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

Migración del modelo de portafolio de la unidad de negocio de soluciones de pago a un entorno de Big Data para la gestión de la cartera morosa en una entidad financiera

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Jean Carlo CANEVELLO SALAZAR

ASESOR

Norberto Ulises ROMÁN CONCHA

Lima, Perú

2021

Referencia bibliográfica

Canevello, J. (2021). *Migración del modelo de portafolio de la unidad de negocio de soluciones de pago a un entorno de Big Data para la gestión de la cartera morosa en una entidad financiera*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Jean Carlo Canevello Salazar
Tipo de documento de identidad	DNI
Número de documento de identidad	46611156
URL de ORCID	https://orcid.org/0000-0001-9626-063X
Datos de asesor	
Nombres y apellidos	Norberto Ulises Román Concha
Tipo de documento de identidad	DNI
Número de documento de identidad	08510560
URL de ORCID	https://orcid.org/0000-0002-3302-7539
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Luzmila Elisa Pró Concepción
Tipo de documento	DNI
Número de documento de identidad	08862360
Miembro del jurado 1	
Nombres y apellidos	Pablo Jesús Romero Naupari
Tipo de documento	DNI
Número de documento de identidad	06182185
Datos de investigación	
Línea de investigación	No Aplica
Grupo de investigación	No aplica
Agencia de 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 13 de diciembre del año 2021, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por la Dra. Pró Concepción Luzmila Elisa (Presidente), Lic. Romero Naupari Pablo Jesus (Miembro) y el Lic. Román Concha Norberto Ulises (Miembro Asesor), usando la plataforma Meet (https://meet.google.com/gfv-qdyi-szt), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: ***"MIGRACIÓN DEL MODELO DE PORTAFOLIO DE LA UNIDAD DE NEGOCIO DE SOLUCIONES DE PAGO A UN ENTORNO DE BIG DATA PARA LA GESTIÓN DE LA CARTERA MOROSA EN UNA ENTIDAD FINANCIERA",*** *por el Bachiller* ***Canevello Salazar Jean Carlo;*** *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.

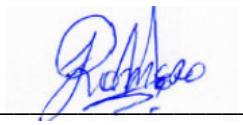
El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

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

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

Siendo las 19:58 horas, se levantó la sesión.

Presidente
Dra. Pró Concepción Luzmila Elisa



Miembro
Lic. Romero Naupari Pablo Jesús

Miembro Asesor
Lic. Román Concha Norberto Ulises

DEDICATORIA

Dedico a mi familia por todo el esfuerzo que hicieron para darme una educación de calidad, por el amor, la paciencia y comprensión que me brindaron, por todos los valores, consejos y palabras de aliento que me motivaban día a día a seguir creciendo y por apoyarme en cada decisión tomada.

AGRADECIMIENTOS

Agradezco a la comisión del programa de titulación por la oportunidad de demostrar mis conocimientos y habilidades con el objetivo de obtener el título profesional.

De igual manera agradezco a Iván Rengifo y Magaly Calderón por brindarme la oportunidad de liderar el proyecto del cual se describe en el presente informe.

Finalmente agradecer al profesor Ulises Román por su asesoramiento y compromiso para el éxito del presente trabajo.

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

MIGRACIÓN DEL MODELO DE PORTAFOLIO DE LA UNIDAD DE NEGOCIO DE SOLUCIONES DE PAGO A UN ENTORNO DE BIG DATA PARA LA GESTIÓN DE LA CARTERA MOROSA EN UNA ENTIDAD FINANCIERA

Autor:	Bach. Canevello Salazar, Jean Carlo
Asesor:	Lic. Román Concha, Ulises
Título:	Trabajo de Suficiencia Profesional
Fecha:	Diciembre 2021

RESUMEN

El presente trabajo de suficiencia profesional plantea la migración del Modelo de Portafolio de la unidad de negocio de Soluciones de Pago desde una plataforma tradicional como Oracle hacia una plataforma Big Data para la gestión de la cartera morosa de una entidad financiera.

Para lograr implementar la solución propuesta se ha utilizado una metodología propia del squad SDP que está dividido en 4 etapas que son el análisis, diseño, implementación y despliegue. Cada una de estas etapa interactúa con distintos roles como arquitectos de datos, seguridad de información, analistas de negocio, modelador de datos y gobierno de datos. A nivel de herramientas se ha utilizado Oracle para crear las tablas de paso en el esquema SDS, DataStage para crear flujos de integración en donde se migren los datos desde Oracle hacia la capa DDV del Data Lake en formato avro, PySpark para crear los procesos que incluyen lógica de negocio y aprovechar todos los recursos de la plataforma Big Data, IBM Infosphere Information Governance Catalog para registrar todo el linaje de datos y mantener un gobierno de datos ordenado, Bitbucket para tener versionado los cambios que se van realizando en el proyecto y finalmente Jira y Jenkins para el flujo ágil del pase a producción.

Como resultado del proyecto se ha logrado implementar el primer Modelo de datos en el Data Lake para la unidad de negocio y como parte del Modelo de Portafolio se implementaron los Modelos Core como la Fact de la cartera, Fact de pagos y Fact de Gestiones por mencionar algunas, esto abre las puertas para que otros modelos del negocio migren al Data Lake en un menor tiempo posible.

Palabras claves: modelo de portafolio, big data, gestión de cartera, entidad financiera, soluciones pago, data lake, migración.

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

MIGRATION OF THE PORTFOLIO MODEL FROM THE PAYMENT SOLUTIONS BUSINESS UNIT TO A BIG DATA ENVIRONMENT FOR THE MANAGEMENT OF THE MOROSA PORTFOLIO IN A FINANCIAL INSTITUTION

Author:	Bach. Canevello Salazar, Jean Carlo
Adviser:	Lic. Román Concha, Ulises
Title:	Professional Sufficiency Work
Date:	December 2021

ABSTRACT

The present work of professional sufficiency proposes the migration of the Portfolio Model of the Payment Solutions business unit from a traditional platform such as Oracle to a Big Data platform for the management of the delinquent portfolio of a financial institution.

In order to implement the proposed solution, a methodology of the SDP squad has been used, which is divided into 4 stages, which are analysis, design, implementation and deployment. Each of these stages interacts with different roles such as data architects, information security, business analysts, data modeler, and data governance. At the tools level, Oracle has been used to create the step tables in the SDS schema, DataStage to create integration flows where data is migrated from Oracle to the DDV layer of the Data Lake in avro format, PySpark to create the processes that include business logic and take advantage of all the resources of the Big Data platform, IBM Infosphere Information Governance Catalog to register the entire data lineage and maintain an orderly data governance, Bitbucket to have versioned the changes that are being made in the project, Jira and Jenkins for the agile flow of the go to production.

As a result of the project, the first data model has been implemented in the Data Lake for the business unit and as part of the Portfolio Model the Core Models were implemented such as the Portfolio Fact, Payment Fact and Management Fact to mention Some, this opens the doors for other business models to migrate to Data Lake in the shortest possible time.

Keywords: portfolio model, big data, portfolio management, financial institution, payment solutions, data lake, migration.

Tabla de Contenido

ÍNDICE DE TABLAS

ÍNDICES DE FIGURAS

INTRODUCCIÓN

El presente informe tiene como objetivo demostrar que el autor tiene la capacidad de aplicar los conocimientos de ingeniería de sistemas en el mundo laboral para poder obtener el título profesional.

El problema que se expone trata sobre la migración de un modelo de datos de la unidad de negocio de Soluciones de Pago desde una plataforma tradicional como Oracle a una plataforma Big Data. La necesidad surge como parte del proceso de integración de todos los modelos de datos del banco en el nuevo ecosistema big data.

El modelo seleccionado por Soluciones de Pago para empezar a migrar los procesos del negocio al nuevo ecosistema es el Modelo de Portafolio, este es un modelo que integra los modelos core del negocio y además sirve de habilitador para otros modelos. Junto a este escenario se une el hecho de que la unidad de negocio estaba en proceso de adquisición de una nueva aplicación para la clasificación y asignación de la cartera morosa, esto significa que el diseño de la solución debe permitir trabajar tanto con la nueva aplicación como con la actual aplicación.

La implementación del modelo se realizó sobre un Data Lake en donde la capa de almacenamiento está sobre HDFS (Hadoop Distributed File System), la capa de procesamiento se realiza con PySpark y también se ha utilizado IBM DataStage para cargar algunos datos desde un sandbox hacia el Data Lake.

El presente informe se encuentra estructurado en 5 capítulos:

El capítulo I trata sobre la trayectoria profesional del autor, se detalla la experiencia profesional, lugares donde ha trabajado, roles desempeñados, tecnologías con las que ha trabajado y los proyectos que le ha ayudado a ganar la experiencia profesional.

En el capítulo II se describe a la empresa donde el autor ha ganado la experiencia profesional desarrollando el proyecto que se describe en el informe, se detalla el área, cargo y funciones desempeñadas.

En el capítulo III se detalla el problema, objetivos, el alcance, la metodología que se usó en el proyecto y los fundamentos teóricos.

En el capítulo IV Se menciona en base a la experiencia la reflexión del sobre el proyecto desarrollado, y finalmente en el capítulo V de detalla las conclusiones y recomendaciones que se pueden dar al proyecto.

CAPÍTULO I:

TRAYECTORIA PROFESIONAL

El autor del presente informe posee el grado de bachiller en Ingeniería de Sistemas de la Universidad Nacional Mayor de San Marcos, con más de 8 años de experiencia trabajando en el ámbito tecnológico de los cuales los últimos 4 años fueron con bachiller.

La experiencia profesional comienza en el 2013 como desarrollador de software hasta el 2017, durante este periodo el autor ha tenido la oportunidad de aprender y aplicar las mejores prácticas de Ingeniería de Software, aprender sobre gestión de proyectos de gran envergadura y desarrollar habilidades blandas para la correcta comunicación con los clientes.

En el 2018, el autor decide incursionar en el mundo del big data ingresando a una entidad financiera como Ingeniero de Datos, durante este periodo ha obtenido conocimientos y experiencia en gestión, gobierno, análisis y explotación de datos bajo un marco de trabajo ágil.

Tabla 1

Listado de la experiencia profesional

EXPERIENCIA PROFESIONAL	
ENCORA	
Periodo	Enero 2021 - Actualidad
Cargo	Ingeniero de datos
Proyectos	**Datahub (Enero 2021 – Actualidad)** Datahub es un proyecto de la fábrica digital del banco Scotiabank soportado por un equipo de Data Engineers cuyo objetivo es brindar una plataforma operativa para el equipo de Data y Analytics. Funciones: - Desarrollar pipelines en Python y Sql Server para automatizar la generación de diversos reportes. - Dar soporte al DataHub para la continuidad operativa de los diversos procesos que existen. - Proponer y realizar mejoras en el DataHub.

	Tecnologías: Python, Sql Server, R, Linux, Git
BANCO DE CRÉDITO BCP	
Periodo	Febrero 2018 – Enero 2021
Cargo	Ingeniero de Datos
Proyectos	**Squad Soluciones de Pago (Febrero del 2018 – Enero 2021)** Es un Squad encargado de la automatización de procesos de información de la unidad de Soluciones de Pago para la cobranza y riesgos de banca minorista bajo un marco de trabajo ágil. Funciones: − Reunirse con el usuario para capturar las necesidades del negocio y orientar las soluciones a los lineamientos de datos del banco. − Realizar el análisis, diseño y la planificación del proyecto. − Extraer, analizar y procesar grandes volúmenes de datos para la creación de modelos de información. − Responsable de la continuidad operativa de los proyectos que se encuentran dentro de las carteras asignadas (activa, castigo, judicial, scores) y también la continuidad del motor de Machine Learning para el equipo de modelamiento de Soluciones de Pago. − Responsable de la migración de los procesos del sandbox de Soluciones de Pago hacia el nuevo ecosistema Data Lake y acorde a los nuevos lineamientos de datos del banco. Tecnologías: Python, Hive, PySpark, DataStage, Power BI, Datameer, Jenkins, Git, Oracle, Sql Server, SSIS, SSRS. **Mesa Berlín (Noviembre del 2018 – Febrero 2019)** Berlín es una mesa de trabajo del Laboratorio de Big Data que se encarga de la búsqueda de nuevas fuentes de datos para ofrecer productos personalizados a los clientes. Funciones: − Desarrollar scripts para la obtención de información por web scraping. − Analizar la calidad de los datos obtenidos. Tecnologías: Python, Azure, Oracle
INSPIRED SOLUTIONS	
Periodo	Mayo 2017 – Enero 2018
Cargo	Analista Programador de Sistemas
Proyectos	**Vault (Noviembre 2017 – Enero 2018)** Vault es una startup de EE.UU para el sector financiero, es una aplicación móvil que funciona como una billetera virtual, los usuarios pueden realizar transferencia de dinero entre ellos y también pueden realizar compras en los sitios afiliados a la aplicación. Funciones:

	– Entender la necesidad del cliente para definir el diseño de una API que será consumida por la aplicación móvil en Android o iOS. – Realizar la planificación para la implementación de la API. – Desarrollar la API y realizar pruebas unitarias. – Coordinar con el equipo de desarrollo móvil para las pruebas integrales. Tecnologías: Django (python), GraphQL, PostgreSQL, Git, Gitlab **Procalidad (Agosto 2017 – Octubre 2017)** Procalidad es un proyecto financiado por el Banco Mundial para la implementación de Sistemas de Autoevaluación de Educación Superior para el Licenciamiento de las universidades en el Perú (SAES y SAESL). Funciones: – Capturar la necesidad del cliente y definir el alcance, tiempo y costo de los requerimientos. – Desarrollar los componentes de software requeridos por el cliente. – Implementar reportes de seguimiento de las autoevaluaciones de las universidades. – Coordinar con el cliente para las pruebas de usuario. – Planificar y ejecutar el pase a producción de los nuevos componentes. Tecnologías: Kohana (PHP), HTML5, CSS3, MySQL, Linux, Git, Gitlab **SeatLabs(Mayo 2017 – Julio 2017)** SeatLabs es un startup de Israel dedicada a la venta de entradas para eventos de entretenimiento. Funciones: – Capturar la necesidad del cliente y definir el alcance, tiempo y costo de los requerimientos. – Desarrollar los componentes de software requeridos por el cliente. – Planificar y ejecutar el pase a producción de los nuevos componentes. Tecnologías: Django (Python), Angular 2, PostgreSQL, Git, Gitlab, Linux
CONSULTOR INDEPENDIENTE	
Periodo	Mayo 2015 – Noviembre 2017
Cargo	Consultor
Proyectos	**Progresa (Enero 2015 – Noviembre 2017)** Es un sistema informático para la gestión de préstamo de microcréditos para personas con pequeño negocio. El sistema soporta múltiples agencias, acceso a diferentes perfiles (cajero, asesor administrador y gerente), evaluación de préstamo, registro de pagos, balance de caja y reportes de seguimiento. Funciones: – Liderar el proyecto para la implementación del sistema. – Capturar los requerimientos del sistema para el análisis y diseño de la arquitectura. – Definir el alcance de las funcionalidades del sistema.

	– Elaborar la línea base del proyecto tales como el alcance, tiempos y costos relacionados a cada requerimiento – Desarrollar los componentes del software. – Coordinar y ejecutar las pruebas de usuario. – Coordinar y ejecutar los pases a producción. – Elaborar el manual de usuario y la documentación. Tecnologías: Kohana (PHP), Bootstrap 3, MySQL, Git, Bitbucket, Linux **Maternelle (Agosto 2015 – Noviembre 2015)** Maternelle es una tienda online dedicada a la venta de ropa y accesorios para bebés. Funciones: – Capturar requerimientos y definir funcionalidades del sistema de gestor de contenidos. – Estimar alcance, tiempo y costos del sistema. – Desarrollar los componentes backend del sistema. – Desarrollar los componentes frontend responsiva a partir de un diseño en Photoshop. – Coordinar las pruebas de usuario. Tecnologías: Wordpress (PHP), MySQL, Html5, CSS3, Linux **Tondero (Mayo 2015 – Julio 2015)** Tondero es un blog de la empresa de entretenimiento Tondero, donde hacen publicaciones de las producciones (películas, teatro, representaciones, etc) que hace esta empresa. Funciones: – Capturar requerimientos y definir funcionalidades del sistema de gestor de contenidos. – Estimar alcance, tiempo y costos del sistema. – Desarrollar los componentes backend del sistema. – Desarrollar los componentes frontend responsiva a partir de un diseño en Photoshop. – Coordinar las pruebas de usuario. Tecnologías: Wordpress (php), MySQL, Html5, CSS3, Linux
SMARTEC	
Periodo	Marzo 2013 – Marzo 2015
Cargo	Programador
Proyectos	**Canal J (Noviembre 2014 – Marzo 2015)** Canal J es un blog que muestra contenido de los diferentes programas de televisión que realiza Jockey Plaza. Principales tareas ejecutadas en el proyecto: – Analizar los requerimientos y estimar el tiempo de implementación. – Desarrollar los componentes de software del sistema de gestión de contenidos (CMS). – Coordinar las pruebas de usuario con usuarios.

	– Coordinar los pases a producción. Tecnologías: Wordpress (php), MySQL, HTML5, CSS3, Javascript, Git, Bitbucket. **E-commerce UPC (Octubre 2014 – Octubre 2014)** Es un e-commerce para la venta de libros propios de la universidad UPC. Principales tareas ejecutadas en el proyecto: – Mantenimiento de los componentes de software. – Coordinar con los UX sobre los cambios solicitados. Tecnologías: PHP, HTML5, CSS3, Git, Bitbucket. **One2One Coaching (Agosto 2014 – Septiembre 2014)** One2One Coaching es un CMS para la publicación de posts sobre la temática del comportamiento de las personas en las organizaciones. Principales tareas ejecutadas en el proyecto: – Analizar los requerimientos y estimar el tiempo de implementación. – Desarrollar los componentes backend de la plataforma. – Desarrollar los componentes frontend responsive de la plataforma a partir de un diseño en Photoshop. Tecnologías: Wordpress (php), HTML5, CSS3, MySQL, Git, Bitbucket. **Promart (Abril 2014 – Julio 2014)** Promart es un e-commerce que se dedica a la venta de productos de construcción. Principales tareas ejecutadas en el proyecto: – Analizar los requerimientos y estimar el tiempo de implementación. – Desarrollar los componentes backend de la plataforma. – Desarrollar los componentes frontend responsive de la plataforma a partir de un diseño en Photoshop. Tecnologías: Prestashop (php), Bootstrap 3, MySQL, Git, Bitbucket. **E-commerce Oltursa (Diciembre 2013 – Marzo 2014)** Es un e-commerce para la venta de pasajes de la empresa de transportes Oltursa. Principales tareas ejecutadas en el proyecto: – Desarrollo de los nuevos componentes de ventas de pasajes. – Desarrollar los cambios solicitados en el frontend. Tecnologías: CakePHP, HTML5, CSS3, Git, Bitbucket. **TuCloset (Mayo 2013 – Noviembre 2013)** TuCloset es un Marketplace de ropa, calzado y accesorios en donde cualquier persona desde cualquier parte del Perú puede afiliarse a la plataforma para la comercialización de productos. Principales tareas ejecutadas en el proyecto: – Desarrollar los componentes backend de la plataforma. – Desarrollar los componentes frontend responsive de la plataforma a partir de un diseño en Photoshop. – Integración de medios de pagos como PayU, 2Checkout, Pago efectivo, Mastercard y Visa.

	Tecnologías: Kohana (php), Bootstrap 3, MySQL, SVN.

Nota. La Tabla 1 muestra el detalle de la experiencia profesional del autor. Fuente: Elaboración propia.

Tabla 2

Formación Académica Profesional

FORMACIÓN ACADÉMICA PROFESIONAL		
Formación	**Institución**	**Periodo**
Bachiller en Ingeniería de Sistemas	Universidad Nacional Mayor de San Marcos	2010 - 2017

Nota. La Tabla 2 detalla las entidades educativas de formación del autor. Fuente: Elaboración propia.

Tabla 3

Cursos y eventos académicos

CURSOS Y EVENTOS ACADÉMICOS		
Curso	**Institución**	**Año**
Google Cloud Big Data and Machine Learning Fundamentals	Google	2021
ThePowerMBA (Gestión y Administración de empresas)	ThePowerMBA	2020 - 2021
Python for Data Science	Edutronic Global Services	2020
Análisis de Datos con Python y Apache Spark	Banco de Crédito BCP	2019
Capacitación en Qlik Sense	Banco de Crédito BCP	2019
Coaching Teatral para el desarrollo de profesionales líderes	Museo de Arte de Lima	2019

Nota. La Tabla 3 describe los cursos y talleres en donde el autor ha participado y le ha ayudado a adquirer más conocimientos. Fuente: Elaboración propia.

Tabla 4

Otras Capacidades

OTRAS CAPACIDADES	
Plataformas	Windows, Linux, Android.
Tecnologías	Python, PySpark, Hive, Power BI, HDFS, Git, DataStage, Datameer, SSIS, SSRS, Django, PHP, HTML5, CSS3, JavaScript, Kotlin.
Base de Datos	Sql Server, Oracle, MySql, PostgreSql.
Otros	Scrum,Kanban, Gestión de datos, Planificación, Bitbucket, Confluence, Jira, AWS, Remedy, ClearQuest, ClearCase, Github, Jenkins, Office, Teams.
Habilidades blandas	Trabajo en equipo, comunicación efectiva, empatía, responsabilidad.

Nota. La Tabla 4 describe los conocimientos adicinales que tiene el autor. Fuente: Elaboración propia.

CAPÍTULO II

CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA

2.1 EMPRESA – ACTIVIDAD QUE REALIZA

Según (www.viabcp.com, 2020) la actividad de la entidad financiera es:

> La empresa es una institución del sistema financiero peruano y es el proveedor líder de servicios financieros en el país. Tiene como objetivo social favorecer el desarrollo de las actividades comerciales y productivas del país, que incluyen la actividad bancaria comercial y de ahorros.
>
> A través de las divisiones de Banca Corporativa y Banca Empresa provee servicios especialmente diseñados para clientes corporativos y empresas medianas, mientras que la Banca Minorista atiende a pequeñas empresa y clientes individuales

Datos de la Empresa:

- **Razón Social:** Banco del Crédito del Perú
- **RUC:** 20100047218
- **Tipo de Empresa:** Sociedad Anónima
- **Dirección:** Cal. Centenario Nro. 156

2.2 VISIÓN

"Ser referentes regionales en gestión, potenciando nuestro liderazgo histórico y transformador de la industria financiera en el Perú." (www.viabcp.com, 2020)

"Ser la empresa peruana que brinda la mejor experiencia a los clientes. Simple, Cercana y Oportuna" (www.viabcp.com, 2020)

"Ser la comunidad laboral de preferencia en el Perú, que inspira, potencia y dinamiza a los mejores profesionales" (www.viabcp.com, 2020)

2.3 MISIÓN

"TRANSFORMAR PLANES EN REALIDAD

Estar siempre contigo, alentando y transformando tus sueños y planes en realidad; y con el Perú, construyendo su historia de desarrollo y superación" (www.viabcp.com, 2020)

2.4 ORGANIZACIÓN DE LA EMPRESA

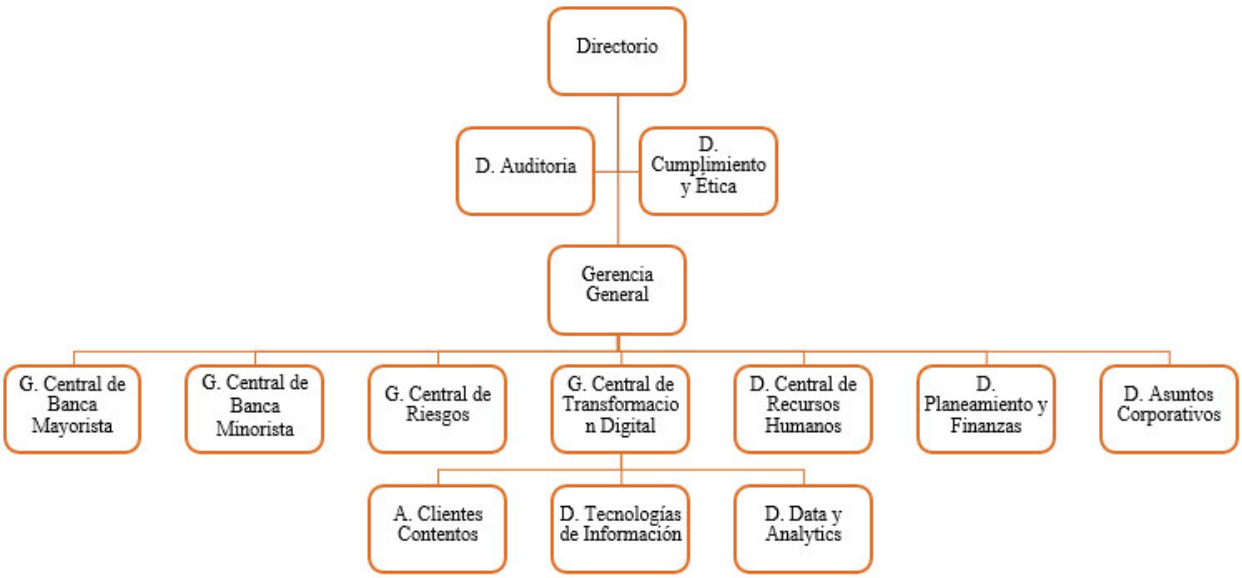


Figura 1. Organigrama General de la Empresa. Fuente: Elaboración propia (adaptado)

- **Gerencia Central de Banca Mayorista**

Según (www.viabcp.com, 2020) la función de la gerencia es:

> La Gerencia Central de Banca Mayorista diseña y gestiona servicios para clientes corporativos y grandes empresas. Los productos ofrecidos por la Banca Mayorista están diseñados para cubrir las necesidades de más de 10,000 clientes en Lima y provincias. Incluyen créditos comerciales, créditos contingentes, productos de comercio exterior, productos de cambio y derivados financieros, y productos pasivos y transaccionales.

- **Gerencia Central de Banca Minorista**

"El objetivo principal de la Banca Minorista es la de gestionar los productos, servicios, segmentos y canales, velando por que éstos sean competitivos, rentables y de calidad percibida por los clientes" (BCP Bolivia, 2009)

- **Gerencia Central de Riesgos**

Según (www.viabcp.com, 2020) la función de la gerencia es:

> La gerencia central de riesgo es la responsable de definir una estrategia alineada a los objetivos y al apetito de riesgo establecidos por el directorio del banco. La gerencia busca fortalecer el proceso de gestión, generar sinergias, optimizar recursos y lograr mejores resultados entre las unidades responsables de la gestión de riesgos no financieros con el resto de las empresas del grupo.

- **Gerencia Central de Transformación Digital**

La gerencia central de transformación digital es el encargado de alinear los procesos y operaciones del banco hacia su digitalización y también de implementar una cultura organizacional orientada hacia la mejora de los modelos de negocio de la empresa centrando todo su esfuerzo hacia la optimización de la experiencia del cliente.

- **División de Cumplimiento y Ética Corporativo**

La división de Cumplimiento y Ética Corporativo es el responsable de "asegurar la implementación de un marco regulatorio en toda la empresa, mediante metodologías y estándares internacionales, con el fin de mitigar riesgos de cumplimiento normativo (sanciones, multas) así como riesgo reputacional" (www.viabcp.com, 2020)

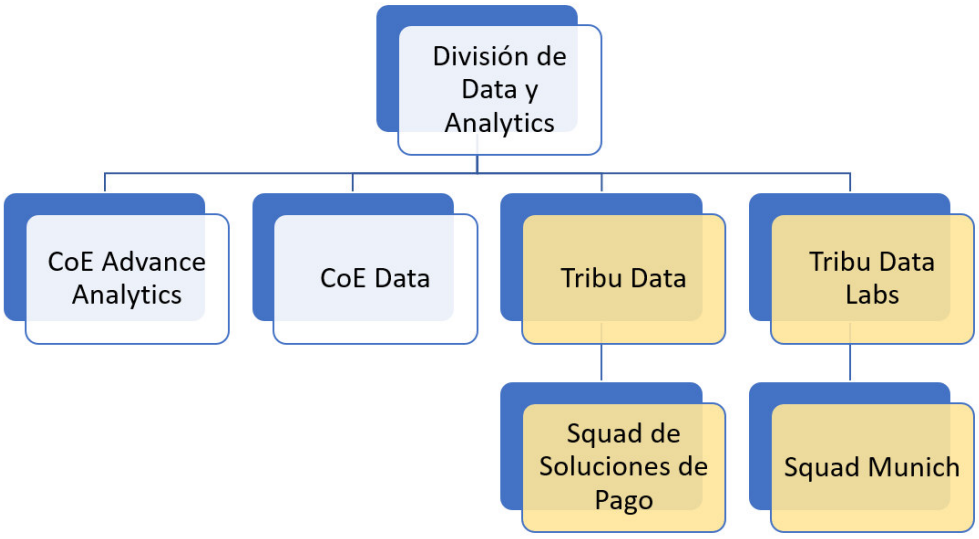


Figura 2. Organigrama del área de la experiencia profesional del autor. Fuente: Elaboración propia.

La División de Data y Analytics se encuentra dentro de la Gerencia Central de Transformación Digital y está dividido en 4 áreas:

- **CoE Advance Analytics:** Se encarga de crear nuevos servicios analíticos para la mejora de los procesos de la corporación.
- **CoE Data:** Son los encargados de la implementación del ecosistema big data, aquí están los arquitectos de datos, modeladores de datos, gobierno de datos, ingenieros de datos.
- **Tribu Data:** Son los encargados de implementar las soluciones de información de las unidades de negocio y también son responsables de poblar de datos el data lake. Está dividido por squads donde cada uno está asignado a una unidad de negocio y otros al poblado de datos. El autor del informe ha estado en el squad de Soluciones de Pagos.
- **Tribu Data Labs**: La tribu es responsable de encontrar nuevas fuentes de datos para mejorar la experiencia de los clientes. El autor del informe ha estado apoyando durante 6 meses al squad de Múnich recopilando datos de distintas fuentes con Web Scraping.

2.5 ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS

El autor del presente informe se desempeñó como Ingeniero de Datos para el Squad de Soluciones de Pago que se encuentra dentro del área de Data y Analytics del banco. El squad es responsable de la creación de soluciones de información para cubrir las necesidades de la unidad de negocio de Soluciones de Pago de Banca Minorista.

Como Ingeniero de Datos nos encargamos de gestionar la implementación de distintas soluciones de información ya sea en entornos tradicionales o en entornos actuales como big

data, interactuamos con distintos stakeholders como arquitectos de datos, analistas de seguridad, gobierno de información, modeladores de datos y negocio, con el objetivo principal de que las soluciones implementadas estén acorde a los lineamientos exigidos por el banco.

Funciones realizadas en la presente entidad financiera:

- Reunirse con el usuario interno para capturar las necesidades del negocio y orientar las soluciones a los lineamientos de datos del banco.
- Realizar la planificación del proyecto que involucra análisis y diseño de la solución en el Data Lake. En la etapa de análisis se realiza una exploración de los datos, limpieza y algunas transformaciones para tener mayor claridad de las fuentes a usar. Luego del análisis exploratorio y teniendo mayor claridad de las fuente se diseña la arquitectura de la solución, es decir qué fuentes se van a utilizar, como se van a leer las fuentes, que capas del data lake o data mart se utilizarán para la solución, el modelado de las tablas y finalmente el reporte que se mostrará al usuario final.
- Implementar soluciones de información para extraer, analizar y procesar grandes volúmenes de datos en el Data Lake. Esta función se encarga de desarrollar pipelines en Oracle o en PySpark, los pipelines incluye limpieza de datos, implementar las reglas de negocio en las lógicas y almacenarlos en las tablas finales.
- Coordinar con los equipos de Arquitectura de Datos, Seguridad de Información y Gobierno de datos para crear soluciones integrales.
- Gobernar los distintos proyectos implementados en el Data Lake para garantizar la continuidad operativa de los procesos.
- Analizar la viabilidad técnica de los procesos implementados con tecnologías tradicionales para migrarlo a un entorno Big Datas. Esta función está alineada al proceso de transformación digital del banco en donde los data mart que se encuentran en una plataforma tradicional como Oracle tienen que ser migrados a el

nuevo ecosistema de big data, esto implica que las soluciones implementadas tienen que ser analizadas para definir si pueden adaptarse al nuevo entorno.

2.6 EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN

Dentro del squad de soluciones de pago los proyectos relacionados con los analistas de negocio se dividen en 5 carteras, cartera activa, castigada, judicial, telefonía y negociación. Adicionalmente el squad también ve proyectos con el equipo de modelamiento, específicamente los temas de Scores de los clientes morosos.

La primera experiencia realizada en la organización para el autor del presente informe fue la implementación de un tablero de control que permite dar seguimiento a las gestiones realizadas en la cartera de judicial. Este proyecto permitió conocer a profundidad las tablas core del banco, conocer a mayor profundidad el modelo de negocio de banca minorista y dar a conocer las habilidades que tiene para la implementación de soluciones de información utilizando diferentes tecnologías.

A lo largo del tiempo, el autor ha participado en proyectos de las distintas carteras de activa, castigada y judicial, así como también en proyectos para el cálculo de scores.

Sin embargo, uno de los proyectos más retadores para el autor fue la implementación de una solución de información para la gestión de portafolio de la cartera activa, castigo y judicial. Este proyecto utilizaba 21 fuentes de información que tenían que ser implementados en un Data Lake utilizando PySpark y siguiendo los lineamientos de seguridad del banco. El proyecto permitió al autor interactuar con diferentes roles del área de data como arquitectos de datos, gobierno de información, seguridad de información y modelamiento de base de datos para garantizar la calidad de la solución.

Además, la experiencia previa del autor en el mundo software contribuyó en tener una visión más clara y amplia al momento de realizar el análisis y diseño de la solución en el Data Lake.

CAPÍTULO III

ACTIVIDADES DESARROLLADAS

3.1 SITUACIÓN PROBLEMÁTICA

Como parte de la transformación digital de la entidad financiera, uno de sus objetivos es convertirse en una empresa data driven, es decir, basar la toma de decisiones en la explotación de los datos que la propia empresa guarda. Sobre este contexto, el banco decide implementar un ecosistema big data que permita centralizar todas las fuentes de información, como archivos planos que dejan los aplicativos core, videos, audios y data estructurada de los distintos sistemas de información.

En consecuencia y como parte del proceso de transformación digital, luego de la implementación de una plataforma de big data en el banco, el siguiente paso es que los data marts de las distintas unidades de negocio migren todos sus procesos al Data Lake, en el caso de soluciones de pago, se ha definido que el Modelo de Portafolio sea el primer proceso en migrar a la nueva plataforma. La elección del modelo se debe a que se necesita de las tablas de hechos core del negocio como la cartera, pagos y gestiones para que el modelo sea implementado, esto significa que la implementación de este modelo es un habilitador para otros modelos de datos del negocio. Junto a este escenario se suma que soluciones de pago estaba en proceso de adquirir un nuevo aplicativo llamado Cyber Financial para la segmentación y asignación de la cartera morosa, y las broads del nuevo aplicativo deberían ir directamente al Data Lake, sin embargo; hasta ese momento no se tenían tales broads, por lo que surge la necesidad de crear una estrategia para que la solución conviva con las antiguas aplicaciones del negocio hasta que la nueva aplicación sea puesta en producción y hacer un switch de las broads antiguas por las nuevas.

3.1.1 DEFINICIÓN DEL PROBLEMA

PROBLEMA PRINCIPAL

No se tiene implementado el modelo de portafolio de la unidad negocio de soluciones de pago en un entorno de big data para la gestión de la cartera activa y judicial.

PROBLEMA SECUNDARIOS

1. Las broads del nuevo aplicativo Cyber Financial no están disponibles al momento de la migración.
2. Los modelos de datos core del negocio como los saldos, pagos y gestiones de los clientes no están implementados en el data lake.
3. El modelo de portafolio no se encuentra implementado en el data lake

3.2 SOLUCIÓN

Implementación del Modelo de Portafolio de la unidad de Soluciones de Pago en el Data Lake considerando trabajar con el aplicativo tradicional Debt Manager utilizando PySpark, IBM Data Stage, Oracle y HDFS.

3.2.1 OBJETIVOS

OBJETIVO GENERAL

Migrar el modelo de portafolio de la unidad de negocio de soluciones de pago a un entorno de big data para la gestión de la cartera morosa en una entidad financiera

OBJETIVOS ESPECIFICOS

1. Implementar el modelo de portafolio utilizando la broad del aplicativo tradicional Debt Manager.
2. Implementar los modelos core del negocio como las tablas de hecho de pagos, gestiones, promesas de pago(pdp), saldos, maestra de cuentas y lookups de productos, rangos de mora, fechas de pago, fechas de cosecha y tipo de acción de cobranza en el Data Lake y utilizando PySpark.

3. Implementar el Modelo de Portafolio en el Data Lake y utilizando PySpark, esto incluye portafolio diario, portafolio mensual y portafolio de cosecha.

3.2.2 ALCANCE

ALCANCE FUNCIONAL

El alcance de este proyecto llega hasta la implementación en el Data Lake de los modelos Core del negocio y el Modelo de Portafolio. Los datos son guardados en tablas modeladas en la capa UDV para que los analistas del negocio puedan hacer exploraciones y armar nuevos tableros de control con las herramientas del ecosistema big data como Datameer, Qlik Sense o Power BI.

ALCANCE ORGANIZACIONAL

El Modelo de Portafolio es usado por los analistas de la unidad de negocio de Soluciones de Pago. El equipo responsable de velar por la continuidad operativa del proceso en el Data Lake es el equipo de Operaciones del Data Lake y el equipo que se encarga de velar por la calidad de los datos es el equipo de Gobierno de datos.

ALCANCE TECNOLÓGICO

Como alcance tecnológico se está utilizando el ecosistema big data del banco. A nivel de almacenamiento se usa HDFS para guardar los datos en formato parquet, a nivel de procesamiento se utiliza PySpark para ejecutar los pipelines desarrollados y también se utiliza Oracle para crear tablas de paso hacia el data lake, a nivel de extracción y carga de fuentes externas se utiliza IBM DataStage.

3.2.3 ETAPAS Y METODOLOGÍA

La implementación del Modelo de Portafolio pasó por las etapas que se detallan en la siguiente tabla utilizando un marco metodológico propio de la unidad de negocio, la correcta ejecución de cada etapa ha permitido realizar un análisis exhaustivo para descartar algunos datos obsoletos del modelo actual y agregar nuevos datos que aporten valor al modelo nuevo

considerando que se deben trabajar con las broads de los aplicativos tradicionales, esto significa que la solución tiene un porcentaje táctico el cual en un futuro tendrá que ser reemplazado por las broads del nuevo aplicativo Cyber Financial.

La metodología ha permitido identificar mejoras y aplicar de manera estratégica las herramientas que brinda el ecosistema de big data como un almacenamiento distribuido (HDFS) en formato avro y parquet para almacenar grandes volúmenes de dato; por otro lado para ejecutar todas las lógicas se está aprovechando todo el potencial del procesamiento en memoria con PySpark y también se está llevando al Data Lake datos procesados desde un data mart utilizando IBM Data Stage.

Tabla 5

Etapas del proyecto

Fase	**Tareas**	**Entregables**	**Técnica Empleada**
Análisis del requerimiento	Definir las variables finales.	Diccionario de negocio de los modelos implementados.	Reuniones de análisis con el negocio.
	Identificación de fuentes	Documento Excel con la trazabilidad de datos.	Matriz de Trazabilidad
Diseño de la Solución	Diseño de la Arquitectura de datos	Documento de arquitectura de datos.	En coordinación con los arquitectos de datos se ha elaborado el diseño de la solución.
	Definir las taxonomías en la capa DDV para HDFS y Linux	Documento de taxonomías	En coordinación con los arquitectos de datos se ha definido la taxonomía.
Implementación de la Solución	Modelado Dimensional	Modelo entidad relación de las tablas	En coordinación con el equipo de modelamiento

			de datos de CoE Data se han aplicado los estándares de Data Lake para definir los nombres de las tablas y campos y también se ha aplicado el modelo de datos Copo de Nieve.
	Desarrollar pipelines	-Scripts en PySpark de los modelos Core y Modelo de Portafolio -Scripts DML y DDL en Hql. -Proyectos de Data Stage	El desarrollo de scripts tanto en PySpark, Oracle y Data Stage siguieron el lineamiento definido por el equipo de arquitectura de datos.
	Validación de resultados	Documento final comparando los resultados finales de los modelos implementados en Oracle vs Data Lake.	Para la validación se han calculado variables de estadística descriptiva que permiten comparar los resultados del modelo nuevo con el modelo actual.
	Implementar el Linaje de datos	- Diccionario de datos. - Visualización del linaje de datos en la herramienta de gobierno de datos	En Coordinación con el equipo de gobierno de datos se ha utilizado la herramienta IBM Infosphere Governance Catalog para implementar

			el linaje de los modelos de datos.
Despliegue	Certificación y Pase a Producción	Aprobación del líder técnico, Product Owner y Seguridad de información	En esta actividad se ha utilizado Jira, Jenkins y Bitbucket para realizar el flujo de certificación.

Nota. La Tabla 5 describe de manera resumida las etapas del proyecto y los entregables por cada etapa ejecutada. Fuente: Elaboración propia

3.2.4 FUNDAMENTOS UTILIZADOS

3.2.4.1 Soluciones de Pago

Soluciones de Pago es el área encargada de gestionar estratégicamente la cobranza a los clientes que ingresan a la cartera morosa.

La buena gestión de la cartera morosa empieza con un correcto y amplio conocimiento del cliente y el otorgamiento de créditos de manera analítica y cuidadosa, a esto se le llama prevención y como resultado se tendrá una eficiencia en los cobros. Las estrategias de cobranza deben están alineados a la realidad económica y política del país; y además se deben considerar ciertos factores antes de aplicar una estrategia como el historial crediticio que tiene el cliente con la empresa, la cantidad de años trabajando con la empresa y las motivaciones que tiene para no pagar. (SANTANA, 2016) (p.25)

Una de las estrategias en la gestión de cobranza es segmentar las cuentas en diferentes tramos y aplicar distintos métodos de cobranzas a cada una de ellas, tal como lo menciona (Córdova, 2005):

> Para garantizar un buen trabajo de cobranza y un adecuado control de las cuentas de los clientes, debe llevarse un estricto registro de aquellas cuentas que han permanecido insolutas excediendo las condiciones normales de venta y ya están vencidas. Esto se llama determinar la antigüedad de los saldos de las cuentas por

cobrar. La antigüedad de cuentas significa su separación en diferentes categorías: primero las cuentas que están dentro de los términos y no están vencidas; después las de 1 a 30 días de vencidas, las de 30 a 60, las de 60 a 90, y así sucesivamente.

3.2.4.2 Modelo de portafolio

Uno de los modelos de datos más importante para soluciones de pago es el Modelo de Portafolio, este modelo de datos consolida la información de la cartera activa y judicial mostrando los cambios de tramo de la cuenta con cortes mensuales o de cosecha desde su ingreso a cobranzas. El modelo de portafolio facilita al negocio en el seguimiento de las gestiones y pagos de la cartera activa y judicial y también les permite desarrollar nuevos tableros de control para la toma de decisiones.

El concepto de tramo de mora es importante en este modelo ya que dependiendo del tramo en el que se encuentra una cuenta se le aplica distintas estrategias de cobranzas. Los tramos de mora definidos por el negocio son los siguientes:

- **Tramo de mora [1 - 8]**

  El grupo de cuentas de este tramo son gestionadas por los canales alternos que son envíos de SMS, cargos automáticos, cargos manuales, etc)

- **Tramo de mora [9 - 30] y [31 - 60]**

  El grupo de cuentas de este tramo son gestionadas por el canal Telefonía, el canal llama a los clientes morosos en este tramo recordándoles que tienen una deuda pendiente o brindándoles facilidades de pago con el objetivo que el cliente se comprometa a pagar su deuda en una fecha determinada, a esta gestión se le llama promesa de pago.

- **Tramo de mora [61 - 90] y [91 - 120]**

  El grupo de clientes de este tramo son gestionados por el canal Campo, es decir que los gestores se acercan al domicilio del cliente para ofrecerles facilidades de pago y obtener una promesa de pago.

- **Tramo de mora [120 a más]**

  El grupo de clientes de este tramo están gestionados por el canal de judicial, el canal utiliza estrategias como venta de la cartera judicial, gestión de la cartera judicial a través de outsourcing o gestión directa.

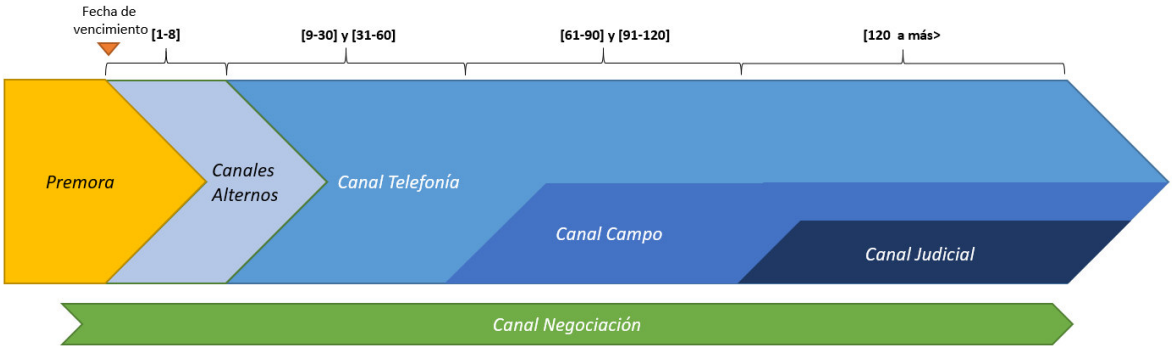


Figura 3: Flujo de gestión de una unidad de soluciones de pago. Fuente: Elaboración propia (adaptado).

Respecto a la información que almacena el Modelo de Portafolio, se guarda datos en 3 momentos y por cada tramo, en el primer momento o ingreso al tramo se guarda la información de la deuda vencida, deuda total y fecha de ingreso. Para el segundo tramo o durante el tiempo que la cuenta permanece en un tramo se guarda la información de pagos y cantidad de gestiones y finalmente el tercer momento o cuando la cuenta sale del tramo se guarda la información de monto de deuda vencida, monto de deuda total, nuevo tramo al que migra, cantidad de gestiones y total de pagos.

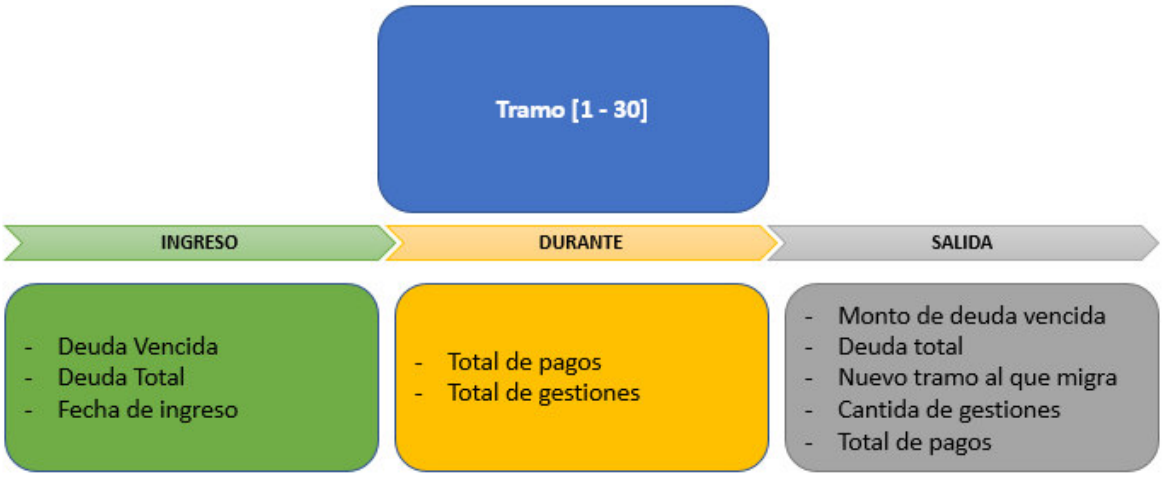


Figura 4. Flujo de datos que se almacena por cada tramo. Fuente: Elaboración propia.

Según los cortes mencionados anteriormente, el Modelo de Portafolio genera las siguientes bases:

- **Modelo de Portafolio Mensual**

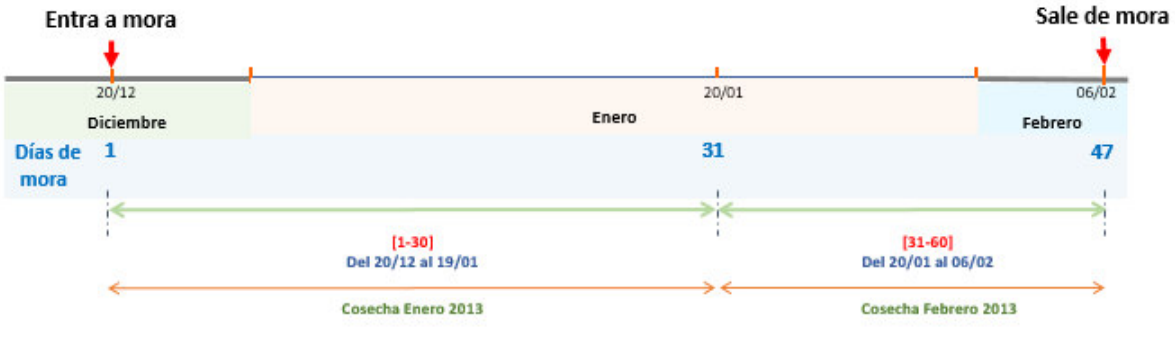
Este modelo guarda la información mencionada anteriormente por cortes mensuales. Por ejemplo, una cuenta entra a mora el día 21 de diciembre, por lo tanto empieza en el tramo [1 - 30], hasta el día 31 de diciembre acumula 11 días de mora y por ser fin de mes se cierra el registro con fecha de salida del 31 y se genera un nuevo registro con fecha de ingreso del 1 de enero con 12 días de mora y en el tramo [1 -30], hasta el 19 de enero se acumulan 30 días de mora lo que significa que va a pasar al tramo [31 – 60], por lo tanto se cierra este registro con fecha de salida del 30 y se genera un nuevo registro con fecha del 20 de enero con el tramo de [31 - 60] y 31 días de mora, y así sucesivamente hasta que la deuda sea cancelada.

Cuenta	Tramo	Mes Ingreso	Fecha Ingreso	Fecha Retiro	Días de Mora	Deuda Vencida al Ingreso	Pago Acumulado	Gestión Acumulada
X	[1-30]	201212	21/12	31/12	11	60	10	4
X	[1-30]	201301	01/01	19/01	30	65	20	4
X	[31-60]	201301	20/01	30/01	41	90	0	5
X	[31-60]	201302	01/02	06/02	47	100	100	2

Figura 5. Datos registrados para el Modelo de Portafolio Mensual Fuente: Elaboración propia.

- **Modelo de Portafolio de Cosecha**

Para este modelo primero se tiene que definir el rango de fechas de una cosecha. Por ejemplo se define que las cosechas para enero del 2013 serán del 20 de diciembre del 2012 al 19 de enero del 2013 y las cosechas para febrero del 2013 serán del 20 de enero del 2013 al 19 de febrero del 2013. Teniendo en cuenta estas fechas de cosechas, pongamos de ejemplo que una cuenta entra en mora el día 20 de diciembre del 2012, por lo tanto entra al tramo de [1 - 30] con 1 día de mora y fecha de ingreso del 20, y debido a que la fecha se encuentra dentro del rango se le asigna el mes de cosecha de enero del 2013. Al llegar el día 19 de enero la cuenta acumula un total de 30 días de mora, cierra el tramo [1 -30] con fecha de salida del 19 y se genera un nuevo registro con fecha de ingreso del 20 de enero en el tramo [31 - 60] con 31 días de mora y con el mes de cosecha asignado para febrero del 2013 por estar dentro del rango. Así sucesivamente hasta que la cuenta salga de mora.



Cuenta	Tramo	Mes Cosecha	Fecha Ingreso	Fecha Retiro	Días de Mora	Deuda Vencida al Ingreso	Pago Acumulado	Gestión Acumulado
X	[1-30]	201301	20/12	19/01	30	60	30	8
X	[31-60]	201302	20/01	06/02	47	90	100	7

Figura 6. Datos registrados para el Modelo de Portafolio de Cosechas. Fuente: Elaboración propia.

3.2.4.3 Big Data

"El Big Data son activos de información de gran volumen, alta velocidad y / o gran variedad que exigen formas rentables e innovadoras de procesamiento de información que permitan una mejor comprensión, toma de decisiones y automatización de procesos." (Gartner, 2021)

Otra definición lo tenemos en la página oficial de (Oracle, 2021), el cuál define a Big Data como:

> El Big Data está formado por conjuntos de datos de mayor tamaño y más complejos, especialmente procedentes de nuevas fuentes de datos. Estos conjuntos de datos son tan voluminosos que el software de procesamiento de datos convencional sencillamente no puede gestionarlos. Sin embargo, estos volúmenes masivos de datos pueden utilizarse para abordar problemas empresariales que antes no hubiera sido posible solucionar.

3.2.4.3.1 Características del Big Data

Definitivamente el Big Data puede beneficiar a cualquier área que sepa cómo obtener valor de los datos almacenados. Según (Camargo-Vega, Camargo-Ortega, & Joyanes Aguilar, 2014), para que un entorno sea considerado Big Data, este debe tener las siguientes dimensiones básicas:

- **Volumen**

  Cada día, las empresas registran un aumento significativo de sus datos (terabytes, petabytes y exabytes), creados por personas y máquinas. En el año 2000 se generaron 800.000 petabytes (PB), de datos almacenados y se espera que esta cifra alcance los 35 zettabytes (ZB) en el 2020. Las redes sociales también generan datos, es el caso de Twitter, que por sí sola genera más de 7 terabytes (TB) diariamente, y de Facebook, 10 TB de datos cada día. Algunas empresas

generan terabytes de datos cada hora de cada día del año, es decir, las empresas están inundadas de datos. (p.66)

- **Variedad**

  Se puede mencionar que va muy de la mano con el volumen, pues de acuerdo con éste y con el desarrollo de la tecnología, existen muchas formas de representar los datos; es el caso de datos estructurados y no estructurados; estos últimos son los que se generan desde páginas web, archivos de búsquedas, redes sociales, foros, correos electrónicos o producto de sensores en diferentes actividades de las personas. (p.66)

- **Velocidad**

  Se refiere a la velocidad con que se crean los datos, que es la medida en que aumentan los productos de desarrollos de software (páginas web, archivos de búsquedas, redes sociales, foros, correos electrónicos, entre otros) (p.66)

Para (Perez, 2015), además de las dimensiones mencionadas anteriormente se deben considerar las siguientes:

- **Veracidad**

  La veracidad tiene que ver con lo incierto o imprecisión de los datos. Cuando utilizamos fuentes de datos de redes sociales como tweets, entradas de Facebook, etc, ¿qué credibilidad podemos o debemos dar a los datos? Muy posiblemente podremos utilizar estos datos para realizar análisis de sentimiento o identificar cambios de tendencias, pero no podremos sacar conclusiones para la toma de decisiones críticas. (p.80)

- **Valor**

  Valor se refiere tanto al coste de la tecnología como al valor obtenido de su uso.

  La variable del coste es importante, ya que es uno de los factores clave que definen la novedad del Big Data. La diferencia es que antes solo los gobiernos y las grandes empresas se podían permitir tener grandes centros de datos, soluciones de gestión de

datos en tiempo real, de análisis de contenidos desestructurados, sistemas de supercomputación, etc. Ahora estas tecnologías son más accesibles.

Por otro lado el valor también se refiere al beneficio obtenido de las iniciativas Big, Data que IDC las clasifica en (p.81, 82):

- Reducción del coste de capital: reducción del coste de hardware, software y otros costes de infraestructuras.
- Eficiencia de las operaciones: reducción de los costes de operaciones, debido a la mejora de los métodos de integración, gestión, análisis y entrega de datos.
- Mejora de los procesos de negocio: aumento en los ingresos o beneficios debido a una mejora de los procesos de negocio, incluyendo mejoras en el diseño y la prestación de servicios a los ciudadanos, en los procesos de licitación de contratos públicos, etc.

Según las investigaciones realizada por (BBVA Api Market, 2020), existen dos nuevas dimensiones que se deben considerar:

- **Variabilidad**

  "La **variabilidad** hace referencia a la variabilidad en el significado, en el léxico. Esto es relevante a la hora de llevar a cabo análisis de percepciones. Los algoritmos deben ser capaces de comprender el contexto y descifrar el significado exacto de cada palabra en su respectivo entorno. Este análisis semántico resulta mucho más complejo".

- **Visualización**

  "La visualización es lograr que toda la cantidad de datos recolectados y analizados sean comprensibles y sencillos de leer. Sin una visualización adecuada, no se puede sacar el máximo rendimiento y aprovechamiento de los datos en bruto."

3.2.4.3.2 Arquitectura de Big Data

Con el fin de soportar análisis de grandes volúmenes de datos, toda arquitectura de Big Data deben tener las siguientes características según (Quiroz Martinez, Aguilar Duarte, & Intriago Cedeño, 2019) (p.242):

- **Almacenamiento:** "Los recursos hardware y software permiten el almacenamiento distribuido y redundante de los datos, lo que facilita su acceso y disponibilidad. Permiten responder la interrogante ¿dónde tener los datos?"
- **Procesamiento:** "Las herramientas y técnicas de procesamiento proveen el soporte tecnológico para operar con grandes volúmenes de datos en tiempo real. Permiten responder la interrogante ¿cómo trabajar con los datos?"
- **Análisis:** "Los métodos y técnicas computacionales realizan el análisis de los datos, lo que favorece la toma de decisiones oportuna y eficiente en las entidades financieras. Permiten responder a la interrogante ¿qué hacer con los datos?"

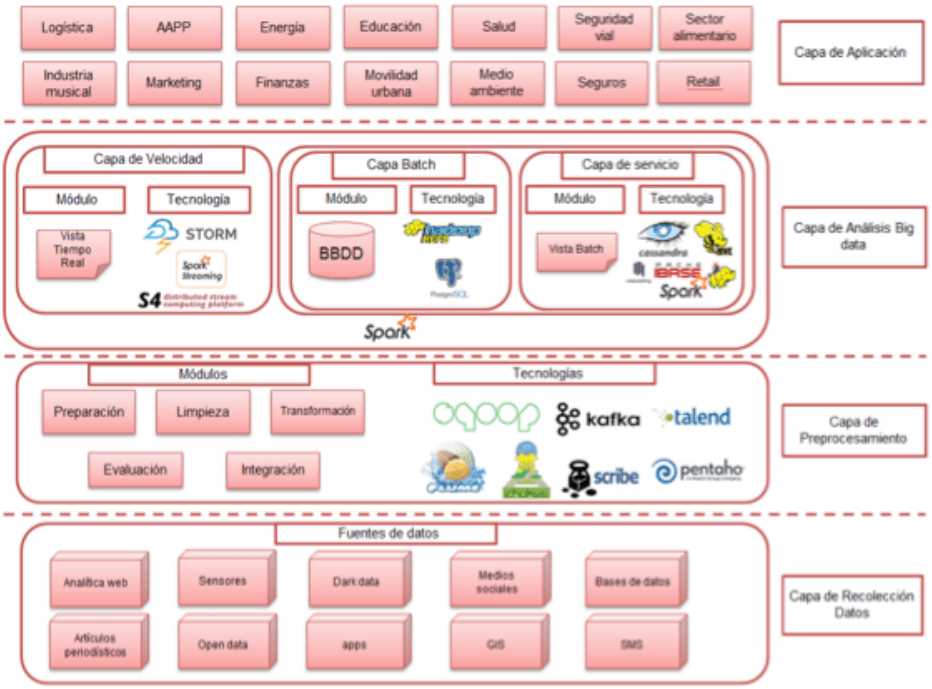


Figura 7. Arquitectura Lambda para Big Data. Fuente: Fractalia

3.2.4.4 Data Lake y Data Mart

3.2.4.4.1 Data Lake

Como comenta (Torres, Aguilar, Martin, & Diaz, 2017) en la Jornada de Automáticas N° 38 en alianza con la Universidad de Oviedo, el Data Lake se define como:

> Una gran cantidad de datos primarios en distintos formatos hasta que resultan necesarios para extracción. Estos lagos de datos pueden ser estructurados y no estructurados, "en el lago y luego permitir que las personas "destilen" sus propias visualizaciones particulares utilizando aquella tecnología que mejor se adapte a la tarea (por ej., SQL o NoSQL, bases de datos basadas en disco o en memoria, MPP o SMP.) Y el usuario crea sus visualizaciones de empresa mediante la compilación y agregación de datos desde múltiples vistas locales" (p.627). Existen riesgos debido a una mala gestión de los datos importantes como la pérdida de contexto significativo.

En nuestro contexto, la entidad financiera tiene implementado un Data Lake que trabaja sobre un cluster de Hadoop y que a su vez está dividido en 4 zonas:

- RDV: Raw Data Vault, esta zona almacena los datos tal cuál es generado desde su origen, es decir que no tiene ninguna alteración, la data en esta zona se guarda de manera histórica en formato avro y está disponible en cualquier momento.
- UDV: Universal Data Vault, esta zona almacena los datos de RDV pero con una estructura y modelado definidos, es una zona donde se encuentra la data disponible para todas las unidades de negocio. Por ejemplo se guarda datos de clientes, cuentas, transacciones, saldos, etc. Se guarda en formato parquet.
- DDV: Dimensional Data Vault, en esta zona se implementan los modelos de datos para las unidades de negocio, es decir, los datos de la capa UDV pasan por un

proceso de transformación y se guardan en un espacio asignado para la unidad de negocio. Los datos se guardan en formato parquet.

- EDV: Experimental Data Vault, es una zona donde los data scientist o data analyst exploran la data disponible y prueban sus modelos de datos. En esta zona no se pueden guardar datos de forma permanente ya que es una zona experimental.

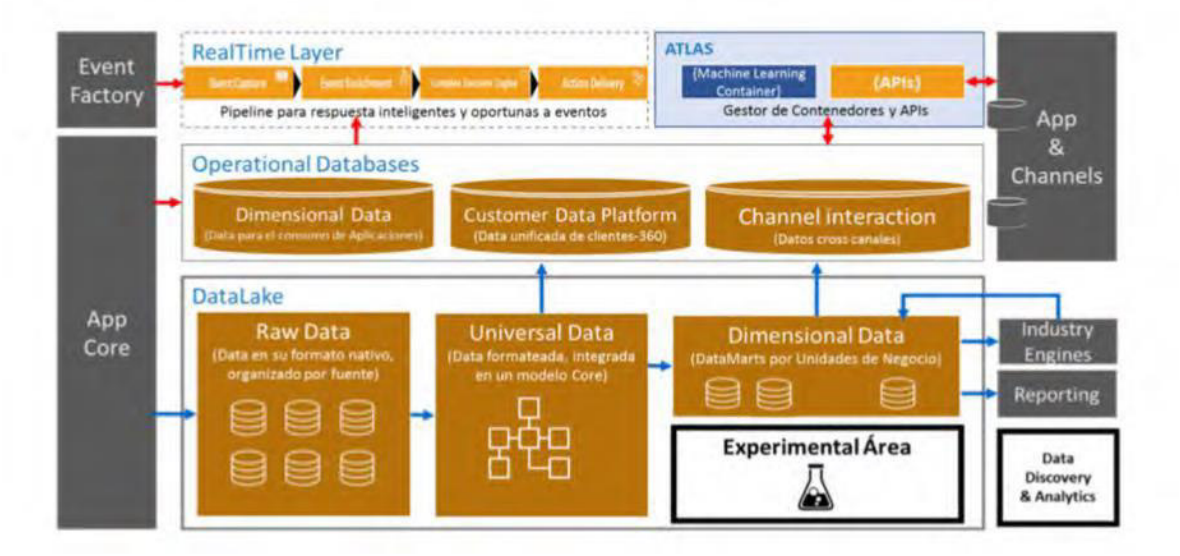


Figura 8. Componentes de una plataforma de datos. Fuente: Entidad Financiera

3.2.4.4.2 Data Mart

Según (Calderón, 2014) conceptualiza a un Data Mart como almacenamiento especializado en una unidad negocio "que se caracteriza por disponer de una estructura óptima de datos para analizar la información al detalle desde todas las perspectivas que afecten a los procesos de dicho departamento. Un Data Mart puede ser alimentado desde los datos de un Data Warehouse o integrar por sí mismo un compendio de distintas fuentes de información". (p.24)

Para el caso de Soluciones de Pago, el data mart está conformado por 3 capas, la primera es la capa Stage que carga en las tablas del data mart los datos que dejan los aplicativos del negocio, como Debt Manager, Nmac, Infinix, Triad y Negociación, y también data entries (excel, csv, txt) sin aplicar alguna regla de negocio ni limpieza de datos. La siguiente es la capa

ODS, en esta capa los datos cargados en la capa Stage pasa por un proceso de limpieza, integración y modelamiento, y en conjunto con las tablas del Data Warehouse y otros data mart se crean las tablas maestras y lookups. Finalmente, la capa BDS es la capa en donde se aplican reglas de negocio para crear las tablas de hechos como la Fact de la cartera, Fact de pagos, Fact de gestiones y Fact de PDP que son las principales en el datamart.

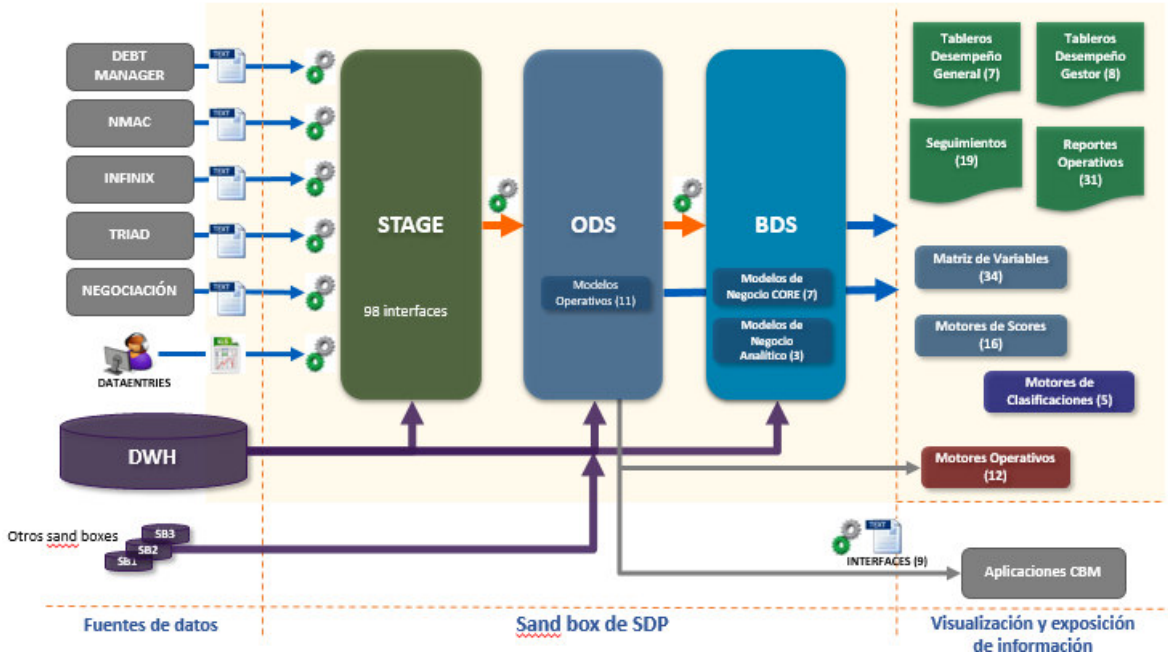


Figura 9. Arquitectura de datos de un data mart. Fuente: Entidad Financiera

3.2.4.5 Herramientas Tecnológicas para la Migración

Para la capa de procesamiento, tenemos principalmente a las siguientes herramientas:

3.2.4.5.1 PySpark

"PySpark es una interfaz para Apache Spark en Python. No solo le permite escribir aplicaciones Spark utilizando las API de Python, sino que también proporciona el shell de PySpark para analizar interactivamente sus datos en un entorno distribuido" (Apache Spark, 2021)

PySpark es un lenguaje que trabaja con la memoria RAM, te permite realiza análisis exploratorio de datos, construir pipelines de machine learning y crear ETL en una plataforma de datos distribuidos. (Towars Data Science, 2018)

Una de las librerías más importantes de PySpark es PySparkSQL, esta librería te permite realizar análisis de tipo SQL a datos estructurados y semiestructurados, además se puede conectar con Apache Hive y convertir la data en dataframe. (Databricks, 2021)

3.2.4.5.2 Hive

Un concepto claro de Hive es la que define (IBM, Apache Hive, 2021):

> Apache Hive es un software de almacenamiento de datos de código abierto para leer, escribir y administrar archivos de conjuntos de datos grandes que se almacenan directamente en el Sistema de archivos distribuido Apache Hadoop (HDFS) u otros sistemas de almacenamiento de datos como Apache HBase. Hive permite a los desarrolladores de SQL escribir declaraciones Hive Query Language (HQL) que son similares a las declaraciones SQL estándar para consultas y análisis de datos. Está diseñado para facilitar la programación de MapReduce porque no es necesario conocer ni escribir un código Java extenso. En su lugar, puede escribir consultas de forma más sencilla en HQL, y Hive puede crear el mapa y reducir las funciones.

3.2.4.5.3 Hue

"Hue es un asistente SQL de código abierto para consultar bases de datos y almacenes de datos y colaborar" (Cloudera, 2021)

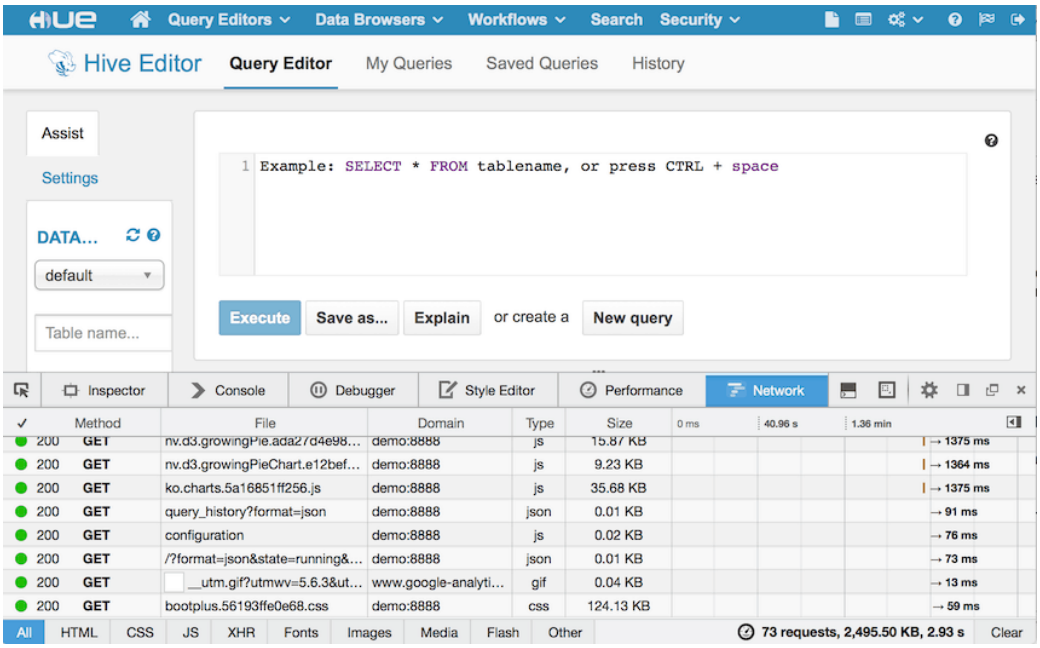


Figura 10: Interfaz web de Hue utilizando Hive. Fuente: Cloudera

3.2.4.5.4 DataStage

"Es una herramienta de integración de datos que sirve para diseñar, desarrollar y ejecutar trabajos que mueven y transforman datos. DataStage admite patrones de extracción, transformación y carga ETL" (IBM, DataStage, 2021)

Otra definición según (PowerData, n.d.) "es una herramienta ETL que se utiliza para extraer datos, transformarlos, aplicar en ellos principios de negocio y luego cargarlos con algún objetivo específico"

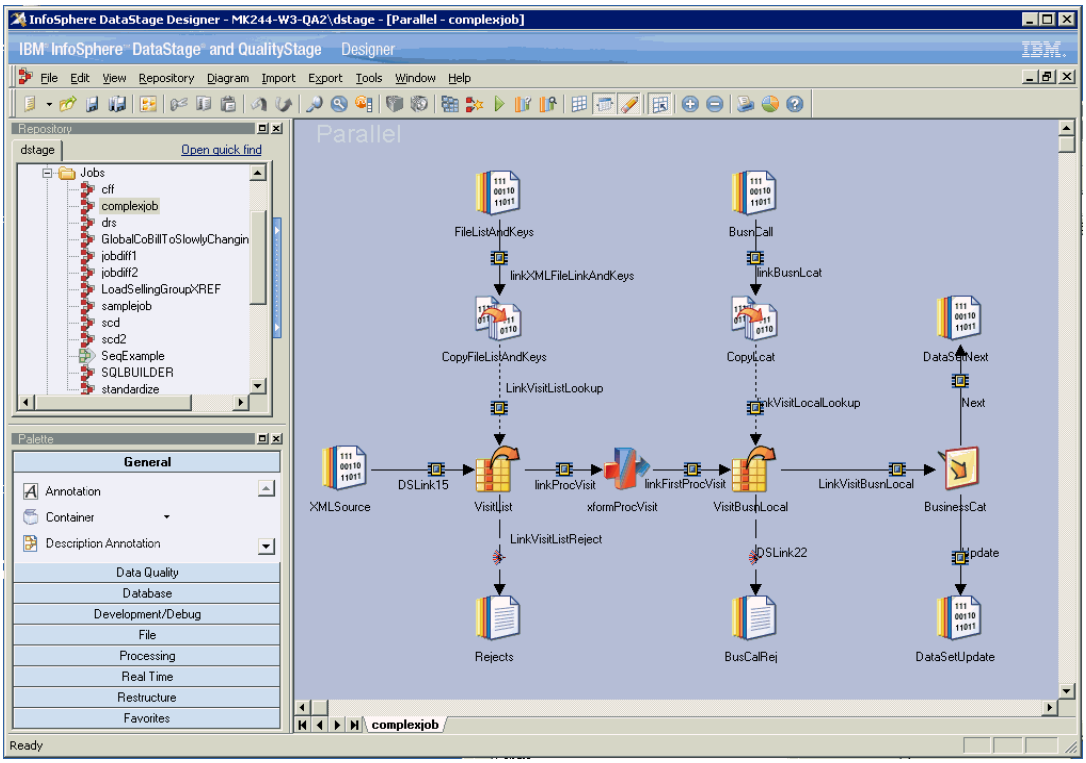


Figura 11: Ejemplo de proyecto DataStage. Fuente: analitica.si

3.2.4.5.5 HDFS

HDFS es una tecnología desarrollada por (Apache Hadoop, 2021) para el almacenamiento de datos de forma distribuida el cuál lo define como:

> Un sistema de archivos distribuido diseñado para ejecutarse en hardware básico. Tiene muchas similitudes con los sistemas de archivos distribuidos existentes. Sin embargo, las diferencias con otros sistemas de archivos distribuidos son significativas. HDFS es

altamente tolerante a fallas y está diseñado para implementarse en hardware de bajo costo. HDFS proporciona acceso de alto rendimiento a los datos de la aplicación y es adecuado para aplicaciones que tienen grandes conjuntos de datos.

3.2.4.5.6 Avro

"Apache Avro es un sistema de serialización de datos que proporciona estructuras de datos enriquecidas, formato de datos binarios e integración con lenguajes dinámicos." (Apache Avro, 2021)

Una definición más extensa lo describe (Vohra, 2016) en su libro Practical Hadoop Ecosystem:

> Apache Avro es un formato de serialización de datos binarios compacto que proporciona estructuras de datos variadas. Avro usa esquemas de notación JSON para serializar / deserializar datos. Los datos de Avro se almacenan en un archivo contenedor (un archivo .avro) y su esquema (el archivo .avsc) se almacena con el archivo de datos. A diferencia de otros sistemas similares, como los búferes de protocolo, Avro no requiere la generación de código y utiliza escritura dinámica. Los datos no están etiquetados porque el esquema va acompañado de los datos, lo que da como resultado un archivo de datos compacto. Avro admite el control de versiones; pueden coexistir diferentes versiones (que tienen diferentes columnas) de los archivos de datos de Avro junto con sus esquemas. Otro beneficio de Avro es la interoperabilidad con otros lenguajes debido a su formato binario eficiente. (p.303)

3.2.4.5.7 Parquet

"Apache Parquet es un formato de almacenamiento en columnas disponible para cualquier proyecto en el ecosistema Hadoop, independientemente de la elección del marco de procesamiento de datos, modelo de datos o lenguaje de programación." (Apache Parquet, 2021)

El principal atributo de parquet es su almacenamiento columnar y su sistema de compresión tal como lo describe (Vohra, 2016):

> Apache Parquet es un formato de archivo binario comprimido, eficiente, estructurado, orientado a columnas (también llamado almacenamiento en columnas). Parquet admite varios códecs de compresión, incluidos Snappy, GZIP, deflate y BZIP2. Snappy es el predeterminado. Los formatos de archivo estructurados como RCFile, Avro, SequenceFile y Parquet ofrecen un mejor rendimiento con soporte de compresión, lo que reduce el tamaño de los datos en el disco y, en consecuencia, los recursos de E/S y CPU necesarios para deserializar los datos. (p.325)

3.2.5 IMPLEMENTACIÓN DE LAS ÁREAS, PROCESOS y SISTEMAS

La unidad de soluciones de pago inició su proceso de migración a la plataforma big data del banco con la implementación del Modelo de Portafolio, empezando con la definición del alcance del proyecto, analizar disponibilidad de fuentes, diseñar la arquitectura de la solución, implementación y despliegue de la solución.

La implementación de la Migración del modelo de portafolio de la unidad de negocio de soluciones de pago a un entorno de big data para la gestión de la cartera morosa en una entidad financiera se ha desarrollado según las siguientes etapas:

3.2.5.1 Análisis del Requerimiento

Durante esta etapa se ha revisado el modelo de Portfolio existente en el data mart de SDP que está implementado en Oracle, se han analizado las variables finales en coordinación con los analistas de negocio y luego de tener definido estas variables se pasó a realizar la trazabilidad de estas para llegar a las fuentes originales y validar si estas fuentes existen en el data lake.

Como resultado de la etapa, se ha generado el diccionario de negocio que ha sido definido por los analistas, también se elaboró un documento de trazabilidad de fuentes y variables finales del modelo actual en Oracle y adicionalmente un documento con los datos homólogos en el data lake.

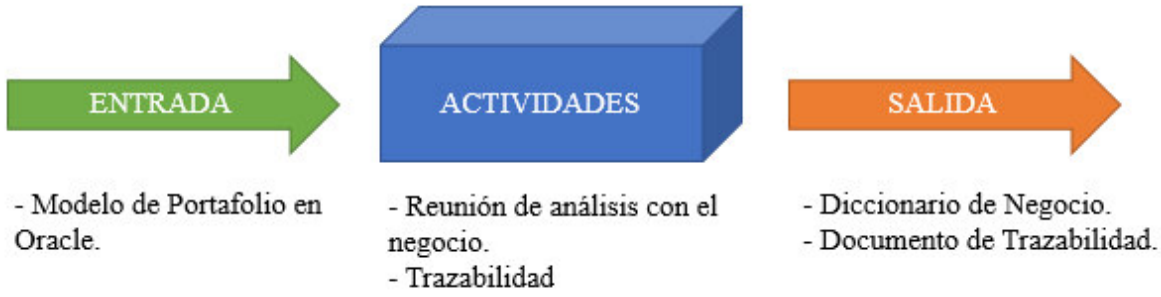


Figura 12. Flujo de entrada y salida de la etapa de análisis del requerimiento. Fuente: Elaboración propia

3.2.5.1.1 Definir las variables finales

Durante esta actividad se programaron reuniones con los analistas de negocio y con los analistas de gobierno de datos para definir cuáles serían las variables finales a implementar en el Modelo de Portafolio generando como entregable el Diccionario de Negocio, este documento tiene como finalidad identificar que campos de las tablas finales tienen mayor nivel de importancia respecto a otros, con el objetivo de priorizar la calidad de los datos de mayor importancia. Las variables que se identifican en el diccionario de negocio para cada campo son:

Tabla 6

Descripción de variables para el documento de negocio

Variable	Descripción
Term Name	Nombre del dato
Steward	Nombre del Owner
Long Description	Descripción completa y bien detallada para el entendimiento de cualquier persona.
Usage	Explicar claramente para que es usado el dato dentro y fuera del dominio.
Example	Colocar muestras de cómo es visualizado el dato.
Dominio	Indicar a que dominio pertenece
Subdominio	Colocar el nombre del subdominio al que pertenece el ED

Código de identificación	Llenar de acuerdo a la tabla de abreviaciones de los dominios
Dato de alta criticidad (DAC)	Indicar con Si/NO si es un dato sensible
Criticidad	Indicar con (EDC/EDNC) si el dato es relevante para el negocio
Sustento de criticidad	Si el ED es señalado como dato critico indicar en este atributo a que se debe la criticidad del Dato.
Nivel de seguridad del dato	Elegir entre publico/privado/restringido
Origen	Seleccionar si es interno o externo
Periodicidad de uso	Seleccionar si es diario/semanal/quincenal/mensual/trimestral/semestral/anual
Uso en reportes regulatorios	Indicar con SI/NO si es un dato que se usa en reportes regulatorios
Otros consumidores de la red	Indicar el nombre de las unidades que usan el dato
Fuente oficial	DWH / Sandbox Nombre del Sandbox
Tabla oficial	Tabla en la fuente Oficial
Campo oficial	Campo en la tabla Oficial

El modelo de portafolio comprende dos tablas finales que son la Fact de Portafolio Mensual y la Fact de Portafolio de Cosechas, para la primera tabla se han identificado un total de 87 campos y para la segunda tiene un total de 92 campos en el diccionario de negocio.

Term Name	Usage	Example	DOMINIO	SUBDOMINIO	DATO DE ALTA CRITICID	CRITICIDAD	SUSTENTO DE CRITICIDAD	NIVEL DE SEGURIDAD DEL	ORIGEN	OTROS CONSUMIDORES	FUENTE OFICIAL	TABLA OFICIAL	CAMPO OFICIAL
Periodo del portafolio asignado a Cobranzas al ingresar a un tramo de mora	Este dato es utilizado en los siguientes cálculos más importante: el roll rate, provisiones, ingresos a mora, etc.	202004	Cobranza Minorista	Portafolio	No	EDNC	Es un dato para medición de resultados y decisiones estratégicas	Privado	Interno	Soluciones de Pago	Proy_Cobranzas	FCR_TM_RM_GC_CL_COS	CODMESCOSECHA
Código identificador de la cuenta del cliente en SDP	Este dato es utilizado en los siguientes cálculos más importante: el roll rate, provisiones, ingresos a mora,	17494325	Cobranza Minorista	Portafolio	No	EDC	Es un dato para medición de resultados y decisiones	Privado	Interno	Soluciones de Pago	Proy_Cobranzas	FCR_TM_RM_GC_CL_COS	CODCLAVEOPECTA
Fecha de ingreso al tramo de mora en el periodo cosecha	Este dato es utilizado en los siguientes cálculos más importante: el roll rate, provisiones, ingresos a mora,	13/4/2020	Cobranza Minorista	Portafolio	No	EDC	Es un dato para medición de resultados y decisiones	Privado	Interno	Soluciones de Pago	Proy_Cobranzas	FCR_TM_RM_GC_CL_COS	FECINGRESORANGOMORACOSEC
código del rango de mora asignado a la cuenta del cliente.	Este dato es utilizado en los siguientes cálculos más importante: el roll rate, provisiones, ingresos a mora,	73	Cobranza Minorista	Portafolio	No	EDC	Es un dato para medición de resultados y decisiones	Privado	Interno	Soluciones de Pago	Proy_Cobranzas	FCR_TM_RM_GC_CL_COS	CODRANGOMORAINGCOS
Código del producto de la cuenta del cliente	Este dato es utilizado en los siguientes cálculos más importante: el roll rate, provisiones, ingresos a mora,	30	Cobranza Minorista	Portafolio	No	EDC	Es un dato para medición de resultados y decisiones	Privado	Interno	Soluciones de Pago	Proy_Cobranzas	FCR_TM_RM_GC_CL_COS	CODPRODUCTORBP
Código del distrito con el cual se registró al cliente	Este dato es utilizado en los siguientes cálculos más importante: el roll rate, provisiones, ingresos a mora,	1316	Cobranza Minorista	Portafolio	No	EDNC	Es un dato para medición de resultados y decisiones	Privado	Interno	Soluciones de Pago	Proy_Cobranzas	FCR_TM_RM_GC_CL_COS	CODDISTRITO
Código de la moneda de la cuenta del cliente	Este dato es utilizado en los siguientes cálculos más importante: el roll rate, provisiones, ingresos a mora,	1001	Cobranza Minorista	Portafolio	No	EDNC	Es un dato para medición de resultados y decisiones	Privado	Interno	Soluciones de Pago	Proy_Cobranzas	FCR_TM_RM_GC_CL_COS	CODMONEDA
Código identificador del cliente en el Banco	Este dato es utilizado en los siguientes cálculos más importante: el roll rate, provisiones, ingresos a mora, etc.	3762770	Cobranza Minorista	Portafolio	No	EDC	Es un dato para medición de resultados y decisiones estratégicas	Privado	Interno	Soluciones de Pago	Proy_Cobranzas	FCR_TM_RM_GC_CL_COS	CODCLAVECIC

Figura 13. Ejemplo de un diccionario de negocio. Fuente: Entidad Financiera.

Tabla 7

Datos finales del diccionario de negocio con alta criticidad para el Modelo de Portafolio Mensual

Dato	Descripción
Código identificador de la cuenta del cliente en SDP	Código identificador que se crea y se asigna a la cuenta del cliente cuando este ingresa a Soluciones de Pago
Fecha de ingreso al tramo de mora en el periodo cosecha	Fecha en la cual la cuenta del cliente ingresa a un tramo de mora determinado dentro del periodo de la cosecha.
Código del rango de mora asignado a la cuenta del cliente.	Código que identifica el tramo de mora en el cual se encuentra el cliente en el periodo cosecha
Código del producto de la cuenta del cliente	Código genérico asignado al producto de la cuenta del cliente, este código se extrae de las tablas de DWH.
Código identificador del cliente en el Banco	Código genérico que permite identificar al cliente en las bases del banco, este código se extrae de las tablas de DWH.
Fecha en la cual la cuenta se retira del tramo de mora.	Fecha en la cual la cuenta del cliente se retira del tramo de mora determinado dentro del periodo de la cosecha.
Código de router en el día en que la cuenta se retira del tramo de mora	Código del router definido en Soluciones de Pago que se asigna a la cuenta del cliente al momento que se retira de un determinado tramo de mora dentro del periodo cosecha
Código de router al cierre del tramo de mora y periodo.	Código del router definido en Soluciones de Pago que se asigna a la cuenta del cliente al cierre del día de salida del tramo de mora dentro del periodo cosecha
Código del Subrango de mora de la cuenta en el día de ingreso al tramo de mora	Código del subrango de mora (tramo) asignado a la cuenta del cliente en el día de ingreso a mora en el periodo cosecha del mes. En el momento que ingresa la cuenta a mora se le asigna un tramo de mora.

Código del Subrango de mora de la cuenta al cierre del tramo de mora	Código del subrango de mora (tramo) asignado a la cuenta del cliente al cierre del día de salida del tramo de mora en el periodo cosecha.
Cantidad de días de mora inicial de la cuenta con que entra al tramo	Cantidad de días en mora con las que el cliente ingresa al periodo en curso. La medición del roll rate y provisiones se basa en la cantidad de clientes que ingresan desde el primer hasta el último día calendario del periodo cosecha
Tipo de bloqueo de la cuenta del cliente al cierre del día de retiro del tramo	Tipo de bloqueo que tiene la cuenta del cliente en el sistema del banco, con este bloqueo se registra la cuenta al momento de retiro de un tramo de mora en el periodo cosecha
Código de subcanal en el día en que la cuenta del cliente ingresa al tramo de mora	Código del subcanal definido en Soluciones de Pago asignado a la cuenta del cliente en el día que ingreso a mora y a un determinado tramo de mora en el periodo cosecha
Monto deuda total en soles de la cuenta del cliente a la fecha de ingreso al tramo de mora	Deuda total en soles de la cuenta que mantiene el cliente con el Banco al ingresar al tramo de mora, el cual incluye interés. Esta deuda es la suma de los montos que se encuentran al día más los montos que se encuentran en mora.
Monto deuda total en soles de la cuenta del cliente al cierre de la fecha de retiro del tramo de mora	Deuda total en soles de la cuenta que mantiene el cliente con el Banco al retirarse del tramo de mora, el cual incluye interés. Esta deuda es la suma de los montos que se encuentran al día más los montos que se encuentran en mora.
Monto deuda vencida en soles de la cuenta del cliente a la fecha de ingreso al tramo de mora	Deuda vencida en soles de la cuenta que mantiene el cliente con el Banco al ingresar al tramo de mora, el cual incluye interés. Esta deuda solo contempla el monto que se encuentra en mora.
Monto deuda vencida en soles de la cuenta del cliente a la fecha de retiro del tramo de mora	Deuda vencida en soles de la cuenta que mantiene el cliente con el Banco al retirarse del tramo de mora, el cual incluye interés. Esta deuda solo contempla el monto que se encuentra en mora.
Suma de los pagos sin payoff en soles acumulado de la cuenta realizados dentro del tramo de mora	Monto total de los pagos en soles realizados por el cliente a su cuenta en mora, Este pago tiene que ser menor o igual que la deuda vencida del cliente dentro del tramo mora
Monto de la deuda posición del cliente en soles, cuentas en etapa morosa y no morosa.	Deuda total en soles (morosa y no morosa) del cliente de la cartera cobranza, solo se considera los productos de ALS y Visión Plus tomada al ingreso a la Cosecha (cierre del día anterior).
Suma de los pagos con cargo automático cude en soles realizados dentro del tramo	Monto total de los pagos en soles cargados por el cargador cude desde las cuentas de ahorros/corrientes del cliente y que son dirigidos a la cuenta en mora que tiene el cliente dentro del tramo de mora.

3.2.5.1.2 Identificación de Fuentes

Primero se ha identificado la arquitectura actual de la solución que se encuentra en Oracle para tener una visión general de lo que ya está implementado, se identifica que la solución pasa por todas las capas del data mart de SDP.

En primera instancia tenemos a la capa ODS que almacena los datos del aplicativo Debt Manager en distintas tablas, pero para el modelo solo se necesitan 3 los cuales son los datos de gestiones, cartera o saldos y Promesas de Pago o PDP.

En la siguiente capa que es la BDS se aplican reglas de negocio para crear en conjunto con algunas tablas de Data Warehouse las tablas de hechos o fact del core como son la Fact de la Cartera, Fact de Gestiones, Fact de PDP y la Fact de Pagos.

En la capa de reportes es donde se han implementado 3 tablas finales del modelo, la primera tabla es la Fact de Portafolio Diario, esta tabla contiene la información de saldos, pagos, gestiones y el tramo de mora en el que se encuentra cada cuenta de manera diaria, es decir es una tabla histórica diaria. A partir de la tabla mencionada anteriormente se crean dos tablas fact como la Fact de Portafolio Mensual y la Fact de Portafolio de Cosechas, ambas tablas contienen datos de saldos, pagos y gestiones, la principal diferencia entre ambas tablas es que una es una histórica mensual y la otra es una histórica mensual de cosechas. Finalmente tenemos la capa de la vista en donde se crean variables de seguimiento para que sean consumidas por las herramientas disponibles como Excel.

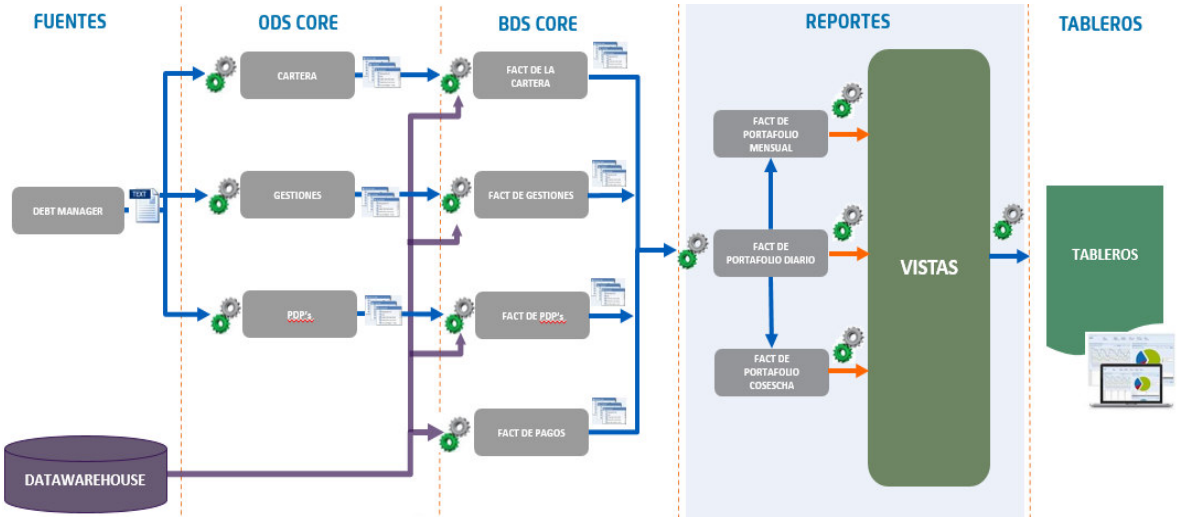


Figura 14. Arquitectura de datos del Modelo de Portafolio. Fuente: Entidad Financiera.

También se ha identificado el modelo dimensional de la solución en Oracle que utiliza fuentes del data mart de SDP y tablas de data warehouse.

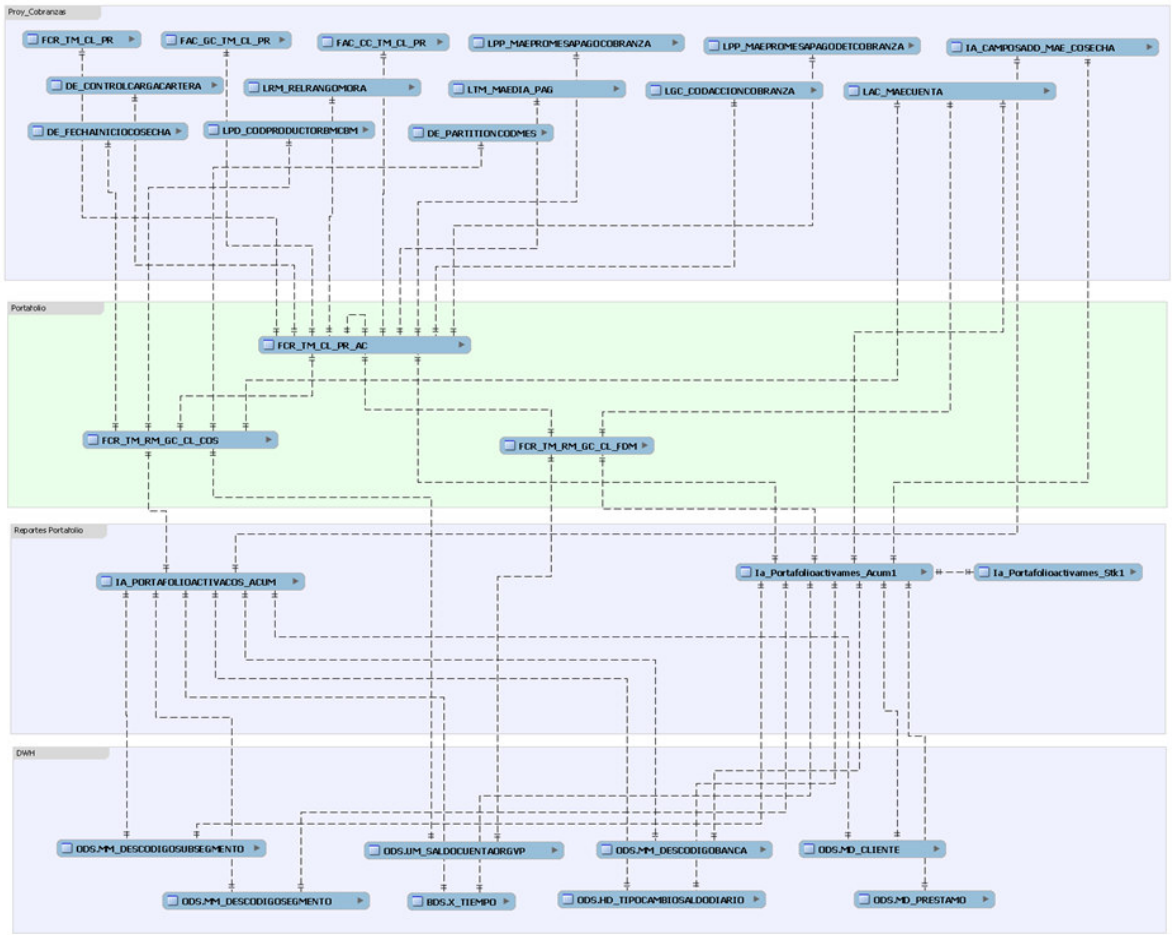


Figura 15. Modelo Dimensional de Modelo de Portafolio. Fuente: Entidad Financiera

Nota: En la zona de proy_cobranzas se encuentran las tablas que pertenecen al data mart de SDP como la Fact de la Cartera, Fact de Gestiones, Fact de Pagos, Fact PDP, maestra de cuentas, lookup de productos, lookup de fechas de cosechas, lookup de pagos y lookup de rango de mora. En la zona DWH están las tablas que se pertenecen a data warehouse del banco como la tabla de tiempo, prestamos, saldos de tarjetas, clientes y segmento. En la capa Portafolio se encuentran las tablas finales como la Fact de Portafolio Diaria, Fact de Portafolio Mensual y Fact de Portafolio de Cosechas. En la capa de Reportes Portafolio se encuentran las vistas generadas para que sean consumidas por herramientas de visualización.

Conociendo la arquitectura y el modelo dimensional de la solución en Oracle ahora debemos hacer la trazabilidad de los datos definidos en el diccionario de negocio para identificar cuáles serían nuestras fuentes finales. Para lograr esto hemos utilizado el método del cangrejo, se ha empezado tomando los campos de la tabla de portafolio y hemos trazado la ruta del campo hacia atrás, identificando tablas intermedias importantes como la Fact de pagos, maestras de cuentas, etc hasta llegar a los campos del aplicativo debt manager o las tablas de data warehouse.

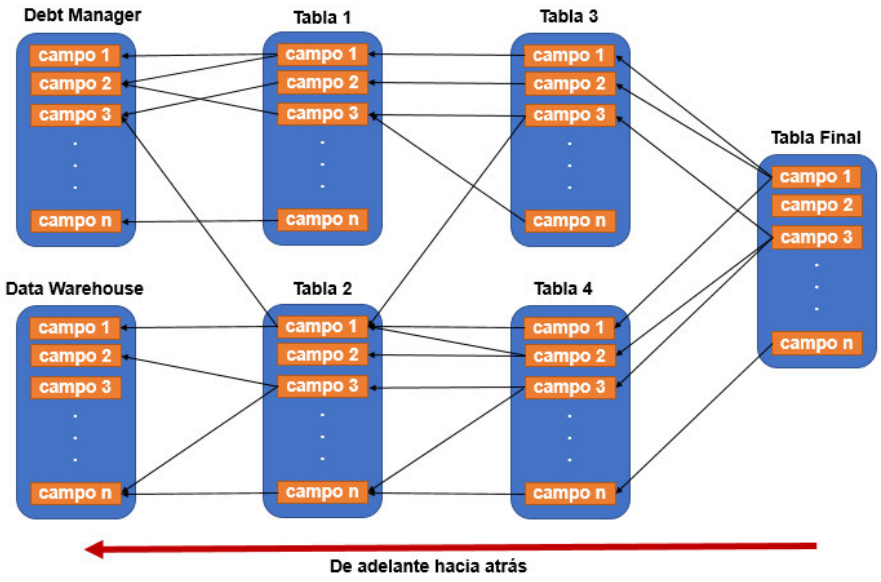


Figura 16. Método de la trazabilidad de datos para identificar las fuentes finales. Fuente: Elaboración propia.

El resultado de la trazabilidad se volcó a un documento con los siguientes campos:

- Tabla Final: Nombre de la tabla que se está mapeando.
- Campo Final: Nombre del campo que se está mapeando.
- ¿Migrar desde Sandbox?: Completa el campo con Si/No que indica el campo en análisis tiene que ser migrado desde el data mart hacia el data lake.
- Tipo de Campo: Indica si el campo ha sufrido transformaciones desde su origen, si ha cambiado se pone “calculado” en caso contrario se coloca “lineal”.
- Tabla Fuente: Nombre de la tabla fuente.
- Camp Fuente: Nombre del campo fuente.

Tabla Final	Campo Final	Migrar de Sbx?	Tipo Cam	Tabla Fuente	Campo Fuente
FCR_TM_RM_GC_CL_FDM	CODMES	NO	Calculado	FCR_TM_CL_PR_AC	FECDIA
FCR_TM_RM_GC_CL_FDM	CODCLAVEOPECTA	NO	Lineal	FCR_TM_CL_PR_AC	CODCLAVEOPECTA
FCR_TM_RM_GC_CL_FDM	FECINGRESORANGOMORAMES	NO	Lineal	FCR_TM_CL_PR_AC	FECINGRESORANGOMORAMES18
FCR_TM_RM_GC_CL_FDM	FECINGRESORANGOMORAMES	NO	Lineal	FCR_TM_CL_PR_AC	FECINGRESORANGOMORAMES130
FCR_TM_RM_GC_CL_FDM	TIPRANGOMORA	NO	Calculado	FCR_TM_CL_PR_AC	CODRANGOMORA130
FCR_TM_RM_GC_CL_FDM	CODRANGOMORAINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODRANGOMORA18
FCR_TM_RM_GC_CL_FDM	CODRANGOMORAINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODRANGOMORA130
FCR_TM_RM_GC_CL_FDM	CODPRODUCTORBP	NO	Lineal	FCR_TM_CL_PR_AC	CODPRODUCTORBP
FCR_TM_RM_GC_CL_FDM	CODDISTRITO	NO	Lineal	FCR_TM_CL_PR_AC	CODDISTRITO
FCR_TM_RM_GC_CL_FDM	CODMONEDA	NO	Lineal	FCR_TM_CL_PR_AC	CODMONEDA
FCR_TM_RM_GC_CL_FDM	CODCLAVECIC	NO	Lineal	FCR_TM_CL_PR_AC	CODCLAVECIC
FCR_TM_RM_GC_CL_FDM	FECSALIDARANGOMORAMES	NO	Calculado	FCR_TM_CL_PR_AC	FECDIA
FCR_TM_RM_GC_CL_FDM	FECCOSECHAASIG	NO	Lineal	FCR_TM_CL_PR_AC	FECCOSECHAASIG
FCR_TM_RM_GC_CL_FDM	CODESCENARIOTRIADINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODESCENARIOCOBRANZATRIAD
FCR_TM_RM_GC_CL_FDM	CODINDICADORCOBRANZAINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODINDICADORCOBRANZA
FCR_TM_RM_GC_CL_FDM	CODESTRATEGIAINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODESTRATEGIA
FCR_TM_RM_GC_CL_FDM	CODSUCAGENUCLEOINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODSUCAGENUCLEO
FCR_TM_RM_GC_CL_FDM	CODGRUPOGESTIONPOSPREVINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODGRUPOGESTIONPOSICIONPREV
FCR_TM_RM_GC_CL_FDM	CODGRUPOGESTIONPOSINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODGRUPOGESTIONPOSICION
FCR_TM_RM_GC_CL_FDM	CODGRUPOGESTIONPOSFINMES	NO	Lineal	FCR_TM_CL_PR_AC	CODGRUPOGESTIONPOSICION
FCR_TM_RM_GC_CL_FDM	CODGRUPOGESTIONPOSCIEFINMES	NO	Lineal	FCR_TM_CL_PR_AC	CODGRUPOGESTIONPOSICIONCIERRE
FCR_TM_RM_GC_CL_FDM	CODGRUPOFUNCIONALPREVINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODGRUPOFUNCIONALPREV
FCR_TM_RM_GC_CL_FDM	CODGRUPOFUNCIONALINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODGRUPOFUNCIONAL
FCR_TM_RM_GC_CL_FDM	CODGRUPOFUNCIONALFINMES	NO	Lineal	FCR_TM_CL_PR_AC	CODGRUPOFUNCIONAL
FCR_TM_RM_GC_CL_FDM	CODGRUPOFUNCIONALCIEFINMES	NO	Lineal	FCR_TM_CL_PR_AC	CODGRUPOFUNCIONALCIERRE
FCR_TM_RM_GC_CL_FDM	CODSUBRANGOMORAPREVINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODSUBRANGOMORAPREV
FCR_TM_RM_GC_CL_FDM	CODSUBRANGOMORAINGMES	NO	Lineal	FCR_TM_CL_PR_AC	CODSUBRANGOMORA
FCR_TM_RM_GC_CL_FDM	CODSUBRANGOMORACIEFINMES	NO	Lineal	FCR_TM_CL_PR_AC	CODSUBRANGOMORACIERRE

Figura 17. Muestra del documento de trazabilidad. Fuente: Entidad financiera

3.2.5.2 Diseño de la Solución

En esta etapa el diseño de la solución tiene como punto inicial el documento de trazabilidad que nos ayudará a identificar si las fuentes que necesitamos se encuentran en la capa UDV del data lake o tenemos que migrar datos calculados desde el data mart de SDP.

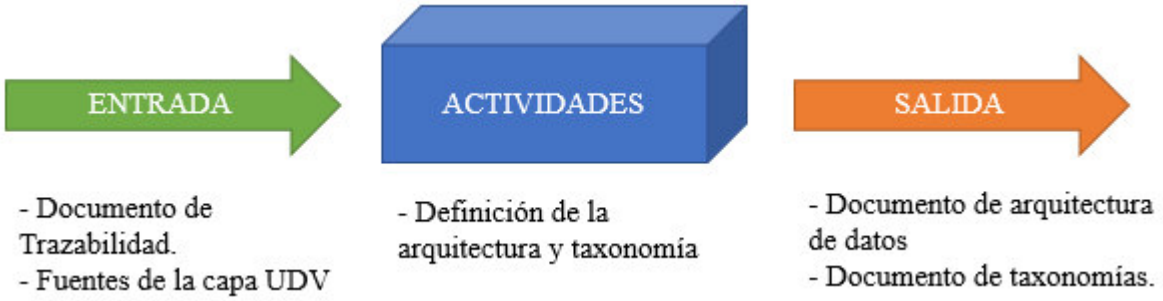


Figura 18. Flujo de entrada y salida de la etapa de Diseño de la Solución. Fuente: Elaboración propia

3.2.5.2.1 Diseño de la Arquitectura de datos

En esta actividad hemos tomado todos los campos cuya fuente viene desde el data warehouse y validamos si se encuentran migrados en la capa DDV del data lake para que pueda ser consumida por el nuevo modelo de portafolio. Luego validamos si las fuentes que se originan en el data mart de SDP se encuentran en el data lake, en este caso ningún aplicativo de la unidad de Soluciones de Pago como el Debt Manager se encuentra en el data lake, por lo tanto estas fuentes serían migradas desde el data mart hacia la capa DDV del data lake. Hay que tener en consideración que las broads de las aplicaciones de SDP no se encuentra en el data lake debido a que el negocio está en proceso de adquirir un nuevo aplicativo llamado Cyber Financial y el objetivo es que estas nuevas broads se almacenen directamente en la nueva plataforma de big data; por lo tanto, para no realizar un doble esfuerzo en migrar las broads de las antiguas aplicaciones al data lake y luego las nuevas broads, la arquitectura de la solución está diseñada de tal manera en un inicio el Modelo de Portafolio trabaje con las broads antiguas y cuando las nuevas broads de Cyber Financial se encuentre en el data lake se realicen ajustes mínimos en la solución para que el modelo trabaje con la nueva aplicación.

Luego de revisar si las fuentes requeridas para el modelo se encuentran en la capa UDV del data lake, se han encontrado que solo los saldos y los pagos están en esta capa y que serán utilizadas para crear las siguientes tablas:

- Fact de Cartera: El 68% de los campos o 45 de 75 campos que se quieren serán creadas por las fuentes del data lake. El 32% de campos restantes serán datos migrados desde el data mart de SDP.
- Fact de Pagos: El 100% de los campos de esta tabla serán creada por las fuentes del data lake.

Una vez identificado las tablas que serán creadas con fuentes del data lake, se tiene que identificar las tablas que serán migradas desde el data mart de SDP hacia el data lake para la construcción del Modelo de Portafolio. La ejecución de esta tarea se ha realizado identificando las tablas que no tienen su fuente en la capa UDV los cuales son:

- Tabla de fechas de carga de SDP.
- Tabla de fechas de cosechas.
- Tabla lookup de acción de gestiones.
- Tabla lookup de productos.
- Tabla maestras de Promesas de Pago.
- Tabla Fact de gestiones.
- Tabla de fechas de pagos.
- Tabla Fact de Cartera: Se migrará solo el 38% de los campos que se requieren ya que sus fuentes son del aplicativo Debt Manager y no se encuentran en el data lake.

Por lineamiento de arquitectura todas las tablas que son migradas desde un sandbox deben aterrizar sobre la zona “terceros” en HDFS del data lake y solo puede ser utilizada por la unidad de negocio.

Finalmente, cuando todas las tablas mencionadas anteriormente se encuentre implementadas en la capa DDV del data lake, se podrán implementar las tablas del Modelo de Portafolio que son:

- Tabla Fact de Portafolio Diario.
- Tabla Fact de Portafolio Mensual.
- Tabla Fact de Portafolio de Cosecha.

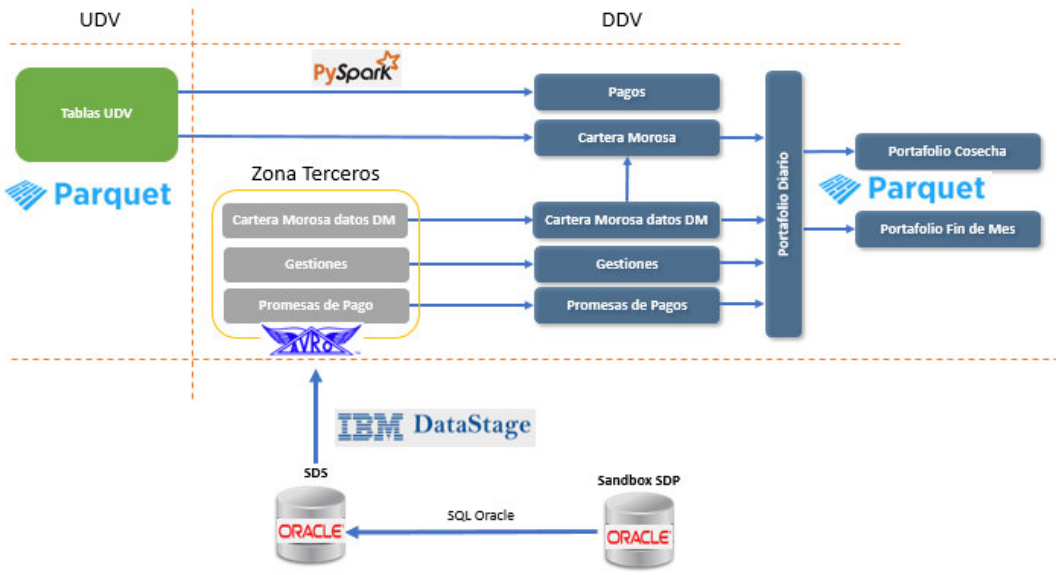


Figura 19. Arquitectura de la solución en el Data Lake. Fuente: Elaboración propia (adaptado)

3.2.5.2.2 Definir las taxonomías en la capa DDV para HDFS y Linux

Para tener los datos organizados en el data lake, la unidad de arquitectura de datos ha definido las taxonomías para HDFS el cuál se debe implementar para cada solución, estas taxonomías definen la ruta donde deben almacenarse los datos.

La taxonomía tiene el siguiente orden jerárquico:

- Ambiente: Se indica si el ambiente es producción (prod), certificación (cert) o desarrollo (desa). En este proyecto se definen la taxonomía para todos los ambientes.
- Organización: Nombre de la empresa.
- DDV: Es la capa donde se va a implementar la solución, este valor no cambia.
- Unidad: Iniciales de la unidad de negocio que está implementando la solución. Para el proyecto las iniciales son “sdp”

- Solución: Iniciales de la solución del negocio, para este proyecto se han definido dos soluciones. El primero es la solución referente al Modelo de Portafolio y se ha definido las iniciales como “cross” porque no solo almacenará el modelo de portafolio sino también otros proyecto de la unidad de negocio. La segunda solución hace referencia a los Modelos Core que son necesario para el modelo de portafolio, por lo tanto las iniciales serán “core”.

  Dentro de esta carpeta se tienen las siguientes subcarpetas:

  - data: El nombre no cambia pero dentro de esta carpeta existe la subcarpeta “out” en donde se guardan los datos no DAC.
  - datadac: El nombre de la carpeta no cambia pero dentro de esta carpeta existe la subcarpeta “out” en donde se guarda los datos DAC.
  - terceros: El nombre no cambia y dentro de esta carpeta existe la subcarpeta “in” en donde se guardan los datos que migraron desde fuera del data lake.
  - reject: El nombre no cambia y aquí se almacenan los datos rechazados por los procesos ETL
  - archive: No cambia el nombre y aquí se guardan información histórica y no activa.
  - temp: No cambia el nombre y aquí se almacena datos temporales para los procesos ETL.

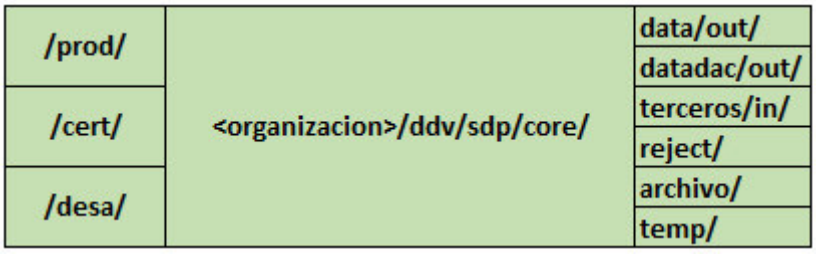


Figura 20. Taxonomía para HDFS para las soluciones core. Fuente: Elaboración propia

/prod/	<organizacion>/ddv/sdp/cross/	data/out/
		datadac/out/
/cert/		terceros/in/
		reject/
/desa/		archivo/
		temp/

Figura 21. Taxonomía para HDFS para las soluciones cross. Fuente: Elaboración propia.

La taxonomía para Linux es igual a la taxonomía de HDFS hasta la carpeta de "solución", las subcarpetas que existen dentro de esta carpeta son:

- input: Dato de entrada.
- output: Datos de salida.
- reject: Datos de rechazados.
- tmp: Datos temporales.
- hql: Scripts en Hive para hacer consultas.
- jar: Ejecutables en java
- ksh: Script en batch.
- log: archivos los de los ejecutables.
- sql: Scripts en pyspark
- xcom: Archivos utilizados por el protocolo XCOM para la transferencia de archivos.
- schema: Esquemas de los archivos avro.

/prod/	<organizacion>/ddv/sdp/core/	input/
		output/
		reject/
/cert/		tmp/
		hql/
		jar/
		ksh/
/desa/		log/
		sql/
		xcom/
		schema/

Figura 22. Taxonomía para Linux de los Modelos Core. Fuente: Elaboración Propia

3.2.5.3 Implementación de la Solución

En esta etapa, toda la fase de implementación se realiza en ambientes de desarrollo.

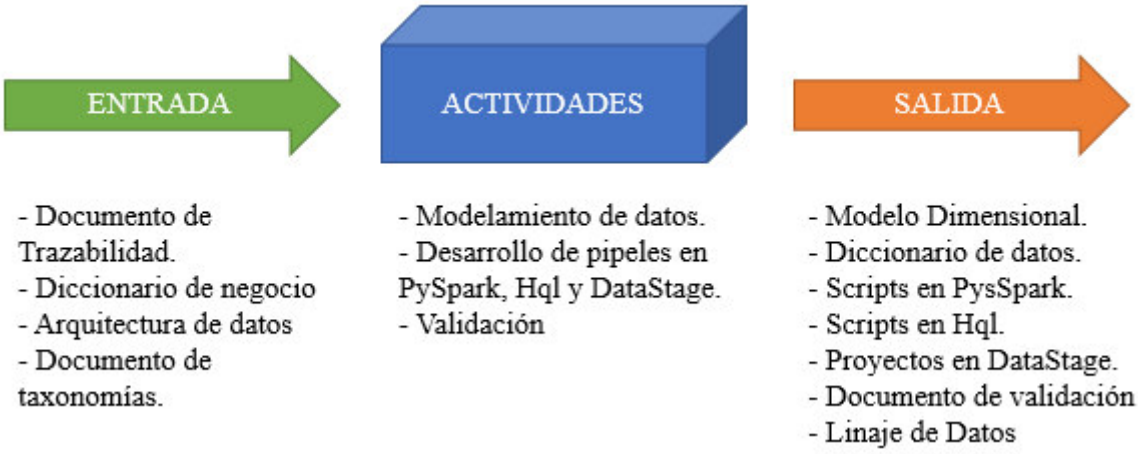


Figura 23. Flujo de entrada y salida de la etapa de Implementación. Fuente: Elaboración propia

3.2.5.3.1 Modelado Dimensional

En esta etapa de modelado dimensional se tienen dos modelos, el primero es el modelado de las tablas en la capa DDV del data lake y el segundo modelado corresponde al esquema SDS en Oracle que va a servir para migrar las tablas desde el data mart de SDP hacia el data lake. Los modelados dimensionales se realizan con la herramienta Erwin Data Modeler.

- **Modelado en la capa DDV**

El modelado en la capa DDV debe estar alineado a las nomenclaturas definidas por el equipo de modelamiento de datos. Dentro de esta nomenclatura existe el término "TTT" que hace referencia al tipo de tabla que se implementa, por ejemplo:

  - MAE: Maestra.
  - COD: Descripción de códigos.
  - TIP: Descripción de tipos.
  - FLG: Descriptivas de flags.

También existe el término “DD” que hace referencia a la dimensiones como por ejemplo:

- CL: Cliente.
- TM: Tiempo.
- PR: Producto.

TIPO DE TABLA	PREFIJO	NOMENCLATURA	EJEMPLO
Look Up	L o X	LDD_TTTNombre X_TTTNombre	LCL_MAECLIENTEFINANCIERO X_MAEAGENCIA
Fact	F	F_NOMBRE FDD_NOMBRE FDD1_NOMBRE	F_SALDOMEDIO FPR_CUENTACORRIENTEPYCF
Relacionales	REL	REL_Nombre1Nombre2	REL_CLIENTECUENTA
Fact Agregadas	F	F_NOMBRE_AGn FDD_NOMBRE_AGn	F_SALDOMEDIO_AG1
Fact Eventual	F_E	F_E_NOMBRE	F_E_PROVISION

Figura 24. Nomenclatura de tablas para la capa DDV del Data Lake. Fuente: Elaboración propia.

Nomenclatura	Descripción	Tipo de dato
CODCLAVE	Llave Subrogadas	VARCHAR(128)
TIP	Tipo	CHAR(20)
COD	Código	VARCHAR(30)
MTO	Monto	DECIMAL(21,4)
FEC	Fecha	DATE
CTD	Cantidad	INT
FLG	Flag	CHAR(1)
DES	Descripción	VARCHAR(256)
NBR	Nombre	VARCHAR(120)

Figura 25. Nomenclatura de los campos para la capa DDV del Data Lake. Fuente: Elaboración propia

Con la información del diccionario de negocio y aplicando las nomenclaturas finalmente se ha creado un modelo dimensional con 39 tablas para la capa DDV. Debido a que son activos de información de la entidad financiera solo se mostrarán un modelo resumen de las tablas finales de Portafolio con los campos principales.

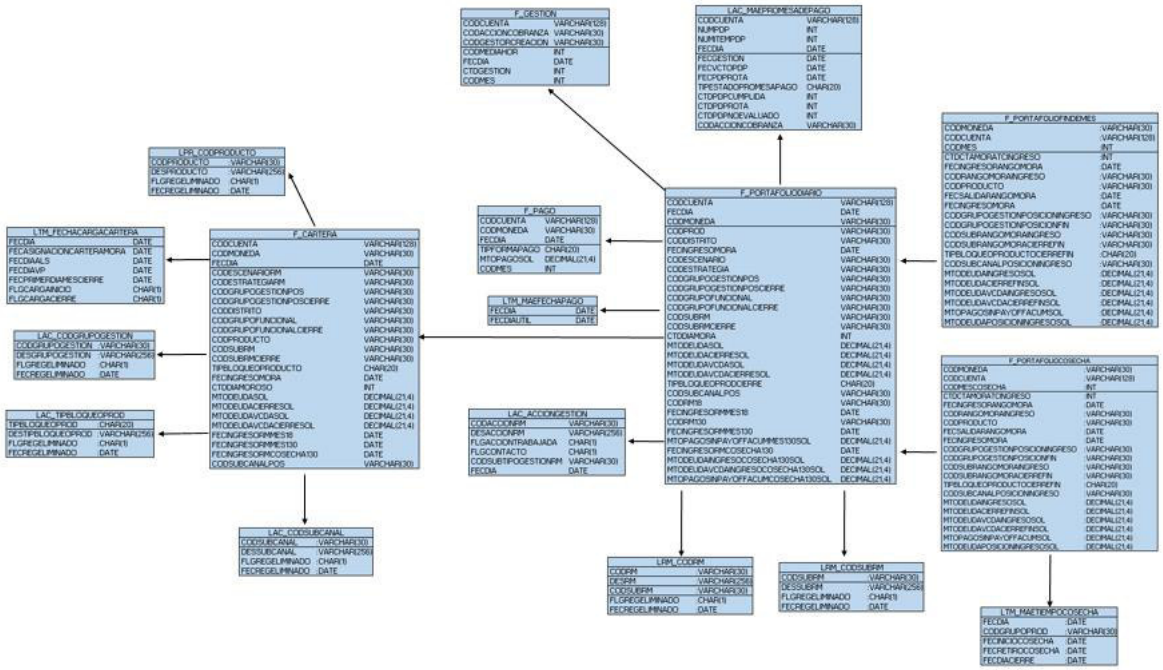


Figura 26. Modelo dimensional lógico de la solución en data lake. Fuente: Elaboración propia (adaptado).

- **Modelado en la capa SDS**

Para el modelado de SDS se han creado tablas sueltas ya que son tablas de paso y solo son útiles para migrar desde datos desde un data mart hacia el data lake y además es parte del lineamiento de arquitectura de datos.

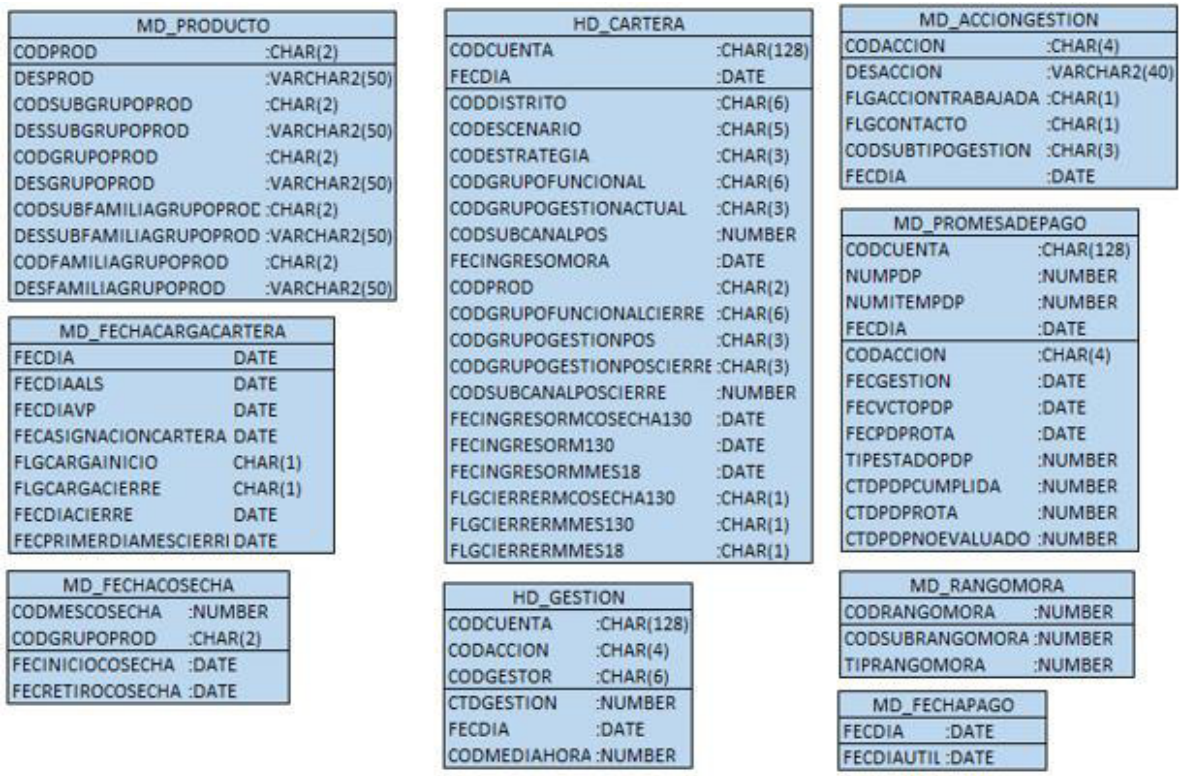


Figura 27. Modelado dimensional de las tablas en SDS: Fuente: Elaboración propia (adaptado)

3.2.5.3.2 Desarrollo de Pipelines

En esta actividad se encuentran 3 fases, la primera fase es llevar las tablas del data mart de SDP al esquema SDS de Data Warehouse, la segunda fase es llevar los datos desde el esquema SDS hacia el Data Lake y la tercera fase es desarrollar el script de Portafolio en PySpark.

- **Crear scripts para cargar los datos al esquema SDS**

Para almacenar las tablas al esquema SDS, el equipo de operaciones de Data Warehouse nos pide desarrollar dos tipos de scripts en Oracle, los DDL y los DML. En los DDL tiene que ir la creación de las tablas y en el DML tienen que ir la lógica de inserción a las tablas.

Tabla 8

Lista de scripts en Oracle para la creación de las tablas en SDS

SCRIPTS DDL
DDL_MD_PRODUCTO.sql
DDL_HD_CARTERA.sql
DDL_HD_GESTION.sql
DDL_MD_PROMESADEPAGO.sql
DDL_MD_ACCIONGESTION.sql
DDL_MD_FECHAPAGO.sql
DDL_MD_RANGOMORA.sql
DDL_MD_FECHACOSECHA.sql
DDL_MD_FECHACARGACARTERA.sql

Tabla 9

Lista de script en Oracle con la lógica de inserción

SCRIPTS DML
DML_MD_PRODUCTO.sql
DML_HD_CARTERA.sql
DML_HD_GESTION.sql
DML_MD_PROMESADEPAGO.sql
DML_MD_ACCIONGESTION.sql
DML_MD_FECHAPAGO.sql
DML_MD_RANGOMORA.sql
DML_MD_FECHACOSECHA.sql
DML_MD_FECHACARGACARTERA.sql

```
CREATE TABLE SDS.HD_GESTION
(
    CODCUENTA           CHAR(128) NOT NULL ,
    CODACCION           CHAR(4) NOT NULL ,
    CODGESTOR           CHAR(6) NOT NULL ,
    CODMEDIAHORA        NUMBER NOT NULL ,
    FECDIA              DATE NOT NULL ,
    CTDGESTION          NUMBER NULL
);

COMMENT ON TABLE SDS.HD_GESTION IS 'Contiene las gestiones realizadas a la cartera cobranza.';
COMMENT ON COLUMN SDS.HD_GESTION.CODCUENTA IS 'Clave generada para identificar a las cuentas.';
COMMENT ON COLUMN SDS.HD_GESTION.CODACCION IS 'Codigo de accion de cobranza utilizada con el cliente.';
COMMENT ON COLUMN SDS.HD_GESTION.CTDGESTION IS 'Cantidad de veces que se ha realizado tal accion para
gestionar la cuenta asignada.';
COMMENT ON COLUMN SDS.HD_GESTION.FECDIA IS 'Indica la fecha a la que pertenece la informacion.';
COMMENT ON COLUMN SDS.HD_GESTION.CODGESTOR IS 'Codigo de gestor que realiza el registro de la gestion.';
COMMENT ON COLUMN SDS.HD_GESTION.CODMEDIAHORA IS 'Codigo de media hora.';
CREATE UNIQUE INDEX SDS.HD_GESTION_U1 ON SDS.HD_GESTION
(CODCUENTA ASC,CODACCION ASC,CODGESTOR ASC,CODMEDIAHORA ASC,FECDIA ASC);
```

Figura 28. Ejemplo de un script DDL del proyecto. Fuente: Entidad financiera (adaptado).

```
-- ||*****************************************************************************
-- || PROYECTO          : INSERCIÓN DE DATOS DE GETIONES EN EL ESQUEMA SDS
-- || NOMBRE            : DML_HD_GESTION.SQL
-- || TABLA DESTINO     : SDS.HD_GESTION
-- || TABLAS FUENTES    : COBRANZAS.HD_GESTION
-- || OBJETIVO          : Pasar de EBI a OBI los reportes de Rastreo Judicial
-- || TIPO              : SQL/PLSQL
-- || REPROCESABLE      : No
-- || SCHEDULER         : -
-- ||
-- || VERSION   DESARROLLADOR           FECHA           DESCRIPCION
-- ||
-- ------------------------------------------------------------------------------
-- || 1         Canevello Salazar JC.   08/05/2020  Creación del proceso
-- ||*****************************************************************************

WHENEVER SQLERROR CONTINUE;
TRUNCATE TABLE SDS.HD_GESTION;
WHENEVER SQLERROR EXIT 1;

INSERT /*+ APPEND */ INTO SDS.HD_GESTION NOLOGGING
(
    CODCUENTA,
    CODACCION,
    CODGESTOR,
    CODMEDIAHORA,
    FECDIA,
    CTDGESTION
)
SELECT
    CODCUENTA,
    CODACCION,
    CODGESTOR,
    CODMEDIAHORA,
    FECDIA,
    CTDGESTION
FROM COBRANZAS.HD_GESTION;
```

Figura 29. Ejemplo de un script DML del proyecto. Fuente: Elaboración propia (adaptado).

- **Crear los flujos de integración para cargar los datos al Data Lake**

En esta parte se ha utilizado la herramienta IBM DataStage para llevar la data de las tablas almacenadas en el esquema SDS hacia el HDFS de la capa DDV en el Data Lake. La data se almacenará en formato avro por lo tanto hay que crear los esquemas o archivos .avsc por cada tabla.

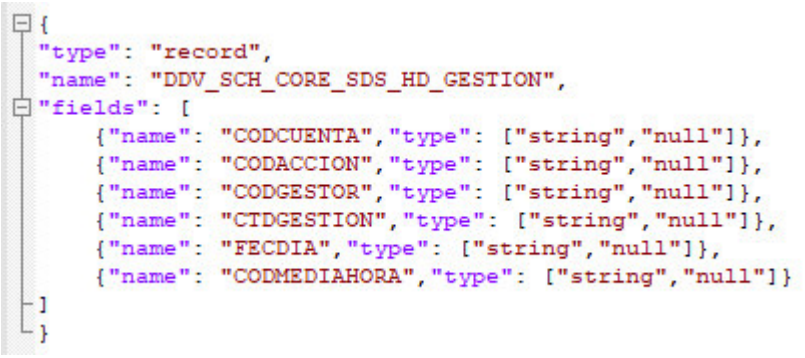


Figura 30. Ejemplo de un esquema de un archivo avro para HD_GESTION. Fuente: Elaboración propia (adaptado)

Por lo tanto se han tenido que crear 9 esquemas.

Tabla 10

Lista de esquemas para los archivos avro's en HDFS.

Archivos AVSC
SCH_DDV_SDP_MD_PRODUCTO.avsc
SCH_DDV_SDP_HD_CARTERA.avsc
SCH_DDV_SDP_HD_GESTION.avsc
SCH_DDV_SDP_MD_PROMESADEPAGO.avsc
SCH_DDV_SDP_MD_ACCIONGESTION.avsc
SCH_DDV_SDP_MD_FECHAPAGO.avsc
SCH_DDV_SDP_MD_RANGOMORA.avsc
SCH_DDV_SDP_MD_FECHACOSECHA.avsc
SCH_DDV_SDP_MD_FECHACARGACARTERA.avsc

Finalmente, en DataStage se configura un componente "external source" para que se conecte a las tablas de SDS y luego se almacene en HDFS en formato avro, para esta última parte se tiene que configurar el componente de "file conector" con la taxonomía para Linux definido en la etapa de arquitectura y los esquemas de los archivos avro's creados anteriormente.

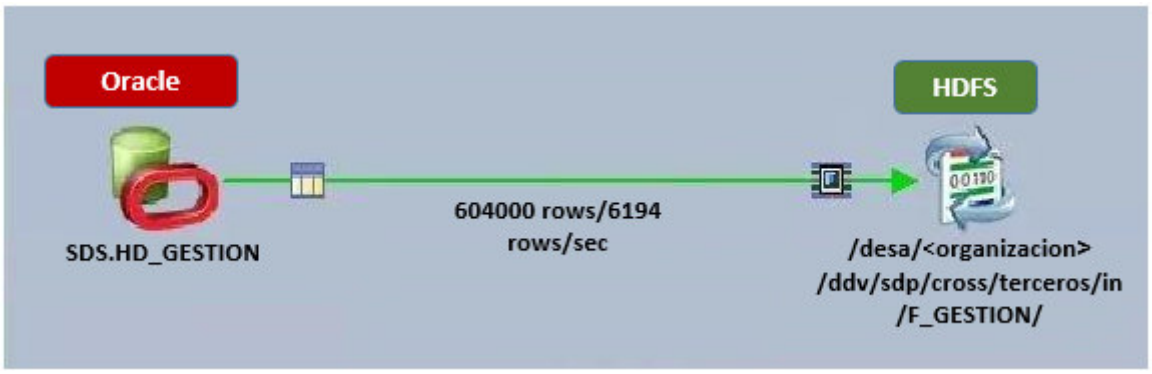


Figura 31. Flujo de trabajo para migrar los datos de gestiones desde Oracle hasta HDFS. Fuente: Elaboración propia (adaptado)

- **Desarrollar el script de Portafolio**

Los scripts se desarrollaron en Hql (Hive) para los DDL y PySpark para los procesos ETL considerando algunos lineamientos definidos por el equipo de arquitectura de datos.

En cuanto a los DDL se crearon los siguientes archivos:

Tabla 11

Lista de archivos DDL para Data Lake

ARCHIVOS DDL PARA HIVE
DDL_LRM_CODRM.hql
DDL_F_PORTAFOLIOCOSECHA.hql
DDL_LPR_CODPROD.hql
DDL_LAC_CODGRUPOGESTION.hql
DDL_LRM_CODSUBRM.hql
DDL_LAC_TIPBLOQUEOPROD.hql
DDL_LAC_CODSUBCANAL.hql
DDL_LTM_MAETIEMPOCOSECHA.hql
DDL_F_PORTAFOLIOFINDEMES.hql
DDL_F_PORTAFOLIODIARIO.hql
DDL_F_CARTERA.hql
DDL_LTM_FECHACARGACARTERA.hql
DDL_F_GESTION.hql
DDL_LAC_ACCIONGESTION.hql
DDL_LAC_MAEPROMESADEPAGO.hql
DDL_LTM_MAEFECHAPAGO.hql

```
SET PRM_HIVE_SCH_DDV=${HIVE_AMBIENTE_LC_SUBSIDIARIA_LC};
SET PRM_HIVE_AMBIENTE=${PRM_AMBIENTE};

DROP TABLE IF EXISTS ${hiveconf:PRM_HIVE_SCH_DDV}_ddv_sdp.F_GESTION ;
CREATE TABLE ${hiveconf:PRM_HIVE_SCH_DDV}_ddv_sdp.F_GESTION
(
    CODCUENTA   VARCHAR(128)
    CODACCIONRM VARCHAR(30)
    CODGESTORCREACION   VARCHAR(30)
    CODMEDIAHOR INT
    FECDIA  DATE
    CTDGESTION  INT
    CODMES  INT
)
STORED AS PARQUET
LOCATION '/${hiveconf:PRM_HIVE_AMBIENTE}/<organizacion>/ddv/sdp/core/data/out/F_GESTION'
TBLPROPERTIES("parquet.compress"="SNAPPY");
```

Figura 32. Ejemplo de un DDL para Hive. Fuente: Elaboración propia

Por otro lado, en los procesos con PySpark se tienen de dos tipos, procesos que migran lo datos del formato avro al formato parquet y los otros procesos son los que aplican reglas de negocio.

Tabla 12

Lista de Scripts en PySpark.

Tipo de proceso	Script
Aplican reglas de negocio	F_PAGO.py
	F_PORTAFOLIOCOSECHA.py
	F_PORTAFOLIOFINDEMES.py
	F_PORTAFOLIODIARIO.py
	F_CARTERA.py
Migrar de Avro a Parquet	LTM_FECHACARGACARTERA.py
	F_GESTION.py
	LAC_ACCIONGESTION.py
	LAC_MAEPROMESADEPAGO.py
	LTM_MAEFECHAPAGO.py
	LRM_CODRM.py
	LPR_CODPROD.py
	LAC_CODGRUPOGESTION.py
	LRM_CODSUBRM.py
	LAC_TIPBLOQUEOPROD.py
	LAC_CODSUBCANAL.py
	LTM_MAETIEMPOCOSECHA.py

3.2.5.3.3 Validación de resultados

Para la validación de resultados, se han comparado algunas variables estadísticas de las tablas finales del Modelo de Portafolio de Oracle vs Data Lake, en los resultados se está considerando una diferencia del 2% en la comparación debido a que se han aplicado algunas reglas de corrección en el data lake que no se encuentran en Oracle. Algunas de las variables a validar son volumen de cuentas, Suma total de saldos, Monto total de pagos y cantidad de gestiones.

Plataforma	Volumen de cuentas	Total de Deuda Vencida	Total de Pagos	Total de Gestiones
Oracle	327 606	S/ 1,342,633,853.00	S/ 712,188,708.00	1 243 547
Data Lake	327 900	S/ 1,342,934,853.00	S/ 712,788,708.00	1 243 647
Diferencia(%)	0.09%	0.02%	0.08%	0.01%

Figura 33. Ejemplo de cuadro de validación de las tablas finales implementadas en las distintas plataformas. Fuente: Elaboración propia.

3.2.5.3.4 Implementación de Linaje de Datos

El linaje de datos se realiza con IBM InfoSphere Information Governance Catalog y se realiza por cada tabla implementada. Para el proyecto se han implementado 17 linajes que corresponden a las tablas en el data lake.

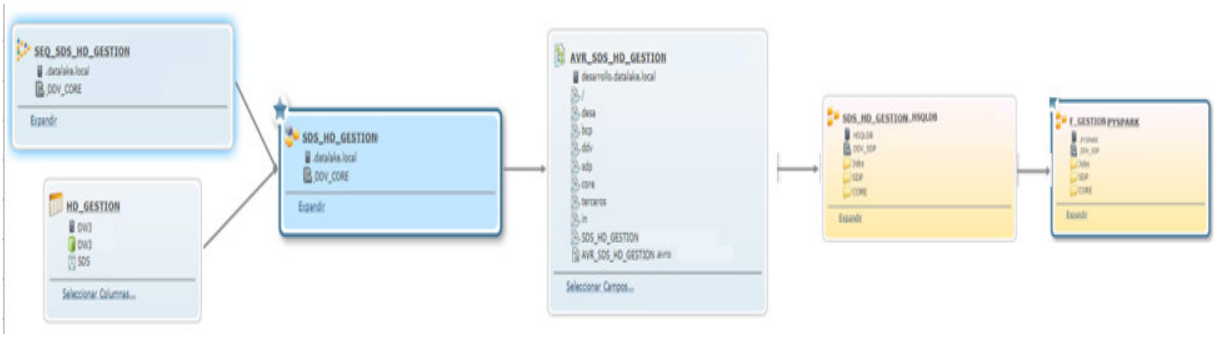


Figura 34. Ejemplo de linaje de datos para la tabla F_Gestion.

3.2.5.4 Despliegue

Para pasar a producción todos los desarrollos que se hicieron, debemos seguir un flujo de integración continua que utiliza las herramientas de Bitbucket, Jira y Jenkins. En Bitbucket se versionan todos los cambios que se van haciendo al proyecto, de esta manera se tiene un

historial de cambios que permite hacer rollback a los scripts, en Jira se registra la solicitud para empezar con el pase a producción y permite que la comunicación sea automatizada desde que empieza la etapa hasta que termina, y en Jenkins se tienen que crear Jobs que permitan automatizar la descarga de los archivos desde BitBucket y copiarlos en los ambientes de certificación o producción y también realizar la ejecución para validar que el pipeline esté correctamente implementado.

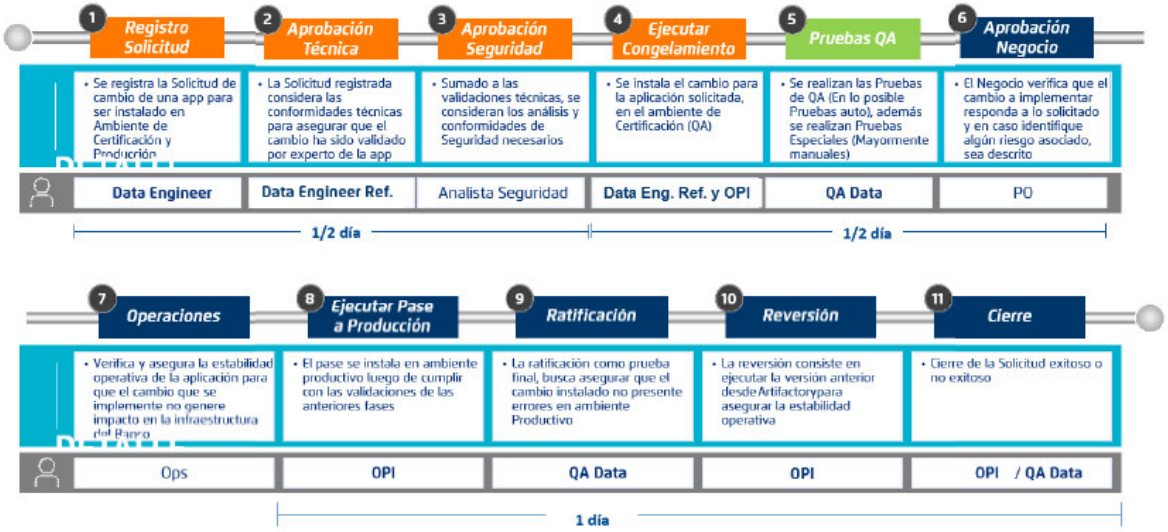


Figura 35. Flujo de pase a producción. Fuente: Entidad financiera

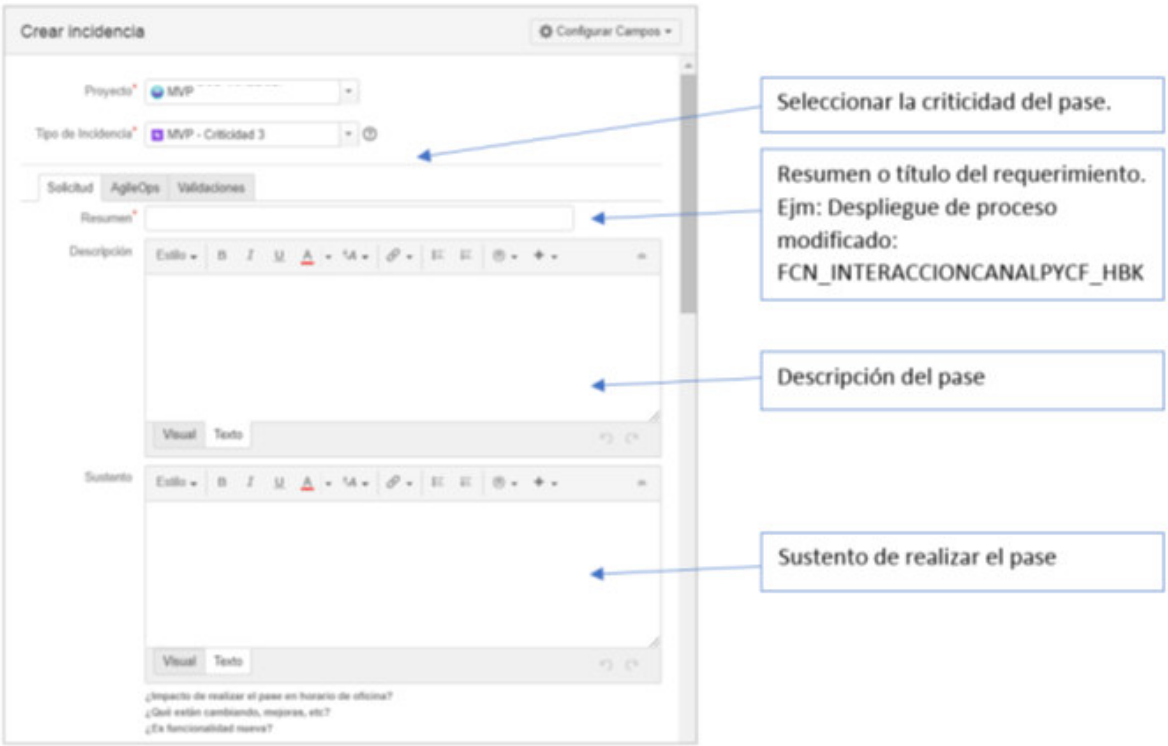


Figura 36. Secciones de la solicitud para pase a producción. Parte 1. Fuente: Entidad financiera.

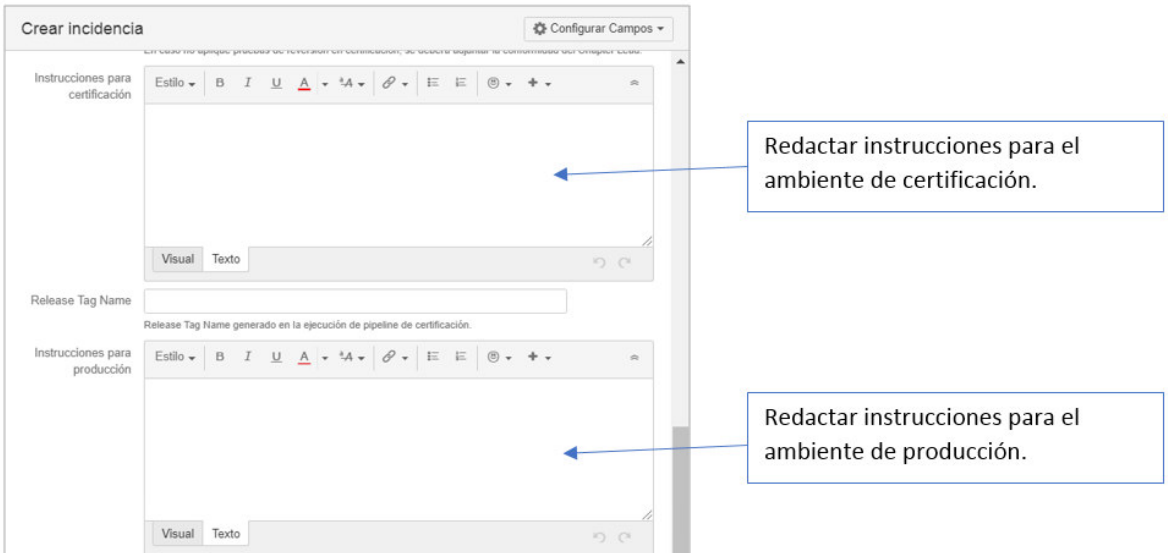


Figura 37. Secciones de la solicitud para pase a producción. Parte 2. Fuente: Entidad financiera.

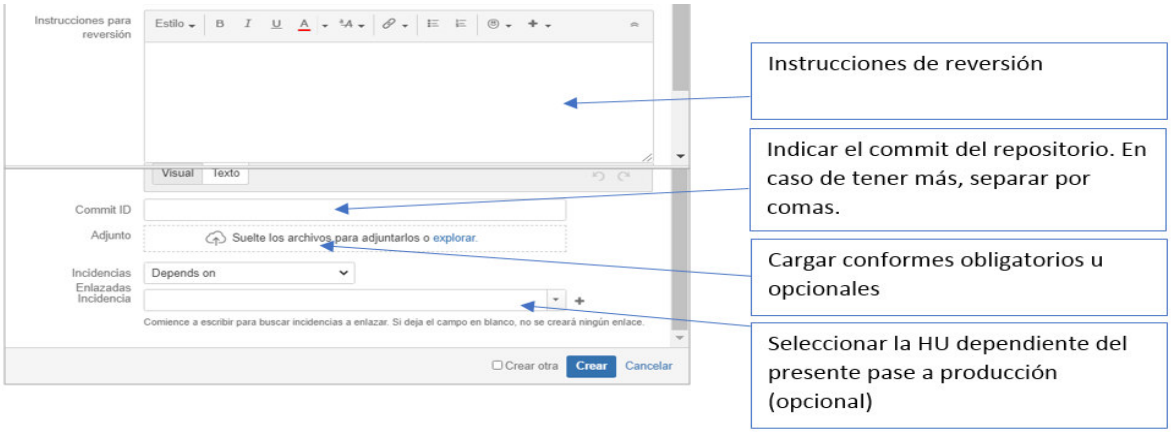


Figura 38. Secciones de la solicitud para pase a producción. Parte 3. Fuente: Entidad financiera.

3.3 EVALUACIÓN ECONÓMICA

3.3.1 EVALUACIÓN DE COSTO

La evaluación económica se centra en la inversión del personal que ha integrado el proyecto en las distintas fases, se necesitó un total de 7 personas con distintos perfiles especializados en cada etapa del proyecto como arquitectos, analistas de negocio seguridad de información, modeladores, gobierno de datos y operaciones para el pase a producción y

operatividad del proyecto. Cada perfil acompañaban al Data Engineer en cada una de las fases. A medida que el proyecto iba avanzando, los distintos perfiles iban dejando el proyecto. El proyecto tuvo un costo de S/ 31 950 soles de los cuales gran parte de la inversión está sobre el Ingeniero de Datos ya que es un rol cross y necesario en todas las fases.

Etapas	Fases	Data Governance	Data Analysis	Data Engineer	Data Security	Data Arquitect	Data Model	DataOps	
Análisis del requerimiento	Definir las variables finales.	S/ 1,500	S/ 1,700	S/ 500	S/ 700				
	Identificación de fuentes		S/ 850	S/ 3,000					
Diseño de la Solución	Diseño de la Arquitectura de datos			S/ 2,700	S/ 700	S/ 1,650			
	Definir las taxonomías en la capa DDV para HDFS y Linux			S/ 1,350		S/ 1,000			
Implementación de la Solución	Modelado Dimensional			S/ 1,350			S/ 1,500		
	Desarrollar pipelines			S/ 8,100					
	Validación de resultados		S/ 850	S/ 550					
	Implementar el Linaje de datos	S/ 500		S/ 1,600					
Despliegue	Certificación y Pase a Producción			S/ 600	S/ 250			S/ 1,000	
Total		S/ 2,000	S/ 3,400	S/ 19,750	S/ 1,650	S/ 2,650	S/ 1,500	S/ 1,000	**S/ 31,950**

Figura 39. Tabla de costos de RR.HH de implementación. Fuente: Elaboración propia.

3.3.2 BENEFICIO PARA LA ORGANIZACIÓN

Implementar el proyecto en la nueva plataforma de Big Data de la entidad bancaria ayuda al plan de transformación digital del banco a tener integrado todos los datos en un solo repositorio, esto con el fin de disponer de ellos en cualquier momento y teniendo la mayor cantidad de datos. Además la implementación de este proyecto ayuda a la unidad de negocio de Soluciones de Pago a empezar a implementar sus soluciones en un entorno Big Data y beneficiarse de las ventajas competitivas de esta.

Como parte de la implementación del Modelo de Portafolio, se tuvo que implementar los Modelos Core de la unidad de negocio para lograr el objetivo del proyecto, esto trae como beneficio que otros Modelos del negocio puedan migrarse al data lake en un menor tiempo ya que se tendría la mayoría de las fuentes disponibles.

Parte de las actividades del Modelo de Portafolio en Oracle y al disponer de un servidor auto-gestionado en donde se ejecutaba el proceso, era el monitoreo del proceso para la continuidad operativa lo que demandaba tiempo y esfuerzo en esta actividad. Con la migración del modelo al Data Lake, el monitoreo y continuidad operativa del proceso pasa a manos del

Equipo de Operaciones de Data Lake, con lo que el negocio gana tiempo y esfuerzo para otras actividades de mayor prioridad.

Con la implementación del Modelo de Portafolio en el Data Lake, la unidad de negocio de Soluciones de Pago tiene las puertas abiertas para utilizar las distintas herramientas analíticas con las que cuenta la plataforma, desde tener un espacio de exploración en el Data Lake llamado EDV, disponer de herramientas de integración de bases de datos como el Datameer hasta tener accesos a la herramientas de visualización oficial del banco que es el Qlik Sense. El acceso a estas herramientas posibilita que los analistas de negocio creen sus propios tableros de control y seguimiento y puedan ser automatizados sin la necesidad de intervención de un Data Engineer.

Finalmente la implementación de este proyecto motiva a otras unidades de negocio a acelerar la migración de sus procesos y beneficiarse de las herramientas de la plataforma de Big Data.

CAPÍTULO IV

REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

La implementación del proyecto en un entorno Big Data evidencia los beneficios y las ventajas competitivas frente a un entorno tradicional como Oracle, Sql Server, etc. Una de las ventajas es la posibilidad de procesamiento de mayor volumen de datos y menor tiempo de ejecución gracias a un sistema distribuido como Hadoop. La plataforma de la entidad financiera se encuentra implementada en sus propios servidores, esto conlleva a una carga operativa y un esfuerzo especializado en big data para mantener en funcionamiento el entorno, y añadido al contexto que la entidad financiera está incursionando en este mundo, durante la implementación del proyecto se presentaron escenarios en donde algunas herramientas de la plataforma se encontraban caídas por ciertas hora lo que ocasionaba retrasos en los entregables.

Durante la implementación del proyecto, en las fases de diseño de arquitectura nos encontramos con algunas necesidades que no se podían satisfacer debido a que no había unos lineamientos definidos, esto nos llevó a reunirnos con equipos de arquitectura de datos y seguridad de información para plantear una solución a nuestra necesidad; sin embargo, al no existir lineamiento alguno estábamos obligados a presentarnos a un comité en donde tenían que aprobar nuestro planteamiento. Pasar por este proceso engorroso significada ampliar el tiempo del proyecto por lo que optamos por una solución táctica y esto se traduce en una deuda técnica para el proyecto el cual debe ser subsanado una vez que los lineamientos se encuentren oficializados

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Se ha logrado identificar más cuentas morosas gracias a la migración del modelo de portafolio de la unidad de negocio de soluciones de pago a un entorno de big data el cual ha ayudado en una gestión más efectiva de la cartera morosa.
2. Se ha habilitado en el data lake los datos de la broads delo aplicativo tradicional como Debt Manager.
3. Se implementó los modelos core del negocio como las tablas de Fact de pagos, gestiones, promesas de pago (pdp), saldos, maestra de cuentas y lookups de productos, rangos de mora, fechas de pago, fechas de cosecha y tipo de acción de cobranza en el Data Lake y utilizando PySpark.
4. Se implementó el Modelo de Portafolio mensual y de cosecha en el Data Lake utilizando PySpark para que el negocio pueda hacer uso de las distintas herramientas analíticas que están en la plataforma de Big Data.

RECOMENDACIONES.

1. Antes de iniciar cualquier proyecto de Big Data en el Data Lake, revisar los lineamientos y documentos de arquitectura y de seguridad de información para asegurarse que cubra todas las necesidades del proyecto.
2. Hacer el despliegue con herramientas de integración continua para que el pase a producción sea realizado en el menor tiempo posible.
3. Migrar el Data Lake a la nube para reducir riesgos operativos por caídas de la plataforma.

BIBLIOGRAFÍA

Apache Avro. (2021). *Documentación de Apache Avro 1.11.0*. Obtenido de Documentación de Apache Avro ™ 1.11.0: https://avro.apache.org/docs/current/

Apache Hadoop. (2021). *Guía de arquitectura HDFS*. Obtenido de Guía de arquitectura HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

Apache Hive. (2021). *Apace Hive*. Obtenido de https://hive.apache.org/

Apache Parquet. (2021). *Apache Parquet*. Obtenido de https://parquet.apache.org/

Apache Spark. (2021). *Apache Spark*. Obtenido de http://spark.apache.org/docs/latest/api/python/

BBVA Api Market. (26 de Feb de 2020). *Las siete 'V' del Big Data*. Obtenido de Las siete 'V' del Big Data: https://www.bbvaapimarket.com/es/mundo-api/las-siete-v-del-big-data/

BCP Bolivia. (2009). *Memoria 2009*. Obtenido de Memoria 2009: https://www.bcp.com.bo/Content/descargas/MemoriaAnual/memoria_2009.pdf

Calderón, M. (2014). *Implementación de una solución de Business Intelligence en una unidad de negocios de cobranza de una entidad financiera.* Lima: Universidad Nacional de Ingeniería.

Camargo-Vega, J. J., Camargo-Ortega, J. F., & Joyanes Aguilar, L. (2014). Conociendo Big Data. *Facultad de Ingeniería*, 15. Obtenido de http://www.scielo.org.co/pdf/rfing/v24n38/v24n38a06.pdf

Cloudera. (2021). *Hue*. Obtenido de https://gethue.com/

Córdova, E. E. (2005). *Crédito y Cobranzas.* México: Universidad Autónoma de México.

Databricks. (2021). *Glosario PySpark*. Obtenido de Glosario PySpark: https://databricks.com/glossary/pyspark

Gartner. (21 de Nov de 2021). *Glosario de Tecnologías de Información*. Obtenido de Glosario de Tecnologías de Información: https://www.gartner.com/en/information-technology/glossary/big-data

IBM. (2021). *Apache Hive*. Obtenido de Apache Hive: https://www.ibm.com/analytics/hadoop/hive

IBM. (2021). *DataStage*. Obtenido de DataStage: https://www.ibm.com/products/datastage

Marz, N., & Warren, J. (2015). *Big Data.*

Oracle. (2021). *¿Qué es Big Data?* Obtenido de ¿Qué es Big Data?: https://www.oracle.com/es/big-data/what-is-big-data/

Perez, F. S. (2015). Big Data.

PowerData. (s.f.). *DataStage*. Obtenido de https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-datastage-y-como-puede-ayudar-a-tu-empresa

Quiroz Martinez, M., Aguilar Duarte, R. A., & Intriago Cedeño, D. (18 de Jul de 2019). Proceso de diseño de una arquitectura Big Data para el análisis de grandes volúmenes de. *Opuntia Brava, 12*(1), 248.

SANTANA, A. (2016). *Análisis de la cartera vencida de la compañia Delta Plastic.* Guayaquil.

Torres, J., Aguilar, R., Martin, C., & Diaz, S. (2017). *Framework para el modelado de un Lago de Datos.* España: Universidad de Oviedo.

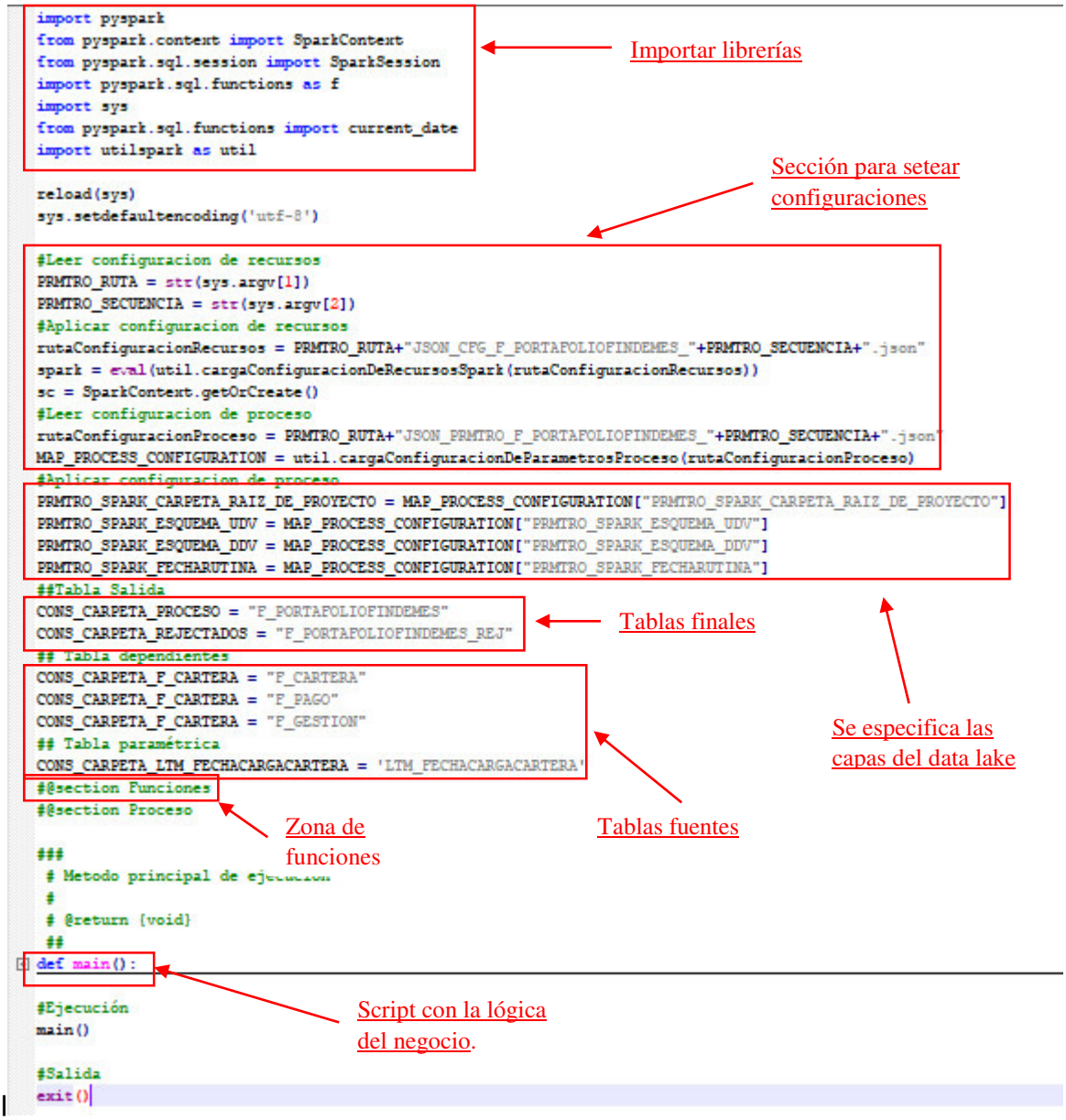
Towars Data Science. (2018). *Una breve introducción a PySpark*. Obtenido de Una breve introducción a PySpark: https://towardsdatascience.com/a-brief-introduction-to-pyspark-ff4284701873

Vohra, D. (2016). *Practical Hadoop Ecosystem.* Apress.

www.viabcp.com. (5 de Nov de 2020). *Memoria Integrada 2020 - BCP.* Obtenido de Memoria Integrada 2020 - BCP: https://www.viabcp.com/wcm/connect/c45ed29f-031d-4748-87ea-1027aafc1016/Memoria+Integrada+BCP+2020.pdf?MOD=AJPERES&CVID=nyIHDGj&attachment=false&id=1617804852779

ANEXO

Anexo 1: Estructura de un archivo PySpark



Anexo 2: Nomenclatura para el modelamiento de tablas y campos en SDS.

Nomenclaturas para tablas

Tipo de Tabla	Prefijo	Nomenclatura
Maestro Mensual	MM	MM_NombreDeTabla
Maestro Semanal	MS	MS_NombreDeTabla
Maestro Diaria	MD	MD_NombreDeTabla
Historico Mensual	HM	HM_NombreDeTabla
Historico Semanal	HS	HS_NombreDeTabla
Historico Diario	HD	HD_NombreDeTabla
Ultimo Mes	UM	UM_NombreDeTabla
Ultima Semana	US	US_NombreDeTabla
Ultimo Dia	UD	UD_NombreDeTabla
Descriptivo Mensual	MM_Des	MM_DesNombreDeTabla
Descriptivo Semanal	MS_Des	MS_DesNombreDeTabla
Descriptivo Diario	MD_Des	MD_DesNombreDeTabla
DataEntry	DE	DE_NombreDeTabla
Temporal Mensual	T	T_NombreDeTabla
Temporal Semanal	TS	TS_NombreDeTabla
Temporal Diaria	TD	TD_NombreDeTabla
Temporal Proceso	TP	TP_NombreDeTabla
Temporal General	TG	TG_NombreDeTabla

Nomenclatura para campos

Tipo de Dato	Descripción
COD	Codigo
TIP	Tipo
FLG	Flag
DES	Descriptivo
MTO	Monto
CTD	Cantidad
NUM	Numero
FEC	Fecha
NBR	Nombre
APE	Apellido

Tipo de datos para los campos

Tipo de Dato	Tipo de Campo
Monto	NUMBER(16,2)
Codigos	CHAR(n)
Fechas	DATE
Descriptivos	VARCHAR2(n)