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

Implementación de un modelo de datos para la identificación de potenciales clientes y optimización del spread comercial en el negocio de cambio de divisas en un entorno Big Data de una entidad bancaria

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Justo Daniel Marlow AYRAS OLANO

ASESOR

César Augusto ALCÁNTARA LOAYZA

Lima, Perú

2021

Referencia bibliográfica

Ayras, J. (2021). *Implementación de un modelo de datos para la identificación de potenciales clientes y optimización del spread comercial en el negocio de cambio de divisas en un entorno Big Data de una entidad bancaria.* [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 autor/ asesor

Datos de autor	
Nombres y apellidos	JUSTO DANIEL MARLOW AYRAS OLANO
Tipo de documento de identidad	DNI
Número de documento de identidad	47322393
URL de ORCID	https://orcid.org/0000-0002-7012-6675
Datos de asesor	
Nombres y apellidos	César Augusto Alcántara Loayza
Tipo de documento de identidad	DNI
Número de documento de identidad	09132297
URL de ORCID	https://orcid.org/0000-0003-3435-4555
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Augusto Parcemón Cortez Vásquez
Tipo de documento	DNI
Número de documento de identidad	08634618
Miembro del jurado 1	
Nombres y apellidos	Joel Fernando Machado Vicente
Tipo de documento	DNI
Número de documento de identidad	40476778
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 19:00 horas del día 21 de diciembre del año 2021, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Cortez Vásquez Augusto Parcemón (Presidente), Mg. Machado Vicente Joel Fernando (Miembro) y el Mg. Alcántara Loayza Cesar Augusto (Miembro Asesor), usando la plataforma Meet (https://meet.google.com/jjy-yahj-fza), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: ***"IMPLEMENTACIÓN DE UN MODELO DE DATOS PARA LA IDENTIFICACIÓN DE POTENCIALES CLIENTES Y OPTIMIZACIÓN DEL SPREAD COMERCIAL EN EL NEGOCIO DE CAMBIO DE DIVISAS EN UN ENTORNO BIG DATA DE UNA ENTIDAD BANCARIA",*** *por el Bachiller* ***Ayras Olano Justo Daniel Marlow;*** *para obtener el Título Profesional de Ingeniero de Sistemas.*

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

El 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, el Bachiller obtuvo la nota de ***18 DIECIOCHO.***

A continuación el Presidente de Jurados el Mg. Cortez Vásquez Augusto Parcemón, declara al Bachiller ***Ingeniero de Sistemas****.*

Siendo las 19:55 horas, se levantó la sesión.

Presidente
Mg. Cortez Vásquez Augusto Parcemon

Miembro
Mg. Machado Vicente Joel Fernando

Miembro Asesor
Mg. Alcántara Loayza Cesar Augusto

DEDICATORIA

Este trabajo se lo dedico a mis padres por todo su apoyo brindado para lograr mis objetivos profesionales.

A mis abuelos por sus consejos y apoyo en todo momento.

A la mujer de mi vida, Leslie, siempre me inspira a seguir creciendo como persona y profesional.

AGRADECIMIENTOS

Agradezco a Dios por siempre estar conmigo, por darme fortaleza para seguir adelante en todo tiempo.

A mi familia, por siempre apoyarme y confiar en mis sueños y anhelos.

A la comisión de titulación por gestionar de manera impecable el programa de suficiencia profesional y brindar la oportunidad de dar a conocer nuestra experiencia profesional.

A mi asesor, Cesar Alcántara, por su guía en la elaboración del presente informe.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

IMPLEMENTACIÓN DE UN MODELO DE DATOS PARA LA IDENTIFICACIÓN DE POTENCIALES CLIENTES Y OPTIMIZACIÓN DEL SPREAD COMERCIAL EN EL NEGOCIO DE CAMBIO DE DIVISAS EN UN ENTORNO BIG DATA DE UNA ENTIDAD BANCARIA

Autor:	Ayras Olano, Justo Daniel Marlow
Asesor:	Alcántara Loayza, Cesar Augusto
Título:	Trabajo de Suficiencia Profesional
Fecha:	Diciembre del 2021

RESUMEN

El presente trabajo de suficiencia profesional se concentra en la implementación de un modelo de datos en el entorno Big Data de una entidad bancaria llevado a cabo en el año 2021. En el 2020 muchos de los clientes de la entidad bancaria preferían hacer sus operaciones de cambio de divisas en otras instituciones financieras, ante esta situación se identificó una oportunidad de incrementar las ganancias en este negocio a través de la explotación de la información de los productos que consumen los clientes. El objetivo del proyecto fue incrementar la rentabilidad del negocio de compra y venta de divisas, identificando a potenciales clientes que requerían realizar ese tipo de operaciones y optimizando el spread comercial. La construcción del modelo se realizó utilizando el marco de trabajo Scrum y se usaron las tecnologías Spark en su versión 2.4 y Python en su versión 3 para los procesamientos de datos. La entidad bancaria tuvo un incremento de clientes en un 3.38% posterior a la implementación del modelo, así mismo se identificó que la rentabilidad del negocio se incrementó en un 4.57% en sus operaciones de cambio de divisas.

Palabras claves: Divisa, Big Data, Data Lake, Spark, Modelo de datos, Potenciales clientes.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

IMPLEMENTATION OF A DATA MODEL FOR THE IDENTIFICATION OF POTENTIAL CUSTOMERS AND OPTIMIZATION OF THE COMMERCIAL SPREAD IN THE FOREIGN EXCHANGE BUSINESS IN A BIG DATA ENVIRONMENT OF A BANKING INSTITUTION

Author:	Ayras Olano, Justo Daniel Marlow
Advisor:	Alcántara Loayza, Cesar Augusto
Title:	Professional Sufficiency Work
Date:	December 2021

ABSTRACT

The present work of professional sufficiency is focused on the implementation of a data model in the Big Data environment of a banking entity carried out in the year 2021. In 2020 many of the bank's customers preferred to make their foreign exchange transactions in other financial institutions, in this situation an opportunity was identified to increase profits in this business through the exploitation of the information of the products consumed by customers. The objective of the project was to increase the profitability of the business of buying and selling foreign exchange, identifying potential clients who required to perform this type of operations and optimizing the commercial spread. The model was built using the Scrum framework and Spark 2.4 and Python version 3 technologies were used for data processing. The bank had a 3.38% increase in customers after the implementation of the model, and it was also identified that the profitability of the business increased by 4.57% in its foreign exchange operations.

Keywords: Forex, Big Data, Data Lake, Spark, Data Model, Potential customers.

INTRODUCCIÓN

En el presente informe se detalla la implementación de un modelo de datos en una entidad bancaria en Perú con el objetivo de incrementar la rentabilidad del negocio de cambio de divisas mediante la identificación de potenciales clientes que requieran realizar este tipo de operaciones y la optimización del spread comercial en las operaciones.

La creación del modelo se realizó en el entorno big data de la entidad bancaria la cual cuenta con un datalake desplegado sobre Hadoop como sistema de archivos distribuidos, las tecnologías big data que se usaron fueron Spark y Python, estas permitieron explotar la información de los servicios financieros de los clientes de la entidad bancaria.

En el CAPÍTULO I se detalla la trayectoria profesional del autor que principalmente se centra en proyectos de big data en entidades financieras y telecomunicaciones, adicionalmente se describen las formaciones académicas, cursos y conocimientos adquiridos durante su experiencia profesional.

En el CAPÍTULO II se detalla la empresa donde el autor brindó sus servicios para el desarrollo del proyecto en el cliente (entidad bancaria), se describen aspectos de la empresa como su visión, misión, organigrama y la experiencia del autor en la empresa.

En el CAPÍTULO III se detalla el problema y la solución en el cliente de la empresa donde laboró el autor, se describen las etapas realizadas durante el proyecto, la metodología utilizada y el beneficio del proyecto en el cliente.

En el CAPÍTULO IV se describe la reflexión crítica del autor sobre su experiencia y los logros conseguidos luego de haber participado en la implementación de la solución en la entidad bancaria.

En el CAPÍTULO V se mencionan las conclusiones y recomendaciones del autor relacionados al proyecto desarrollado.

CAPÍTULO I
TRAYECTORIA PROFESIONAL

El autor posee el grado de Bachiller en la carrera de Ingeniería de Sistemas de la Universidad Nacional Mayor de San Marcos, los últimos 5 años se ha dedicado a la explotación de bases de datos en el marco de trabajo Big Data en proyectos con metodologías ágiles (Scrum, Kanban) en el sector Telecomunicaciones y Banca. Actualmente ocupa el cargo de Big Data Engineer Ssr. en Indra Perú, contribuye con la adopción de lineamientos, nuevas tecnologías, estándares de desarrollo y en el asesoramiento sobre la factibilidad técnica de soluciones, es responsable del procesamiento de datos mediante arquetipos como Spark-Scala, PySpark-Scaffolder, así mismo es integrante del equipo de coaching de Indra Perú para temas de procesamientos de datos.

Ha ocupado distintos cargos enfocados en el análisis y desarrollo de reglas de transformación para la ingesta de datos (Oracle, DataStage, Hive, Spark, Python, Scala), elaboración de dashboards para la toma de decisiones (QlikSense), soporte técnico antes incidencias de las soluciones productivas y planificación y seguimiento de tareas (Jira[1]).

Adicionalmente, cuenta con habilidades blandas para el trabajo en equipo con una excelente comunicación asertiva, siempre orientado a resultados y pensamiento analítico.

El autor tiene experiencia profesional principalmente en dos empresas de consultoría de Tecnologías de Información con participación en clientes del sector Bancario y Telecomunicaciones, se detalla la experiencia del autor en la tabla 1.

***Tabla 1*: Experiencia profesional**

Empresa	Área	Cargo	Fecha
Indra Perú	Tecnologías Avanzadas	Big Data Engineer Ssr.	Abril 2021 - Actualidad

Funciones:

- Analizar y procesar datos con Spark, Python y Scala.
- Proponer y desarrollar soluciones para el procesamiento de datos.

[1] Jira es una herramienta online ayuda a la administración de tareas de un proyecto.

- Planificar y dar seguimiento de las tareas en el marco de la metodología ágil Scrum.
- Capacitar a los Data Engineer del equipo sobre tecnologías Big Data.
- Elaborar diccionarios de datos.
- Crear mallas para la ejecución automática de procesos batch.

Everis Perú	IT Solutions & Services	Big Data Engineer	Noviembre 2019 – Marzo 2021

Funciones:

- Coordinar con las mesas de trabajo los lineamientos de la arquitectura Tecnológica de Datos para su cumplimiento, alineadas a la visión estratégica del banco.
- Planificar y dar seguimiento a las tareas en el marco de la metodología Scrum.
- Aplicar el uso de herramientas del ecosistema Hadoop (HDFS, MapReduce, Hue, Impala, Hive).
- Construir procesos de carga de información al Data Lake usando Hive, Impala, Datastage, PySpark y QlikSense.
- Interpretar procesos de ingesta en Oracle para el desarrollo de procesos ETL en el Datalake mediante Hive, Spark, Datastage, etc.
- Elaborar documentos técnicos (alcance, mapeo, trazabilidad, formato de pases a producción) de los procesos desarrollados.

Everis Perú	IT Solutions & Services	Solutions Assistant DataStage	Abril 2019 – Octubre 2019

Funciones:

- Analizar y desarrollar reglas de transformación para la posterior creación de los Parallel y Sequence Jobs de los procesos de ETL en DataStage.
- Coordinar con los analistas funcionales para definir las reglas de negocio aplicados en los procesamientos de datos.
- Optimizar el tiempo de ejecución y el uso de recursos en los Parallel y Sequence Jobs.
- Definir las reglas de validación de la integridad de data de los datasets entregados al usuario final.

Everis Perú	IT Solutions & Services	Solutions Assistant Business Intelligence	Febrero 2018 – Marzo 2019

Funciones:

- Coordinar con los usuarios finales para definición de las reglas de negocio de las aplicaciones.
- Diseñar y desarrollar informes y dashboards a nivel gerencial, táctico y operativo a través de QlikView y QlikSense.
- Analizar y modelar la información presente en base de datos SQL Server (Encuestas, Averías, Reclamos, Clientes, Ventas).
- Configurar y programar tareas de ejecución de procesos automáticos en el servidor cloud.

***Fuente*. Elaboración propia**

A continuación, en la tabla 2 se detalla la formación académica profesional del autor:

***Tabla 2*: Formación académica profesional**

Formación	Institución	Periodo
Bachiller en Ingeniería de Sistemas	Universidad Nacional Mayor de San Marcos – Facultad de Ingeniería de Sistemas e informática – Escuela Académico Profesional de Ingeniería de Sistemas	2012 - 2017

***Fuente*. Elaboración propia**

Dentro de la formación académica complementaria, el autor llevó a cabo programas y cursos que se detallan en la tabla 3.

Tabla 3: Formación académica complementaria

Formación recibida	Institución que acredita	Horas de estudio	Año de estudio
Ética y Competencia 2021	Indra Perú	1	2021
Scrum y Management 3.0 – Gestión de proyectos de transformación digital	Gesap	44	2019
Programa de Especialización en Big Data	Big Data Academy Perú	48	2018
Business Intelligence SQL Server 2016	Cibertec	80	2017
Agente de Innovación Programa San Marcos Innova 2° Generación	Universidad Nacional Mayor de San Marcos	44	2017
Idioma Inglés fase básica	Asociación Cultural Peruana Británica	ND	2016 - 2017

***Fuente*. Elaboración propia**

Entre otras capacidades que el autor ha ido adquiriendo a lo largo de su trayectoria profesional se encuentran las siguientes:

***Tabla 4*: Otras capacidades**

Herramientas de tecnologías de información	
Tecnologías Big Data	Apache Hadoop, Hive, Impala, Spark, Scala, PySpark, HBase, Phoenix, Kafka, IBM InfoSphere DataStage
Lenguajes de programación	Java, C++, Visual Basic
Bases de datos	Oracle, SQL Server, MySql, Postgres
Gestores de bases de datos	SQL Developer, PL SQL Developer
Metodologías	RUP, Scrum, Kanban
Herramientas de Modelado	Rational Rose, Bizagi
Herramientas de desarrollo integrado	Netbeans, IntelliJ IDEA
Herramientas de integración continua (IC)	Jira, Bitbucket, Jenkins
Herramientas BI	Power BI, Qlik Sense
Sistemas Operativos	Windows, Linux

***Fuente.* Elaboración propia**

CAPÍTULO II
CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA

2.1. Empresa – actividad que realiza

Indra es una empresa de consultoría que brinda servicios de tecnología y se caracteriza por ser socio estratégico de sus clientes, entre sus principales segmentos de mercado se encuentra transporte y defensa, transformación digital y tecnologías de información.

"Su modelo de negocio está basado en una oferta integral de productos propios, con un enfoque end-to-end, de alto valor y con un elevado componente de innovación" (Indra, 2021).

Entre los principales clientes de Indra Perú se encuentran los siguientes:

***Tabla 5*: Clientes de Indra Perú**

Servicio	Cliente
Aplicaciones	ONP Clínica Ricardo Palma
Energía & Utilities	Enel Repsol
Servicios Financieros	BBVA BCP Mapfre Rimac
Telecom & Media	Telefónica Entel
Soluciones de Gestión Empresarial	Confipetrol
Transporte & Tráfico	CORPAC

***Fuente.* Elaboración propia**

Datos de la empresa Indra, sede Perú:

- RUC: 20100123411
- Razón Social: INDRA PERU S.A.
- Página Web: https://www.indracompany.com
- Nombre Comercial: Indra

- Tipo Empresa: Sociedad Anónima
- Actividades Comerciales: Servicios y tecnologías de la información

2.2. Visión

"Ser una empresa innovadora y del conocimiento en las relaciones con nuestros públicos internos y externos (accionistas, empleados, clientes, etc.), así como con las instituciones que lo cultivan y desarrollan, y con las comunidades en las que actuamos" (Indra, 2021)

2.3. Misión

De acuerdo a la misión de Indra (2021), nos manifiesta lo siguiente:

Gestionar el conocimiento y la innovación que puede llegar a desplegar con sus clientes es el foco principal de Indra, ofrecer una oferta de valor que incluye desde la consultoría, implementar proyectos y la integración de sistemas y aplicaciones y el outsourcing de sistemas de información, la oferta se compone de dos pilares importantes: Soluciones y Servicios.

2.4. Organización de la empresa

A continuación, se presenta el organigrama de la empresa:

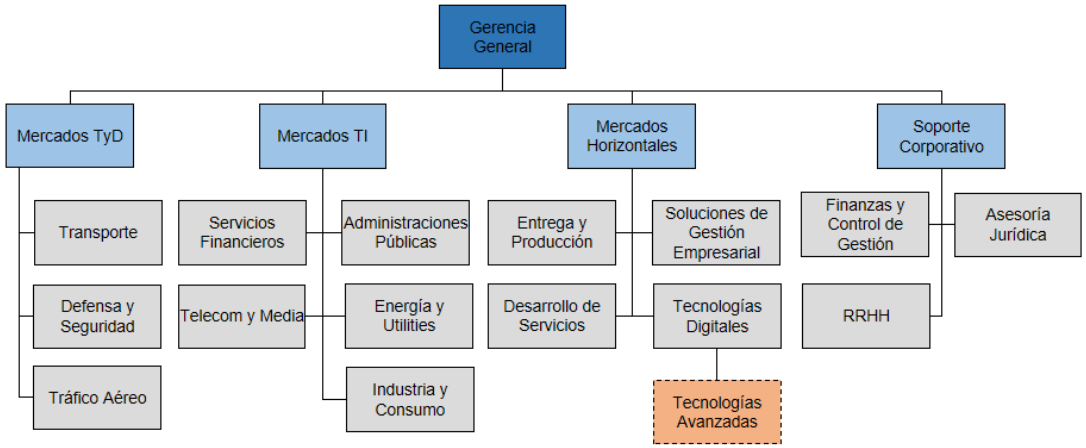


***Figura 1*: Organigrama de la empresa Indra Perú. Elaboración propia**

2.5. Área, cargo y funciones desempeñadas

El autor actualmente tiene el cargo de Big Data Engineer Ssr. en el área de Tecnologías Avanzadas en Indra Perú para un cliente del sector Bancario, desde el 29 de marzo del 2021 hasta la actualidad.

Entre las funciones que desempeña el autor se encuentran las siguientes:

- Entender y acompañar en el levantamiento de los requerimientos o necesidades del mercado de datos (técnicos y funcionales).
- Realizar el desarrollo y/o consultoría Big Data en proyectos punteros y en el área de Tecnologías avanzadas, realizando implantaciones e integraciones.
- Participar en todas las etapas del ciclo de vida del dato: Análisis y comprensión del negocio bancario, exploración de datos, normalización y limpieza, diseñar procesos de ingesta del dato, tanto el disponible en fuentes internas como el capturado externamente, desarrollo de componentes analíticos, tanto en entornos informacionales como transaccionales.
- Trabajar en equipo para el desarrollo de proyectos de desarrollo de componentes analíticos y de tratamiento de la información.
- Apoyar y ejecutar la habilitación de herramientas y componentes de data. (estandarizar, definir su uso y lineamientos necesarios).

2.6. Experiencia profesional realizada en la organización

El autor pertenece al área de Tecnología Avanzadas de Indra Perú, actualmente brinda servicios para un cliente del sector Bancario, donde realiza actividades en un proyecto Big Data, la experiencia se detalla a continuación:

- Planificar y dar seguimiento de las tareas en el marco de la metodología ágil Scrum.
- Desarrollar notebooks PySpark para el pre procesamiento de las fuentes de datos en acompañamiento del Product Owner.
- Crear mallas para la ejecución de procesos de ingesta/reproceso de información de forma automática.
- Elaborar diccionarios de datos para la ingesta de datos en las distintas capas del Datalake.
- Procesar datos mediante el arquetipo Spark-scala y PySpark-Scaffolder definido por el framework de arquitectura.
- Capacitar a los Data Engineer del equipo sobre las tecnologías Big Data.
- Realizar el despliegue de soluciones con las herramientas de integración continua (IC) como Bitbucket, Jenkins y Jira.

CAPÍTULO III
ACTIVIDADES DESARROLLADAS

3.1. Situación problemática

La entidad bancaria negocia con operaciones de cambio de divisas, cuenta con canales de atención estratégicamente distribuidos para sus clientes, cuenta con un motor de precios para el cálculo de spreads comerciales en tiempo real para las operaciones de cambio de divisas, estos se actualizan minuto a minuto y determinan el mejor tipo de cambio por segmentos de clientes y está disponible las 24 horas.

A pesar del esfuerzo de la entidad bancaria por estar a la vanguardia en los requerimientos del mercado, la competencia cada vez más se posiciona en el mercado, las casas de cambio, cajas financieras y Fintech[2] están ofreciendo un mejor tipo de cambio, en consecuencia, incrementan su cartera con clientes de la entidad bancaria.

Como se observa en la figura 2, la entidad bancaria ha superado el ROF[3] planificado desde el 2016 hasta el 2019 y la tendencia fue positiva, sin embargo en el 2020 no se logró cumplir con lo planificado, esto impactó en los objetivos anuales de la entidad bancaria.

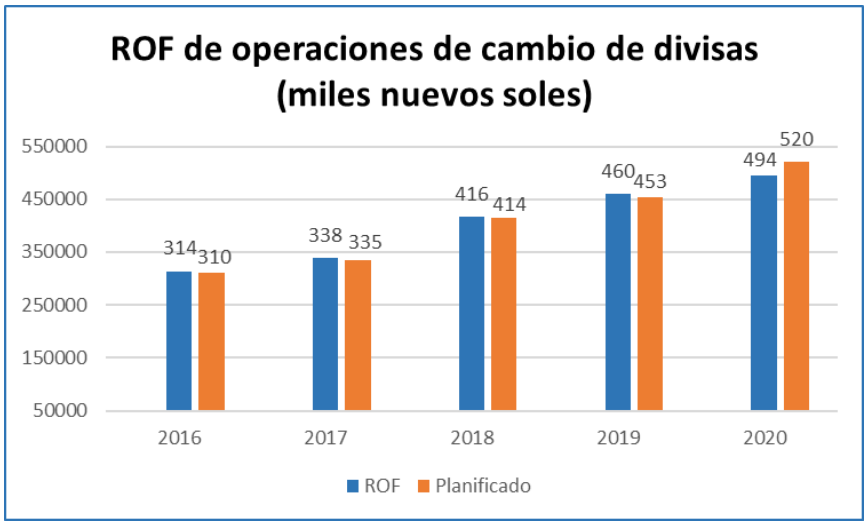


Figura 2: **ROF de operaciones. Elaboración propia[4]**

[2] Modelo de negocio que ofrecen productos o servicios financieros, mediante el uso intensivo de las tecnologías de información.
[3] Resultado de las operaciones financieras, incluye ganancias o pérdidas por ventas de activos.
[4] Datos obtenidos en la reunión de Kick Off para justificar la ejecución del proyecto

Según Anthony Polin, CEO de Inka Money[5], manifiesta lo siguiente, "En el Perú compiten 72 casas de cambio online, cuando en el 2018 solo había 25" (Andina, 2021). En el contexto de la pandemia por el covid-19 y la crisis política que atravesó el Perú en el 2021, el posicionamiento de estas empresas aumentó significativamente.

"Si antes de la pandemia solo el 11% de los cambios de moneda se realizaban de manera online, hoy esto representa el 40%" (Andina, 2021). El mercado es ahora más demandante, buscan que sus operaciones sean inmediatas, virtuales, seguras y sobre todo un tipo de cambio preferencial y mejor al que ofrecen los bancos.

Las Fintech son empresas que ofrecen servicios financieros a través de plataformas tecnológicas, uno de los recursos que usan las Fintech son las plataformas virtuales de cambios de divisas, según un estudio de la Asociación Fintech de Perú durante el contexto de la pandemia covid-19, manifiesta lo siguiente, "El 53,8% de las fintech han percibido una mayor facturación con respecto al mismo mes del año pasado, lo que muestra que ha aumentado la demanda de servicios financieros digitales durante la cuarentena Covid19" (Fintechs, 2020).

Las Fintech se caracterizan por el uso de las tecnologías de información, "El 73,8% de las fintech cuentan con su plataforma tecnológica basada en una web y solo el 13% lo tiene en una aplicación móvil. Las otras tecnologías que usan son Blockchain, APIs de Open Banking y Analítica/Big Data" (Fintechs, 2020).

Entre las tecnologías que usan las Fintech se encuentran las siguientes:

***Tabla 6*: Tecnologías usadas en Fintechs**

Tecnologías	% de uso
Inteligencia Artificial, Big Data, API	4.3%
Open Banking / API	4.3%
Blockchain & Crypto	4.3%
Aplicación Móvil & UX	13%
Plataforma Web	73.8%

***Fuente*. Adaptado de Asociación Fintech de Perú** (2020)

[5] Es una plataforma digital de cambio de divisas.

Es evidente que para la entidad bancaria hay una competencia que cada vez más va tomando un papel determinante en el mercado, las casas de cambios empiezan a formar grupos económicos, las Fintech invierten en tecnología, la competencia está ofreciendo un mejor tipo de cambio, estas situaciones provocan que la entidad bancaria pierda posicionamiento en el mercado.

3.1.1. Definición del problema

La entidad bancaria no cuenta con un mecanismo que permita identificar cuáles de sus clientes estarían realizando sus operaciones de cambio de divisas con la competencia y adicionalmente, que permita al motor de precios calcular un mejor spread comercial y personalizado por cliente.

3.2. Solución

Desde el área de Corporate & Investment Banking de la entidad bancaria se planteó implementar un modelo de datos que permita identificar a los potenciales clientes que cuentan con una alta probabilidad de realizar operaciones de cambio de divisas, basado en los servicios del banco, movimientos de cuentas y transferencias.

Como se observa en la figura 3, el modelo tiene el objetivo de analizar los servicios financieros que consumen los clientes para identificar en cuales el servicio se realiza con moneda extranjera y compararlo con las fuentes de operaciones de cambio de divisas para obtener la información de los clientes que no están realizando sus operaciones de cambio con la entidad bancaria.

Cuando se realizó el análisis de los servicios financieros, se calcularon una serie de variables que facilitaron el análisis personalizado por cliente, estas variables permitieron conocer características del cliente y la frecuencia en que estos operan en la entidad bancaria, esto ayudó a la captación y búsqueda de potenciales clientes.

Parte de la solución también permitió calcular un mejor spread comercial para los clientes que realizaban operaciones de compra y venta de divisas con frecuencia con la entidad bancaria, un spread que resulte más rentable a la entidad bancaria, esto implicó analizar el historial de operaciones de los clientes, transferencias y productos del banco para calcular variables a nivel de cliente, estas variables permitieron al motor de precios calcular un mejor spread comercial y maximizar las ganancias de la entidad bancaria.

Figura 3: **Fuentes de entrada potenciales clientes. Elaboración propia**

La implementación del modelo permitió al banco identificar a los potenciales clientes que necesitaban hacer operaciones de compra y venta de divisas y a los clientes que ya venían operando con frecuencia calcular un mejor spread comercial que le resultara más rentable a la entidad bancaria.

3.2.1. Objetivos

3.2.1.1. Objetivo general

Implementar un modelo de datos para el cálculo de variables en base a las operaciones de los clientes, movimientos de cuentas, transferencias y productos de la entidad bancaria para la identificación de potenciales clientes para operaciones de cambio de divisas y que permita la optimización del cálculo del spread comercial en las operaciones de los clientes que operan con más frecuencia a fin de aumentar la rentabilidad del negocio.

3.2.1.2. Objetivos específicos

Para identificar los potenciales clientes y la optimización del cálculo del spread comercial en el negocio de cambio de divisas, se definieron los siguientes objetivos específicos:

- Analizar los productos que consumen los clientes, conocer los canales de operación y variables de operación como las ganancias, cantidad de operaciones, spreads comerciales, etc.

- Procesar las fuentes de información que permitirán crear las variables del modelo de datos mediante las tecnologías big data que están certificadas en la entidad bancaria.
- Diseñar los flujos de procesamientos de datos para construir el modelo de datos en el entorno big data de la entidad bancaria.

3.2.2. Alcance

El proyecto se centra en la explotación de fuentes de información internas relacionados a los productos financieros que están alojados en la capa master data del datalake de la entidad bancaria, a fin de crear un modelo de datos con variables que ayuden en dos aspectos fundamentales que se indican a continuación:

- Identificación de potenciales clientes que requieran realizar operaciones de cambio de divisas.
- Optimización del spread comercial en las operaciones de cambio de divisas de los clientes que operan con más frecuencia.

El proyecto no incluye la adaptación del motor de precios dinámicos de operaciones de cambio de divisas para el consumo de las variables creadas en el modelo de datos.

3.2.3. Etapas y metodología

El proyecto de implementación del modelo de datos fue realizado en base al marco de trabajo Scrum, se tuvieron las siguientes consideraciones:

- El proyecto tuvo una duración de 2 trimestres desde el jueves 1 de abril 2021 hasta el jueves 16 de septiembre del 2021.
- Los Sprints tuvieron una duración de 10 días hábiles.
- El autor del presente informe estuvo presente durante los 2 trimestres de duración del proyecto y fue parte del scrum team con el rol de Big Data Engineer Ssr.
- Como se observa en la figura 4, el proyecto tuvo 6 etapas en total, estas van desde el pre procesamiento de las fuentes de información hasta la automatización de los procesamientos que generan el modelo.

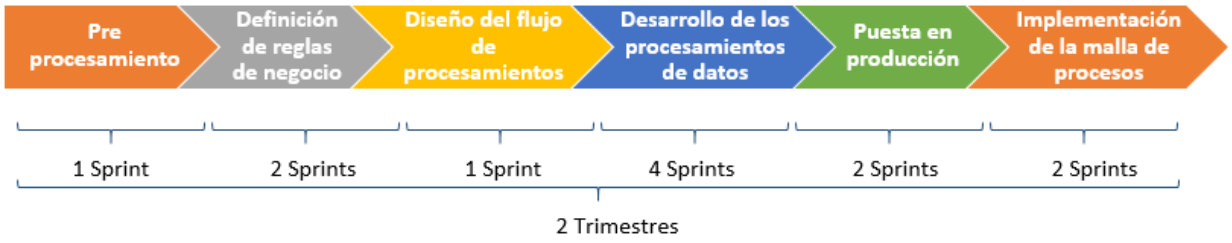


***Figura 4*: Etapas del proyecto. Elaboración propia**

Como se observa en la figura 5 el equipo encargado del proyecto estuvo conformado por los roles de scrum master, product owner y scrum team, este último fue un equipo de diferentes empresas proveedoras.

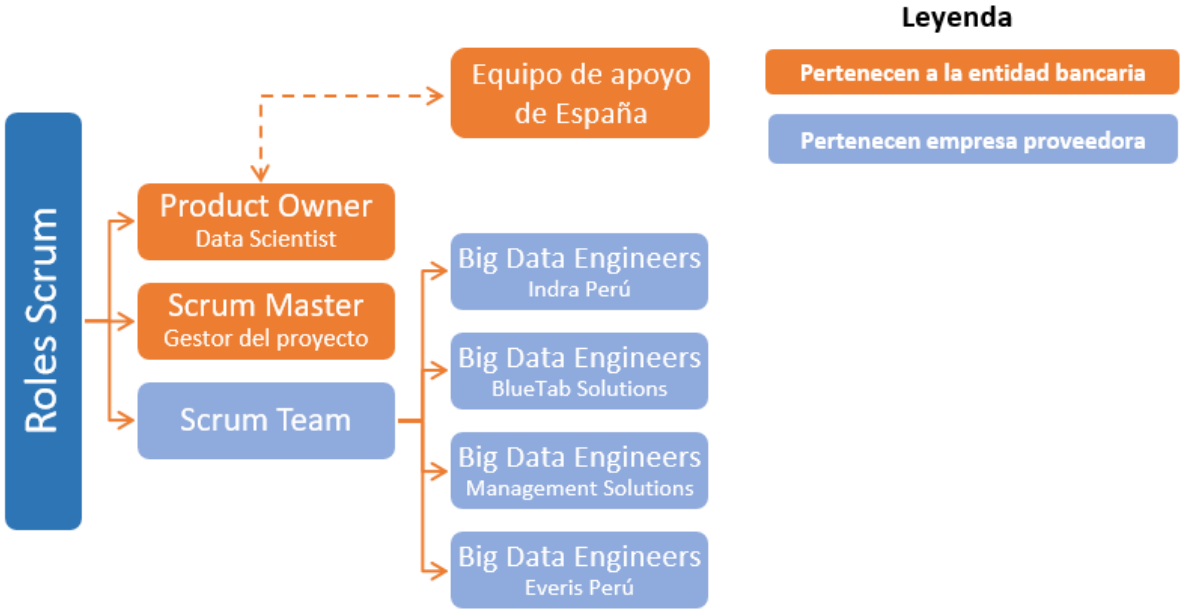


***Figura 5*: Roles Scrum del proyecto. Elaboración propia**

A continuación, se describen los roles del equipo Scrum del proyecto:

- **Product Owner**: Pertenecía a la entidad bancaria, como Data Scientist fue el experto en las fuentes de información que permitieron crear el modelo, tenía el conocimiento funcional de las fuentes y priorizaba las historias de usuario que se abordaron en cada Sprint.
- **Equipo de apoyo de España**: El proyecto contó con el apoyo de un equipo de la sede principal de entidad bancaria ubicada en España, este apoyo fue canalizado por el product owner, el apoyo de este equipo fue de gran impacto al proyecto porque ellos habían implementado un modelo similar al que se requería en Perú.

- **Scrum Master**: Pertenecía a la entidad bancaria, facilitaba los accesos de herramientas al Scrum Team, así mismo gestionó el proyecto y llevaba un seguimiento de las historias de usuario.
- **Scrum Team**: Durante el primer trimestre estuvo compuesto por 9 Big Data Engineer y durante el segundo trimestre por 5 Big Data Engineer, en la tabla 7 se detallan los recursos del Scrum Team por trimestre y las empresas proveedoras que intervinieron en el proyecto.

***Tabla 7*: Recursos Scrum Team**

Trimestre	Fábrica de Software	Cantidad de recursos
1er trimestre	Indra Perú	1 (El autor)
	Everis Perú	2
	BlueTab Solutions	5
	Management Solutions	1
	TOTAL	**9**
2do trimestre	Indra Perú	1 (El autor)
	BlueTab Solutions	3
	Management Solutions	1
	TOTAL	**5**

***Fuente*. Elaboración propia**

En la tabla 8 se presenta el cronograma del proyecto de implementación del modelo de datos, en el cronograma se detalla por cada etapa los Sprints y las fechas involucradas, el proyecto inició el jueves 1 de abril y terminó el jueves 16 de septiembre del año 2021.

***Tabla 8*: Cronograma del proyecto**

Etapa	Sprint	Ceremonias	Inicio	Fin	Duración en días hábiles
Pre procesamiento	Sprint 1	Sprint Planning	Jueves 1 abril	Jueves 1 abril	1
		Ejecución del Sprint	Viernes 2 abril	Martes 13 abril	8

		Sprint Review	Miércoles 14 abril	Miércoles 14 abril	1
		Sprint Retrospective			
Definición de reglas de negocio	Sprint 2	Sprint Planning	Jueves 15 abril	Jueves 15 abril	1
		Ejecución del Sprint	Viernes 16 abril	Martes 27 abril	8
		Sprint Review	Miércoles 28 abril	Miércoles 28 abril	1
		Sprint Retrospective			
	Sprint 3	Sprint Planning	Jueves 29 abril	Jueves 29 abril	1
		Ejecución del Sprint	Viernes 30 abril	Miércoles 12 mayo	8
		Sprint Review	Jueves 13 mayo	Jueves 13 mayo	1
		Sprint Retrospective			
Diseño del flujo de procesamientos	Sprint 4	Sprint Planning	Viernes 14 mayo	Viernes 14 mayo	1
		Ejecución del Sprint	Lunes 17 abril	Miércoles 26 mayo	8
		Sprint Review	Jueves 27 mayo	Jueves 27 mayo	1
		Sprint Retrospective			
Desarrollo de los procesamientos de datos	Sprint 5	Sprint Planning	Viernes 28 mayo	Viernes 28 mayo	1
		Ejecución del Sprint	Lunes 31 mayo	Miércoles 9 junio	8
		Sprint Review	Jueves 10 junio	Jueves 10 junio	1
		Sprint Retrospective			
	Sprint 6	Sprint Planning	Viernes 11 junio	Viernes 11 junio	1
		Ejecución del Sprint	Lunes 14 junio	Miércoles 23 junio	8
		Sprint Review	Jueves 24 junio	Jueves 24 junio	1
		Sprint Retrospective			
	Sprint 7	Sprint Planning	Viernes 25 junio	Viernes 25 junio	1

		Ejecución del Sprint	Lunes 28 junio	Miércoles 7 julio	8
		Sprint Review	Jueves 8 julio	Jueves 8 julio	1
		Sprint Retrospective			
	Sprint 8	Sprint Planning	Viernes 9 julio	Viernes 9 julio	1
		Ejecución del Sprint	Lunes 12 julio	Miércoles 21 julio	8
		Sprint Review	Jueves 22 julio	Jueves 22 julio	1
		Sprint Retrospective			
Puesta en producción	Sprint 9	Sprint Planning	Viernes 23 julio	Viernes 23 julio	1
		Ejecución del Sprint	Lunes 26 julio	Miércoles 4 agosto	8
		Sprint Review	Jueves 5 agosto	Jueves 5 agosto	1
		Sprint Retrospective			
	Sprint 10	Sprint Planning	Viernes 6 agosto	Viernes 6 agosto	1
		Ejecución del Sprint	Lunes 9 agosto	Miércoles 18 agosto	8
		Sprint Review	Jueves 19 agosto	Jueves 19 agosto	1
		Sprint Retrospective			
Implementación de la malla de procesos	Sprint 11	Sprint Planning	Viernes 20 agosto	Viernes 20 agosto	1
		Ejecución del Sprint	Lunes 23 agosto	Miércoles 1 septiembre	8
		Sprint Review	Jueves 2 septiembre	Jueves 2 septiembre	1
		Sprint Retrospective			
	Sprint 12	Sprint Planning	Viernes 3 septiembre	Viernes 3 septiembre	1
		Ejecución del Sprint	Lunes 5 septiembre	Miércoles 15 septiembre	8
		Sprint Review			1

	Sprint Retrospective	Jueves 16 septiembre	Jueves 16 septiembre

***Fuente*. Elaboración propia**

De acuerdo con los autores (Duque-Jaramillo & Villa-Enciso, 2020) las etapas para abordar proyectos de big data es mediante un flujo de tareas, las cuales se detallan a continuación:

- **Personas**: Los autores indican que las personas son la base principal de los procesos porque a través de ellas se generan los datos.
- **Generación de datos**: La generación de datos se generan por la interacción de las personas con los sistemas de información de una organización.
- **Almacenamiento en bodegas**: En esta etapa se centralizan los datos generados en el repositorio principal de la organización que posteriormente permita analizar los datos.
- **Análisis**: Los datos almacenados y centralizados permite a la organización analizar y procesar los datos para ser más competitivos.
- **Big Data**: En esta etapa se aplican los conceptos y tecnologías que brinda big data, para procesar grandes volúmenes de datos y ofrecer información en tiempo real a la organización.
- **Decisiones y acciones**: Los autores mencionan que big data servirá como apoyo a la organización para tomar decisiones y definir acciones.

A continuación, se describe de forma general lo desarrollado en cada etapa del proyecto de implementación del modelo de datos:

- **Pre procesamiento:** Fue la etapa inicial del proyecto donde el scrum team y el product owner llevaron a cabo el análisis de las fuentes de información que permitirían crear el modelo de datos, el análisis consistió en conocer a detalle cada fuente de información, tener claro que tipo de información contienen y que variables se pueden crear por cada fuente.
- **Definición de reglas de negocio:** Producto del análisis de las fuentes en la etapa anterior, el product owner definió las reglas de negocio por cada fuente de información que permitieron crear las variables del modelo de datos, en esta etapa el product owner recibió apoyo del equipo de colaboración de España ya

que ellos tenían experiencia en la creación de modelos de datos para el aumento de la rentabilidad del negocio de cambio de divisas.

- **Diseño del flujo de procesamientos:** Durante esta etapa el scrum team diseñó la mejor forma de secuenciar los procesamientos tomando en cuenta las fuentes de entrada de cada procesamiento, las dependencias entre ellos e identificando cuales se pueden ejecutar en paralelo. Esto permitió optimizar el tiempo de ejecución de todo el proceso que genera el modelo.
- **Desarrollo de los procesamientos de datos:** El scrum team desarrolló los procesamientos de datos tomando en cuenta las reglas de negocio definidas y el flujo de los procesamientos. Durante el desarrolló se fueron afinando las reglas de negocio con apoyo del product owner.
- **Puesta en producción:** El scrum team estructuró los desarrollos realizados en la etapa anterior con el arquetipo Scaffolder definido por el área de Arquitectura de datos de la entidad bancaria, así mismo se interactuó con el equipo DQA quienes validaron que los desarrollos cumplan con las buenas prácticas de desarrollo y lineamientos de la entidad bancaria, fue de gran impacto el uso de herramientas de integración continua porque evitó realizar trabajos manuales para la puesta en producción de los desarrollos. Se desplegaron en el ambiente de producción de la entidad bancaria todos los procesamientos que permitieron crear el modelo de datos.
- **Implementación de la malla de procesos:** En esta última etapa se automatizó las ejecuciones de todos los procesos desplegados en producción, el diseño del flujo de procesamientos permitió tener una visión general de las dependencias entre cada procesamiento, esto facilitó armar la malla de procesos. Para lograr la instalación de la malla el scrum team interactuó con el equipo de Soporte Control-M de México ya que en ese país se tienen físicamente los servidores y en Perú no hay equipo que certifique e instale mallas en Control-M.

En la figura 6 se observa la interacción que hubo entre los equipos de España, Perú y México, el equipo de Perú fue el encargado de implementar el modelo datos, el equipo de España apoyó en la definición de las variables que debía contemplar el modelo de datos y el equipo de Soporte Control-M de México apoyó en automatizar la ejecución de los procesos Scaffolder.

***Figura 6*: Equipos de varios países. Elaboración propia**

3.2.4. Fundamentos utilizados

3.2.4.1. Scrum

"Scrum es un marco ligero que ayuda a las personas, los equipos y las organizaciones a generar valor a través de soluciones adaptables para problemas complejos" (Scrum, 2021).

El marco de trabajo Scrum, se caracteriza por desarrollos de corta duración de tiempo donde se crean pequeños entregables que van aportando a la creación de un producto final, a estos periodos de tiempo se les denomina Sprints que por lo general duran 2 semanas, como se observa en la figura 7, Scrum tiene un ciclo de vida donde intervienen una serie de roles, artefactos y ceremonias que se detallan a continuación.

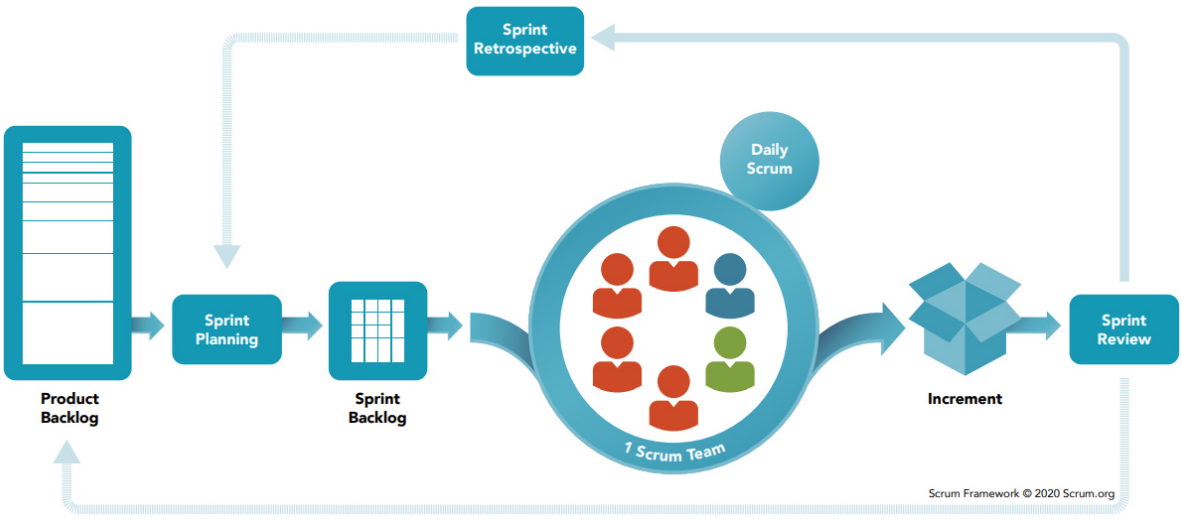


***Figura 7*: Ciclo de vida Scrum. Adaptado de Scrum** (2021)

En la tabla 9 se describen los roles que intervienen en el marco de trabajo Scrum.

***Tabla 9*: Roles Scrum**

Roles	Descripción
Product Owner	Es el responsable de administrar el Product Backlog definiendo los requerimientos que se plasman en las historias de usuario, así mismo de priorizar las historias que se van abarcar en cada iteración o sprint.
Scrum Master	Es el encargado de facilitar los recursos necesarios al Scrum Team para evitar problemas al momento de desarrollar las historias de usuario.
Scrum Team	Es el equipo responsable de desarrollar las historias de usuario y crear el producto final.
Stakeholder	Son las personas interesadas en el producto, generalmente son los dueños o directores de la organización.

***Fuente*. Elaboración propia**

En la tabla 10 se describen las ceremonias que forman parte de Scrum.

***Tabla 10*: Ceremonias Scrum**

Ceremonias	Descripción
Sprint Planing	Es la etapa donde se planifican las historias que serán abordadas durante un Sprint, la priorización de historias se da por el Product Owner.
Daily Scrum	Reunión diaria para el seguimiento de tareas desarrolladas durante el Sprint, también se analizan los posibles problemas que impidan el desarrollo de historias de usuario.
Sprint Review	Consiste en una reunión que se da al final de cada Sprint, se explica a los Stakeholder el avance y los logros conseguidos durante el Sprint.
Sprint Retrospective	Durante esta reunión el Scrum Team comenta su apreciación sobre el Sprint pasado a fin de ver las cosas que han funcionado e identificar puntos de mejora. El objetivo de esta reunión es promover la mejora continua.

***Fuente*. Elaboración propia**

En la tabla 11 se describen los artefactos que forman parte de Scrum.

***Tabla 11*: Artefactos Scrum**

Artefactos	Descripción
Product Backlog	Conjunto de tareas denominadas historias de usuario donde se describen de forma concreta las tareas que deben realizarse durante los Sprints para cumplir con el objetivo del proyecto.
Sprint Backlog	Es un conjunto de historias de usuario parte del Product Backlog que son abordadas durante un Sprint.

***Fuente*. Elaboración propia**

3.2.4.2. Big data

Se puede definir que big data es un marco de trabajo que permite procesar grandes volúmenes de datos a gran velocidad, datos que pueden variar en el tiempo y que pueden ser estructurados como no estructurados. Big data es considerado como un marco de trabajo porque ofrece dos cosas básicas, conceptos y tecnologías.

Entre los conceptos que más caracterizan a big data son las 5V, big data se asocia a la capacidad de procesar grandes volúmenes de información a gran velocidad, sin embargo, existen otras características que se detallan a continuación:

- **Volumen**: La capacidad de procesar grandes conjuntos de datos es uno de los principales atributos del big data, “Big data describe colecciones de datos de un tamaño difícil de procesar con técnicas tradicionales de gestión de datos” (Kayser, Nehrke, & Zubovic, 2018).
- **Variedad**: Existen muchos tipos o formas en la que se pueden encontrar los datos, no solo existen los datos transaccionales de bases de datos estructurados. “Si bien muchas definiciones de big data se concentran en el aspecto del volumen que se refiere a la escala de datos disponibles, big data trae, en particular, formatos heterogéneos y un amplio espectro de posibles fuentes de datos” (Kayser, Nehrke, & Zubovic, 2018). Por ejemplo, como datos no estructurados se encuentran audios, videos o imágenes.
- **Velocidad**: La velocidad facilita generar datos casi al instante y viajar lo más rápido posible para procesarse. La ventaja de procesar datos a gran velocidad

permite a la organización de que algunos de sus servicios puedan ofrecerse en tiempo real y ser más competitivo, así mismo generar información actualizada para la toma de decisiones.

- **Veracidad**: Es necesario que los datos enriquecidos pasen por validaciones de calidad para que la información que se brinde tenga un alto grado de fiabilidad. "Para que el prototipo sea profesionalizado, los resultados deben ser aceptados y comprendidos, y la unidad de negocio debe estar continuamente involucrada en el proceso" (Kayser, Nehrke, & Zubovic, 2018), para que la información sea veraz es necesario que las unidades de negocios sean parte de la validación de los resultados.
- **Valor**: Los beneficios del big data están relacionados a la reducción de costos, incremento de la rentabilidad y mejoras del negocio. "Según nuestro entendimiento, el valor se genera mediante el análisis de datos dentro de un contexto determinado, con una declaración de problema relacionada con un requisito empresarial que impulsa la necesidad de innovación" (Kayser, Nehrke, & Zubovic, 2018). Big data da pie a la innovación por las bondades que esta permite, generar nuevas soluciones e incrementar la eficiencia de los servicios y productos de la organización.

3.2.4.3. Hadoop

Es un sistema de archivos distribuido, gestiona el almacenamiento de archivos en clústers con cientos o miles de máquinas y es tolerante a fallos, permite almacenar y procesar grandes volúmenes de datos estructurados y no estructurados, a continuación, se describen algunas de sus características:

- "Los archivos grandes se dividen en múltiples bloques y cada uno de ellos se replica en múltiples servidores, dependiendo del factor de replicación de archivos" (Kalmukov, Marinov, Mladenova, & Valova, 2021). Esta característica de Hadoop permite tener disponible los datos incluso si uno o más servidores fallan.
- "Lo utilizan con éxito muchas empresas transnacionales, como Facebook, LinkedIn, Twitter, eBay, Samsung, J.P. Morgan, AOL y muchas más" (Kalmukov, Marinov, Mladenova, & Valova, 2021). Debido a las bondades de Hadoop como su alta escalabilidad, almacenar grandes volúmenes de datos,

procesar a grandes velocidades, etc. grandes empresas como Facebook lo implementan en sus sistemas de información.

- "El ecosistema incluye una variedad de herramientas (Hive, Pig, HBase, Oozie, Zookeeper, Sqoop, Flume, Spark, Kafka, Impala) que permiten a las empresas almacenar, procesar y analizar grandes volúmenes de datos" (Kalmukov, Marinov, Mladenova, & Valova, 2021). Hadoop se puede integrar con muchas herramientas que permiten procesar diversos tipos de datos, cabe resaltar que estas herramientas también son de código libre.

3.2.4.4. Apache Spark

Es un motor que permite a gran velocidad el procesamiento de datos, es de código libre y está construido con el lenguaje de programación Scala, trabaja principalmente con la RAM de los workers permitiendo trabajar con grandes volúmenes de datos como petabytes de data.

En base a una investigación sobre la detección de fraudes en tiempo real se manifiesta lo siguiente, "En comparación con MapReduce, la velocidad de procesamiento de datos de Spark es exponencialmente más rápida porque MapReduce escribe y lee en los discos duros. Por esta razón, decidimos utilizar la implementación en memoria de Spark en nuestro estudio" (Hasanin, Khoshgoftaar, Joffrey L., & Bauder, 2019).

Antes de Spark, Hadoop ofrecía procesamientos de datos mediante su motor MapReduce, el cual lo hacía haciendo procesos de escritura y lectura en disco, es notable que para procesamientos Spark permite hacerlo a una gran velocidad a comparación de MapReduce por su particularidad en el uso exclusivo de la memoria de forma distribuida.

3.2.4.5. Python

Es un lenguaje de programación que se integra fácilmente en un entorno big data por la cantidad de librerías que presenta, existen actualmente librerías que permiten usar funciones de Apache Spark a través de Python para el procesamiento y visualización de datos.

En un proyecto de machine learning, Python fue utilizado para el procesamiento de los datos y visualización en un entorno big data de forma exitosa, en un artículo relacionado a dicho proyecto se menciona lo siguiente, "En base a la

gran cantidad de datos generados, los métodos de machine learning automático y los métodos de programación en lenguaje Python se utilizan para analizar y procesar los datos" (Wu, Wang, & Tang, 2021).

Python permite procesar volúmenes grandes de datos, usar librerías que brindan la opción de crear funciones para el machine learning como funciones de regresión lineal, procesar y visualizar datos. Python tienen muchos atributos como los mencionados anteriormente que facilitan el trabajo en proyectos de big data.

3.2.4.6. Jupyter Notebook

Es una herramienta web que permite ejecutar código en diferentes lenguajes de programación como Python, Scala, etc., es de código libre y se puede integrar a entornos big data. El uso de Jupyter Notebook es muy interactivo y practico debido a su composición por celdas, la ventaja de esta herramienta es que permite conectarse a un nodo del datalake y ejecutar porciones de código de forma inmediata, esto abstrae al usuario en temas de configuración de recursos y conexiones al datalake,

Debido a sus características permite explorar datos y procesarlos de formar amigable al usuario, en una empresa de Alemania, DESY, se usó Jupyter Notebook para la exploración de datos en un proyecto de física de partículas, ellos manifiestan lo siguiente, "Originalmente, el alcance de los Jupyter Notebook era principalmente proporcionar tutoriales prácticos fáciles de usar, proporcionar flujos de trabajo de referencia o desarrollar (principalmente) código Python. Este tipo de aplicaciones no requieren alta disponibilidad ni recursos informáticos dedicados" (Reppin, y otros, 2021). Es notable la facilidad de uso de la herramienta para procesos de exploración de datos y evitar invertir recursos dedicados para dichos propósitos.

3.2.4.7. Datalake

El concepto de datalake busca centralizar la información de una organización que permita procesos de analítica de datos, según Buer, y otros autores, manifiestan lo siguiente sobre su experiencia en procesamientos de datos en un datalake, "Se ha demostrado que puede escalar a petabytes de datos, lo que permite a las organizaciones almacenar y procesar datos a una escala que antes no era factible sin sistemas dedicados muy costosos" (2021).

La ventaja del datalake con otros sistemas de procesamiento de datos es que permite almacenar y explotar grandes volúmenes de datos como petabytes de datos, este nivel de procesamiento requiere mayores costos a nivel de infraestructura para los sistemas de datos tradicionales.

En la experiencia de Buer, y otros autores, el proyecto que realizaron tuvo muchas fuentes de entrada para el procesamiento de datos, debido a esto definieron 3 capas básicas en el datalake, estas son:

- **DropZone**: Capa donde aterrizan los datos de las fuentes de la organización mediante las herramientas necesarias.
- **LandingZone**: Capa donde se almacenan los datos de forma inmutable, solo se realiza un cambio de formato, en el caso de su proyecto fue parquet, para esta ingesta usaron SQL, Spark y WebHDFS.
- **IntegrationZone**: Capa donde se realizan el enriquecimiento de los datos, para las necesidades particulares de los proyectos.

Las capas del datalake de la entidad bancaria conceptualmente es muy similar a lo indicado en la referencia anterior, la diferencia principal es que el datalake de la entidad bancaria cuenta con una capa adicional, esta cumple la función de zona donde se almacenan los datos enriquecidos, a continuación, se detallan las capas del datalake de la entidad bancaria:

- **Staging Data**: Capa donde llega la información de las aplicaciones core de la entidad bancaria.
- **Raw Data**: Capa donde se almacenan los datos en crudo, sin ninguna transformación, solo se realiza un cambio de formato a Avro.
- **Master Data**: Capa donde se almacenan los datos preparados y optimizados para su explotación en formato Parquet.
- **Staging Out**: Capa donde se almacenan los datos preparados y optimizados para el uso externo como aplicaciones, webs, reportes u otros.

3.2.4.8. Formatos de almacenamiento Avro y Parquet

En una investigación sobre los tipos de formato en un entorno big data se manifiesta lo siguiente, “Avro es un formato de almacenamiento de datos orientado a filas. Contiene un esquema en formato JSON, que permite operaciones de lectura e

interpretación más rápidas. La estructura del archivo consta de una cabecera y bloques de datos" (Belov, Tatarintsev, & Nikulchev, 2021).

Con respecto al formato parquet se manifiesta lo siguiente, "Apache Parquet es un formato binario orientado a columnas. Permite definir esquemas de compresión a nivel de columna y agregar nuevas codificaciones a medida que aparecen" (Belov, Tatarintsev, & Nikulchev, 2021).

Como se observa en las figuras 8 y 9 los formatos avro y parquet permiten procesar a mayor velocidad procesamientos de filtros y agrupación de datos que otros de formatos de datos.

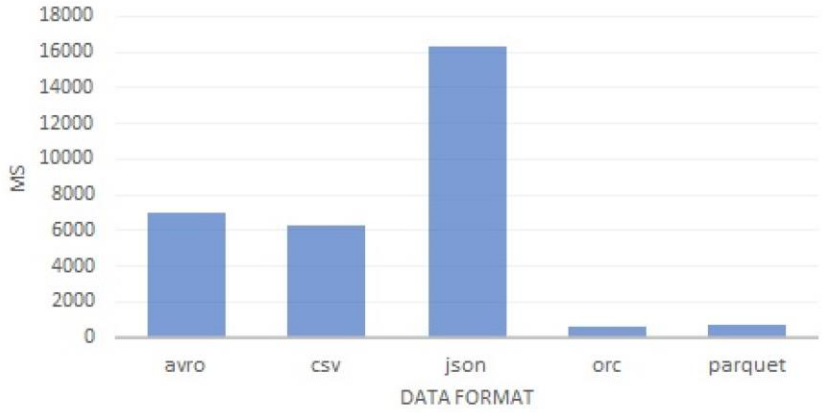


***Figura 8*: Filtrar datos. Adaptado de Belov, Tatarintsev, & Nikulchev** (2021)

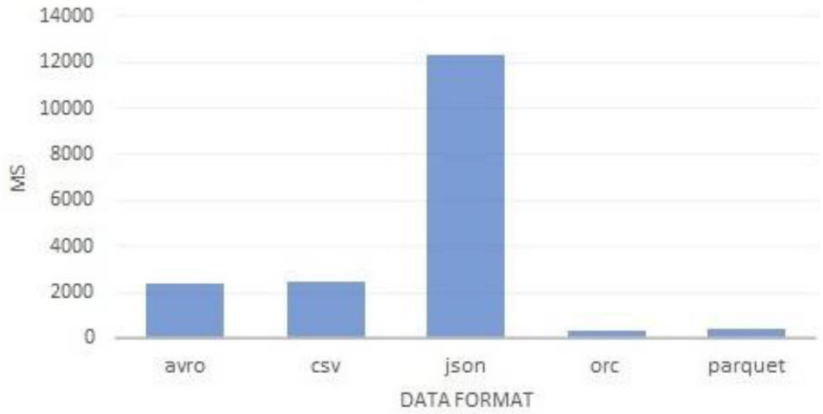


***Figura 9*: Agrupar datos. Adaptado de Belov, Tatarintsev, & Nikulchev** (2021)

Avro permite el rápido procesamiento de grandes volúmenes de datos y puede trabajar con estructuras complejas de datos y el formato parquet es mucho más rápido que avro para el procesamiento de datos y es considerado el formato estándar para trabajar en procesamiento de datos en la entidad bancaria.

3.2.4.9. Integración continua

Durante el ciclo de vida del desarrollo de software hay una fase donde lo desarrollado pasa a ser parte de un sistema, es en este proceso donde el concepto de integración continua recalca dos aspectos fundamentales, la integración automática de los nuevos entregables y las pruebas de calidad de dichos entregables.

"La IC permite a las empresas de software tener un ciclo de lanzamiento más corto y frecuente, mejorar la calidad del software y aumentar la productividad de sus equipos. Esta práctica incluye la construcción y las pruebas de software automatizadas" (Shahin, Ali Babar, & Zhu, 2017). Al ser automático los despliegues, permite que el equipo de desarrollo tenga más tiempo para invertirlo en la producción del producto, así mismo permite conocer el nivel de calidad de los entregables a través de las pruebas automatizadas.

3.2.4.10. Jenkins

"Jenkins es un servidor de automatización de código abierto autónomo que se puede utilizar para automatizar todo tipo de tareas relacionadas con la creación, prueba y entrega o implementación de software" (Jenkins, 2021). Jenkins permite que los despliegues sean automatizados, es una herramienta que permite la integración continua de los entregables creados por el equipo de desarrollo.

La ventaja de Jenkins es que permite la visualización de las etapas del despliegue por medio del plugin Blue Ocean, con respecto a Blue Ocean en un artículo de investigación sobre la visualización de flujos de tareas con Jenkins se manifiesta lo siguiente, "Blue Ocean es un plugin instalable para Jenkins con una interfaz de usuario rediseñada y moderna. Blue Ocean también tiene una interfaz gráfica de usuario web para crear y editar tuberías que resulta en archivos declarativos de pipeline de Jenkins" (Révész & Pataki, 2021).

3.2.4.11. Control-M

Control-M es una herramienta que permite automatizar las ejecuciones de procesos batch, unos de los componentes principales de Control-M es el Job Scheduling Definition el cual permite especificar el criterio de ejecución y que debe suceder con cada job o proceso, adicionalmente permite organizar el flujo de tareas según las dependencias que estas tienen entre sí.

Los parámetros que reciben los jobs son los siguientes:

- Parámetros generales: Información general del job.
- Parámetros básicos de Scheduling: Es el criterio de ejecución del job.
- Parámetros pre proceso: Son las condiciones previas que deben suceder para que un job se ejecute.
- Parámetros post proceso: Son las acciones posteriores que se van a realizar posterior a la ejecución de un job.

Para el proyecto descrito en el presente informe, Control-M permite automatizar los flujos de datos entre las capas del datalake de la entidad bancaria.

3.2.4.12. Arquetipo Scaffolder

Un arquetipo permite que el código fuente de una solución se estructure por un patrón de diseño determinado, básicamente es una plantilla preparada para plasmar todo lo desarrollado con ciertas tecnologías y lenguajes de programación, estas plantillas contienen archivos de configuración que permiten la conexión automática con el servidor donde se hará el despliegue de lo desarrollado.

En la figura 10 se observa la estructura del arquetipo base, esta cuenta con las siguientes secciones:

- Los archivos transformations.py y utils.py se definen las funciones de negocio y utilitarios, en esta sección se definen determinadas transformaciones para el tratamiento de los datos como, por ejemplo, filtros, cruces entre tablas, agrupamientos, entre otros.
- En el archivo constants.py se definen las constantes que serán usadas durante todo el desarrollo, una constante es un valor que no cambia en el tiempo.
- En el archivo fields.py se definen los campos de las fuentes de entradas, Python permite crear diccionarios donde se pueden agrupar un conjunto de constantes, los diccionarios en esta sección permiten agrupar los campos por cada fuente de entrada.
- El archivo application.conf permite definir parámetros de entrada por parte de Control-M para que los procesos al ser ejecutados reciban parámetros como fechas de lectura, rutas de lectura y/o escritura.
- El archivo app.py permite orquestar todas las funciones de negocio y utilitarias según una determinada lógica, en la figura 10 se muestra la estructura interna

de este archivo donde se pueden definir las secciones de importación de librerías, lectura de parámetros de entrada, documentación del proceso y finalmente el código.

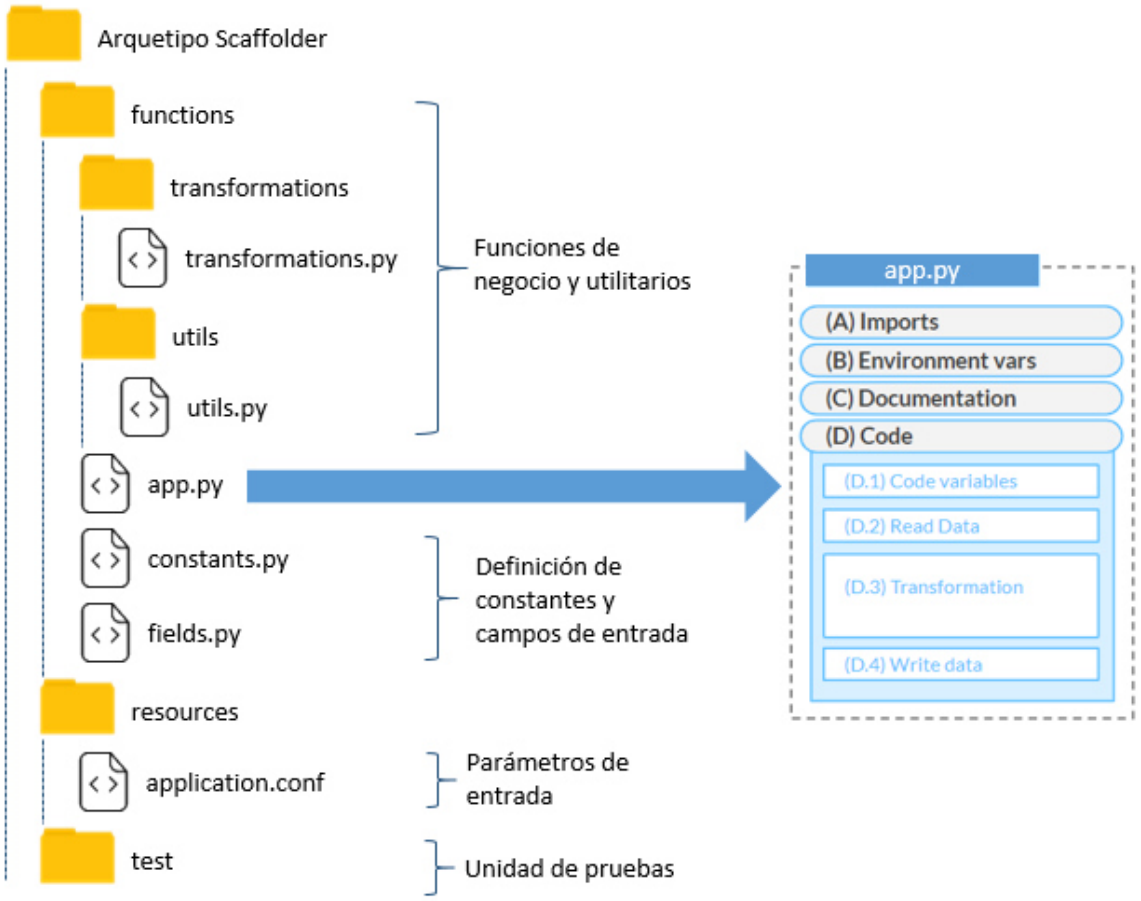


***Figura 10*: Estructura arquetipo Scaffolder. Elaboración propia**

Entre los principales beneficios de usar un arquetipo es tener un estándar de desarrollo para los proyectos de la organización, al ser reutilizado un arquetipo por varios proyectos permite a otros equipos identificar rápidamente el objetivo del código fuente y realizar un mantenimiento fluido en caso sea necesario.

En la entidad bancaria donde se realizó el proyecto del presente informe el arquetipo que utilizó el equipo de desarrollo se llamó Scaffolder, el cual permitió estructurar el código desarrollado con Python y Spark.

3.2.4.13. Data quality assurance

El aseguramiento de la calidad de datos tiene el propósito de garantizar que un conjunto de datos cumpla con ciertas reglas derivadas de su propia naturaleza.

El incumplimiento de cualquiera de estos implicaría que los datos no son fiables o válidos para su correcto uso, ya sea desde un punto de vista técnico o funcional.

En un artículo donde abordan el tema de aseguramiento de calidad en tecnologías big data manifiestan lo siguiente, "…el aseguramiento de la calidad es una actividad que se aplica a todo el proceso de big data. El desarrollo de aplicaciones de big data es una tarea de ingeniería de sistemas que incluye requisitos, análisis, diseño, implementación y pruebas." (Shunhui, Qingqiu, Wennan, Pengcheng, & Henry, 2020). El objetivo es prevenir que los datos tengan errores técnicos o funcionales que puedan afectar el correcto funcionamiento de los servicios de una empresa o en la toma de decisiones.

Cuando un científico de datos o un desarrollador escribe código, este puede tener errores que el desarrollador no previo durante el desarrollo. Es importante garantizar algunos estándares en el proyecto, por ejemplo, sobre cómo documentar, probar código, convenciones de nomenclatura, etc.

En la entidad bancaria el equipo Data Quality Assurance (DQA), designa uno o más revisores para verificar el código desarrollado. Los revisores deben verificar que el código cumpla con el propósito del autor y que el código tenga documentación, pruebas exhaustivas y siga las convenciones de estilo de codificación correctas.

3.2.4.14. Operaciones de cambio de divisas

Las entidades financieras que ofrecen el servicio de cambio de divisas permiten que los usuarios como personas naturales o jurídicas puedan hacer la compra o venta de diversas divisas.

En una investigación sobre el uso de redes neuronales para la predicción del tipo de cambio de monedas extranjeras manifiestan lo siguiente, "El mercado de divisas (FOREX) es el mercado de cambio de divisas más grande del mundo. Los comerciantes comercian con billones de dólares por día" (Saiful Islam & Hossain, 2020). El mercado de divisas es muy amplio donde entidades financieras, empresas hasta individuos diariamente están realizando operaciones de compra y venta de divisas.

"Debido a la complejidad, volatilidad y alta fluctuación, es bastante difícil adivinar el precio antes del momento real. Los comerciantes e inversores buscan continuamente nuevos métodos para superar al mercado y obtener mayores

ganancias" (Saiful Islam & Hossain, 2020). A diario es un reto determinar el mejor tipo de cambio para la compra y venta de divisas por la naturaleza propia de este mercado, sin embargo, en la actualidad existen diversos métodos para predecir el comportamiento del mercado como machine learning o minería de datos.

3.2.4.15. Spread comercial

El spread comercial es la diferencia entre el precio de costo y venta de un determinado bien, para el presente informe hace referencia al spread de las divisas, es decir, la diferencia entre los precios de compra y venta de las divisas, esta representa la ganancia que obtiene la entidad bancaria como intermediario para las operaciones de cambio de divisas.

3.2.5. Implementación de las áreas, procesos, sistemas y buenas prácticas

A continuación, se describe a detalle cada etapa del proyecto de implementación del modelo de datos.

3.2.5.1. Pre procesamiento

Durante esta etapa el scrum team analizó en conjunto con el product owner 14 fuentes de información internas que fueron fuentes de entrada para la construcción de las variables del modelo de datos, se tomaron en cuenta los siguientes puntos:

- El objetivo fue analizar el comportamiento de la data y tener claro que variables se podían crear a través de estas.
- El análisis de las fuentes se realizó a través de Jupyter Notebook con Apache Spark versión 2.4 y Python versión 3 en el entorno[6] big data de la entidad bancaria.

Para el análisis de las fuentes se llevaron a cabo las siguientes actividades:

- Conocer a que nivel de granularidad están los registros, por ejemplo, a nivel de cliente, transacciones, órdenes de pago, divisas, entidad, etc.
- Identificar que campos son los que permitirán crear las variables.
- Validar si existe duplicidad de registros.

[6] En la etapa de desarrollo de los procesamientos de datos se puede observar la arquitectura a alto nivel de la solución en el entorno big data de la entidad bancaria.

- Conocer la estructura de las fuentes y tener un conocimiento funcional de los datos más relevantes.
- Detectar datos anómalos que no tenía contemplado el product owner.
- Proponer la creación de variables que no tenía contemplado el product owner que agreguen valor al modelo.
- Conocer la periodicidad de carga, historia y el tipo de ingesta de las fuentes.

En las siguientes figuras se observa la estructura de los Jupyter Notebook que el scrum team utilizó para realizar el análisis exploratorio de las fuentes de información.

En la figura 11 se observa la sección de encabezado donde se indica que el objetivo de la Notebook es el pre procesamiento y el nombre de la fuente de información que se va analizar, adicionalmente como buena práctica se centralizaron las librerías necesarias durante todo el análisis al inicio de la Notebook.

***Figura 11*: Pre procesamiento - sección librerías. Elaboración propia**

En la figura 12 se observa la sección de definición de las funciones que se utilizaron en el análisis exploratorio de las fuentes de información, en la figura se muestran a manera de ejemplo dos funciones.

2. Funciones

```
[2]: def filter_last_day_month(df,col_date="fecha_ejecucion"):
         df_dates = df.select(col_date).distinct()\
                         .withColumn('year_month',
                                     F.regexp_replace(
                                         F.substring(F.col(col_date),1,7),"-",""))\
                         .groupby('year_month').agg(F.max(col_date).alias(col_date))

         df_final = df.alias('a').join(df_dates.alias('b'),
                                       on = col_date ,
                                       how = 'inner')\
                         .select('a.*')
         return df_final
```

```
[3]: def count_empty_values(df,columns=["default_all"], empty_string=False, porcentange=True, n_rows=0):
         if(columns[0]=="default_all"):
             columns = df.columns
         if(n_rows == 0 & porcentange):
             n_rows = df.count()
         for column in columns:
             nulls = df.select(column).filter(F.col(column).isNull()).count()
             if(empty_string):
                 empty_str = df.select(column).filter(F.col(column)=="").count()
                 nulls = nulls + empty_str
             if(porcentange):
                 porc_nulls = round(nulls/n_rows * 100,2)
                 print(column+ ": " + str(nulls) + "| " + str(porc_nulls) +"%")
             else:
                 print(column + ": " + str(nulls))
```

***Figura 12*: Pre procesamiento-sección funciones. Elaboración propia**

A continuación, se describen las dos funciones que se muestran en la figura 12:

- La función filter_last_day_month permitió filtrar las fuentes de información para calcular la volumetría mensual de las fuentes.
- La función count_empty_values permitió conocer el porcentaje de datos nulos por cada campo de la fuente.

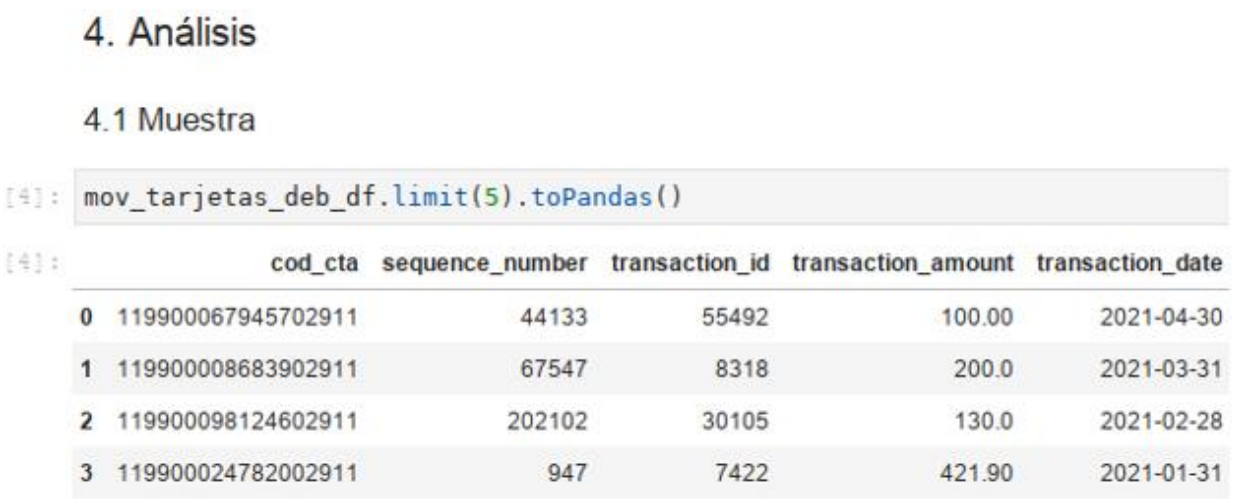
En la figura 13 se observa la sección donde se realiza la lectura de las fuentes de información, tal como se indicó en el alcance del proyecto, las fuentes de información se ubican en la capa master data del datalake y se encuentran en formato parquet.

3. Lectura BD

```
[3]: mov_tarjetas_deb_df = spark.read.parquet('/capa/master/data/mov_tarjetas_debito')
```

***Figura 13*: Pre procesamiento-sección lectura parquets. Elaboración propia**

En la figura 14 se observa la sección donde se realiza el análisis de las fuentes de información, donde se inició con lo más básico como observar una muestra de la data, en esta sección se invocan a las funciones definidas en la sección de definición de funciones, así mismo se usan comandos de Python y Spark para el análisis a detalle.



4. Análisis

4.1 Muestra

```
[4]: mov_tarjetas_deb_df.limit(5).toPandas()
```

[4]:

	cod_cta	sequence_number	transaction_id	transaction_amount	transaction_date
0	1199000679457029 11	44133	55492	100.00	2021-04-30
1	1199000086839029 11	67547	8318	200.0	2021-03-31
2	1199000981246029 11	202102	30105	130.0	2021-02-28
3	1199000247820029 11	947	7422	421.90	2021-01-31

***Figura 14*: Pre procesamiento-sección análisis. Elaboración propia[7]**

Los datos obtenidos de la exploración de las fuentes se centralizaron en un Excel online compartido entre todos los integrantes del equipo scrum con la estructura que se muestra en la tabla 12, la ventaja de centralizar toda la información permitió al product owner tener una visión general de todas las fuentes analizadas.

***Tabla 12*: Estructura de Excel con detalle del análisis de fuentes**

N°	Campo	Descripción	Ejemplo
1	Fuente	Nombre de la fuente analizada.	Comercio exterior
2	Descripción	Descripción funcional de la data que contiene la fuente.	Tabla que contiene información histórica única de la maestra de contratos de comercio exterior
3	Responsable	Nombre de la persona que realiza el análisis.	Daniel Ayras Olano
4	Ruta input	Ruta donde se encuentra la data de la fuente de información.	/capa/master/data/mov_tarjetas_debito
5	Número columnas	Cantidad de campos que tiene la fuente.	15

[7] Los datos mostrados en la figura son una simulación creada por el autor

6	Prom. Mensual	Cantidad mensual promedio de registros que contiene la fuente.	5 millones
7	Historia	Fecha de inicio de la ingesta de la fuente.	2015-03-01
8	Periodo ingesta	Frecuencia de carga de la fuente como diaria, mensual o anual.	Diaria
9	Días ingesta	Indica si la ejecución de carga de la fuente es en día hábil o en día calendario.	Hábil
10	Tipo ingesta	Indica si la ingesta de la fuente es acumulado o si no lo es.	Acumulado
11	Nivel data	Nivel de granularidad de la fuente.	Cliente
12	Observación	Detalle adicional como particularidades o características de las fuentes de información.	- La información del 2016-10-31 hasta 2019-04-30 con un periodo mensual. - A partir de 2019-05-02, es una ejecución diaria. - El acumulado de la data es por mes.

***Fuente*. Elaboración propia**

En la tabla 13 se describen las fuentes de información que se analizaron.

***Tabla 13*: Fuentes de información**

N°	Fuentes	Descripción	Volumetría mensual
1	Comex	Tabla que contiene información histórica de contratos de comercio exterior.	500 mil
2	Confirming	Tabla maestra de contratos de cesión de pagos a proveedores, servicio financiero que gestiona los pagos de una empresa a sus proveedores.	2.1 millones
3	Recaudos	Tabla de los convenios de recaudo monetario que existen entre distintas empresas con el banco, este servicio financiero consiste en el cobro de los recibos o facturas de clientes.	14 millones
4	Cartas fianzas	Tabla que contiene información maestra de las cartas fianza así como de avales.	7 millones
5	Seguros	Tabla maestra que almacena información de los contratos de seguros comercializados de tipo préstamos (desgravamen, inmuebles, vehiculares), optativos (retiros, protección tarjetas), garantías y leasing.	14 millones

6	Movimientos de cuentas	Tabla del movimiento de las cuentas de los clientes.	4.6 millones
7	Préstamos	Tabla de los préstamos que tienen los clientes del banco.	6 millones
8	Tarjetas	Tabla maestra de tarjetas de crédito y débito.	12 millones
9	Movimientos de tarjetas de débito y crédito	Tabla que almacena los movimientos de tarjetas de crédito y débito.	650 mil
10	Transferencias al exterior	Tabla que contiene información de las transferencias que realizan los clientes desde Perú hacia afuera.	100 mil
11	Leasing	Tabla que contiene la información maestra de los contratos de Leasing.	75 mil
12	Operaciones de cambios de divisas	Tabla que almacena las operaciones de cambio de divisas por canal de operación.	2 millones
13	Maestra de participantes	Tabla que contiene la relación de los contratos con el código del cliente.	87 millones
14	Maestra de clientes	Tabla que contiene información de los clientes.	9 millones

***Fuente*. Elaboración propia**

3.2.5.2. Definición de reglas de negocio

En esta etapa en base al conocimiento obtenido del pre procesamiento, con apoyo del product owner y el scrum team se definieron las reglas de negocio para el cálculo de las variables, se definió como abordar las fuentes y que variables eran factibles crear por cada fuente, así mismo se definieron los campos finales del modelo que serían plasmados en el diccionario de datos del modelo a fin de asegurar el gobierno de datos en la entidad bancaria.

A continuación, se detallan las actividades realizadas durante esta etapa:

- El product owner delegó las 14 fuentes de información analizadas en la etapa 1 a cada big data engineer del scrum team para abordar los procesamientos de datos para crear las variables.
- Se definieron las transformaciones que debían realizarse como cruces, eliminación de duplicados, filtros, agrupamientos, etc. a fin de calcular las variables por cada fuente.

- El product owner se reunió con cada big data engineer para analizar si estas fuentes contaban con el código del cliente.
- Se identificó que las fuentes que no contaban con el código del cliente podían obtenerlo a través de 2 fuentes, la maestra de Participantes y Clientes.

El apoyo del equipo de España fue de alto valor en esta etapa porque el product owner pudo conocer que variables habían considerado ellos y como las calcularon, ese conocimiento sumado análisis realizado en el pre procesamiento realizado permitió definir las variables por cada fuente de información que se muestran en la tabla 14, cabe resaltar que las variables fueron creadas a nivel de cliente.

***Tabla 14*: Variables del modelo**

N°	Fuentes	Variables
1	Comex	Número de contratos vencidos comex. Número de contratos vigentes comex. Importe total de contratos vigentes financiados de comex.
2	Confirming	Número total de órdenes de pago confirming. Número meses por vencer del contrato más antiguo confirming. Importe de órdenes de pago pendientes por pagar confirming.
3	Recaudos	Número de convenios vigentes de recaudo. Número de convenios vencidos de recaudo. Importe de movimientos de convenios vigentes de recaudo. Número de movimientos de convenios de pago de recaudo. Importe de movimientos de convenios de pago de recaudo.
4	Cartas fianzas	Saldo actual de las cartas fianzas activas
5	Seguros	Número de seguros vigentes. Importe total de seguros vigentes mensual.
6	Movimientos de cuentas	Número de cuentas activas en dólares. Importe promedio de movimientos de ingresos últimos 6 meses. Número de movimientos de ingresos de últimos 6 meses. Importe promedio de movimientos de egresos últimos 6 meses. Número de movimientos de egresos de últimos 6 meses.
7	Préstamos	Número total de préstamos activos. Número de créditos de consumo activos del cliente. Número de créditos hipotecarios activos del cliente.

		Número de créditos comerciales activos del cliente. Importe disponible de los prestamos activos.
8	Tarjetas	Número de tarjetas débito vigentes. Número de tarjetas crédito vigentes.
9	Movimientos de tarjetas de débito y crédito	Número de movimientos de tarjeta debito en dólares últimos 6 meses. Importe promedio de movimientos de tarjeta debito en dólares últimos 6 meses. Número de movimientos de tarjeta crédito en dólares últimos 6 meses. Importe promedio de movimientos de tarjeta crédito en dólares últimos 6 meses.
10	Transferencias al exterior	Importe de transferencias al exterior de últimos 6 meses. Importe promedio de transferencias al exterior de últimos 6 meses. Número de transferencias al exterior de últimos 6 meses.
11	Leasing	Número de leasing en estado activo. Número máximo de meses por pagar de leasing. Importe total pagos de leasing.
12	Operaciones de cambios de divisas	Número de movimientos liquidados. Número de movimientos cotizados. Porcentaje por canal de operación en los últimos 6 meses. Número de movimientos en mercado abierto del último mes. Número de movimientos en mercado cerrado del último mes.
13	Maestra de participantes	Fuente auxiliar para obtener el código del cliente.
14	Maestra de clientes	Fuente auxiliar para obtener el código del cliente.
TOTAL DE VARIABLES		**41**

***Fuente*. Elaboración propia**

3.2.5.3. Diseño del flujo de procesamientos

En esta etapa el Scrum Team analizó la forma más óptima de secuenciar los procesos a construir, cabe resaltar que los procesos consistieron en desarrollar las variables por cada fuente de información según las reglas de negocio definidas en la etapa anterior y que se muestran en la tabla 14.

Se identificaron las transformaciones que se tenían en común en varios procesamientos a fin de evitar desarrollos redundantes, es decir no realizar las mismas transformaciones en cada procesamiento, agrupándolas en procesos previos.

Esto mejoró el rendimiento de la ejecución de los procesamientos. Por ejemplo, en la figura 15 se observa lo siguiente:

- Las fuentes de Comex y Recaudos no tienen el campo del código de cliente.
- Estas fuentes dependen de las maestras de Participantes y Clientes para obtener el código del cliente.
- Esas dos maestras deben pasar por un proceso de transformación y limpieza de datos para obtener correctamente el código del cliente.
- Para evitar hacer el mismo proceso de transformación y limpieza de ambas maestras en los procesos de Comex y Recaudos, se optó por hacerlo en procesos previos que generen una tabla temporal que sirva como fuente de entrada a los procesos de Comex y Recaudos.

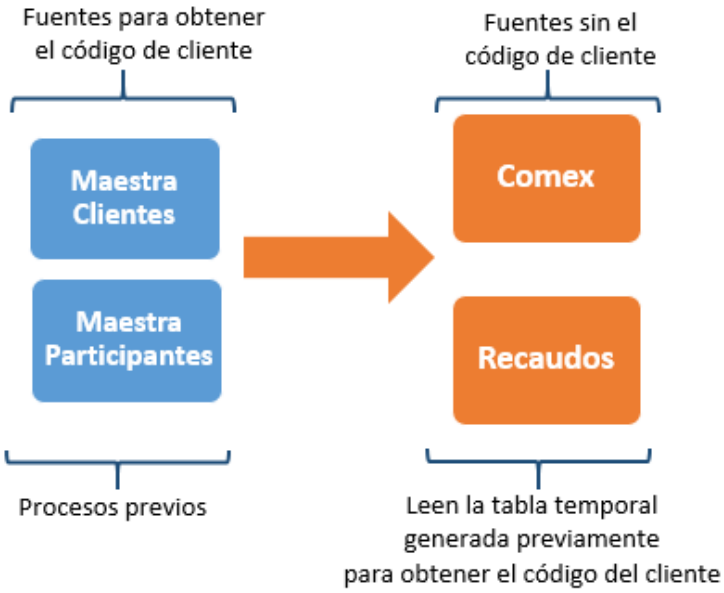


***Figura 15*: Evitar redundancia de procesamientos. Elaboración propia**

Las actividades llevadas a cabo durante esta etapa son las siguientes:

- Se listaron las fuentes de entrada por cada procesamiento para identificar que transformaciones se tienen en común en los procesamientos.
- Se analizó que procesos podían ejecutarse en paralelo.
- Se validó que fuentes no contaban con el código del cliente para que los procesos de esas fuentes dependan de los procesos de las maestras de Participantes y Clientes.

Como consecuencia de lo señalado anteriormente se llegó a las siguientes conclusiones:

- Se identificaron un total de 15 procesos para creación de variables del modelo, de los cuales 2 procesos permitieron obtener el código del cliente, 12 procesos calcular las variables del modelo y 1 proceso consolidar las variables y formar finalmente la tabla del modelo.
- Se identificaron 7 fuentes que dependían de las maestras de Participantes y Clientes para obtener el código del cliente.
- La maestra de Participantes permitió obtener la relación entre el código del cliente y el código de contrato de los servicios financieros.
- La maestra de Clientes permitió obtener la relación entre el código del cliente y el documento de identificación del cliente.
- Se identificaron 7 procesos que podían ejecutarse en paralelo, de los cuales 5 procesos fueron para crear variables que si contaban con el código del cliente y 2 son los procesos correspondientes a las maestras de Participantes y Clientes.
- Se decidió crear un proceso final para consolidar todas las variables y crear la tabla del modelo final, cabe resaltar en este punto que los 14 procesos previos a este último generaron tablas temporales y solo este último generó la tabla persistente con las variables del modelo.

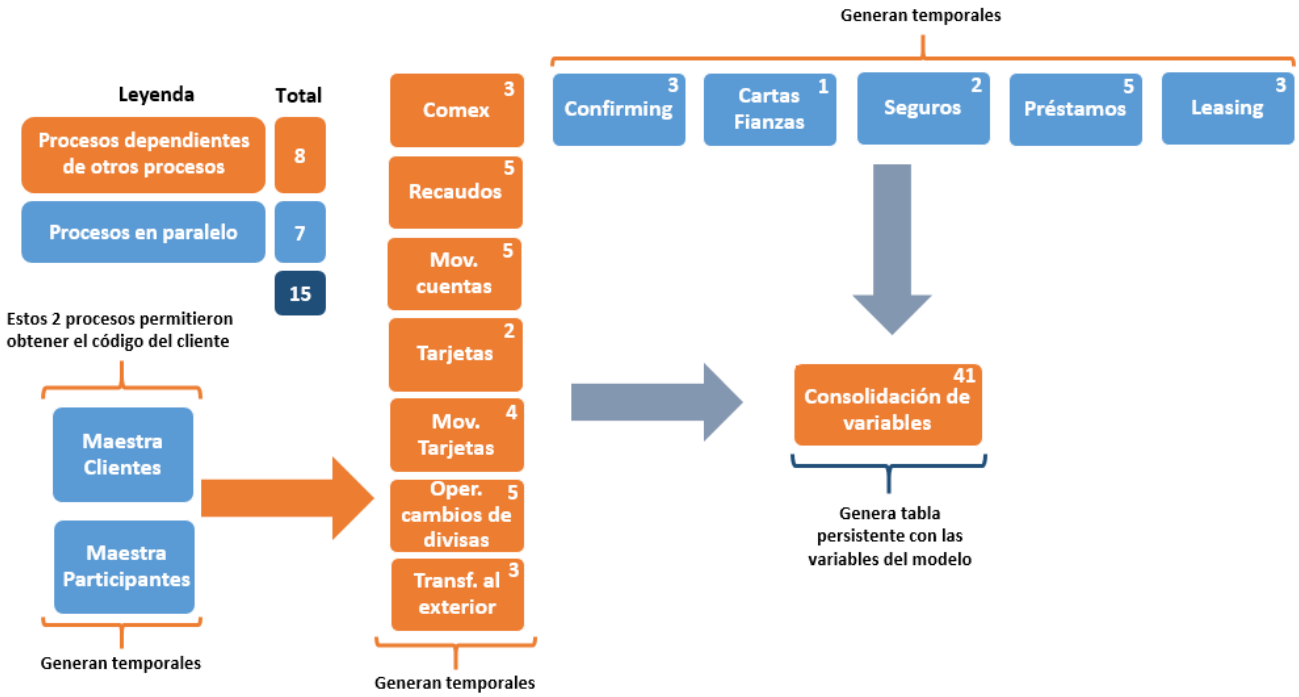


***Figura 16*: Diseño flujo de procesamientos. Elaboración propia**

En la figura 16 se puede observar el diseño a alto nivel producto del análisis realizado en esta etapa con respecto a los 15 procesos identificados, en la esquina superior derecha se indica la cantidad de variables generadas por cada proceso.

3.2.5.4. Desarrollo de los procesamientos de datos

En esta etapa el scrum team abordó el desarrollo de los procesamientos de las fuentes de información para el cálculo de las variables, durante esta etapa se desarrollaron las siguientes actividades:

- Se desarrollaron los 15 procesamientos identificados en la etapa anterior con las tecnologías Apache Spark versión 2.4. y Python versión 3 a través de Jupyter Notebook disponibles en el ambiente Sandbox del entorno big data de la entidad bancaria.
- Con apoyo del product owner se afinaron las reglas de negocio en base a las observaciones de los datos que se fueron identificando durante el desarrollo.
- Los desarrollos de los procesamientos de datos se hicieron en base a las buenas prácticas de desarrollo definidas por la entidad bancaria. Se describe con más detalle este punto en la siguiente etapa de puesta en producción.
- Se contempló el desarrollo de pruebas unitarias de las funciones desarrolladas en los procesamientos a fin de asegurar la calidad de estos.
- En el ambiente local[8] se estructuraron los procesamientos según el arquetipo Scaffolder proporcionado por el área de Arquitectura de la entidad bancaria, esta estructuración fue necesaria para la puesta en producción de los procesamientos.

En la figura 17 se puede observar la arquitectura a alto nivel de la solución, en los siguientes puntos se detalla en que consistió el desarrollo de los procesamientos.

[8] El ambiente local hace referencia a la computadora o laptop personal del programador.

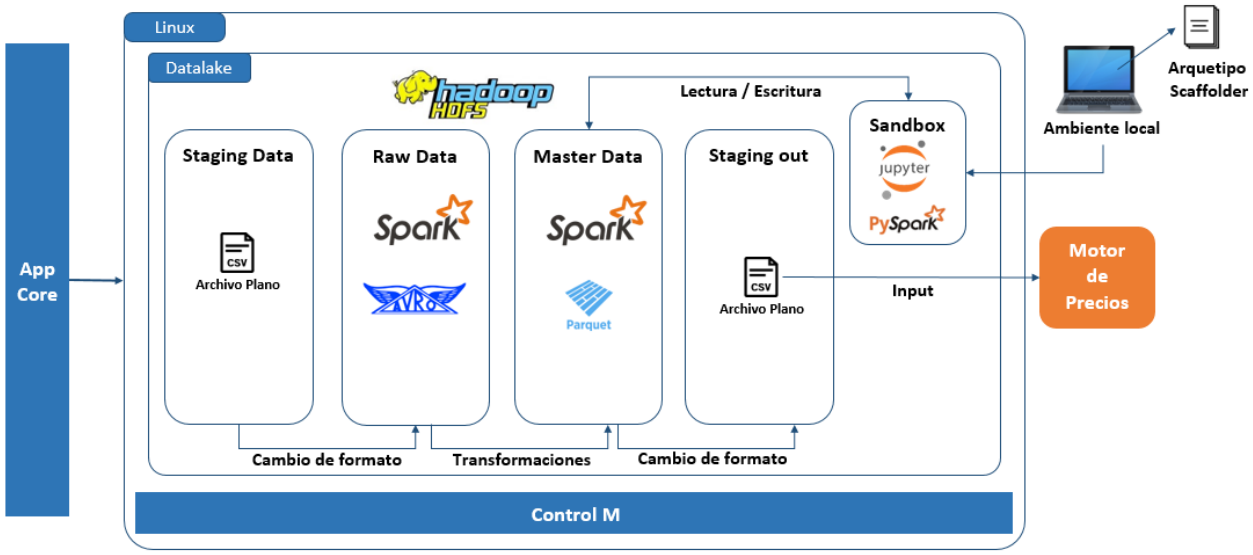


***Figura 17*: Arquitectura alto nivel de la solución. Elaboración propia**

A continuación, se describe la arquitectura a alto nivel de la solución mostrada en la figura 17:

- La entidad bancaria cuenta con un datalake que está desplegado sobre un clúster on premise, con Hadoop como sistema de archivos distribuido.
- El datalake cuenta con 4 capas:
  - **Staging Data**: Capa donde llega la información de las aplicaciones core de la entidad bancaria, por ejemplo, información de operaciones de cambio de divisas, seguros, prestamos, movimientos de tarjetas, etc. en general de todas las fuentes que se muestran en la tabla 13.
  - **Raw Data**: Capa donde se almacenan los datos en crudo, sin ninguna transformación, solo se realiza un cambio de formato a Avro.
  - **Master Data**: Capa donde se almacenan los datos preparados y optimizados para su explotación en formato Parquet, la explotación de datos que se realizó en el proyecto fue realizado en esta capa.
  - **Staging Out**: Capa donde se almacenan los datos preparados y optimizados para el uso externo, se generó un archivo con formato csv para que sirva como fuente de entrada para el motor de precios.
- Las 14 fuentes que permitieron crear las variables se encuentran en la capa master data del datalake, en archivos con formato Parquet y compresión Snappy.

- Los desarrollos de los 15 procesamientos se realizaron sobre la plataforma Sandbox.
- El ambiente Sandbox cuenta con la herramienta web Jupyter Notebook, esta permitió la lectura, transformación y escritura de datos en la capa master data del Datalake.
- La estrategia que se siguió para el desarrollo de variables fue construir un Notebook por cada proceso, es decir se desarrollaron 15 Notebooks en total, 12 Notebooks para la creación de variables, 2 Notebooks para el procesamiento de las fuentes que permitieron obtener el código del cliente y 1 Notebook para la consolidación de las variables.
- En cada Notebook se crearon las funciones que permitieron abordar las reglas de negocio definidas en la etapa de definición de reglas de negocio.
- El ambiente local permitió la estructuración de las funciones desarrolladas en las Notebooks usando el arquetipo Scaffolder, este arquetipo permitió que todos los procesos desarrollados puedan desplegarse en el ambiente de producción del datalake ya que contaba con los archivos de configuración necesarios para la conexión al ambiente productivo del datalake.
- Se desarrolló un proceso Scaffolder por cada Notebook, es decir se desarrollaron 15 procesos Scaffolder.

En este punto cabe resaltar que el modelo final es una fuente de entrada para el motor de precios de la entidad bancaria (esto se puede observar en la figura 16), la generación del spread comercial para las operaciones de cambio de divisa se hacía por segmentos de clientes, ahora con la creación del modelo de datos el motor tiene una nueva fuente de entrada con variables personalizadas a nivel de cliente.

En la figura 18 se puede observar la estructura del arquetipo Scaffolder, se compone de apartados para ordenar las funciones de negocio y utilitarias, centralizar las constantes y campos de entrada de las fuentes procesadas, invocar las funciones según la lógica del procesamiento y escritura de los datos resultantes de todas las transformaciones desarrolladas en el datalake y un espacio para realizar pruebas unitarias de las funciones creadas.

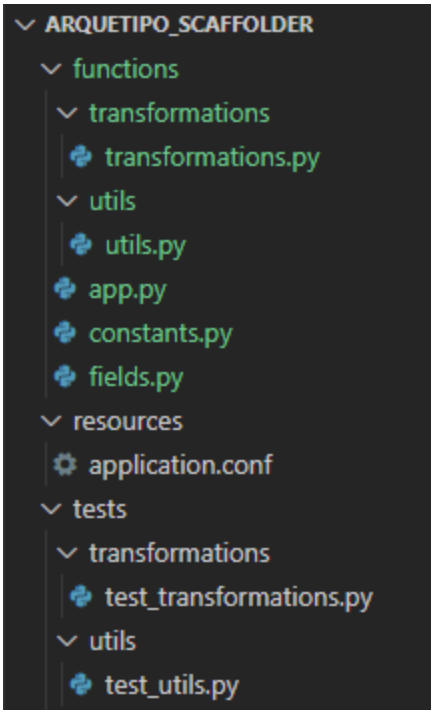


***Figura 18*: Arquetipo Scaffolder. Elaboración propia**

En los siguientes puntos se describe a detalle los apartados del arquetipo:

- En el archivo **transformations.py** se implementaron las reglas de negocio para la creación de variables, por ejemplo, filtros, deduplicación de registros, cruces entre fuentes, en general todas las transformaciones necesarias para crear las variables del modelo de datos. En la figura 19 se observa un ejemplo de una función de negocio donde se agrupan los datos de un dataframe a nivel de cliente y se calcula la sumatoria de montos de las operaciones realizadas.

```
def group_variables_by_total_customer(self, total_variables_channel_df):
    """
    Operations are grouped by customer and service channel.
    :param total_variables_channel_df: dataframe with customer operations
    :return: grouped dataframe
    """
    total_variables_channel_df = total_variables_channel_df \
        .groupBy(Fields.CUSTOMER_ID_FIELD,
                 Fields.FOREIGN_EXCHANGE_FIELDS['VARIABLES_CHANNEL_TYPE']) \
        .agg(sum(Constants.TOTAL_LOCAL_CURRENCY_AMOUNT).alias(Constants.TOTAL_PAYMENT_AMOUNT))
    return total_variables_channel_df
```

Agrupar por cliente

Suma de montos

***Figura 19*: Función transformación. Elaboración propia**

- En el archivo **utils.py** se implementaron las funciones utilitarias para el procesamiento de las fuentes de información, por ejemplo, funciones para crear

campos auxiliares, artificios para el tratamiento de los datos, lectura o escritura de archivos parquets o asignar el tipo de dato a los campos de un dataframe. En la figura 20 se observa una función utilitaria que donde se cambian los tipos de datos de los campos de un dataframe.

```
def cast_output_fields(self, final_variables_customer_df):
    """
    The dataframe fields are matched with the dynamic price variables.
    :param final_variables_customer_df: dataframe with dynamic pricing variables
    :return: dataframe with caste data types
    """
    final_variables_customer_df = final_variables_customer_df \
        .select(col(Fields.CUSTOMER_ID_FIELD).cast('string'),
                col(Fields.PROCESS_DATE_PERIOD_ID_FIELD).cast('string'),
                col(Fields.VARIABLES_FOREIGN_FIELDS['TOTAL_CLOSE_MARKET_NUMBER']).cast('integer'),
                col(Fields.VARIABLES_FOREIGN_FIELDS['TOTAL_OPEN_MARKET_NUMBER']).cast('integer'),
                col(Fields.VARIABLES_FOREIGN_FIELDS['TOTAL_LIQUIDATED_CHANGE_NUMBER']).cast('integer'),
                col(Fields.VARIABLES_FOREIGN_FIELDS['TOTAL_QUOTED_CHANGE_NUMBER']).cast('integer'),
                col(Fields.VARIABLES_FOREIGN_FIELDS['OPERATIONS_ANALYTICS_PERCENTAGE']).cast('decimal(23,10)'),
                col(Fields.VARIABLES_FOREIGN_FIELDS['OPERATIONS_CHANGE_PERCENTAGE']).cast('decimal(23,10)'),
                col(Fields.VARIABLES_FOREIGN_FIELDS['OPERATIONS_MASSIVE_PERCENTAGE']).cast('decimal(23,10)'),
                col(Fields.VARIABLES_FOREIGN_FIELDS['OPERATIONS_ADVANCE_PERCENTAGE']).cast('decimal(23,10)'),
                col(Fields.VARIABLES_FOREIGN_FIELDS['OPERATIONS_CASHIER_PERCENTAGE']).cast('decimal(23,10)')) \
        .cache()
    return final_variables_customer_df
```

Casteo de datos

***Figura 20*: Función utilitaria. Elaboración propia**

- En el archivo **constants.py** se definieron las constantes que se usaron en el procesamiento, como buena práctica es recomendable tener los valores en duro que se van a usar durante todo el procesamiento centralizados en un archivo, con nombres claros, si en un futuro se requiera hacer un mantenimiento del desarrollo, esto facilitará interpretar el código desarrollado. En la figura 21 se observa la definición de constantes y diccionarios que son usados durante el desarrollo de un procesamiento.

```
DUPLICATE_CUSTOMER_CODE_NUMBER = 1
COUNTER_NUMBER = 0
MAXIMUM_ANALYSIS_PERIOD_NUMBER = 6
DOLLAR_CURRENCY_TYPE = 'USD'
INDICATOR_SUCCESS_TYPE = '_SUCCESS'

DATE_PROCESS_NUMBERS = {
    'MONTH_JANUARY': 1,
    'MONTH_JUNE': 6,
    'MONTH_DECEMBER': 12,
    'ADD_ONE_MONTH': 1,
    'MINUS_ONE_YEAR': 1,
    'ADD_ONE_YEAR': 1,
    'MINUS_ONE_DAY': 1
}
```

Constante para la moneda dolar

Dicionario para el tratamiento de fechas

***Figura 21*: Definición de constantes. Elaboración propia**

- En el archivo **fields.py** se definieron los campos de las fuentes de entrada del procesamiento, Python permite crear diccionarios de datos para agrupar ítems que son parte de un grupo con un mismo concepto. En la figura 22 se observa la definición de un diccionario con 3 constantes, estos representan los campos de entrada de una fuente de información para un determinado procesamiento.

```
SPECTRUM_AUXILIARY_FIELDS = {
    'CUSTOMER_CODE_ID': 'auxiliary_info_desc',
    'INTERNAL_CUSTOMER_CODE_ID': 'field_id',
    'TOTAL_CUSTOMER_NUMBER': 'total_customer_number'
}
```

***Figura 22*: Definición de campos de entrada. Elaboración propia**

- En el archivo **app.py** se invocaron a las funciones de negocio y utilitarias para realizar las transformaciones necesarias para crear las variables del modelo de datos y finalmente la escritura en la capa Master Data del datalake. En la figura 23 se observa la invocación de las funciones que permiten crear variables que serán parte del modelo de datos, en las últimas líneas se observa la escritura del dataframe en hdfs con formato parquet.

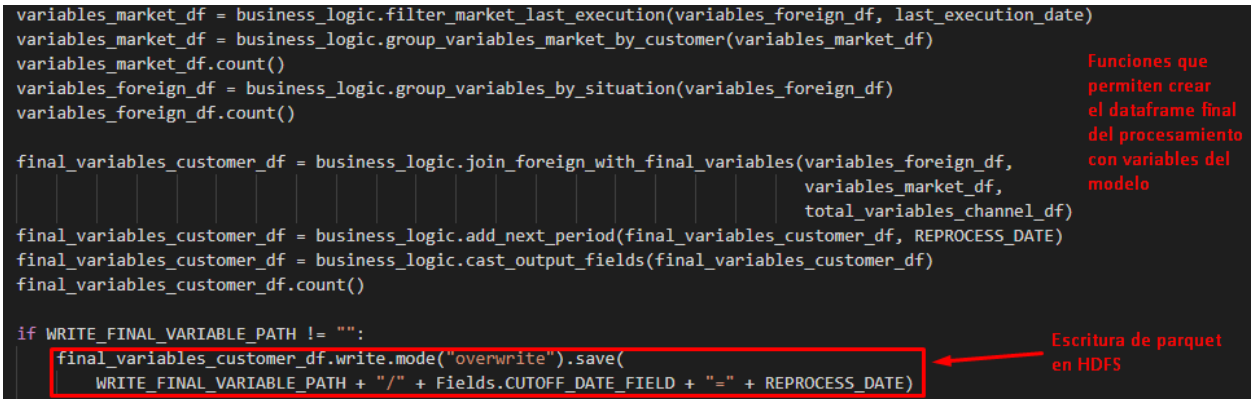


***Figura 23*: Invocación de funciones y escritura. Elaboración propia**

- En el archivo **application.conf** se definieron los parámetros de entrada por parte de Control-M, esto con el objetivo que el proceso tenga una ejecución dinámica, el parámetro más importante en todos los procesos fueron las fechas de lectura de las fuentes de entrada, cada vez que se ejecuten los procesos, Control-M está configurado para que envié como parámetro la fecha de lectura

correspondiente de las fuentes. En la figura 24 se observan los parámetros de un archivo de configuración como fechas y rutas.

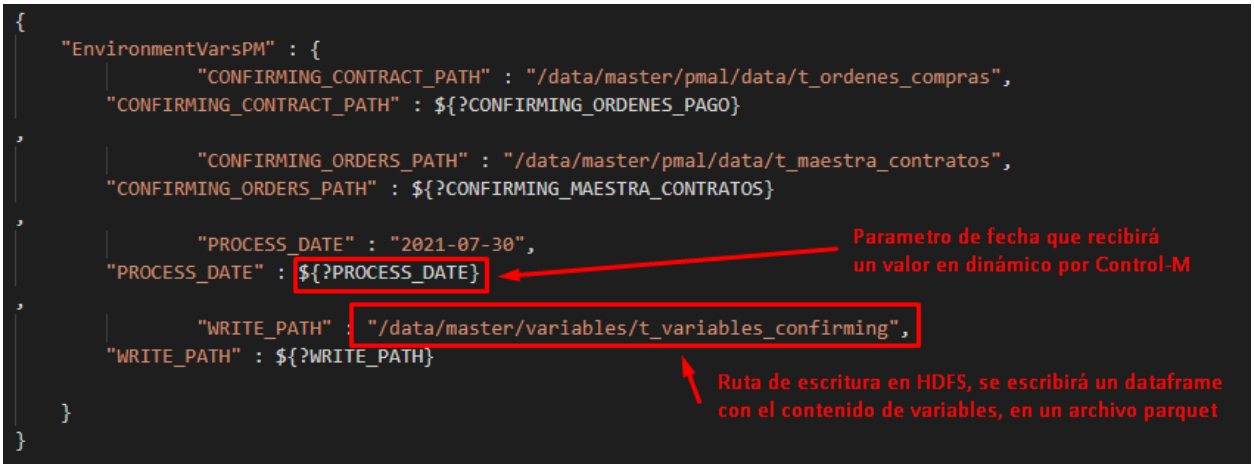


***Figura 24*: Archivo de configuración. Elaboración propia**

- En los archivos **test_transformations.py** y **test_utils.py** se desarrollaron las pruebas unitarias de las funciones de negocio y utilitarias respectivamente, esto con el objetivo de asegurar que las funciones definidas van a ejecutar correctamente y realizar la transformación esperada. En la figura 25 se observa una prueba unitaria de una función la cual se realiza mediante la librería pytest que proporciona Python, básicamente la función mediante un subtract (similar a un minus en SQL) valida los valores que debe devolver una función.

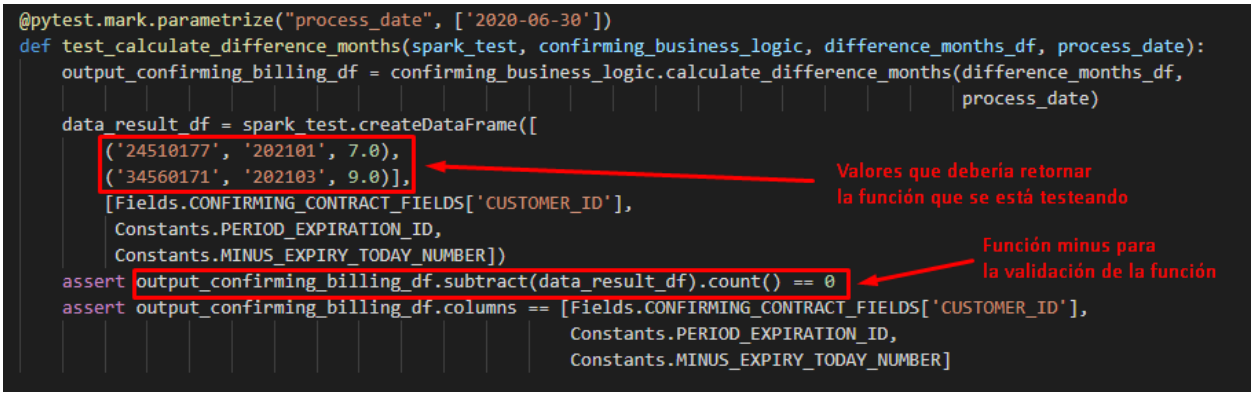


***Figura 25*: Pruebas unitarias. Elaboración propia**

3.2.5.5. Puesta en producción

En esta etapa se pasó a producción los 15 procesamientos de datos desarrollados con el arquetipo Scaffolder, para este objetivo se usaron herramientas de integración continua que facilitaron la puesta en producción de los procesos

Scaffolder, así mismo se interactuó con el equipo de DQA que permitió asegurar la calidad de los procesamientos de datos desarrollados.

En la figura 26 se puede observar el diagrama de interacción con el equipo DQA para la puesta en producción de los procesos Scaffolder.

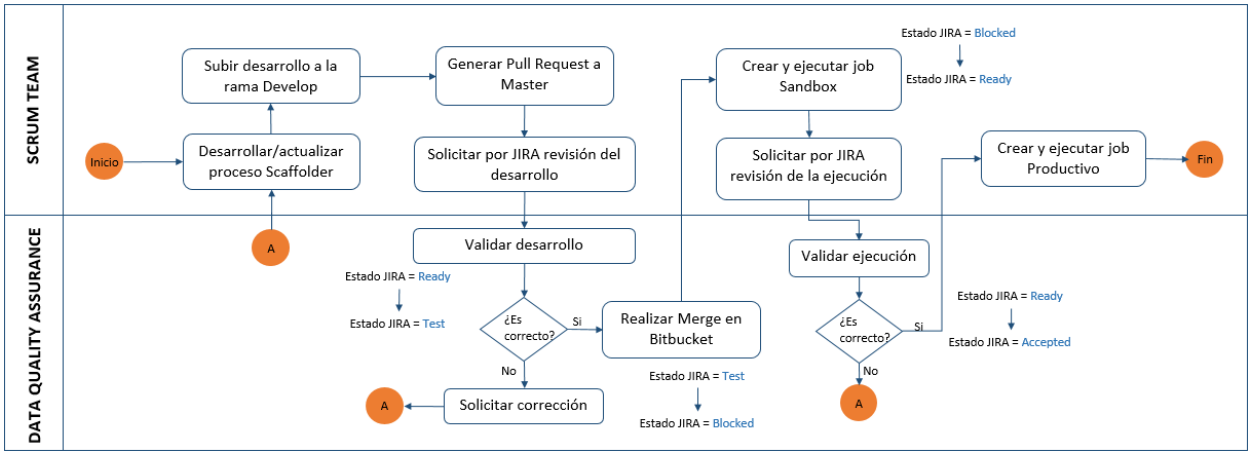


***Figura 26*: Diagrama de interacción con DQA. Elaboración propia**

El scrum team se encargó de subir cada proceso Scaffolder del ambiente local a la rama develop de un repositorio Bitbucket, posteriormente a través de Jira se solicitó la revisión de una pull request (PR) para que el equipo DQA pueda revisar el desarrollo, adicionalmente en Jira se adjuntó un documento Excel donde se detalló cada campo de la tabla resultante del proceso Scaffolder (ver anexo 2).

La revisión del equipo DQA consistió en validar si se cumplieron los estándares de desarrollo para procesamientos de datos, a continuación, se muestran algunos de los estándares:

- No utilizar "*" para importar librerías. Importar solo librerías que se usarán en el desarrollo. Por ejemplo, se puede observar en la parte izquierda de la figura 27, la importación de librerías que serán usadas durante el desarrollo y en la parte derecha el uso del comodín "*" para importar todo el contenido de algunas librerías.

CORRECTO

```
from pyspark.sql.functions import col
from datetime import datetime
from decimal import Decimal
from pyspark.sql.types import StructType, StructField, DateType
```

INCORRECTO

```
from pyspark.sql.functions import *
from pyspark.sql.types import *
```

***Figura 27*: Definición de librerías. Elaboración propia**

- Las constantes se escriben en mayúsculas con guiones bajos que separan las palabras, el nombre de la constante debe dar a entender claramente el uso de esta. Por ejemplo, se puede observar en la parte izquierda de la figura 28, la definición de constantes con nombres claros y dan a entender un determinado concepto y en la parte derecha se definen constantes con nombres que no dan a entender un concepto claro.

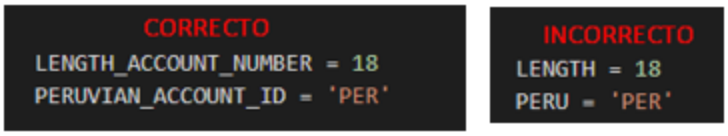


***Figura 28*: Definición de constantes. Elaboración propia**

- Las variables se escriben en minúsculas con guiones bajos que separan las palabras, el nombre de la variable debe dar a entender claramente el uso de esta. Por ejemplo, se puede observar en la parte superior de la figura 29, la definición de una variable para un dataframe con un nombre claro y da a entender un determinado concepto y en la parte inferior se define una variable con un nombre muy genérico sin un concepto claro.

***Figura 29*: Definición de variables. Elaboración propia**

- Solo métodos, funciones y clases deben tener comentarios. Este comentario será para describir su funcionalidad. En la figura 30 se observan los comentarios realizados sobre una función, esta se compone de 3 apartados donde se describe el objetivo de la función, la descripción de los parámetros de entrada y que retorna la función.

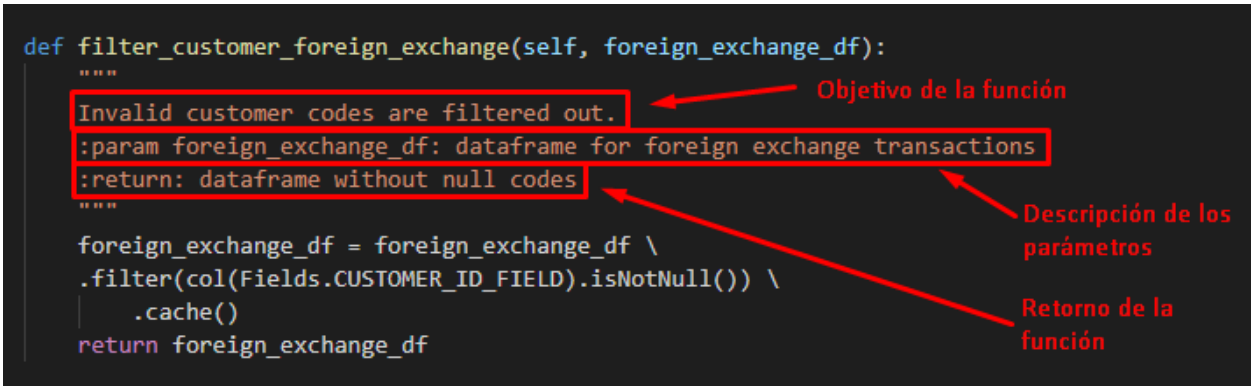


***Figura 30*: Comentarios de una función. Elaboración propia**

- Las transformaciones deben cumplir con los siguientes puntos:
    - Utilizar Datasets and DataFrames de Spark para aprovechar Spark, en lugar de los RDD.
    - Comprobar que no haya funciones de impresión de datos sensibles en consola durante la ejecución del procesamiento.
    - Evitar el abuso de la función withColumns porque genera nuevas tablas a partir del original en cada llamada.
    - Seleccionar solo las columnas necesarias para los cruces.
    - Las tablas en las rutas temporales se deben eliminar periódicamente.
    - Usar el módulo Spark SQL en lugar de custom UDFs porque es un SQL amigable y natural basado en una API y ofrece optimizaciones adicionales.

Con relación a la figura 26, cuando DQA encontró falencias en el código fuente de los procesos Scaffolder, estos fueron observados y devueltos al scrum team para ser actualizados, los pasos posteriores de interacción con el equipo DQA son los siguientes:

- Cuando los desarrollos ya no tenían observaciones por parte del equipo DQA, realizaron el merge de la PR el cual consiste en pasar el desarrollo de la rama develop a la rama master en el repositorio Bitbucket.
- Luego del merge de la PR el scrum team creó y ejecutó los jobs asociados a los desarrollos Scaffolder en el ambiente Sandbox del datalake.
- Posteriormente el equipo DQA revisó las evidencias de la ejecución y si no encontraba errores de ejecución daba como aceptado el desarrollo Scaffolder.

- Luego el scrum team creó y ejecutó los jobs productivos, estos jobs se asociaron a las rutas productivas de las fuentes de información ubicadas en la capa Master Data del datalake, es decir, los jobs en este punto ya estaban preparados para procesar la información del ambiente de producción de las fuentes que se muestran en la tabla 12.

Las herramientas de IC fueron de gran impacto en el proyecto para que el scrum team y el equipo DQA eviten procesos manuales e inviertan más tiempo en otras tareas productivas, las herramientas IC que se usaron en el proyecto fueron Bitbucket, Jira y Jenkins.

En la figura 31 se observa el flujo de interacción con las herramientas de IC.

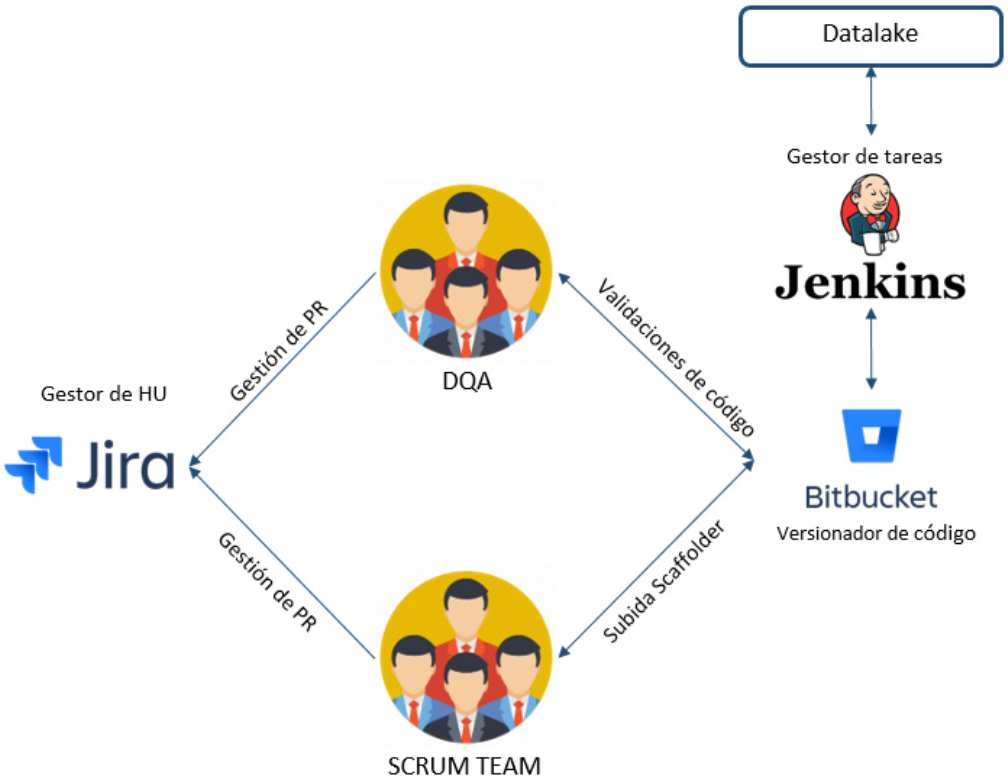


***Figura 31*: Flujo de interacción con herramientas de IC. Elaboración propia**

A continuación, se describe la interacción que hubo con las herramientas de integración continua:

- El repositorio Bitbucket permitió al scrum team exportar las Notebooks desarrolladas en el ambiente Sandbox del datalake a un ambiente local.

- En el ambiente local el scrum team pudo estructurar todas las funciones desarrolladas en las Notebooks en el arquetipo Scaffolder y posteriormente subirlo al repositorio Bitbucket.
- A través de Jira el scrum team y el equipo DQA pudieron gestionar las solicitudes de revisión de los desarrollos en Bitbucket y ejecuciones en Sandbox (ver anexo 1).
- Bitbucket también permitió que el equipo DQA pueda validar los desarrollos, a través de esta herramienta pudieron indicar observaciones en caso los desarrollos tuvieran falencias.
- Jenkins se encargó de realizar de forma automática el despliegue y las ejecuciones de los jobs asociados a los procesos Scaffolder en el ambiente sandbox del datalake. También permitió crear y ejecutar los jobs en el ambiente productivo del datalake.

En la figura 32 se observan las tareas de despliegue de forma gráfica, esto permitió al scrum team validar de forma visual si el despliegue realizado por Jenkins se hizo correctamente.

***Figura 32*: Despliegue en Jenkins. Elaboración propia**

A continuación, se detallan las tareas que se programó en Jenkins para el despliegue de los procesamientos con el arquetipo Scaffolder:

- **Checkout Global Library**: Jenkins desde el inicio del despliegue se encarga de validar las librerías utilizadas en el procesamiento, la versión de Apache Spark y Python que serán utilizados en el desarrollo.
- **Samuel Pre Build Stage**: Luego por medio del motor de calidad (Samuel) se valida que el código fuente cumpla con reglas de seguridad como no exponer datos sensibles.
- **Python Worker Verify**: Luego reserva un worker, con ciertos recursos computacionales como la memoria RAM y CPU para la ejecución del procesamiento.

- **Python Worker Container Build**: Se copian los archivos de BitBucket al datalake para su posterior ejecución.
- **Python Worker Container Test & Package**: Se ejecutan las pruebas unitarias desarrolladas en los archivos de test, test_transformations.py y test_utils.py.
- **Samuel Pre Deploy Stage**: Se valida que los resultados obtenidos de las pruebas unitarias sean satisfactorios, requisito obligatorio para la ejecución del archivo app.py.
- **Python module Deploy**: Se ejecuta el archivo principal del arquetipo Scaffolder, el archivo app.py el cual invoca todas las funciones de negocio y utilitarias, posteriormente se realiza la escritura de los dataframes resultantes en hdfs.
- **Release**: Se genera un reporte indicando si fue o no satisfactorio las ejecuciones de las tareas anteriores.
- **Publish on Version Tracker**: Se envía un indicador a Bitbucket si todas las tareas ejecutadas se realizaron correctamente.

3.2.5.6. Implementación de la malla de procesos

En esta etapa se elaboró la malla (conjunto de jobs de Control-M) que permitió automatizar la ejecución de los 15 procesamientos de datos que fueron desplegados en producción en la etapa anterior, para este proceso se utilizó la herramienta Control-M, con esta herramienta se organizó el flujo de los procesamientos en el orden y con las dependencias necesarias para crear el modelo de datos.

Cabe resaltar que los jobs de los procesamientos creados en la etapa de puesta en producción (procesos Scaffolder) no se pueden ejecutar de forma automática por sí mismos, es por eso la importancia de los jobs Control-M para que estos se encarguen de ejecutar de forma automática los procesos Scaffolder que van a crear las variables del modelo de datos.

La entidad bancaria por lineamientos internos indicó que todos los proyectos de Perú relacionados a procesamientos de datos en el datalake deben canalizar la creación de mallas con el equipo de Soporte Control-M de la sede de México de la entidad bancaria.

En la figura 33 se observa la interacción entre el scrum team y el equipo de Soporte Control-M para la implementación de la malla.

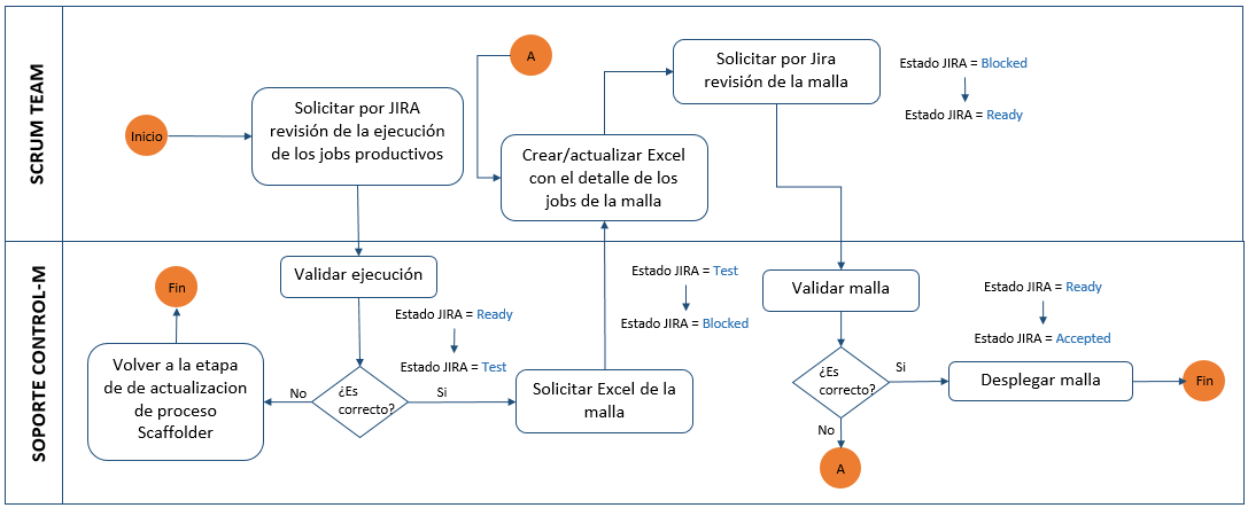


***Figura 33*: Diagrama de interacción con Soporte Control-M. Elaboración propia**

En los siguientes puntos se describe la interacción entre el scrum team y el equipo de Soporte Control-M con el propósito de instalar la malla que permita automatizar la ejecución de los procesos Scaffolder:

- Como punto de partida el scrum team envió las evidencias de ejecución de los 15 jobs productivos al equipo de Soporte Control-M para que validen si los jobs ejecutaron satisfactoriamente, la solicitud se hizo mediante Jira.
- Cuando el equipo de Soporte Control-M validó que todos los jobs se ejecutaron correctamente, solicitaron por medio de Jira un Excel con el detalle de todos los jobs que se requerían automatizar con Control-M, cabe resaltar en este punto que los jobs se configuraron para que ejecuten de forma mensual.
- En esta etapa fue fundamental el diseño a alto nivel que se realizó en la etapa de diseño del flujo de procesamientos porque el diseño elaborado sirvió como base para armar la malla de procesos, específicamente para las condiciones de entrada y salida de los jobs, en la tabla 15 se detallan estos dos puntos.

En la tabla 15 se observa la estructura del Excel que solicitó el equipo de Soporte Control-M para la implementación de la malla de procesos, se describe y se dan ejemplos por cada campo del Excel.

***Tabla 15*: Estructura de Excel con detalle de los jobs Control-M**

N°	Campo	Descripción	Ejemplos
1	Nombre UUAA	Unidad aplicativa[9] asignada al proyecto, es un código de 4 dígitos que se utiliza como referencia en las rutas de escritura en los procesamientos.	PFAN, PKOG, PRED
2	Nombre Productivo	Nombre asignado al job productivo.	PFANCP9E01, PKOGC0020, PREDCP9F90
3	Frecuencia de Ejecución	Se indica la frecuencia de ejecución del job.	DIARIO, MENSUAL, ANUAL
4	Tipo de Job	Alias del servidor donde se va a ejecutar el job.	DATALAKE, APX
5	Sub Tipo de Job	Capa donde se va ejecutar el job.	RAW, MASTER, STAGING_OUT
6	Nombre del Objeto a Invocar	Id del job productivo, es autogenerado al crear el job productivo.	pfan-pe-psp-biz-pvarmstrpeoplefxpbifw28wkvv-01
7	Nombre del Insumo	Nombre del objeto donde va escribir el job.	tabla_maestra_clientes, tabla_movimientos_tarjetas
8	Tipo de Calendario	Se indica si la ejecución será en días hábiles o en días calendario.	HÁBIL, CALENDARIO
9	Nro. Máximo de Días en Activo	Cantidad de días que permanecerá el log de la ejecución disponible.	0, 1, 2, 3
10	Condición de Entrada	Condiciones requeridas para que el job inicie su ejecución. Son los jobs que deben terminar de ejecutar antes de la ejecución del job, van separados por comas.	PREDCP9F90, PKOGC0020
11	Condición de Salida	Nombre de los jobs que pueden ejecutarse posterior a la ejecución del job, van separados por comas.	PSAGCP9E01, PSAGCP9044
12	Parámetro 1	Campos libres para asignar variables, valores o fórmulas.	Se admiten 10 parámetros como máximo. Los parámetros son dinámicos, es decir, no van valores en hard code(literal), van definidos por nomenclaturas definidas por la entidad bancaria, por ejemplo:
13	Parámetro 2		
14	Parámetro 3		
15	Parámetro 4		
16	Parámetro 5		

[9] Conjunto de fuentes de información afines.

17	Parámetro 6		Fecha del día de ejecución en el servidor: %%$OYEAR-%%OMONTH-%%ODAY
18	Parámetro 7		
19	Parámetro 8		Fecha de ejecución del servidor menos un mes: %%$CALCDATE %%$OYEAR.%%OMONTH.01 -1
20	Parámetro 9		
21	Parámetro 10		
22	Nro. Registro del Autor	Id del data engineer responsable de la malla.	DE62839

***Fuente*. Elaboración propia**

Cuando se envió el Excel con el detalle de cada job, el equipo de Soporte Control-M validó si los datos enviados son correctos y analizó el impacto de los nuevos jobs que se deseaban añadir en el ambiente de producción, finalmente desplegaron los jobs de Control-M que permitieron ejecutar los procesos Scaffolder para crear las variables del modelo de datos.

3.3. Evaluación de resultados

En este subcapítulo se detalla el beneficio del modelo de datos en la organización, se describe el cambio que sufrió el motor de precios dinámicos al interactuar con el modelo de datos y como esta nueva fuente de entrada para el motor de precios permitió al negocio obtener nuevos clientes y optimizar el spread comercial y de esta manera el negocio de cambio de divisas fuera más rentable para la entidad bancaria.

3.3.1. Beneficio para la organización

Como se indicó en el alcance del proyecto de implementación del modelo de datos, no se tenía contemplado la adaptación del motor de precios para el consumo del modelo de datos, otro proyecto que se llamó “Precios dinámicos” desarrolló los componentes necesarios para que el motor de precios consuma el modelo de datos, el equipo de colaboración de España fue de gran valor porque transmitieron su experiencia en la adaptación del motor de precios de España con un modelo de datos similar al que se desarrolló en Perú, cabe resaltar que ambos proyectos se desarrollaron en paralelo.

En Perú, en el mes de octubre del año 2021 se puso en marcha el motor de precios con el nuevo modelo de datos como fuente de entrada para el cálculo del spread comercial para las operaciones de cambio de divisas.

El autor realizó un análisis comparativo en 3 aspectos claves para conocer el impacto del modelo de datos como nueva fuente de entrada al motor de precios en el negocio de cambio de divisas de la entidad bancaria, a continuación, se presentan los cálculos porcentuales[10]:

- Las ganancias se incrementaron en un 4.57% con respecto al promedio mensual de las ganancias obtenidas desde enero hasta septiembre.
- La cantidad de operaciones de compra y venta de divisas se incrementaron en un 15.08% con respecto al promedio mensual de la cantidad de operaciones realizadas desde enero hasta septiembre.
- La cantidad de clientes que operaron se incrementaron en un 3.38% con respecto al promedio mensual de la cantidad de clientes que operaron desde enero hasta septiembre.

[10] Los porcentajes calculados son en base a los datos internos disponibles de la entidad bancaria.

CAPÍTULO IV
REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

El uso de las tecnologías big data permitió a la entidad bancaria incrementar su rentabilidad en el negocio de compra y venta de divisas, el datalake contaba con las fuentes de información necesarias para crear el modelo de datos, el ambiente Sandbox tenía las tecnologías necesarias para el tratamiento de los datos, la infraestructura contaba con los recursos computacionales necesarios, este conjunto de elementos permitió que el proyecto se pueda desarrollar con éxito.

La experiencia del scrum master en gestión de proyectos en la entidad bancaria ayudó a detectar con anticipación que pasos seguir durante las etapas del proyecto, así mismo el conocimiento del product owner con relación a las fuentes de información y su comunicación clara fue de gran apoyo para la creación del modelo de datos.

El autor del presente informe durante el desarrollo del proyecto se esforzó por sacar adelante el proyecto, su experiencia en procesamiento de datos con tecnologías big data y habilidades blandas fue clave para el proyecto, fue felicitado por el líder técnico del proyecto por no solo abocarse en sus tareas, sino también por capacitar y asesorar al equipo en tecnologías big data y ser referente técnico al equipo y a otros proyectos de la entidad bancaria.

La experiencia del autor en otros proyectos de big data en el sector bancario le ayudó a realizar las tareas en un menor tiempo lo que le permitió desarrollar otras habilidades como liderazgo, apoyar al scrum master en gestionar el proyecto y ser parte del equipo de coaching de Indra Perú.

El trabajo en equipo y la colaboración entre todos los integrantes del scrum team fue muy importante para lograr el objetivo del proyecto, a pesar de que el equipo estuvo compuesto de varias empresas proveedoras se rompió con la barrera de no pertenecer a una misma empresa y se pudo hacer un excelente trabajo en equipo, esto es un claro ejemplo de coopetencia.

Durante el proyecto el autor del presente informe tuvo que lidiar con la actitud pasiva de algunos integrantes del scrum team, se sabe que al no tener experiencia en ciertas herramientas tecnológicas hay una curva de aprendizaje, es en

esta etapa donde se deben esforzar por aprender rápido y ser capaces de asociar sus experiencias con lo que se va aprendiendo.

El proyecto tuvo apoyo de un equipo de la sede principal de la entidad bancaria ubicada en España esto fue muy importante porque ellos ya habían creado un modelo de datos similar al que se requería en Perú y su experiencia iba a ser de gran impacto en el proyecto, sin embargo, la disponibilidad del equipo de apoyo era limitada, además la diferencia horaria que existe entre estos países no permiten una coordinación fluida, esto ocasionó retrasos en el proyecto.

El equipo perteneciente a la entidad bancaria encargada de promover el uso del arquetipo Scaffolder en los proyectos no tuvo la buena práctica de validar el "end to end" del arquetipo, es decir, no validó que todos los pasos involucrados en el uso del arquetipo hasta la puesta en producción estén habilitados correctamente, esto ocasionó que en el transcurso del proyecto se detecten problemas técnicos no previstos con el arquetipo, el autor apoyó a ese equipo en solucionar los problemas detectados con tal de no afectar a los tiempos establecidos del proyecto.

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- El modelo de datos permitió identificar potenciales clientes y optimizar el spread comercial en el negocio de cambio de divisas ya que en el primer mes del uso del modelo de datos se logró incrementar las ganancias en un 4.57%, la cantidad de operaciones aumentaron en un 15.08% y los clientes aumentaron en un 3.38%.
- Se logró identificar las fuentes de información que ayudaron a crear el modelo de datos gracias a la experiencia del product owner en los productos financieros que brinda la entidad bancaria y al apoyo del equipo de España en la definición de las variables del modelo de datos.
- El ambiente sandbox de la entidad bancaria permitió que el equipo de desarrollo pueda conectarse de forma rápida al datalake ya que este contaba con las configuraciones y herramientas necesarias para realizar los procesamientos de datos mediante Spark y Python.
- La unidad responsable de la entidad bancaria encargada del negocio de cambio de divisas fue de gran valor al proyecto al momento de definir las variables, sin su conocimiento del negocio y su experiencia en el mercado de divisas no hubiera sido factible definir las variables del modelo de datos que el motor de precios debía contemplar en el cálculo del spread comercial.
- El uso del arquetipo Scaffolder permitió estructurar el código fuente de los procesamientos de datos bajo un estándar de desarrollo definido por el área de Arquitectura de la entidad bancaria y facilitó el despliegue en producción ya que contaba con los archivos de configuración necesarios para la conexión al ambiente productivo del datalake.
- Los desarrollos realizados pasaron por una revisión a detalle por parte del equipo DQA, esto permitió que se cumplan con las buenas prácticas de desarrollo según los lineamientos de desarrollo de procesos con tecnologías big data que definió la entidad bancaria.
- El apoyo del equipo de México fue importante para la automatización de los procesamientos de datos, al centralizar los temas de automatización de

procesos con ellos permitió que el proyecto pueda culminar en los tiempos establecidos.

- Durante el desarrollo del proyecto se interactuó con equipos de otros países como España y México incluso el equipo de desarrollo estuvo compuesto de diferentes proveedores, esto permite a la entidad bancaria desarrollar una red de contactos utilizándola para conseguir los objetivos de negocio.

5.2. Recomendaciones

- En el caso de crear un arquetipo para otras tecnologías es necesario realizar pruebas de concepto para validar su factibilidad técnica, la idea central es desarrollar un prototipo o pequeño proyecto para detectar posibles complicaciones técnicas y puedan ser resueltas antes de iniciar un proyecto real, a fin de evitar retrasos en los tiempos planificados del proyecto.
- Con el paso del tiempo los modelos de datos se degradan, se debería revisar periódicamente el beneficio del modelo de datos en el negocio de cambio de divisas ya que en el tiempo el comportamiento del mercado puede variar por temas sociales, económicos o políticos.
- Se podrían usar fuentes externas como información de redes sociales o datos de navegación de los clientes en las diversas plataformas de la entidad bancaria para detectar los clientes que estén cotizando tipos de cambio y enviar en tiempo real un correo, mensaje o notificación animando al cliente en realizar su operación de compra o venta de divisas con la entidad bancaria.

BIBLIOGRAFÍA

Andina. (Junio de 2021). *Cambios de moneda*. Obtenido de https://andina.pe/: https://andina.pe/agencia/noticia-cuatro-cada-10-cambios-moneda-se-realizan-a-traves-del-canal-online-848934.aspx

Asociación Fintech de Perú. (2020). *Fintechs.* Obtenido de https://fintechperu.com/: https://fintechperu.com/#/blog/6154f12e7e9078b2adeda230

Bauer, D., Froese, F., Garcés-Erice, L., Giblin, C., Labbi, A., A. Nagy, Z., . . . Wespi, A. (2021). Building and Operating a Large-Scale Enterprise Data Analytics Platform. *Big Data Research, 23*. Obtenido de https://www.sciencedirect.com/science/article/pii/S2214579620300496

Belov, V., Tatarintsev, A., & Nikulchev, E. (2021). Choosing a Data Storage Format in the Apache Hadoop System Based on Experimental Evaluation Using Apache Spark. *Symmetry, 13*. doi:10.3390/sym13020195

Duque-Jaramillo, J., & Villa-Enciso, E. (2020). Big Data: desarrollo, avance y aplicación en las Organizaciones de la era de la Información (Big Data: Development, Advancement and Implementation Organizations in Information Age). *Revista CEA, 2*(4), 27-45. Obtenido de https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3519567#

Hasanin, T., Khoshgoftaar, T., Joffrey L., L., & Bauder, R. (2019). Severely imbalanced Big Data challenges: investigating data sampling approaches. *Journal of Big Data, 6*. doi:10.1186/s40537-019-0274-4

Indra. (2021). *Empresa Global de Tecnonlogía y Consultoría*. Obtenido de https://www.indracompany.com/: https://www.indracompany.com/es/indra

Indra. (2021). *Empresa Global de Tecnonlogía y Consultoría.* Obtenido de https://www.indracompany.com/: https://www.indracompany.com/sites/default/files/TOMO%20II%20RESP%20CORP.pdf

Jenkins. (2021). *Jenkins*. Obtenido de https://www.jenkins.io: https://www.jenkins.io/doc/#what-is-jenkins

Kalmukov, Y., Marinov, M., Mladenova, T., & Valova, I. (2021). Analysis and Experimental Study of HDFS Performance. *TEM Journal, 10*, 806-814. doi:10.18421/TEM102-38

Kayser, V., Nehrke, B., & Zubovic, D. (2018). Data Science as an Innovation Challenge: From Big Data to Value Proposition. *TECHNOLOGY INNOVATION MANAGEMENT REVIEW, 8*, 16-25. doi:10.22215/timreview/1143

Reppin, J., Beyer, C., Hartmann, T., Schluenzen, F., Flemming, M., Sternberger, S., & Kemp, Y. (2021). Interactive analysis notebooks on DESY batch resources: Bringing Juypter to HTCondor and Maxwell at DESY. *Computing and Software for Big Science, 5*. doi:10.1007/s41781-021-00058-y

Révész, Á., & Pataki, N. (2021). Visualisation of Jenkins Pipelines. *Acta Cybernetica*. doi:10.14232/actacyb.284211

Saiful Islam, M., & Hossain, E. (2020). Foreign Exchange Currency Rate Prediction using a GRU-LSTM Hybrid Network. *Soft Computing Letters*. doi:doi.org/10.1016/j.socl.2020.100009

Scrum. (2021). *Fundación Scrum.* Obtenido de https://www.scrum.org/: https://www.scrum.org/resources/what-is-scrum

Shahin, M., Ali Babar, M., & Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access, 5*, 3909 - 3943. doi:10.1109/ACCESS.2017.2685629

Shunhui, J., Qingqiu, L., Wennan, C., Pengcheng, Z., & Henry, M. (2020). Quality Assurance Technologies of Big Data Applications: A Systematic Literature Review. *Applied Sciences, 10*. doi:10.3390/app10228052

Wu, S., Wang, C., & Tang, R. (2021). Analysis and Research on the Performance of Solar Concentration Based on Big Data and Machine Learning. *Journal of Physics: Conference Series, 2026*. doi:10.1088/1742-6596/2026/1/012028

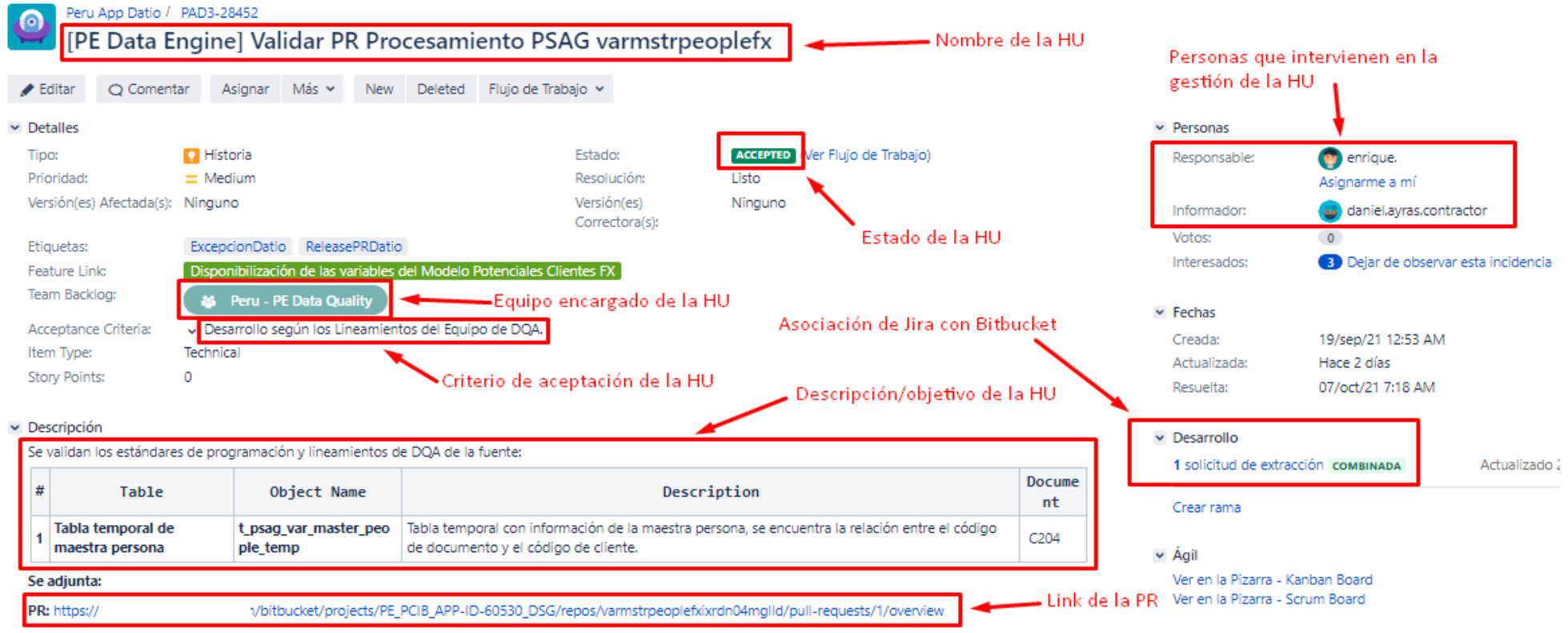
GLOSARIO

- **Apache Spark**: Motor de procesamientos de datos que trabaja en memoria, proporciona una librería para trabajar con el lenguaje de programación Python, debido a su trabajo en memoria permite realizar procesamientos a gran velocidad.
- **Arquetipo**: Plantilla que permite estructurar el código fuente de una solución bajo un patrón de diseño determinado y que facilita el despliegue a producción.
- **Big data**: Es un marco de trabajo que permite procesar grandes volúmenes de datos a gran velocidad, datos que pueden variar en el tiempo y que pueden ser estructurados como no estructurados.
- **Cambio de divisas**: Una operación de cambio de divisas permite convertir de una moneda local a una extranjera o viceversa.
- **Datalake**: Centraliza la información de una organización, permite procesos de analítica de datos estructurados como no estructurados.
- **Hadoop**: Sistema de archivos distribuidos, gestiona el almacenamiento de archivos en clústers con cientos o miles de máquinas y es tolerante a fallos, permite almacenar y procesar grandes volúmenes de datos estructurados y no estructurados
- **Integración continua**: Permite tener un ciclo de lanzamiento más corto y frecuente, mejorar la calidad del software y aumentar la productividad de sus equipos.
- **Python**: Es un lenguaje de programación de código libre, permite crear aplicaciones y es de fácil sintaxis y fácil de entender porque es muy cercana al lenguaje natural.
- **Sandbox**: Es una plataforma que permite conectarse a un espacio del datalake y realizar tareas exploratorias de fuentes de información y procesar datos. Brinda una conexión a ciertas herramientas de big data como Spark, Hadoop, Python, Scala, etc. que estén desplegadas sobre el datalake.
- **Spread comercial**: Es la diferencia entre los precios de compra y venta de una divisa, representa la retribución para la entidad bancaria como intermediario para el cambio de divisas.

ANEXOS

Anexo 1: Estructura de una historia de usuario para la gestión de pases a producción.

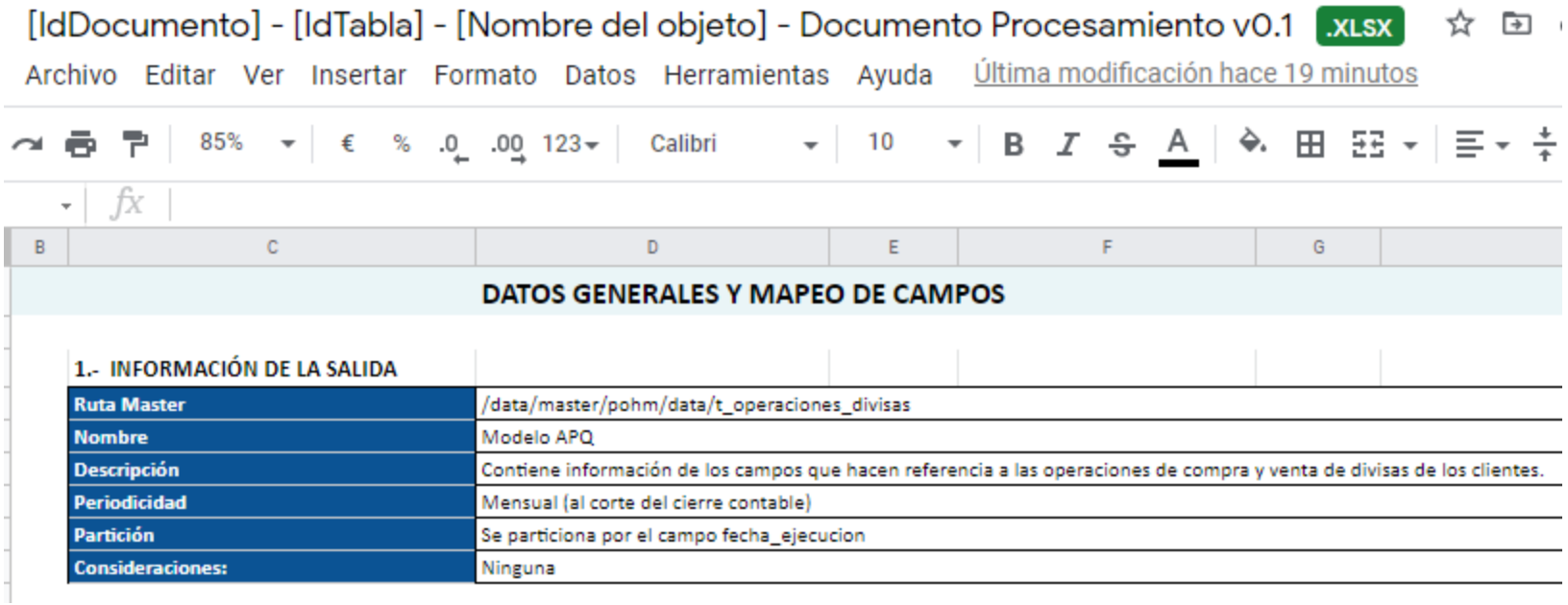
En el anexo 1 se observa la estructura de una historia de usuario creado en Jira, la cual se puede asociar con Bitbucket, esto facilita la gestión de pases a producción de procesos Scaffolder.



Anexo 2: Documento Excel donde se detalla el procesamiento realizado por Scaffolder

En el anexo 2 se observa la estructura del documento Excel que detalla el procesamiento realizado por el arquetipo Scaffolder, se compone de 3 secciones las cuales se describen a continuación:

- La primera sección contiene información de la fuente de salida del proceso Scaffolder como la ruta donde se va escribir en hdfs, la descripción del objetivo del procesamiento, la periodicidad de ejecución del proceso, entre otros.
- La segunda sección contiene una descripción funcional de cada campo de la tabla resultante del proceso Scaffolder.
- La última sección contiene información de relaciones o cruces de tablas realizadas en el proceso Scaffolder.



[IdDocumento] - [IdTabla] - [Nombre del objeto] - Documento Procesamiento v0.1 .XLSX

Archivo Editar Ver Insertar Formato Datos Herramientas Ayuda Última modificación hace 19 minutos

85% € % .0 .00 123 Calibri 10

fx

B	C	D	E	F	G

DATOS GENERALES Y MAPEO DE CAMPOS

1.- INFORMACIÓN DE LA SALIDA

Ruta Master	/data/master/pohm/data/t_operaciones_divisas
Nombre	Modelo APQ
Descripción	Contiene información de los campos que hacen referencia a las operaciones de compra y venta de divisas de los clientes.
Periodicidad	Mensual (al corte del cierre contable)
Partición	Se particiona por el campo fecha_ejecucion
Consideraciones:	Ninguna

2.- MAPEO DE CAMPOS

DESTINO			ORIGEN : Información de Operaciones de cambio de divisas			
Campo	Nombre	Llave	Tabla Origen	Query	Campo de la Tabla / Lógica de Carga	Observación
fecha_ejecucion	FECHA DE LA EJECUCION DE LA TRANSACCION	X	t_maestra_clientes t_operaciones_clientes	CALCULO		
id_operacion	CODIGO DE IDENTIFICACION UNICA DE UNA OPERACION	X				
codigo_cliente	CODIGO DEL CLIENTE QUE REALIZO LA OPERACION		t_maestra_clientes	CALCULO		
documento_cliente	NUMERO DEL DOCUMENTO DEL CLIENTE		t_maestra_clientes	CALCULO		
pais	NACIONALIDAD DEL CLIENTE		t_maestra_clientes	DIRECTO		
fecha_transaccion	FECHA EN QUE SE REALIZO LA OPERACION			FIJO		
monto_operacion	MONTO DE LA OPERACION			FIJO		
id_sede	ID DE LA SEDE DONDE SE HIZO LA OPERACION		t_sedes_oficinas	DIRECTO		

3. RELACIONES ENTRE FUENTES

#	Tabla Principal (P)	Tabla Relacionada (R)	TIPO	Campos (On)	Condiciòn (Where)
R1	t_maestra_clientes	t_operaciones_clientes	Left Join	P .codigo_cliente = R.codigo_cliente	R.flag_activo = 'S'
R2	t_maestra_clientes	t_sedes_oficinas	Left Join	P.id_sede = R.id_sede	
R3	t_detalle_operaciones	t_operaciones_cambio	Left Join	P.id_operacion = R.id_operacion	
R4	t_maestra_clientes	t_detalle_operaciones	Left Join	P.codigo_cliente= R.codigo_cliente	
R5	t_catalago_monedas	t_operaciones_cambio	Left Join	P.id_moneda = R.id_moneda	

Anexo 3: Documento del flujo de procesamientos

El anexo 3 contiene un diagrama a alto nivel del flujo de información y procesos considerados en el modelo de datos implementado en el proyecto, el cual detalla las fuentes de entrada, la creación de variables y finalmente la escritura en hdfs del modelo de datos.