ya que el pase a producción fue exitoso.

J. Cierre

Este es el carril final y es aquí donde llegó el despliegue ya que culminó todas las etapas exitosamente

3.2.5.4. Comparación del tiempo de despliegue

Tomando como referencia el mismo despliegue que se utilizó para describir las secciones anteriores, se muestra en el grafico 27 una comparación del tiempo que demoró realizar el despliegue utilizando el flujo con integración continua versus el flujo tradicional.

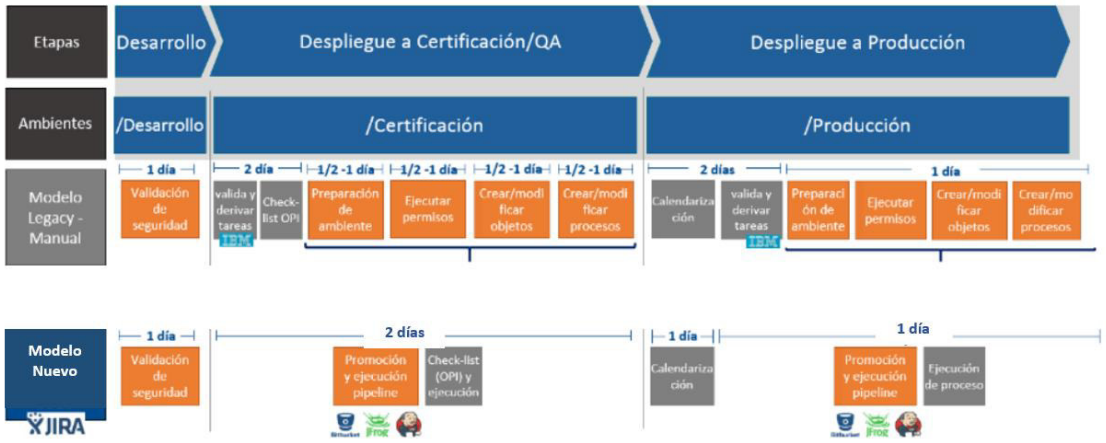


Figura 27. Flujo comparativo de tiempos de despliegue entre modelo tradicional vs. Modelo Nuevo
Nota: Tomado de fuente interna de la entidad financiera

Como se puede observar, hubo una reducción de 8 a 5 días lo que significa una reducción del 37.5% del tiempo del despliegue. Con ello observamos el principal beneficio de la integración continua y uso de la Metodología Kanban que es la reducción del tiempo de despliegue lo que permite una mayor eficiencia operativa. La figura 27 se detalla en el anexo2.

3.3. EVALUACION ECONÓMICA

En la evaluación económica consideraremos el análisis de costos y los beneficios:

3.3.1 ANÁLISIS DE LOS COSTOS:

Para realizar esta evaluación se tomará en cuenta el pago mensual a los roles participantes del piloto durante un trimestre. Esto se muestra en la tabla x

Tabla 11

Análisis de costos del piloto en el 2do trimestre

Rol	Abril	Mayo	Junio	Total
Product Owner	9,000	9,000	9,000	27,000
Data Engineer - LT	4,000	4,000	4,000	12,000
Data Engineer - QA	4,000	4,000	4,000	12,000
Data Engineer - Dev	4,000	4,000	4,000	12,000
Agile Operator	5,000	5,000	5,000	15,000
Analista de seguridad	5,000	5,000	5,000	15,000
				93,000

Nota. Elaboración propia

3.3.2 BENEFICIOS PARA LA ORGANIZACIÓN:

Como uno de los principales beneficios que trajo el nuevo flujo usando integración continúa y metodología Kanban a la organización se encuentra la reducción en los tiempos de despliegue lo que permite cumplir con la alta demanda de pases a producción que experimenta la entidad financiera.

Adicionalmente hay otros beneficios que se derivan también de este nuevo flujo como son la centralización de entregables de pase a producción ahora se encuentra organizada en repositorios Bitbucket.

También cabe mencionar que gracias a Jenkins se consiguió un nivel de automatización importante, lo que favoreció las pruebas en desarrollo, así como los despliegues en certificación y producción.

Finalmente, este flujo resulta mucho más sencillo de comprender por lo que al ingresar nuevos team members la adopción del flujo sería mucho más rápida.

CAPÍTULO IV
REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

El uso del flujo de integración continua bajo la metodología Kanban trajo muchos beneficios a la organización como se mencionó anteriormente, sin embargo, aun hay procesos que no se encuentran automatizados en Jenkins por lo que sería muy importante continuar con la automatización del resto de procesos para lograr obtener el máximo beneficio de las herramientas.

El piloto inició con algunos squads en la División de Data & Analytics, sin embargo, sería recomendable que este uso se extienda a todos los equipos de la División e incluso a otras áreas de la entidad financiera ya que no solo puede ser aplicada a big data sino también a otro tipo de procesos como son los SQL que se usan en muchos equipos de la entidad financiera.

Adicionalmente, podemos comentar que a pesar que la capacitación se dio a los roles LT del squad y luego a los otros team members, hubiera sido más beneficioso brindar una mayor capacitación a los otros team members para que conozcan y aprovechen mucho más el flujo.

CONCLUSIONES

1. Se logró reducir el tiempo de despliegue de los procesos de big data haciendo uso de las herramientas de integración continua y la metodología Kanban en la entidad bancaria. Gracias a la metodología se consiguió la visualización de todas las etapas del despliegue de los procesos de big data facilitando la identificación de los cuellos de botella.
2. Los pases a producción de los procesos de big data culminaron con una cantidad reducida de errores debido al uso de repositorios integrados (Bitbucket) que facilitaron la revisión de pares y la actualización de versiones.
3. Los tiempos de desarrollo y pruebas de procesos de big data disminuyeron gracias al uso de pipelines automatizados mediante Jenkins, esto debido a que las pruebas se realizan desde el desarrollo lo que permite detectar observaciones rápidamente.
4. Debido a la facilidad de la adopción de las herramientas de integración continua los tiempos para capacitar a nuevos roles son menores logrando que los despliegues se realicen en un periodo más corto de aproximadamente 37,5%.

RECOMENDACIONES

1. Se recomienda expandir el uso del flujo de integración continua bajo el enfoque de la metodología Kanban a otras áreas de la entidad financiera ya que es adaptable a otros tipos de proceso y no solo de big data.
2. Para obtener más beneficios es aconsejable seguir automatizando más procesos para que los equipos puedan utilizar este flujo en el 100% de los despliegues.
3. En base a la experiencia, sería beneficioso brindar una capacitación más detallada a los analistas de calidad de cada squad a fin de que puedan detectar los errores en los despliegues mucho más rápido.
4. Para mantener un ordenamiento en la creación de los repositorios y pipelines de Jenkins se recomienda que todo squad siga los lineamientos de nomenclatura de manera obligatoria ya que esto permite identificar a que tipo de proceso se hace referencia.

REFERENCIAS BIBLIOGRAFICAS

- Atlassian. (Noviembre de 2021). *¿Para qué sirve Jira?* Obtenido de https://www.atlassian.com/es/software/jira/guides/use-cases/what-is-jira-used-for
- Atlassian. (Noviembre de 2021). *Bitbucket*. Obtenido de https://www.atlassian.com/es/software/bitbucket
- AWS, A. W. (11 de 2021). *¿Qué es la integración continua?* Obtenido de https://aws.amazon.com/es/devops/continuous-integration/
- Git. (Noviembre de 2021). *Git*. Obtenido de https://git-scm.com/
- Hofmann, C., Lauber, S., Haefner, B., & Lanza, G. (2018). Development of an agile development method based on Kanban for distributed part-time teams and an introduction framework. *Procedia Manufacturing*, 45-50.
- Kumar Bhadani, A., & Jothimani, D. (2017). Big Data: Challenges, Opportunities and Realities. En *Effective Big Data Management and Opportunities for Implementation.*
- Miloslavskaya, N., & Tolstoy, A. (2016). Big Data, Fast Data and Data Lake Concepts. *Procedia Computer Science*, 300-305.
- Sentrio. (16 de Setiembre de 2021). *Introducción a Jenkins*. Obtenido de https://sentrio.io/blog/que-es-jenkins/
- Snaplogic. (Noviembre de 2021). *How to get valuable insights on data stored in Azure Data Lake Store*. Obtenido de https://www.snaplogic.com/blog/valuable-insights-on-data-stored-in-azure-data-lake-store
- Turner, R., Ingold, D., Lane, J. A., Ray, M., & Andersond, D. (2012). Effectiveness of kanban approaches in systems engineering within rapid response environments. *Procedia Computer Science*, 309-314.

GLOSARIO

- **BACKLOG**: Son las actividades pendientes que se deben realizar para cubrir la necesidad del negocio, el backlog puede ir variando según nuevas necesidades del negocio.
- **CAPACITY**: Es la capacidad en horas / recursos disponibles del equipo para atender requerimientos.
- **COMMIT**: Es el código con el que se puede identificar la última versión de archivos del repositorio, se usa para indicar que el desarrollo concluyó y se puede iniciar el congelamiento.
- **HISTORIA DE USUARIO**: Es una de las actividades que componen el backlog, generalmente una historia de usuario es tomada por un desarrollador hasta llevar un producto a producción.
- **LINEAMIENTO**: Son normas que se deben aplicar a los desarrollos de manera obligatoria en la entidad bancaria.
- **LOG**: Es un archivo que contiene las evidencias de los procesos ejecutados.
- **SQUAD**: Es un equipo de la entidad financiera que trabaja bajo una metodología ágil.
- **WIP**: en ingles es “Work in Progress”, se refiere a la cantidad de tareas que se pueden atender en un carril del tablero Jira con enfoque Kanban, esto dependerá del capacity del equipo y permite identificar cuellos de botella rápidamente.

ANEXOS

ANEXO 1 – Comandos Git

- Comando para la clonación del repositorio a la máquina local:

  GIT_SSL_NO_VERIFY=true git clone “URL_REPOSITORIO”

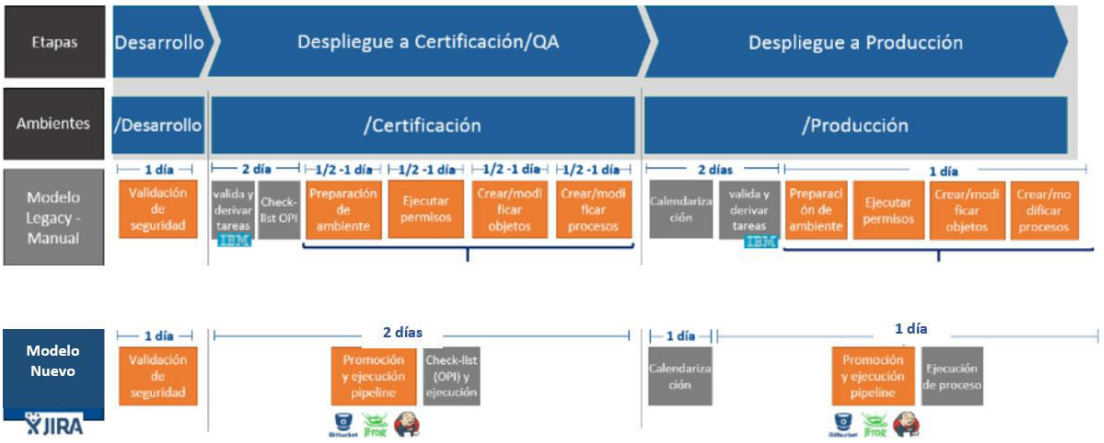
- Ubicarnos en la rama de “construcción”

  cd “NOMBE_REPOSITORIO”
  git checkout "construccion"

- Subir los cambios al repositorio remoto Bitbucket

  git add .
  git commit -m "update"
  GIT_SSL_NO_VERIFY=true git push

ANEXO 2 – Comparación de flujos



En el grafico se muestra la comparación de tiempos de despliegues entre el flujo tradicional de la entidad financiera versus el nuevo flujo usando la integración continua.

Como se puede observar en la etapa de desarrollo se consideran en ambos casos la actividad de validación de seguridad la cual se mantuvo en tiempos ya que el analista de seguridad debe realizar las mismas validaciones en ambos casos. En el caso de la etapa de congelamiento se reduce de entre 4 a 5 días hasta solo 2 días, la diferencia principal es que en el flujo tradicional todas las actividades se realizaban de manera independiente y manual y en el nuevo flujo solo se ejecuta Jenkins de forma automatizada y con ello se puede desplegar el cambio completo.

Finalmente, en producción, se redujo de 3 a 2 días, esto de manera similar al congelamiento ya que todo el despliegue en ambiente productivo es automatizado mientras que en el flujo normal se tomaba más tiempo ya que debían derivarse tareas manuales a diferentes responsables.

Como se observa, la ventaja principal de la integración continua es la automatización en las ejecuciones parametrizando las variables por ambiente.