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

Desarrollo de una solución de Big Data en una entidad bancaria para refactorizar sus procesos de migración y toma de decisiones

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Luis Alfredo HEREDIA GUERREROS

ASESOR

Norberto Ulises ROMÁN CONCHA

Lima, Perú

2021

Referencia bibliográfica

Heredia, L. (2021). *Desarrollo de una solución de Big Data en una entidad bancaria para refactorizar sus procesos de migración y toma de decisiones*. [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	LUIS ALFREDO HEREDIA GUERREROS
Tipo de documento de identidad	DNI
Número de documento de identidad	44181872
URL de ORCID	https://orcid.org/0000-0001-9213-5102
Datos de asesor	
Nombres y apellidos	NORBERTO ULISES ROMAN 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	8862360
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	Financiamiento Propio

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

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

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 20:00 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 Jesús (Miembro) y el Lic. Román Concha Norberto Ulises (Miembro Asesor), usando la plataforma Meet (https://meet.google.com/gfv-qdyi-szt), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: ***"DESARROLLO DE UNA SOLUCIÓN DE BIG DATA EN UNA ENTIDAD BANCARIA PARA REFACTORIZAR SUS PROCESOS DE MIGRACIÓN Y TOMA DE DECISIONES",*** *por el Bachiller* ***Heredia Guerreros Luis Alfredo;*** *para obtener el Título Profesional de Ingeniero de Sistemas.*

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

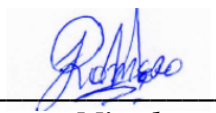
El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

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

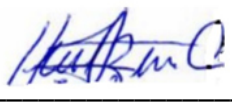
A continuación la Presidente de Jurados la Dra. Pró Concepción Luzmila Elisa, declara al Bachiller ***Ingeniero de Sistemas****.*

Siendo las 20:48 horas, se levantó la sesión.

_____
Presidente
Dra. Pró Concepción Luzmila Elisa



_____
Miembro
Lic. Romero Naupari Pablo Jesús



_____
Miembro Asesor
Lic. Román Concha Norberto Ulises

DEDICATORIA

A mis padres por ser mi guía y siempre brindarme el soporte para lograr mis objetivos y nuevos retos en la vida

AGRADECIMIENTO

A mis compañeros de trabajo por las experiencias y las grandes enseñanzas brindadas que han servido para poder realizar este informe profesional

Finalmente quiero agradecer al profesor Ulises Román quien me ha brindado su experiencia y conocimientos para la realización del presente informe.

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

DESARROLLO DE UNA SOLUCIÓN DE BIG DATA EN UNA ENTIDAD BANCARIA PARA REFACTORIZAR SUS PROCESOS DE MIGRACIÓN Y TOMA DE DECISIONES

Autor: Heredia Guerreros, Luis Alfredo
Asesor: Román Concha, Ulises

Título: Trabajo de Suficiencia Profesional
Fecha: Diciembre. 2021

RESUMEN

El Trabajo de Suficiencia Profesional (TSP), describe el desarrollo de una solución de Big Data en una entidad bancaria para refactorizar sus procesos de migración y toma de decisiones. Presentando una propuesta de solución que permite resolver los problemas actuales que presenta la organización como son las lentitudes de procesamiento, inconsistencia de los datos y la necesidad de contar con datos que aporten valor al negocio.

La metodología utilizada para este proyecto fue SCRUM, la cual ha permitido a la organización agilizar las actividades y adoptar progresivamente nuevas tecnologías en el entorno Data Lake utilizando buenas prácticas de Big Data, lo cual ha permitido mejorar los procesos actuales brindándole: Procesamiento distribuido, Almacenamiento distribuido, Alta disponibilidad, Encriptación de datos y Gobierno de datos.

Finalmente, se tiene como resultados esperados la creación de un modelo de solución para refactorizar los procesos de migración bajo el entorno Big Data, brindándole solución a la aplicación de un modelo de datos que permita al área de riesgos identificar que clientes podrían acceder a un crédito.

Palabras claves: Big Data, Entidad bancaria, Procesos de migración, Toma de decisiones, Data Lake, Refactorizar, Hadoop.

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

DEVELOPMENT OF A BIG DATA SOLUTION IN A BANKING ENTITY TO REFACTORIZE ITS MIGRATION AND DECISION-MAKING PROCESSES

Author: Heredia Guerreros, Luis Alfredo
Adviser: Román Concha, Ulises

Title: Professional sufficiency work
Date: December, 2021

ABSTRACT

The Professional Sufficiency Work (TSP), describes the development of a Big Data solution in a bank to refactor its migration and decision-making processes. Presenting a solution proposal that allows solving the current problems that the organization presents such as processing slowness, data inconsistency and the need to have data that add value to the business.

The methodology used for this project was SCRUM, which has allowed the organization to streamline activities and progressively adopt new technologies in the Data Lake environment using good Big Data practices, which has allowed to improve current processes by providing: Distributed processing, Storage Distributed, High Availability, Data Encryption and Data Governance.

Finally, the expected results are the creation of a solution model to refactor the migration processes under the Big Data environment, providing a solution to the application of a data model that allows the risk area to identify that customers could access a credit.

Keywords: Big Data, Banking entity, Migration processes, Decision making, Data Lake, Refactorize, Hadoop.

ÍNDICE

ÍNDICE FIGURAS

ÍNDICE TABLAS

INTRODUCCIÓN

Muchas organizaciones actualmente utilizan las TI para el análisis, visualización de grandes volúmenes de información y su posterior toma de decisiones. El trabajo que presentamos se basa en fundamentar la situación actual del área de negocios de una empresa del rubro financiero la cual actualmente maneja procesos utilizando tecnologías tradicionales las cuales en el transcurso del tiempo se han visto impactadas en tiempos de ejecución, redundancia de información y poca capacidad de almacenamiento y procesamiento de información lo que les ha conllevado a tener retrasos en los entregables, ,información incorrecta y por consiguiente un mal análisis de sus datos.

El objetivo que busca el autor de este informe es proponer una migración de los procesos actuales utilizando el framework Big Data, el cual le servirá de apoyo al área de riesgos brindándole una solución a los problemas que presenta actualmente y resolviendo las falencias actuales proponiendo una refactorización de sus procesos, velocidad de procesamiento, alta capacidad de almacenamiento, disponibilidad de la información, calidad en los datos y alta escalabilidad en sus soluciones tecnológicas, utilizando plataforma Cloudera y el motor de procesamiento Spark. Lo que permitirá administrar de manera adecuada sus procesos y brindar calidad de datos aportando valor a la organización.

El trabajo de suficiencia profesional expuesto se divide en cinco capítulos los cuales contienen los siguientes puntos:

En el Capítulo I se expone mi formación académica y experiencia profesional y las tecnologías adquiridas durante los últimos tres años.

En el Capítulo II se presenta el área de negocio en estudio donde plasmo mi experiencia laboral y se detalla las actividades que realiza un Data Engineer.

En el Capítulo III se define el problema y se da una propuesta de solución, poniendo en claro los objetivos, alcance, metodología, fundamentos teóricos y la implementación de la solución. Finalmente se plantea la evaluación técnica financiera del proyecto.

En el Capítulo IV se realiza una crítica sobre la solución planteada, así como las situaciones e inconvenientes que se presentaron.

En el Capítulo V se exponen las conclusiones y sugerencias, la bibliografía y un glosario de términos.

CAPÍTULO I

TRAYECTORIA PROFESIONAL

El autor del trabajo es un Bachiller en ingeniería de Sistemas e informática con una experiencia de aproximadamente 4 años desempeñándose con responsabilidad y compromiso en el rubro de consultoría, lo cual le ha permitido adquirir una amplia experiencia ofreciendo servicios a clientes del rubro Bancario, Comercial y Telecomunicaciones. La trayectoria profesional del autor del presente informe se detalla en el siguiente cuadro:

Tabla 1

Experiencia profesional

<table>
<tr><th colspan="5">EXPERIENCIA PROFESIONAL</th></tr>
<tr><th>Entidad</th><th>Área</th><th>Cargo</th><th>Fecha</th><th>Tiempo</th></tr>
<tr><td rowspan="2">INDRA</td><td>Tecnologías avanzadas</td><td>Data Engineer</td><td>Abril 2021 - Actualidad</td><td>8 meses</td></tr>
<tr><td colspan="4">Funciones:
Análisis y diseño de la solución
Procesamiento de grandes volúmenes de información
Participar y proponer soluciones en el proyecto de migración Data Lake
Refactorizar procesos
Cumplir con los lineamientos y buenas prácticas de la organización
Proponer soluciones
Disponibilizar la información
Despliegue de procesos</td></tr>
<tr><td>EVERIS</td><td>Solutions</td><td>Data Engineer</td><td>Noviembre 2019 - Marzo 2021</td><td>16 meses</td></tr>
</table>

<table>
<tr><td></td><td colspan="4">Funciones:
Análisis y diseño de la solución
Procesamiento de grandes volúmenes de información
Gestión del Data Lake
Automatización de procesos
Despliegue de procesos</td></tr>
<tr><td rowspan="2">GLOBAL HITSS</td><td>BI</td><td>Analista Datawarehouse</td><td>Mayo 2019 - Julio 2019</td><td>3 meses</td></tr>
<tr><td colspan="4">Funciones:
Análisis y diseño de la solución
Toma de requerimientos de los clientes
Automatización de procesos
Generación de reportes</td></tr>
<tr><td rowspan="2">BLUE OCEAN</td><td>BI</td><td>Analista Inteligencia de negocios</td><td>Julio 2018 - Mayo 2019</td><td>10 meses</td></tr>
<tr><td colspan="4">Funciones:
Servicios tercerizados a demanda
Toma de requerimientos
Creación de esquemas para tablas HIVE y Script Shells
Comandos HDFS y LINUX
Creación de cuadros y reporteria
Creación de Scripts en Qlikview</td></tr>
<tr><td rowspan="2">EVERIS</td><td>Solutions</td><td>Analista de Inteligencia de negocios</td><td>Enero 2017 - Julio 2018</td><td>18 meses</td></tr>
<tr><td colspan="4">Funciones:</td></tr>
</table>

	• Análisis e implementación de soluciones de Business Intelligence • Desarrollo de aplicaciones en SQL Server • Implementación de soluciones en Qlikview y Qliksense • Training en Qlikview, Qliksense.

Nota. Elaboración propia

Tabla 2

Formación Académica Profesional

FORMACIÓN ACADÉMICA PROFESIONAL			
Formación Recibida	Institución que Acredita	Documento de acreditación	Fecha
Ingeniería de Sistemas - Facultad de Ingeniería de Sistemas e Informática	UNMSM	Diploma de Grado Académico de Bachiller	2012 - 2017

Nota. Elaboración propia

Tabla 3

Cursos y Eventos académicos

CURSOS Y EVENTOS ACADÉMICOS		
Curso	Institución	Año
Spark Professional	Big Data Academy Perú	2020
Big Data on AWS Specialist	EDUTRONIC	2020
Google Cloud Platform Big Data and Machine Learning Fundamentals	COURSERA	2019
Programa de Especialización en Big Data	Big Data Academy	2018

	Perú	

Nota. Elaboración propia

Tabla 4

Otras Capacidades

OTRAS CAPACIDADES	
Lenguajes de Programación	Python, Spark
Motores de Bases de Datos	SQL Server, Oracle
Gestores de Bases de Datos	SQL Server tools, Toad for Oracle, SQL Developer
Metodologías	Scrum, Kanban
Herramientas de Modelado	Erwin, Rational Rose
Herramientas de Desarrollo	IntelliJ IDEA, Pycharm
Herramientas BI	Integration Services, Qlikview , Qliksense
Sistemas Operativos	Windows, Linux

Nota. Elaboración propia

CAPÍTULO II
CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA

2.1. EMPRESA - ACTIVIDAD QUE REALIZA

De acuerdo con INDRA (2021) , nos informa lo siguiente:

Indra es una de las empresas Lideres en consultoría y tecnología y cuenta con socios estratégicos en todo el mundo. Provee soluciones globales en las áreas de Transporte y Defensa, y una de las empresas con mayor reconocimiento en consultoría con reconocimiento en Europa y Latinoamérica

Líder en transformación digital y Tecnologías con su filial Minsait. Su modelo de negocio está basado en una oferta integral de productos propios, con un enfoque end-to-end, de alto valor y con un elevado componente de innovación.

Indra tuvo unos ingresos de 3.043 millones euros durante el año 2020 Cuenta con una planilla de aproximadamente 48.000 colaboradores, con oficinas en más de 46 países (13 América, 32 Europa, 1 en Perú) y operaciones comerciales en más de 140 países.

Dentro de sus principales clientes Indra cuenta con las más reconocidas empresas del rubro Bancario y de telecomunicaciones

- BCP
- BBVA
- BANBIF
- BANCO DE COMERCIO
- BANCO PICHINCHA
- SCOTIABANK
- MOVISTAR
- ENTEL
- CLARO DEL PERU

2.2. VISIÓN

De acuerdo con la Misión de INDRA (2021) nos informa lo siguiente:

Crear en conjunto con sus colaboradores conocimiento en innovación dándole a sus clientes una consultoría completa, proponiéndole proyectos de integración de software de sistemas y aplicaciones para procesos de negocios. Esta oferta se estructura en dos segmentos principales: Soluciones y Servicios.

2.3. MISIÓN

De acuerdo con la Visión de INDRA (2021) nos informa lo siguiente:

Ser sostenible en el largo plazo acompañado al buen comportamiento de la empresa brindando su compromiso con el público en el ámbito económico, medioambiental y social. Para Indra, la responsabilidad de la empresa debe ir en línea con su actividad natural, la creación de riqueza, y en su caso a través de la generación de soluciones y servicios, y de aquello que les es propio y distintivo: la innovación

2.4. ORGANIZACIÓN DE LA EMPRESA

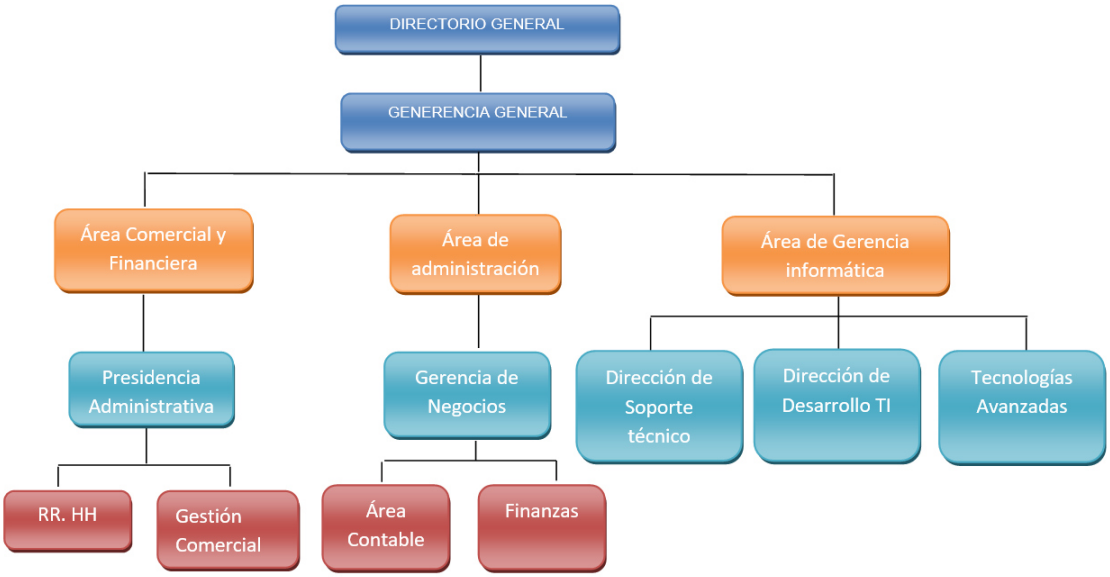


Figura 1. Organigrama general
fuente: Elaboración propia adaptado de indracompany.com

2.5. ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS

El creador de este trabajo de suficiencia profesional (TSP), ejerce el cargo como Data Engineer en el área de Tecnologías Avanzadas para la consultora Indra Perú, ofreciendo los servicios al Equipo de Data Riesgos Migración (DRM) en una empresa del sistema financiero bancario.

Dentro de las principales actividades del área de Riesgos se tiene como objetivo principal disponibilizar la información correcta y confiable para la toma de decisiones de las diferentes áreas de la organización bancaria.

Esta área en mención tiene como uno de sus objetivos brindar información del cliente como son, obtener información de clientes que podrían acceder a aprobación de un crédito, realizar la creación de modelos que permitan determinar que clientes podrían ser buenos pagadores y quienes podrían caer en endeudamiento.

El autor dentro de sus funciones como Data Engineer tiene las siguientes responsabilidades:

- Participar en el proyecto de migración Data Lake definidos por el área de riesgos.
- Refactorización y optimización proponiendo mejoras a los procesos actuales utilizando las buenas prácticas de Big Data
- Cumplir con los lineamientos y estándares establecidos por la empresa para el desarrollo de las soluciones
- Coordinación con el dueño del producto para el uso correcto de las reglas del negocio
- Disponibilizar las soluciones a los clientes finales en el entorno Data Lake
- Monitorear los estadísticos de los datos generados por el proceso de migración viendo la completitud e integridad de los datos
- En coordinación con el dueño del producto se crean pilotos con la finalidad de mejorar y agregarle valor a los datos

Es importante mencionar que el área está dividida en dos grandes equipos conformado aproximadamente por catorce personas, uno de los cuales está integrado por analistas quienes tienen la responsabilidad de conocer el negocio y ser los dueños del

producto; y otro grupo que está integrado por Data Engineers y Modeladores los cuales proponen solución a los problemas de área apoyándose en herramientas tecnológicas. A continuación, se detallan los principales roles en el área:

- Owner: Es el dueño del producto y aquel que da respuestas a las consultas del negocio.
- Chapter: Es el líder del producto y el encargado de que se cuenten con la disponibilidad y tecnologías necesarias.
- Modelador: Es aquella persona que define los nombres de tablas y campos de las bases de datos (Oracle 12c). Conoce del negocio y sigue los lineamientos y políticas de la entidad financiera.
- Data Engineer: Es aquel que se encarga de realizar las tareas de análisis, desarrollo y mejoramiento de procesos de las sistemas y aplicaciones existentes, basándose en las reglas de negocio establecida por los usuarios.
- Quality Assurance (QA): Valida y verifica realizando las pruebas rendimiento, integración y de sistemas, que cumplan todas las reglas establecidas por el cliente final
- Gestor de malla: Es aquel que se encarga de diseñar el flujo de ejecución de los procesos y el que genera las tareas para el pase a producción.

2.6. EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN

El autor del presente trabajo ha tenido como experiencia dentro de la organización la participación en diferentes proyectos como Data Engineer:

- Portafolio de clientes: Proyecto que consiste en integrar los clientes provenientes de Personas naturales y Tarjetas ofrecidas por la entidad financiera.
- Universo Pyme: Proyecto que consiste en crear un modelo para estudiar el comportamiento de los clientes desde diferentes fuentes como Registro Consolidado de Crédito (RCC), Portafolio de clientes y SUNAT.
- Motor de calidad: Proyecto que consiste en crear un framework que permite generar visualización y estadísticos de los datos en diferentes capas del Data Lake (como: RDV-Raw Data Vault, UDV-Universal Data Vault, DDV-Dimensional Data Vault)

- Refactorizar procesos y proponer soluciones que permitan optimizar los tiempos de ejecución de los procesos de desarrollo de Oracle migrados a Data Lake.
- Apoyo a los nuevos integrantes en la capacitación y en la preparación del ambiente para los desarrollos en el Data Lake y su migración posterior.
- Disponibilizar tablas input en el Data Lake para su posterior procesamiento utilizando SPARK.

CAPÍTULO III
ACTIVIDADES DESARROLLADAS

3.1 SITUACIÓN PROBLEMÁTICA

Hoy en día las organizaciones no cuentan con las herramientas y tecnologías necesarias para generar valor en el negocio basado en los datos que permita tomar decisiones en los tiempos esperados.

La entidad financiera del caso de estudio, desde hace algunos años lleva proponiendo modelos de datos que han brindado solución a muchas problemáticas de la organización, todas estas soluciones han sido propuestas utilizando Bases de datos tradicionales y entornos Datawarehouse. Estos modelos siguen actualmente en producción y están presentado problemas de procesamiento, tiempos de ejecución y espacios en disco lo cual ha provocado problemas en el análisis de la información, procesos recurrentes y en la toma de decisiones en el negocio.

Actualmente existe mucha competencia en los distintos rubros bancarios, donde se están proponiendo soluciones innovadoras apoyándose en las tecnologías actuales, los cuales les han permitido obtener resultados en un corto tiempo, lo que ha permitido una buena toma de decisiones ofreciendo mejores servicios que les ha permitido tomar buenas decisiones, captar nuevos clientes, muchos de ellos nuestros.

Además, se presenta un problema de encriptación de datos, ya que actualmente los usuarios en el área de riesgos tienen acceso a información sensible del usuario, lo cual muchas actualmente es un riesgo.

3.1.1 DEFINICIÓN DEL PROBLEMA

Problema principal

No se cuenta con una solución de Big data que permita una correcta toma de decisiones en un menor tiempo

Problemas secundarios

1. Falta de integración de datos, pues cada usuario genera tablas aisladas, lo cual genera redundancia en la información.
2. Velocidad de procesamiento lento, se tiene mucha información, los motores de procesamiento actuales no son los óptimos.
3. No se cuenta con información confiable, ya que la mayoría de la información se encuentra desactualizada.
4. Poca capacidad de almacenamiento, porque muchos usuarios utilizan las herramientas y sobrecargan la base de datos y los procesos en ejecución se ven perjudicados.

Se describe algunas causas:

- Los usuarios del negocio están identificados con las herramientas tradicionales y son resistentes al cambio.
- No se tiene un correcto gobierno del dato.
- Se crean tablas de acuerdo con el desarrollo o el requerimiento del área, lo que genera en el tiempo tablas e información redundante en sus bases de datos.
- Por requerimiento de espacio, se borra información histórica, lo que conlleva a una mala toma de decisiones
- Los procesos por su naturaleza actual en estos entornos requieren mucho tiempo de procesamiento y generan dependencia en procesos recurrentes

3.2 SOLUCIÓN

Desarrollar una solución de Big Data en una entidad bancaria para refactorizar sus procesos de migración y toma de decisiones utilizando la infraestructura HADOOP. Almacenando los datos en HDFS, el procesamiento con el motor SPARK y la metadata de las tablas usando HIVE.

3.2.1 OBJETIVOS

Objetivo General

Desarrollar una solución de Big Data que permita refactorizar sus procesos de migración y la toma de decisiones para solucionar los problemas del área y obtener datos con valor en una

entidad bancaria**.**

Objetivos Específicos

1. Analizar y refactorizar los procesos de integración de datos evitando la redundancia de información y el correcto gobierno de los datos.
2. Retar a los usuarios en el uso de las tecnologías de Big Data para procesamiento y almacenamiento de datos usando el motor de procesamiento SPARK el cual trabaja en memoria y con grandes volúmenes de información
3. Definir un modelo de datos que me permita gestionar correctamente la información y la toma de decisiones.
4. Utilizar las capacidades de Hadoop y su componente HDFS el cual permitirá tener un almacenamiento distribuido masivo.

3.2.2 ALCANCE

- Alcance Funcional

El alcance del presente informe consiste en migrar los modelos de bases de datos actuales a un entorno Data Lake lo que permitirá centralizar toda la información del área de riesgos, procesamiento y almacenamiento distribuido, Escalabilidad y finalmente poder obtener información con valor para el negocio.

- Alcance Organizacional

El proyecto permitió que otras áreas puedan implementar sus soluciones integrándolas a este nuevo proceso, soluciones tales como:

Tabla 5

Impacto en las áreas de negocio

AREA IMPACTADA	SOLUCION
Banca Negocios	Se adopto la solución propuesta para los clientes de la pequeña empresa, para la solución de los clientes de negocio.

Finanzas	Utilizar la información del modelo de clientes para la generar otro modelo, como el de modelo COVID-19 de clientes.
Pyme	Procesos recurrentes de cálculo de indicadores para clientes de pequeñas y medianas empresas

Nota. Elaboración propia

3.2.3 ETAPAS Y METODOLOGÍA

Las etapas tomadas en cuenta para el desarrollo de la solución Big Data se llevaron bajo el marco de metodología SCRUM.

En esta metodología las tareas se realizarán mediante Sprint en los cuales se obtendrá un entregable para el negocio. En cada sprint se tendrán en cuenta las siguientes fases: Definición del problema, diseño, desarrollo, pruebas y por último el despliegue.

Bajo esta metodología se ven involucrados diferentes roles como son:

1. Product Owner: Es el dueño del producto y el encargado de definir el alcance del proyecto y priorizar el backlog
2. Scrum Máster: Es el facilitador, encargado de organizar las ceremonias, refinar el backlog y realizar la retrospectiva
3. Equipo desarrollo: Data Engineers a cargo del análisis y puesta en marcha del backlog.

Dentro del proyecto de migración me encuentro con el rol de Data Engineer y mis actividades se detallarán a continuación en el siguiente trabajo de suficiencia profesional. Por temas de confidencialidad, algunos de los entregables no se podrán mostrar en este documento.

Número de Quarters

Se realiza la programación anual donde el año se dividirá en 4 Quarters.

Número de Sprint

El proyecto se llevó a cabo en 6 sprints en el intervalo del Q2 y Q3

Tiempo de duración del Sprint

Cada sprint se planificada durante dos semanas

Roles involucrados en el proyecto

En el proyecto participaron Product Owner, Chapter Leader, Scrum máster y el equipo de desarrollo.

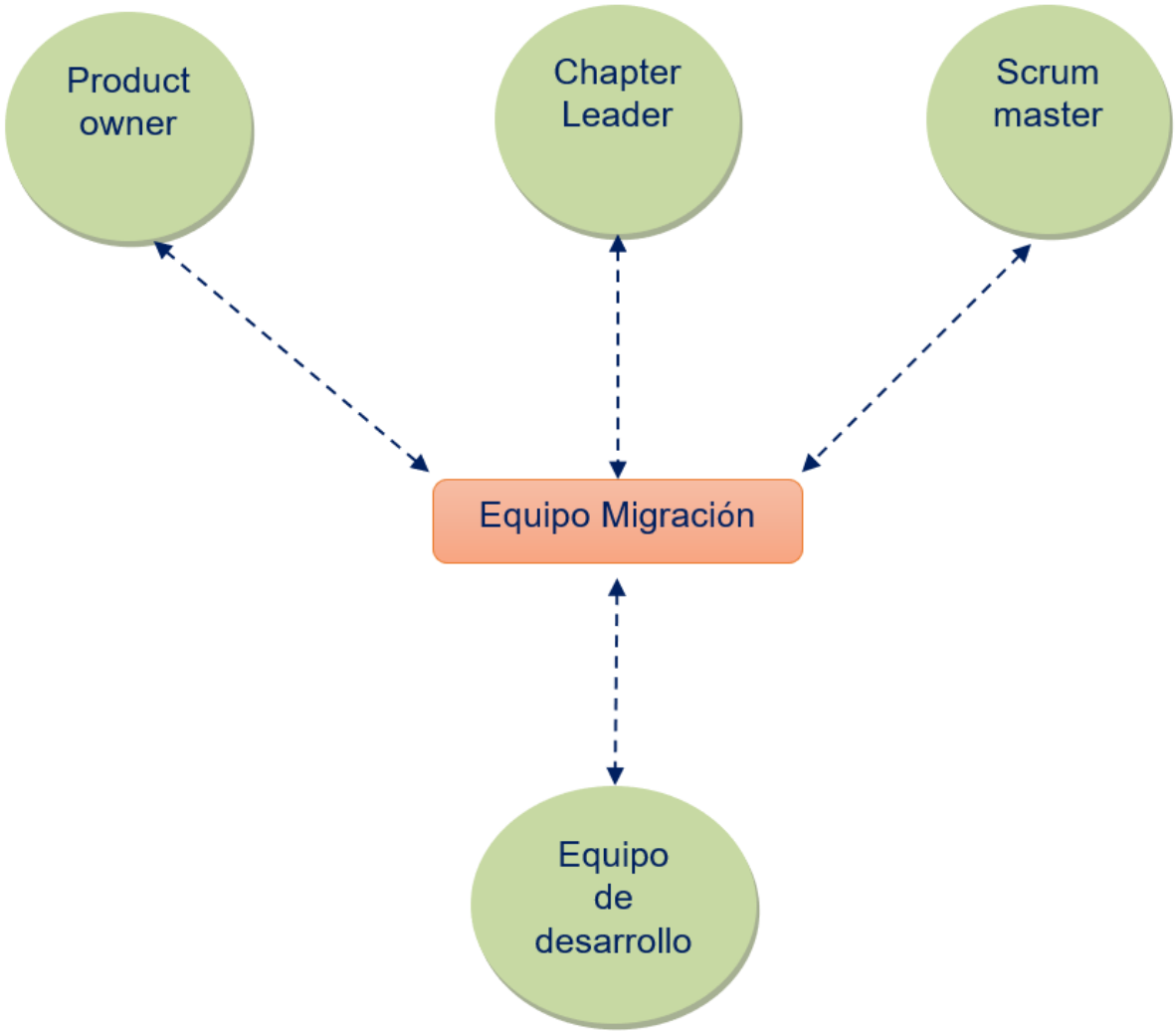


Figura 2. Roles del proyecto
fuente: Elaboración propia

Cronograma de las iteraciones

El cronograma nos muestra las iteraciones realizadas, actividades y responsables, esta actividad se inició el 15 de abril del 2021 al 15 de julio del 2021

Tabla 6
Cronograma de actividades por sprint

#sprint	Evento	#Dias	Inicio	Fin	Responsables
1	Sprint Planning	1	15/04/2021	15/04/2021	Producto owner Scrum Máster Equipo de desarrollo
1	Inicio del Sprint	9	16/04/2021	28/04/2021	Equipo de migración
2	Sprint Planning	1	30/04/2021	30/04/2021	Producto owner Scrum Máster Equipo de desarrollo
2	Inicio del Sprint	9	01/05/2021	13/05/2021	Equipo de migración
3	Sprint Planning	1	17/05/2021	17/05/2021	Producto owner Scrum Máster Equipo de desarrollo
3	Inicio del Sprint	9	18/05/2021	28/05/2021	Equipo de migración
4	Sprint Planning	1	01/06/2021	01/06/2021	Producto owner Scrum Máster Equipo de desarrollo

4	Inicio del Sprint	9	02/06/2021	14/06/2021	Equipo de migración
5	Sprint Planning	1	16/06/2021	16/06/2021	Producto owner Scrum Máster Equipo de desarrollo
5	Inicio del Sprint	9	17/06/2021	29/06/2021	Equipo de migración
6	Sprint Planning	1	01/07/2021	01/07/2021	Producto owner Scrum Máster Equipo de desarrollo
6	Inicio del Sprint	9	02/07/2021	14/07/2021	Equipo de migración

Nota. Elaboración propia

3.2.4 FUNDAMENTOS UTILIZADOS

3.2.4.1 Big Data

Características

De acuerdo con lo mencionado por Seliya (2021) indica que el paradigma del Big Data se basa en cumplir los desafíos de las 5vs, las cuales son

- **Volumen:** "El volumen es la dimensión más obvia al caracterizar grandes colecciones de datos creadas para diferentes usos y propósitos. El almacenamiento de Big Data supone el reto más inmediato, ya que la primera responsabilidad es la de preservar todos los datos generados en el ámbito de actuación del sistema".
- **Velocidad:** "La velocidad indica la velocidad a la que entran los datos y cómo podría cambiar potencialmente las características del volumen de Big data"
- **Variedad:** "se refiere a los diferentes grados de estructura (o falta de ella) que pueden encontrarse en una colección considerada Big Data. La colección puede integrar datos

procedentes de múltiples fuentes como las redes de sensores, logs generados en servidores web"

- **Veracidad:**" Hace referencia a que la inconsistencia y falta de completitud de los datos impacta negativamente en la veracidad y confiabilidad de estos. Además, menciona que es necesario trabajar con datos que no sean falsos ni estén corrompidos y que provengan de una fuente confiable"
- **Valor:** "El valor a menudo se considera el aspecto más importante de Big Data, y eso se debe a que la extracción de un corpus de datos tan grande debería producir resultados prácticos y valor comercial para el usuario final"

Ecosistema

Según lo expuesto por Cravero (2020) el ecosistema del Big Data se define como:

> Este conjunto de componentes interrelacionados se puede definir como el Ecosistema Big Data que se ocupa de la evolución de los datos, los modelos e infraestructura de apoyo durante todo el ciclo de vida de Big Data. Por otro lado, definen un Marco de Arquitectura de Big Data, que incluye 5 componentes que abordan distintos aspectos del ecosistema: (1) modelo de datos, estructuras y tipos; (2) administración de Big Data; (3) herramientas de análisis; (4) Infraestructura; y (5) seguridad de Big Data.

La siguiente figura muestra la relación entre los componentes.

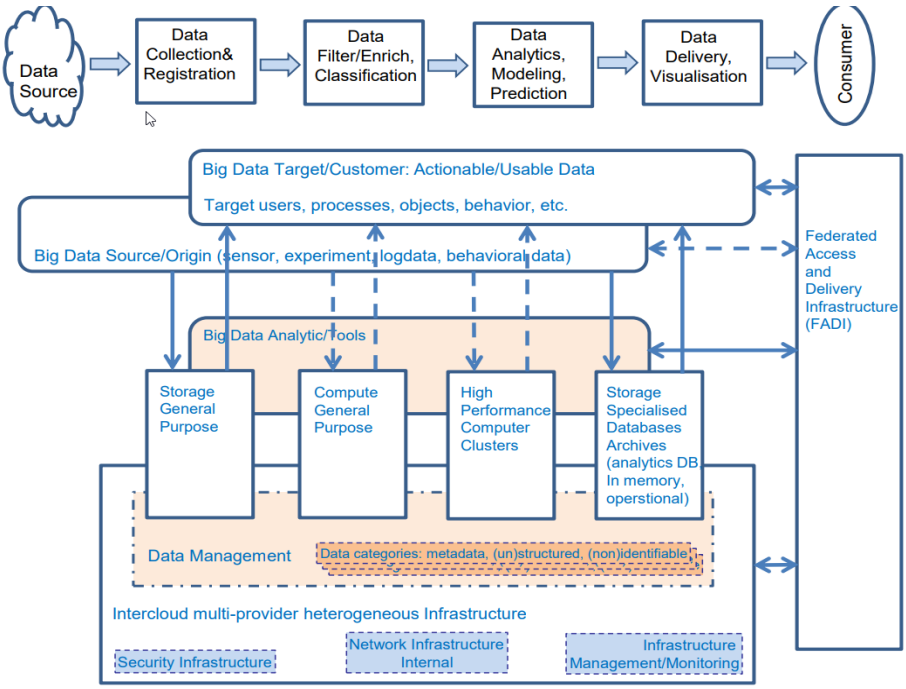


Figura 3. Arquitectura general del Big Data
fuente: Cravero (2020)

Componentes del Big Data

- **Data Lake:** Según lo expuesto por Pwint (2018)

  En la era del Big data, un nuevo término llamado "Data Lake" apareció en el universo digital. La intención más simple del lago de datos es mezclar todos los datos producidos por una organización para brindan información más valiosa con una granularidad más fina. Las tecnologías de Big Data son a veces consideradas como tecnologías destructivas ya que revolucionaron las formas tradicionales de hacer cosas en esta era de uso intensivo de datos. Se vuelven a aplicar los conceptos del sistema distribuido y paralelo como la base de Big Data, como los paradigmas MapReduce para manejar las Big Vs características - volumen, velocidad, variedad, valor y valor. Las bases de datos SQL incumbentes con características ACID son desafiadas (y algunas veces incluso reemplazadas) por NoSQL bases de datos con características BASE. Ahora, el concepto de Data Lake intenta desafiar los almacenes de datos tradicionales y fiables para almacenar datos complejos heterogéneos

- **Ingesta:** Por lo mencionado por Meehan (2017)

  La ingestión de datos es el proceso de obtener datos de su fuente a su sistema doméstico de la manera más eficiente y correcta posible. Este siempre ha sido un problema

importante y ha sido el objetivo de muchas iniciativas de investigación anteriores, como integración de datos, de duplicación, mantenimiento de restricciones de integridad y carga masiva de datos. La ingestión de datos es con frecuencia discutido bajo el nombre de Extraer, Transformar y Cargar.

Tecnologías utilizadas

- **Hadoop:** Por otro lado, Honar (2021) define como un marco para administrar un clúster de computadoras con procesamiento distribuido en MapReduce. Siendo sus componentes principales MapReduce (para procesamiento paralelo y distribuido) y Hadoop Distributed File System (HDFS) (como almacenamiento de datos en un sistema de archivos distribuido)

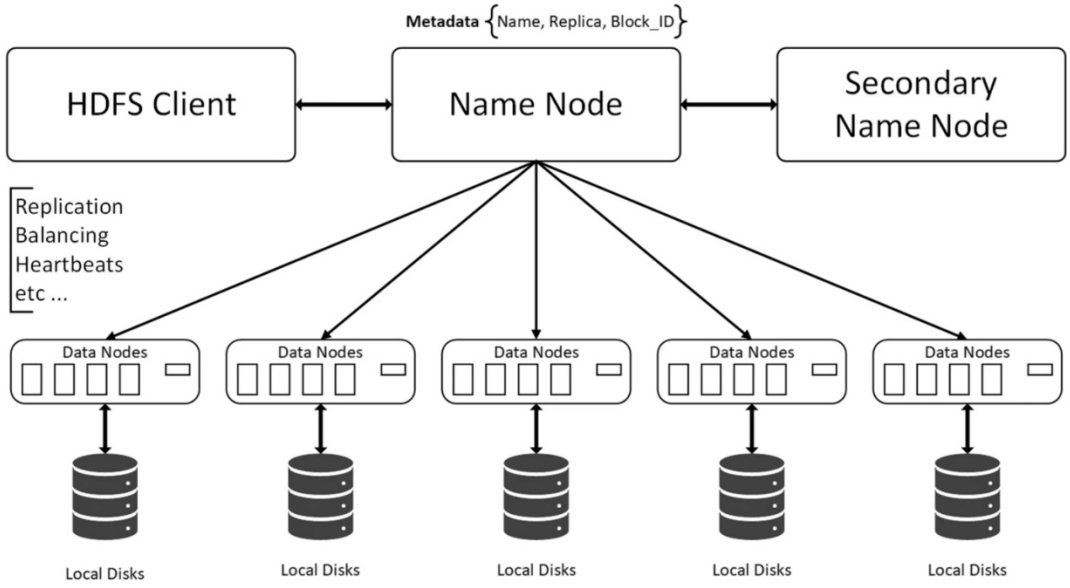


Figura 4. Estructura HDFS
fuente: Honar (2021)

- **Apache Spark**: Según lo expuesto en por Shaikh (2019) "Apache Spark es una potente plataforma de procesamiento de Big data que adapta el marco híbrido. Un marco híbrido ofrece soporte para capacidades de procesamiento tanto por lotes como por secuencias. Aunque Spark usa muchos principios similares a Hadoop El motor MapReduce, Spark supera a este último en términos de rendimiento. Por ejemplo, dado el mismo procesamiento por lotes carga de trabajo, Spark puede ser más rápido que MapReduce debido a la "completa "cálculo en memoria" que utiliza Spark en comparación a la lectura y escritura tradicionales en el disco utilizado por Mapa reducido. Spark puede ejecutarse en modo independiente o puede ser combinado con

Hadoop para reemplazar el motor MapReduce". Spark soporta lenguajes de programación como Python, Scala y R

Tipos de almacenamiento

- **Avro:** Según Nexla (2018) "Apache Avro fue lanzado por el grupo de trabajo de Hadoop en 2009. Es un formato basado en filas que es altamente divisible. La característica innovadora y clave de Avro es que el esquema viaja con datos. La definición de datos se almacena en formato JSON mientras los datos se almacenan en formato binario, lo que minimiza el tamaño del archivo y maximiza la eficiencia. Avro cuenta con un sólido soporte para la evolución de esquemas mediante la gestión de campos, campos faltantes y campos que han cambiado. Esto permite que el software antiguo leer los nuevos datos y el nuevo software para leer los datos antiguos, una característica fundamental si sus datos tienen el potencial de cambiar.
- **Parquet:** Según Nexla (2018) "Lanzado en 2013, Parquet fue desarrollado por Cloudera y Twitter (e inspiró por el sistema de consultas Dremel de Google) para servir como un almacén de datos en columnas optimizado en Hadoop. Parquet es especialmente experto en analizar conjuntos de datos amplios con muchas columnas. Cada archivo Parquet contiene datos binarios organizados por "grupo de filas". Por cada fila grupo, los valores de los datos están organizados por columna. Esto permite beneficios de compresión que describimos anteriormente. El parquet es una buena opción para cargas de trabajo de gran lectura"

3.2.4.2 Proceso de migración de datos

Según lo expuesto por Pérez (2020) menciona lo siguiente: "El proceso de Migración de Datos es cada vez más utilizado y demandado, debido a la necesidad de tomar los datos de unos repositorios y trasladarlos a otros más eficientes, con mayores capacidades de almacenamiento, con mejores mecanismos de seguridad y provistos de mejores posibilidades de explotación de la data. Debido a su magnitud e importancia empresarial estos procesos son tratados como proyectos paralelos a los propios de desarrollo. Algunas de las razones por las cuales se aborda este proceso son: Cambio de plataforma tecnológica, Cambio y actualización de aplicativos informáticos, Mejoramiento en tiempos de respuesta, Mejores Políticas de seguridad, Compatibilidad con otros aplicativos. Facilitar el intercambio de información,

Optimización de ambientes de TI, Aplicación de nuevas reglas del negocio, Adaptabilidad a exigencias del mercado”

Los procesos de migración en la entidad bancaria consisten en los siguientes pasos:

- Identificar el proceso Oracle a migrar
- Realizar un análisis de los componentes
- Creación del script
- Realizar el proceso de migración

3.2.4.3 Toma de decisiones

De acuerdo con lo mencionado por Peñalosa (2010) “La toma de decisiones es una situación que está presente en nuestras vidas desde que despertamos hasta que nos acostamos, solamente al despertar debemos elegir entre levantarnos o no, cuando nos levantamos elegimos si nos ponemos o no zapatos para caminar dentro el dormitorio y así sucesivamente, nuestra vida está llena de elecciones, unas más difíciles que otras, con más o menos implicancia, pero al final siempre estamos decidiendo. Es por esta razón que desde hace bastante tiempo las personas vienen estudiando este tema, tratando de facilitar la toma de decisiones y reducir el riesgo al mínimo posible cuando de elegir se trata.”

Según lo expuesto por Rodríguez (2017) “El modelo distingue las cuatro etapas/fases de toma de decisiones y precisa los procesos cognitivos que intervienen en el mismo: percepción organizacional, creación de conocimiento, negociación y aprendizaje organizacional. También presenta los procesos informacionales que garantizan un adecuado uso de información: búsqueda y selección, procesamiento y análisis de Información”

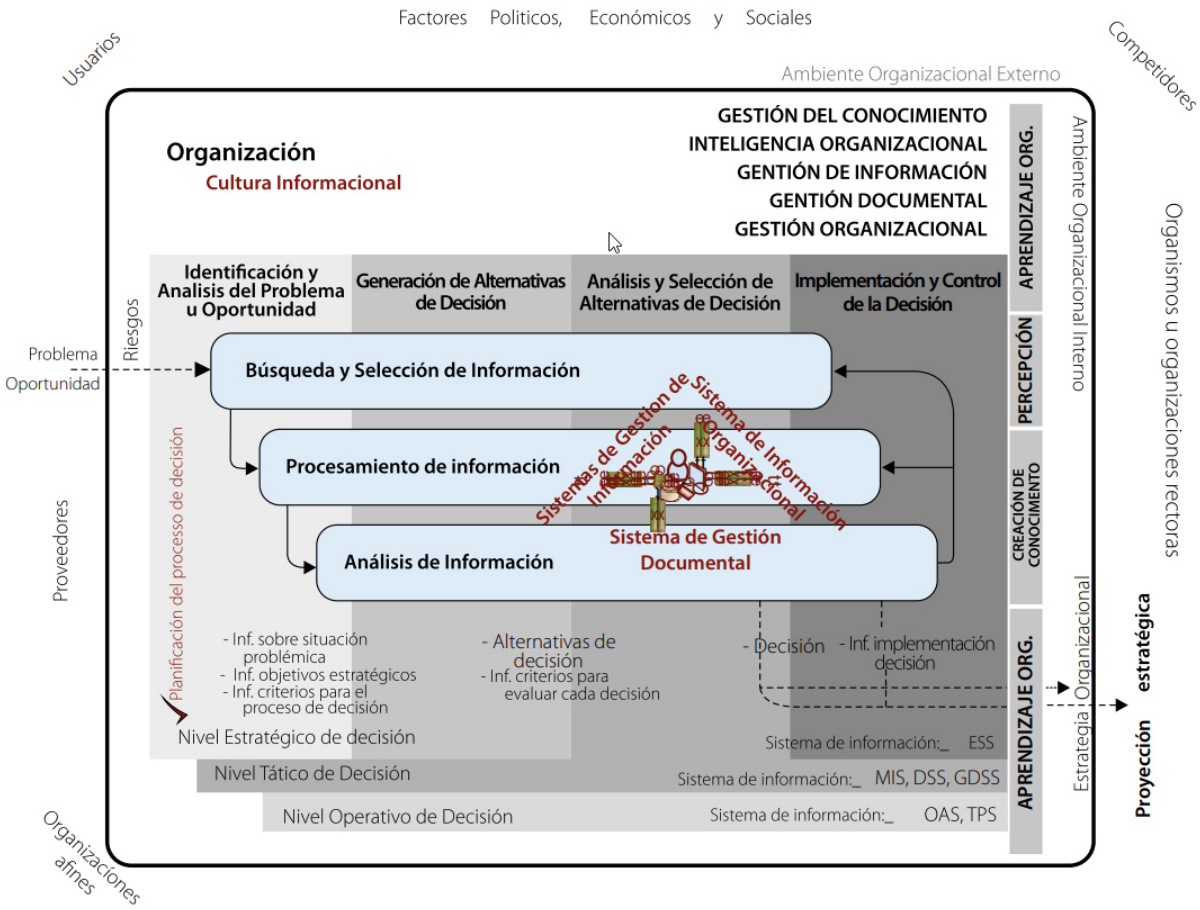


Figura 5. Modelo de uso de información para la toma de decisiones
fuente: Rodriguez (2017)

3.2.4.4 Gestión bancaria

El Big data cumple un papel trascendental en la gestión bancaria pues de acuerdo con lo mencionado por Fernandez (2019) “hay tecnologías que resultan especialmente útiles en el sector financiero para sistemas de pronóstico y de análisis de comportamiento de la clientela mediante la explotación del llamado Big data, que está generando ingente información que es fácilmente aprovechable por nuevos competidores de la banca: las grandes compañías tecnológicas, que son las grandes recolectoras de datos y que ha llevado a identificar a esos datos como el motor de la actualidad”

3.2.5 IMPLEMENTACIÓN DE LAS ÁREAS, PROCESOS Y SISTEMAS

De acuerdo con lo expuesto en las etapas y metodologías definido en TSP, en este capítulo se da el detalle de la implementación y los procesos realizados durante el desarrollo del proyecto.

Tabla 7

Actividades por sprint

#sprint	Evento	Actividades	Responsables
1	Sprint Planning	Plantear la necesidad de poder migrar un modelo para analizar a los clientes y no clientes de la entidad bancaria	Producto owner Scrum Máster Equipo de desarrollo
1	Inicio del Sprint	Identificar y definir las fuentes que se migraran inicialmente dependiente su criticidad.	Equipo de migración
2	Sprint Planning	Plantear la necesidad de migrar los clientes de la fuente SUNAT	Producto owner Scrum Máster Equipo de desarrollo
2	Inicio del Sprint	Realizar una carga táctica de la tabla SUNAT de Oracle a Data Lake	Equipo de migración
3	Sprint Planning	El dueño del producto plantea la necesidad de migrar por prioridad las fuentes Cliente riesgos y RCC	Producto owner Scrum Máster Equipo de desarrollo
3	Inicio del Sprint	Realizar la migración a Data	Equipo de migración

		Lake refactorizando el proceso Cliente riesgos y RCC	
4	Sprint Planning	El dueño del producto plantea la necesidad de migrar el Portafolio de clientes PV y LSA	Producto owner Scrum Máster Equipo de desarrollo
4	Inicio del Sprint	Realizar la migración a Data Lake refactorizando el proceso del Portafolio de clientes PV y LSA	Equipo de migración
5	Sprint Planning	Al tener todas las fuentes disponibles se plantea realizar la integración de todos los procesos del cliente.	Producto owner Scrum Máster Equipo de desarrollo
5	Inicio del Sprint	Realizar Integración del modelo para el control de clientes y no clientes integrando las fuentes de RCC, SUNAT, Portafolio y Riesgos	Equipo de migración
6	Sprint Planning	El negocio tiene la necesidad de productivizar los procesos actuales, por lo cual se realizará el despliegue de toda la solución utilizando las herramientas con las que dispone la organización.	Producto owner Scrum Máster Equipo de desarrollo
6	Inicio del Sprint	Despliegue y pase a producción	Equipo de migración

Nota. Elaboración propia

Sprint 1

Planning sprint 1

El negocio se siente en la necesidad de poder migrar un modelo para analizar a los clientes y no clientes de la entidad bancaria, por lo que se analizara las cuatro principales fuentes como son: Clientes SUNAT, Clientes RCC, Portafolio de Clientes y riesgos. Como primera actividad lo que se requiere es analizar el estatus actual del proceso e identificar la estrategia de migración del proceso.

Actualmente el proceso presenta los siguientes problemas:

- Largo tiempo de procesamiento
- El proceso sufre caídas por falta de memoria, lo cual genera dependencias en otros procesos e impacta en procesos recurrentes en el negocio
- Los datos proporcionados no están actualizados
- Se tienen problemas de reprocesamiento
- No se guarda información histórica pues sus procesos ocupan mucho espacio por lo que solo se dispone de información del mes actual y del mes anterior al proceso
- Se esperan integrar más fuentes al proceso actual y poder estudiar al cliente desde otras perspectivas
- El proceso no está modularizado por lo cual el mantenimiento de este proceso es un riesgo para el negocio

Tabla 8

Actividades del sprint 1

Ticket	Actividad	Responsable
TAREA-810	Identificar y definir las fuentes que se migraran inicialmente dependiente su criticidad.	Data Engineer

Nota. Elaboración propia

Inicio sprint 1

a) Primer paso: análisis del script

Para iniciar esta actividad recibo el script de Oracle de la solución propuesta, se requirió realizar la revisión del script actual e identificar todas las tablas involucradas en el proceso

```
INSERT INTO  TP_UNIVERSO
(
  CODMES,
  CODCLAVE,
  CTD
)
SELECT
  &V_MES1 AS CODMES,
  CODCLAVE,
  COUNT(1) AS CTD
FROM (
    SELECT
  CODMES, CODCLAVE,
    SUM(CASE WHEN substr(CODCTA,1,2)='14' AND substr(CODCTA,4,1) in ('1') AND substr(CODCTA,5,2) in ('02','13','12')
    SUM(CASE WHEN substr(CODCTA,1,2)='72' AND substr(CODCTA,4,1) IN ('5') AND SUBSTR(CODCTA,7,2) IN ('02','12','13')
    FROM TBT.DM_DETALLERCC
    WHERE CODMES>=&V_MES13 and CODMES<=&V_MES2
    AND CODCLAVE>0
    AND (
      (substr(CODCTA,1,2)='14' AND substr(CODCTA,4,1) in ('1') AND substr(CODCTA,5,2) in ('02','13','12'))
        OR
      (substr(CODCTA,1,2)='72' AND substr(CODCTA,4,1) IN ('5') AND SUBSTR(CODCTA,7,2) IN ('02','12','13'))
      )
    GROUP BY CODMES, CODCLAVE
)
WHERE DEUDA_PYME_VIGENTE>0 OR LINEA_PYME_NOUTIL>0
```

Figura 6. Script Inicial Cliente Pyme
Fuente: Referencia del proyecto en la entidad bancaria

b) Segundo paso: Identificación de las fuentes.

Una vez revisado el código fuente, se va identificando todas las tablas involucradas en el proceso y realizando un gráfico del estado actual del proceso, lo cual nos permitirá poder identificar oportunidades de mejora en el proceso.

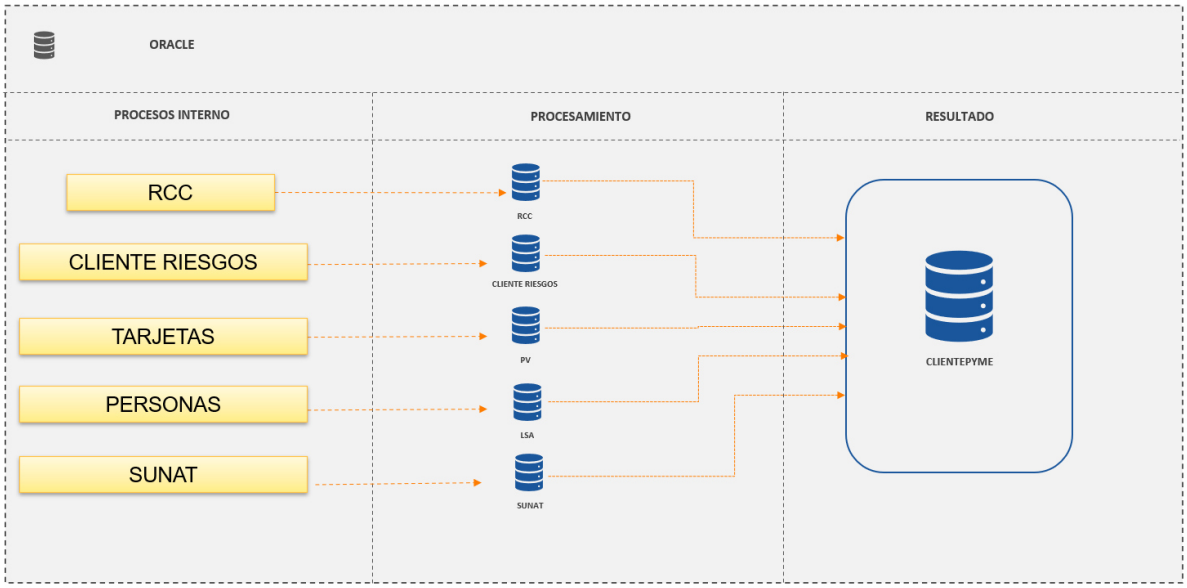


Figura 7. Bosquejo inicial del proceso
Fuente: Elaboración propia

c) Tercer paso: Una descripción de los procesos.

Una vez obtenido el grafico inicial se tendrá que entender la información que almacena cada proceso, esto con la finalidad de poder identificar si es que se pueden unificar procesos con comportamientos similares, a continuación, una breve descripción de cada proceso:

- Proceso RCC: Es aquel proceso que nos permite obtener información del endeudamiento de los clientes
- Proceso Cliente Riesgos: Es aquel proceso que nos permite obtener información personal del cliente
- Tarjetas: Nos permite obtener información del movimiento de las tarjetas del cliente
- Personas: Nos permite obtener información de los movimientos de las tarjetas de personas que son no clientes de la entidad
- SUNAT: Información de los clientes reportados con deuda

d) Cuarto paso: Primera planificación para el proceso de migración.

Una vez identificado cada proceso se procedió a realizar la planificación de los posteriores sprints, en los cuales se ira refactorizando y migrando cada uno de los procesos utilizando el motor de procesamiento SPARK.

El plan de migración se iniciará con la fuente de información SUNAT la cual es una fuente que se encuentra disponible en Oracle.

Sprint 2

Planning sprint 2

Por estrategia del negocio, se realizará una migración de la fuente SUNAT, esta fuente es adquirida por la organización y almacenada mensualmente en la base de datos Oracle, por lo cual se requirió realizar una migración de esta tabla desde el origen Oracle a el destino Data Lake utilizando un conector el cual permita migrar la información mensual de los clientes.

Tabla 9

Actividades del sprint 2

Ticket	Actividad	Responsable
TAREA-811	Realizar una carga táctica de la tabla SUNAT de Oracle a Data Lake	Data Engineer

Nota. Elaboración propia

Inicio sprint 2

Para poder realizar esta actividad, se requirió investigar el proceso de cómo se Migra una tabla de Oracle a Datalake en un ambiente de Cloudera, el cual detallare a continuación:

a) Crear un archivo json de configuración que tenga los parámetros de conexión al servidor Oracle. Estructura del archivo "credentials.json"

```
{
  "username": "MATRICULA",
  "hostname": "XX.XXX.XXX.XX",
  "port": "1521",
  "database": "NOMBREBD",
  "url_root": "jdbc:oracle:thin:@",
  "pass_file": "ora.password"
}
```

Figura 8. Configuración de credenciales
Nota: Elaboración propia

b) Creación del archivo password.jceks en cual almacenara la contraseña del Oracle en el datalake. Con este conjunto de comandos se podrá realizar la conexión al entorno creando una llave de encriptación la cual permitirá crear una contraseña de Oracle en el ambiente Data Lake

```
kinit
hadoop credential delete ora.password -provider jceks://hdfs/user/BMATRICULA/password.jceks
hadoop credential create ora.password -provider jceks://hdfs/user/BMATRICULA/password.jceks
hadoop credential list -provider jceks://hdfs/user/BMATRICULA/password.jceks
```

Figura 9. Creación de la contraseña
Nota: Elaboración propia

c) Para poder realizar la ingesta de la tabla de Oracle, se requiere crear una tabla en hive en Data Lake de tipo external, la cual permita almacenar la información ingestada desde Oracle al entorno Lake

```
DROP TABLE IF EXISTS  ${hiveconf:AMBIENTE}_clientepyme.sunat;
CREATE EXTERNAL TABLE ${hiveconf:AMBIENTE}_clientepyme.sunat
(
     CODCLAVE                                   VARCHAR(128),
     TIPDOC                                     CHAR(20),
     FECRUTINA                                  TIMESTAMP,
     FECACTUALIZACIONREGISTRO                   TIMESTAMP
)
PARTITIONED BY (CODMES INT)
STORED AS PARQUET
LOCATION '/${hiveconf:AMBIENTE}/empresa/clientepyme/sunat'
TBLPROPERTIES ('parquet.compression'='SNAPPY');
```

Figura 10. Creación de la tabla SUNAT
Nota: Elaboración propia

d) Creación del script en Pyspark el cual me permitirá realizar la migración de la tabla Oracle a Data Lake y poder almacenarla en la tabla HIVE creada.

```
spark = SparkSession.builder \
          .appName("carga sunat") \
          .config("spark.driver.memory", "1g")\
          .config("spark.executor.cores", "6")\
          .config("spark.executor.memory", "8g")\
          .config("spark.dynamicAllocation.maxExecutors", "20")\
          .config("spark.executor.memoryOverhead", "2g")\
          .config("spark.jars", "ojdbc7.jar")\
          .config("spark.driver.extraClassPath", "ojdbc7.jar")\
          .config("spark.executor.extraJavaOptions", "-Doracle.net.crypto_checksum_client=REQUESTED,-Dorac
          .config("spark.driver.extraJavaOptions", "-Doracle.net.crypto_checksum_client=REQUESTED,-Doracle
          .config("spark.hadoop.hadoop.security.credential.provider.path","jceks://hdfs/user/BMATRICULA/pa
          .enableHiveSupport() \
          .getOrCreate()
spark.conf.set("spark.sql.shuffle.partitions", "11")
spark.conf.set("spark.default.parallelism", "15")

dbName = '${hiveconf:AMBIENTE}_clientepyme'
tableName =  'sunat'

#2. Abrir el archivo de credentials
with open('credentials.json', 'r') as f:
    credentials = json.load(f)



#3. Abrir el passfile
x = spark.sparkContext._jsc.hadoopConfiguration()
a = x.getPassword(credentials['pass_file'])
passw = ""
for i in range(a.__len__()):
   passw = passw + str(a.__getitem__(i))

url = credentials['url_root']+credentials['hostname']+":"+credentials['port']+":"+credentials['database']


sqlQuery = """(SELECT CODCLAVE, TIPDOC, FECRUTINA, FECACTUALIZACIONREGISTRO FROM SUNAT)"""
```

Figura 11. Script Pyspark SUNAT
Nota: Elaboración propia

Este proceso contiene inicialmente la definición de los valores de tunning para la ejecución del proceso Spark, posteriormente se define los parámetros de configuración que permitirán conectarnos al Data Lake haciendo uso de la llave de encriptación antes explicada. Finalmente se realiza la lectura del archivo ".json" que contiene los parámetros de conexión a Oracle y posteriormente se realiza la consulta o Query la cual me permitirá extraer la información de Oracle y disponibilizarla en Data Lake

e) Para consolidar el proceso y aprendizaje se consolida en un gráfico todo el proceso de migración de una tabla Oracle a Data Lake
   - Creación de la tabla HIVE
   - Lectura del archivo con la configuración del motor de base de datos
   - Obtención de la contraseña Oracle encriptada en datalake
   - Lectura de la tabla SUNAT mediante Spark, convirtiéndola en un dataframe
   - Escritura del dataframe en una ruta HDFS

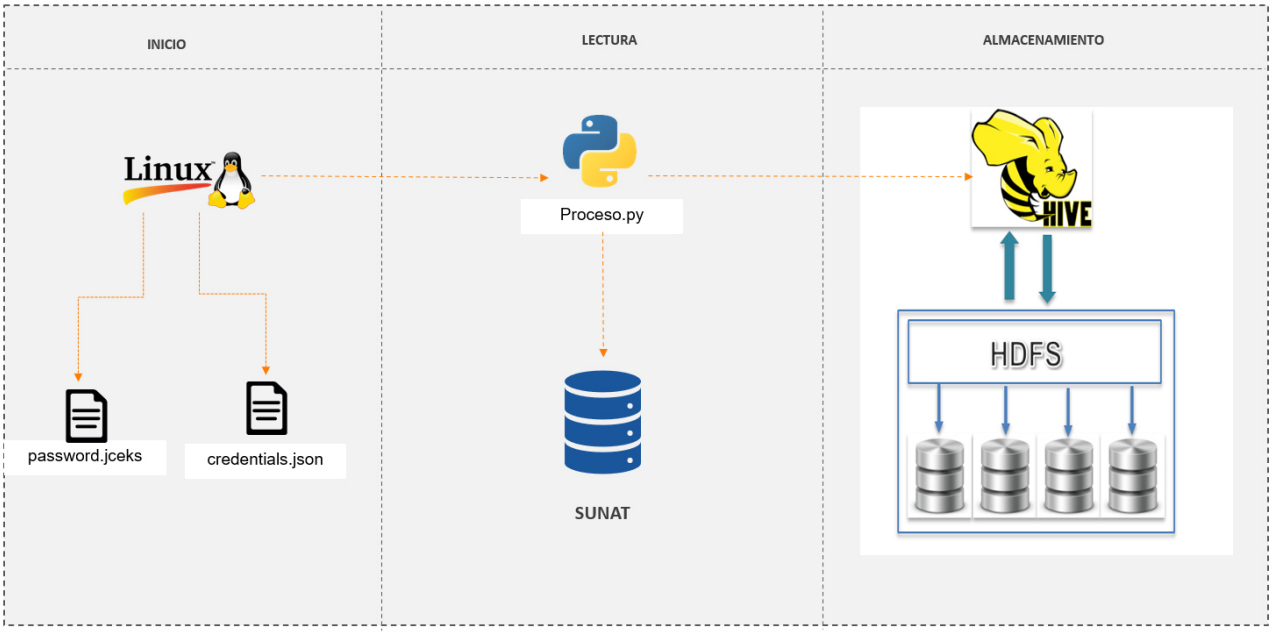


Figura 12. Migración de tabla SUNAT
Nota: Elaboración propia

f) Finalmente se deberá realizar la consulta de la Tabla final ingestada en Hive

	codclave	tipdoc	fecrutina	fecactualizacionregistro
1	a4473ab2eb	6	2021-08-19 17:50:30.0	2021-08-19 17:50:30.0
2	04e114edee	1	2021-08-19 17:50:30.0	2021-08-19 17:50:30.0
3	9216ea8fc1	1	2021-08-19 17:50:30.0	2021-08-19 17:50:30.0
4	6450136a3a	1	2021-08-19 17:50:30.0	2021-08-19 17:50:30.0
5	72cb8a8028	6	2021-08-19 17:50:30.0	2021-08-19 17:50:30.0
6	92197ee9eb	1	2021-08-19 17:50:30.0	2021-08-19 17:50:30.0

Figura 13. Tabla migrada SUNAT
Nota: Elaboración propia

Sprint 3

Planning sprint 3

Se requiere contar con información de los clientes y su endeudamiento, por lo cual se realizará la carga mensual de esta información en el Data Lake, esta información es enviada mensualmente por lo cual se desea construir un flujo que permita disponibilizar información mensual de RCC en el Data Lake.

Adicionalmente a esto se requiere contar con información de las tablas que contengan la lista de Clientes y No clientes de la entidad financiera la cual se encuentra en la tabla Cliente riesgos, por lo cual se requiere realizar la migración adicional de esta tabla.

Tabla 10

Actividades del sprint 3

Ticket	Actividad	Responsable
TAREA-812	Realizar la migración a Data Lake refactorizando el proceso Clientes RCC	Data Engineer
TAREA-813	Realizar la migración a Data Lake refactorizando el proceso Cliente riesgos	Data Engineer

Nota. Elaboración propia

Inicio sprint 3

a) El flujo ya se conoce pues se realizó un plan piloto con la fuente SUNAT por lo cual se requerirá de lo siguiente para poder realizar la migración:

- Creación de la tabla HIVE para la fuente RCC
- Diseño del proceso de migración
- Creación del script Pyspark RCC

b) Se realizo el diseño de la tabla HIVE, el diseño de esta tabla se realizó tomando como referencia el script de la Figura 6:

```
DROP TABLE IF EXISTS ${hiveconf:AMBIENTE}INT.deudorsbs;
CREATE EXTERNAL TABLE ${hiveconf:AMBIENTE}INT.deudorsbs(
    CODCLAVEDEUDORSBS                VARCHAR(128),
    ROLDEUDORSBS                     CHAR(20),
    CREDITO                          CHAR(20),
    CODCLAVEEMPSISTEMAFINANCIERO     VARCHAR(128),
    ROLEMPSISTEMAFINANCIERO          CHAR(20),
    CODCLAVECONTABLESBS              VARCHAR(128),
    CTDDIAMOROSIDAD                  INT,
    CODSBS                           VARCHAR(30),
    CREDITOSBS                       CHAR(20),
    DESCREDITO                       VARCHAR(256),
    CODCONTABLESBS                   VARCHAR(30),
    DESCONTABLESBS                   VARCHAR(256),
    CODEMPSISTEMAFINANCIERO          VARCHAR(30),
    NBREMPSISTEMAFINANCIERO          VARCHAR(120),
    CLASIFRIESGO                     CHAR(20),
    DESCLASIFRIESGO                  VARCHAR(256),
    CODCLAVECLI                      VARCHAR(128),
    ROLCLI                           CHAR(20),
    CODCLAVEUNICOCLI                 VARCHAR(128),
    MONEDA                           CHAR(20),
    DESMONEDA                        VARCHAR(256),
    MTODEUDASOL                      DECIMAL(21,4),
    FECACTUALIZACIONREGISTRO         TIMESTAMP,
    FRECUENCIAREGISTRO               CHAR(1),
    FECRUTINA                        STRING,
    FECDIA                           STRING
)
STORED AS PARQUET
LOCATION '/${hiveconf:AMBIENTE}/empresa/int/deudorsbs'
TBLPROPERTIES ('parquet.compression'='SNAPPY');
```

Figura 14. Creación de la tabla RCC
Nota: Elaboración propia

c) Se realizo el esquema del proceso de migración para la fuente RCC

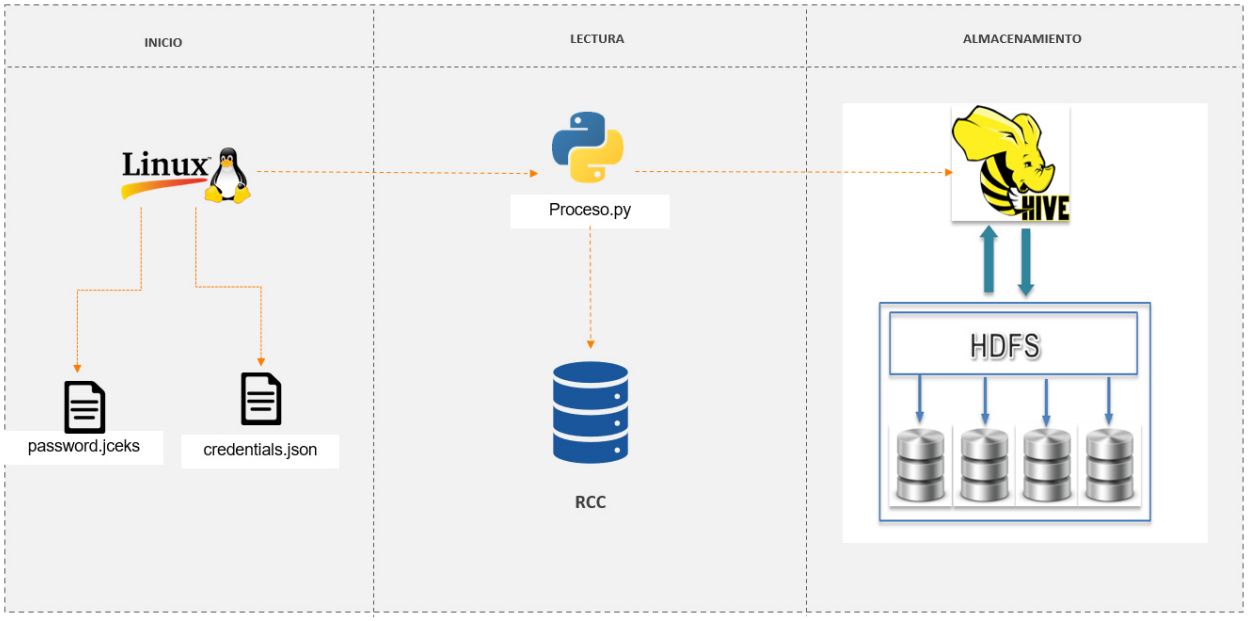


Figura 15. Migración de tabla RCC
Nota: Elaboración propia

d) Elaboración y ejecución del script

```
spark = SparkSession.builder \
          .appName("carga deudorsbs") \
          .config("spark.driver.memory", "1g")\
          .config("spark.executor.cores", "6")\
          .config("spark.executor.memory", "8g")\
          .config("spark.dynamicAllocation.maxExecutors", "20")\
          .config("spark.executor.memoryOverhead", "2g")\
          .config("spark.jars", "ojdbc7.jar")\
          .config("spark.driver.extraClassPath", "ojdbc7.jar")\
          .config("spark.executor.extraJavaOptions", "-Doracle.net.crypto_checksum_client=REQUESTED,-Dora
          .config("spark.driver.extraJavaOptions", "-Doracle.net.crypto_checksum_client=REQUESTED,-Doracl
          .config("spark.hadoop.hadoop.security.credential.provider.path","jceks://hdfs/user/BMATRICULA/p
          .enableHiveSupport() \
          .getOrCreate()
spark.conf.set("spark.sql.shuffle.partitions", "11")
spark.conf.set("spark.default.parallelism", "15")

dbName = '${hiveconf:AMBIENTE}int_deudorsbs'
tableName =  'deudorsbs'

#2. Abrir el archivo de credentials
with open('credentials.json', 'r') as f:
    credentials = json.load(f)



#3. Abrir el passfile
x = spark.sparkContext._jsc.hadoopConfiguration()
a = x.getPassword(credentials['pass_file'])
passw = ""
for i in range(a.__len__()):
   passw = passw + str(a.__getitem__(i))

url = credentials['url_root']+credentials['hostname']+":"+credentials['port']+":"+credentials['database']


sqlQuery = """(SELECT CODCLAVE, TIPFEC, TIPROL,
 CODSBS, DESSBS,
 FECRUTINA,
 FECACTUALIZACIONREGISTRO FROM RCC)"""
```

Figura 16. Script Pyspark RCC

Nota: Elaboración propia

e) Se realizo la creación del Script para la migración de la tabla RCC del ambiente Oracle a Data Lake

codclavedeudorsbs	roldeudorsbs	credito	codclaveempsistemafinanciero	rolempsistemafinanciero	codclavecontabl
cc0a4eefcd	0003	NR	2289768293	0014	b463c536f9
c998b91600	0003	NR	2153a4ae1a	0014	b463c536f9
8f2fcfb7d8	0003	NR	2289768293	0014	3a996e5f26
7e43ccf448	0003	NR	0ef20551d6	0014	17072b9e3a
0621030dab	0003	NR	0ef20551d6	0014	17072b9e3a
df3772674f	0003	NR	2153a4ae1a	0014	b463c536f9
f86ab16ea0	0003	NR	0ef20551d6	0014	17072b9e3a
aa2f2c6fdf	0003	NR	0ef20551d6	0014	17072b9e3a
9d5bef11c7	0003	NR	2153a4ae1a	0014	b463c536f9
30d47898da	0003	NR	0ef20551d6	0014	17072b9e3a

Figura 17. Tabla migrada RCC
Nota: Elaboración propia

Se realizo la migración de las tablas Oracle a Data Lake con información de los clientes con endeudamiento de los clientes, los datos en Data Lake se mantendrán encriptados para brindar seguridad a la información sensible del cliente, así como los campos necesarios para la ejecución del proceso Cliente Pyme.

Adicionalmente se debe mencionar que para la tabla CLIENTE RIESGOS se repite el mismo método de migración expuesto para las tablas SUNAT Y RCC en el sprint 2 y sprint 3

Sprint 4

Se tiene la necesidad de realizar la migración del proceso Portafolio Clientes, iniciando con los clientes PV que contiene información del consumo de las tarjetas de crédito de los clientes.

Tabla 11

Actividades del sprint 4

Ticket	Actividad	Responsable
TAREA-814	Realizar la migración a Data Lake refactorizando el proceso del Portafolio de clientes PV	Data Engineer
TAREA-815	Realizar la migración a Data Lake refactorizando el proceso del Portafolio de clientes LSA	Data Engineer

Nota. Elaboración propia

a) Estado actual

Inicialmente se realizó la revisión del código Oracle para el proceso de clientes PV

```
round(nvl(d.mtoprincipal_soles,0),2) as mtoprincipal_soles,
round(nvl(d.mtointeres_soles,0),2) as mtointeres_soles,
round(nvl(e.mtobalanceactual_soles,0),2) as mtobalanceactual_soles,
round(nvl(f.mtolineacredito_soles,0),2) as mtolineacredito_soles,
CASE WHEN round(nvl(d.mtoprincipal_soles,0),2)>0 THEN 1 ELSE 0 END FLG_MTOPRINCIPAL_MAYOR_CERO,
CASE WHEN round(nvl(f.mtolineacredito_soles,0),2)>0 THEN 1 ELSE 0 END FLG_MTOLINEACREDITO_MAYOR_CERO,
CASE WHEN B.TIPESTCTA IN ('A ','AC','D ') THEN 1 ELSE 0 END FLG_TIPESTCTA_ACTIVO,
G.FLG_CASTIGADO,
G.FLG_BLOQUEO,
H.FECAPERTURA,
CASE WHEN H.FECAPERTURA IS NULL THEN NULL ELSE MONTHS_BETWEEN (TO_DATE (A.CODMES, 'YYYYMM'), (TO_DATE
J.CODPRODUCTO
FROM TP_UNIV_CTA_VP_1_&V_MES  A,
     TP_UNIV_CTA_VP_ESTCTA_3_&V_MES  B,
     TP_UNIV_CTA_VP_BLOQ_3_&V_MES  C,
     TP_UNIV_CTA_VP_PRINC_1_&V_MES  D,
     TP_UNIV_CTA_VP_BAL_1_&V_MES  E,
     TP_UNIV_CTA_VP_LIN_2_&V_MES  F,
     PROY_RBP.MM_BLOQUEOS_CASTIGOS G,
     MD_CUENTA H,
     MD_CUENTA J,
     MD_CLIENTE I
WHERE A.CODCTATARJETA=B.CODCTATARJETA(+)
  AND A.CODCTATARJETA=C.CODCTATARJETA(+)
  AND A.CODCTATARJETA=D.CODCTATARJETA(+)
```

Figura 18. Script PV
Fuente: Referencia del proyecto en la entidad bancaria

Este código cuenta con las reglas de negocio y las tablas que están en Oracle con la información de tarjetas para los clientes de la entidad, por lo cual se realizó un gráfico para

poder identificar todos los inputs e identificar: la redundancia de tablas y la integración de cálculos en el procesamiento actual.

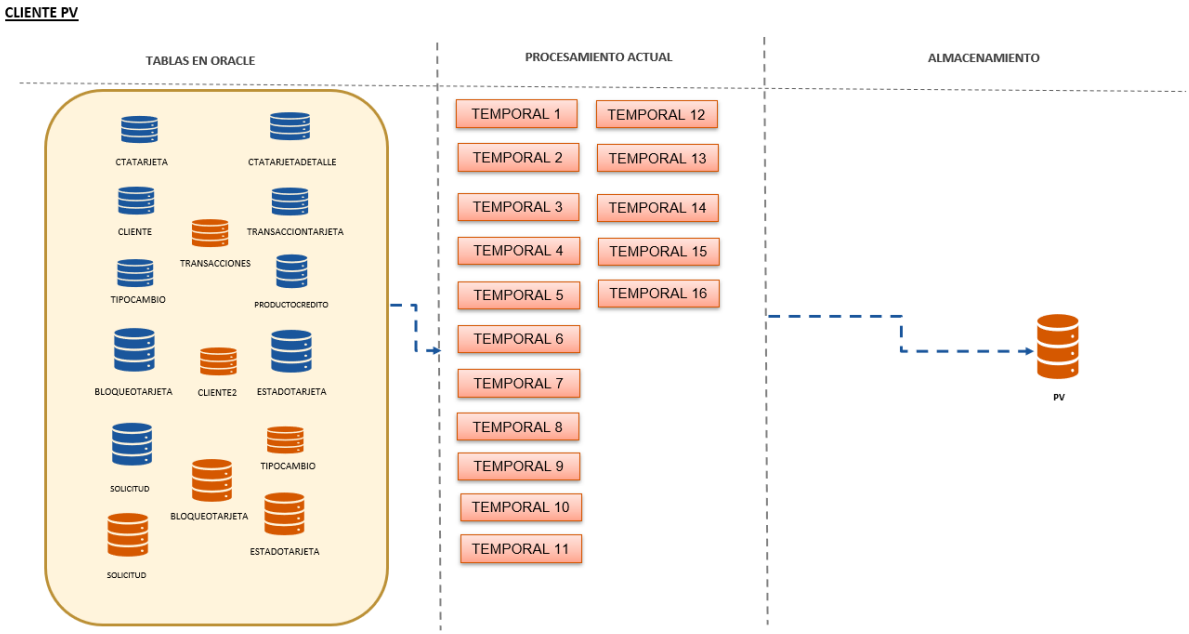


Figura 19. Diseño actual PV
Nota: Elaboración propia

En el siguiente grafico representa la situación actual del proceso PV, en el cual se tienen las tablas de Oracle, en las cuales se encuentran tablas redundantes, en la sección de procesamiento se representa las 16 tablas temporales creadas para realizar el proceso y finalmente como se almacena la información en Oracle.

b) La propuesta de solución a este problema se planteó de la siguiente manera:

- Se identificaron tablas maestras
- Se identificaron cálculos innecesarios que no generaban valor al producto final
- Existen tablas que no se utilizan en el proceso.

Con todo ello se realizó la propuesta de solución, de la siguiente manera:

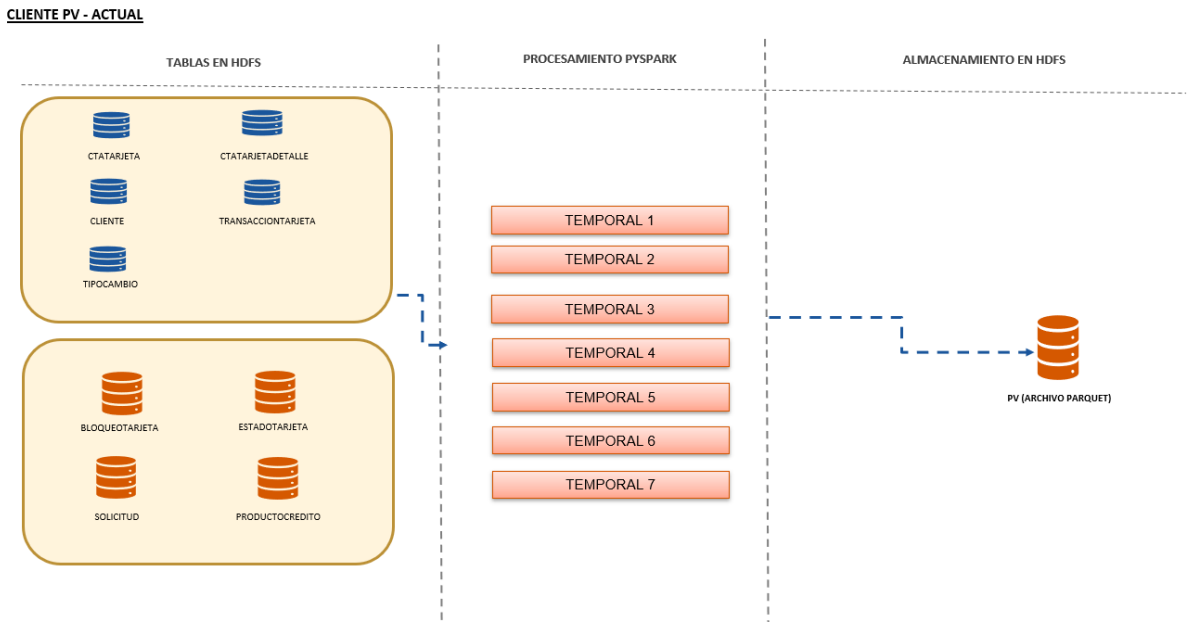


Figura 20. Diseño final PV
Nota: Elaboración propia

Se definieron las tablas input necesarias para el proceso, estas tablas input estuvieron a cargo de los data Engineer, quienes migraron estas tablas al HDFS para poder ser consumidas en el proceso PV. El proceso de refactorización fue realizado por el expositor de este informe. Al no ser PV una tabla de consumo final se decidió guardar esta información en HDFS en formato PARQUET.

c) Creación del script PV

De acuerdo con las buenas prácticas de Big Data el procesamiento más optimo es a través de la tecnología Spark, por lo cual se optó por utilizar Pyspark nativo. Con esto se logró:

- Refactorizar el proceso final a 7 tablas temporales
- Utilización recurrente de tablas maestras
- Aprovechar el procesamiento en memoria propio de spark
- Almacenar la información en formato Parquet con compresión Snappy, lo cual permitirá un menor tamaño de almacenamiento y una mayor calidad de procesamiento.

```
dfTarjetaConsolidado = dfTarjetaConsolidado.join(dfMdSolicitudMicPorTarjetaOriginal, on=['CODCLAVECTAORIGINALSOLICITUD'] , how='left_outer')

dfTarjetaConsolidado = dfTarjetaConsolidado.withColumn('CODSOLICITUD', coalesce('CODSOLICITUDMIC','CODSOLICITUD'))

dfTarjetaConsolidado = dfTarjetaConsolidado.join(dfTipCredito, on=['CODPRODUCTOCREDITO'], how='left_outer').\
                                            join(dfMmRelacionProducto, on=['CODPRODUCTORBM'], how='left_outer').\
                                            join(dfMmBloqueosCastigos, on=['TIPBLOQUEOPRODUCTO'], how='left_outer')

dfTarjetaConsolidado = dfTarjetaConsolidado.withColumn('FLGMTOSALDOCAPITALMAYOR0',when(round(col('MTOSALDOCAPITALSOL'),2) > 0, lit('1')).otherwise(lit('0'))).\
                                            withColumn('FLGMTOLINEACREDITOMAYOR0',when(((round(col('MTOLINEACREDITOSOL'),2) > 0) | (round(col('MTOLINEACREDITODOL'),2) > 0)), lit('1')).othe
                                            withColumn('FLGESTADOCTAACTIVA',when(col('TIPESTADOCTA').isin(['A','AC','D']), lit('1')).otherwise(lit('0'))).\
                                            withColumn('FLGCTAVALIDA',when(((col('FLGCAST') == 0) & (col('FLGESTADOCTAACTIVA') == '1') & ((col('FLGMTOSALDOCAPITALMAYOR0') == '1') | ((col('
                                            withColumn('FLGDEF30BLOQUEOATRASO',when(((col('CTDDIAATRASO') > 30) | substring(col('TIPBLOQUEOPRODUCTO'), 1, 1).isin(['E','Y','U','G','R','I','
                                            withColumn('FLGDEF60BLOQUEOATRASO',when(((col('CTDDIAATRASO') > 60) | substring(col('TIPBLOQUEOPRODUCTO'), 1, 1).isin(['E','Y','U','G','R','I'])
                                            withColumn('FLGDEF90BLOQUEOATRASO',when(((col('CTDDIAATRASO') > 90) | substring(col('TIPBLOQUEOPRODUCTO'), 1, 1).isin(['E','Y','U','G','R'])), 1
                                            withColumn('FLGDEF120BLOQUEOATRASO',when(((col('CTDDIAATRASO') > 120) | substring(col('TIPBLOQUEOPRODUCTO'), 1, 1).isin(['E','Y','U','G','R'])),


dfTarjetaConsolidado = dfTarjetaConsolidado.withColumn('FLGCTASPERSONAREV', when(((col('CODPRODUCTONIVEL0RBM') == 'TARJETA') & (col('FLGCTAVALIDA')=='1')), lit('1')).otherwise(lit('0'))).\
                                            withColumn('TIPRANGOLINEA', when(col('FLGCTASPERSONAREV')=='1', when((col('MTOLINEACREDITOSOL') < 1000), lit(1)).
                                                                                                            when(((col('MTOLINEACREDITOSOL') >= 1000) & (col('MTOLINEACREDITOSOL') < 3000)),
                                                                                                            when(((col('MTOLINEACREDITOSOL') >= 3000) & (col('MTOLINEACREDITOSOL') < 6000)),
                                                                                                            when((col('MTOLINEACREDITOSOL') >= 6000), lit(4)).
                                                                                                            otherwise(lit(9))
                                                                    ).otherwise(lit(None))).\
                                            withColumn('NUMPRODUCTONIVEL1RBM', when(col('FLGCTASPERSONAREV')=='1', when((col('CODPRODUCTONIVEL1RBM') == '01. Tarjeta Visa'), lit(1)).
                                                                                                                    when((col('CODPRODUCTONIVEL1RBM') == '02. Tarjeta Amex'), lit(2)).
                                                                                                                    when((col('CODPRODUCTONIVEL1RBM') == '03. Tarjeta Mastercard'), lit(3)).
                                                                                                                    otherwise(lit(9))
```

Figura 21. Script Pyspark PV
Nota: Elaboración propia

g) Como parte de la siguiente tarea de migración, se realizó la revisión del código LSA el cual contiene información de los préstamos solicitados por los clientes:

```
  A.CTDDIAATRASO,
      round(nvl(A.MTOPRINCIPAL_SOLES,0),2) AS MTOPRINCIPAL_SOLES,
      round(nvl(A.MTOBALANCEACTUAL_SOLES,0),2) AS MTOBALANCEACTUAL_SOLES,
      round(nvl(A.MTOPRINCIPAL_DOLARES,0),2) AS MTOPRINCIPAL_DOLARES,
      round(nvl(A.MTOBALANCEACTUAL_DOLARES,0),2) AS MTOBALANCEACTUAL_DOLARES,
      round(nvl(A.MTOTOTALINTERES_SOLES,0),2) AS MTOTOTALINTERES_SOLES, --<<
      CASE WHEN round(nvl(A.MTOPRINCIPAL_SOLES,0),2)>0  THEN 1 ELSE 0 END
      AS FLG_MTOPRINCIPAL_MAYOR_CERO,
      CASE WHEN TIPESTCTA IN ('A','AC','D') THEN 1 ELSE 0 END
      AS FLG_TIPESTCTA_ACTIVO,
      B.FLG_CASTIGADO,
      B.FLG_BLOQUEO,
      A.FECAPERTURA,
      A.NUMEDADMADURACION,
      A.FECDESEMBOLSOORIGINAL,
      A.CODMES_FECDESEMBORIG
FROM TP3 A,
     BLOQUEOS_CASTIGOS B
WHERE TRIM(A.TIPBLOQUEOPRODUCTO)=TRIM(B.TIPBLOQUEOPRODUCTO(+));
```

Figura 22. Script LSA
Fuente: Referencia del proyecto en la entidad bancaria

h) Se identifico que las tablas para el proceso PV eran las mismas que las del proceso LSA con diferente lógica de negocio, pero la tabla final tenía la misma estructura. El script actual del proceso LSA se ve representando en el siguiente grafico:

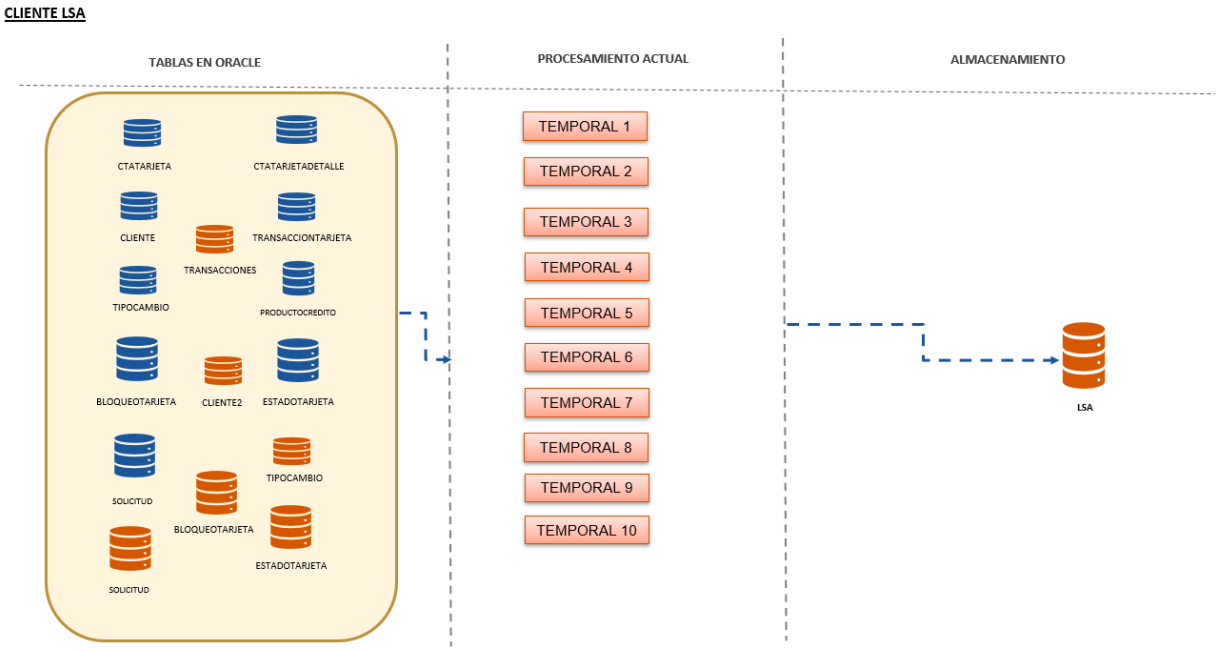


Figura 23. Diseño actual LSA
Nota: Elaboración propia

La propuesta de solución a este problema se planteó de la siguiente manera:

- Se reutilizaron tablas maestras del proceso PV
- Se identificaron cálculos innecesarios que no generaban valor al producto final

Con todo ello se realizó la propuesta de solución, de la siguiente manera:

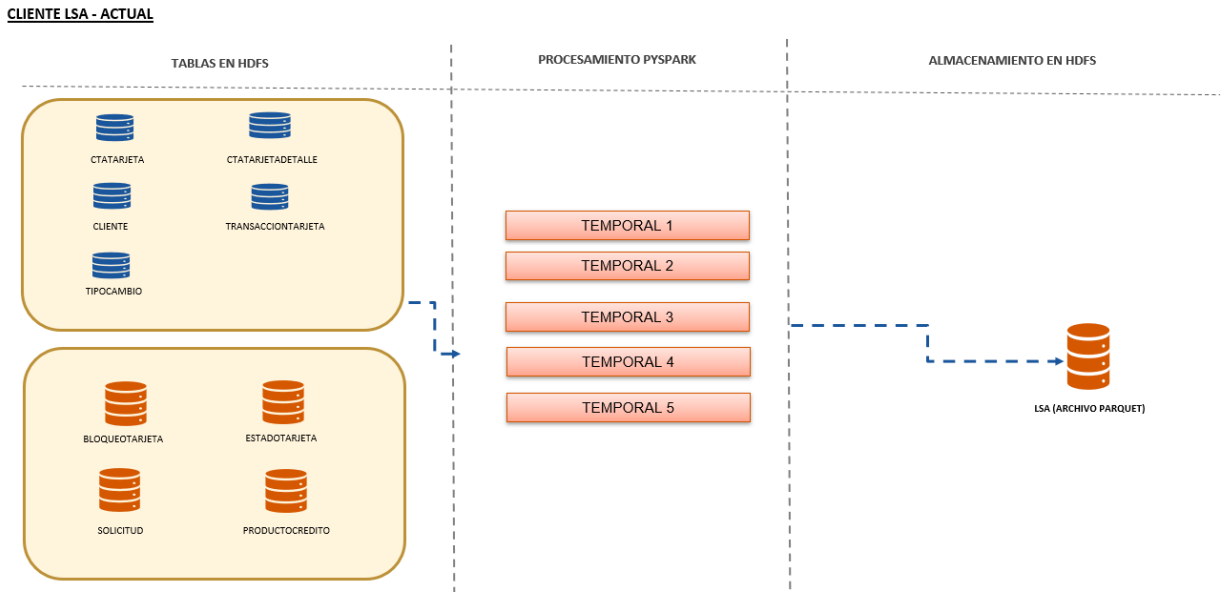


Figura 24. Diseño final LSA
Nota: Elaboración propia

Al no ser LSA una tabla de consumo final se decidió guardar esta información en HDFS en formato PARQUET.

i) Creación del script LSA

El procesamiento se realizó con pyspark nativo, realizando en general lo siguiente:

- Refactorizar el proceso final a 5 tablas temporales
- Reutilizar las tablas maestras del proceso PV
- Aprovechar el procesamiento en memoria propio de spark
- Almacenar la información en formato Parquet con compresión Snappy, lo cual permitirá un menor tamaño de almacenamiento y una mayor calidad de procesamiento.

```
dfTpPortafolioCreditoPrestamos = dfTpPortafolioCreditoPrestamos.withColumn('FLGMTOSALDOCAPITALMAYOR0', when((col('MTOSALDOCAPITALSOL') > 0), '1').otherwise(lit('0'))).\
                                 withColumn('FLGESTADOCTAACTIVA', when((col('TIPESTADOCTA').isin('A', 'AC', 'D')), '1').otherwise(lit('0'))).\
                                 withColumn('FLGCTAVALIDA', when((col('TIPESTADOCTA').isin('A', 'AC', 'D') & (col('FLGCAST') == 0) & (col('MTOSALDOCAPITALSOL') > 0)), '1').otherwise(lit('0'))).\
                                 withColumn('FLGDEF30BLOQUEOATRASO', when((col('CTDDIAATRASO') > 30) | (substring(col('TIPBLOQUEOPRODUCTO'), 1, 1).isin('E', 'Y', 'U', 'G', 'R')), '1').otherwise(lit('0'))).\
                                 withColumn('FLGDEF60BLOQUEOATRASO', when((col('CTDDIAATRASO') > 60) | (substring(col('TIPBLOQUEOPRODUCTO'), 1, 1).isin('E', 'Y', 'U', 'G', 'R')), '1').otherwise(lit('0'))).\
                                 withColumn('FLGDEF90BLOQUEOATRASO', when((col('CTDDIAATRASO') > 90) | (substring(col('TIPBLOQUEOPRODUCTO'), 1, 1).isin('E', 'Y', 'U', 'G', 'R')), '1').otherwise(lit('0'))).\
                                 withColumn('FLGDEF120BLOQUEOATRASO', when((col('CTDDIAATRASO') > 120) | (substring(col('TIPBLOQUEOPRODUCTO'), 1, 1).isin('E', 'Y', 'U', 'R')), '1').otherwise(lit('0'))).\
                                 withColumn('FECACTUALIZACIONREGISTRO', current_timestamp()).\
                                 withColumn('FECRUTINA', lit(PRM_SPARK_FECHARUTINA)).\
                                 withColumn('CTDMESMADURACION', when(col('FECAPERTURA').isNull(), '').otherwise(lit(months_between(to_date(col('FECRUTINA')), to_date(substring(col('FECAPERTURA'), 1, 7)))).cast('int'))).\
                                 withColumn('CODMESCOSECHA', date_format('FECAPERTURA', 'yyyyMM')).\
                                 withColumn('CODPRODUCTONIVEL1RBM', when((isNull('CODPRODUCTONIVEL1RBM') & (col('CODPRODUCTOCREDITO') == 'EPCV')), lit('13.Congelamiento Covid')).otherwise(col('CODPRODUCTONIVEL1RBM'))).\
                                 withColumn('CODCAMPANIA', coalesce(col('CODCAMPANIA'), when(col('CODPRODUCTONIVEL1RBM')=='12.Refinanciado Pyme', lit('REF')).
                                                                                        when(col('CODPRODUCTONIVEL1RBM')=='13. Reprogramado Pyme', lit('REP')).
                                                                                        when(substring(col('CODSOLICITUD'),1,3)=='YMC', concat(substring(col('CODSOLICITUD'),1,1),substring(col('CODSOLICITUD'),4,2))))
                                 ).\
                                 withColumn('DESCAMPANIA', lit(' ')).\
                                 withColumn('FLG_CONGELAMIENTO', when(col('CODPRODUCTOCREDITO') == 'EPCV', lit('1')).otherwise(lit('0'))).\
                                 withColumn('CODAPP', lit('ALS'))

dfTpPortafolioCreditoPrestamos = dfTpPortafolioCreditoPrestamos.join(dfProductoPyme, on=['CODCAMPANIA'], how='left_outer')

dfTpPortafolioCreditoPrestamos = dfTpPortafolioCreditoPrestamos.withColumn('FLG_PYME_NOREV', when(col('CODPRODUCTONIVEL0RBM')=='PYME NO REVOLVENTE', lit('1')).otherwise(lit('0'))).\
                                  withColumn('CODPRODUCTONIVEL0PYME', when(col('FLG_PYME_NOREV')=='1', when(col('FLG_CONGELAMIENTO')=='1', lit('WK')).otherwise(col('CODPRODUCTONIVEL0PYME')))).\
                                  withColumn('CODPRODUCTONIVEL1PYME', when(col('FLG_PYME_NOREV')=='1', when(col('FLG_CONGELAMIENTO')=='1', lit('CONGELAMIE')).otherwise(col('CODPRODUCTONIVEL1PYME')))).\
                                  withColumn('CODPRODUCTONIVEL2PYME', when(col('FLG_PYME_NOREV')=='1', when(col('FLG_CONGELAMIENTO')=='1', lit('WK')).otherwise(col('CODPRODUCTONIVEL2PYME'))))
```

Figura 25. Script Pyspark LSA
Nota: Elaboración propia

Posteriormente a la creación de los procesos PV y LSA, se realizó la integración de ambos procesos para la creación de la tabla final de Portafolio de clientes de la entidad, tomando como solución final el siguiente proceso:

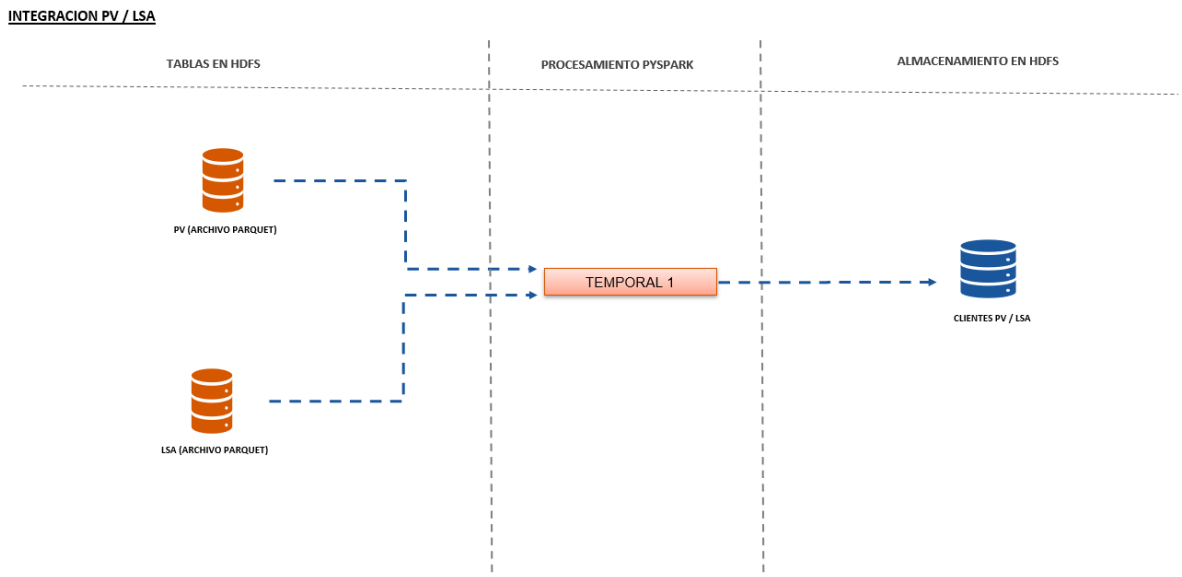


Figura 26. Diseño final integración PV/LSA
Nota: Elaboración propia

```
dfPortafolioPrestamos = spark.read.format("parquet").load(PRM_SPARK_FILE_INPUT_LOCATION + "/" + PRM_SPARK_FILE_PARQUET_TP_HD_PORTAFOLIOCREDITOPRESTAMOS)
dfPortafolioPrestamosUnion = dfPortafolioPrestamos.withColumn('CODCLAVECTAORIGINALSOLICITUD', lit(None)).\
                                        withColumn('MTOLINEACREDITO', lit(None)).\
                                        withColumn('MTOLINEACREDITOSOL', lit(None)).\
                                        withColumn('MTOLINEACREDITODOL', lit(None)).\
                                        withColumn('MTOLINEACREDITOINICIAL', lit(None)).\
                                        withColumn('CTDTRXCARGOMTOMAYOR0', lit(None)).\
                                        withColumn('CTDTRXCARGOMTOMAYOR100', lit(None)).\
                                        withColumn('CTDTRXCARGOMTOMAYOR1000', lit(None)).\
                                        withColumn('MTOTOTALTRXCARGO', lit(None)).\
                                        withColumn('MTOTOTALTRXABONO', lit(None)).\
                                        withColumn('MTOBTSOL', lit(None)).\
                                        withColumn('MTOEPSOL', lit(None)).\
                                        withColumn('MTOOPECOMPRASOL', lit(None)).\
                                        withColumn('MTODISPOSICIONEFECTIVOSOL', lit(None)).\
                                        withColumn('MTOCOMISIONTARJETACREDITOSOL', lit(None)).\
                                        withColumn('FLGMTOLINEACREDITOMAYOR0', lit(None)).\
                                        withColumn('CODLINEAPRODUCTORBM', lit(None)).\
                                        withColumn('DESLINEAPRODUCTORBM', lit(None)).\
                                        withColumn('CODGRUPOPRODUCTORBM', lit(None)).\
                                        withColumn('DESGRUPOPRODUCTORBM', lit(None)).\
                                        withColumn('TIPRANGOLINEA', lit(None)).\
                                        withColumn('DESTIPRANGOLINEA', lit(None)).\
                                        withColumn('NUMPRODUCTONIVEL1RBM', lit(None)).\
                                        withColumn('FLGLANPASS', lit(None)).\
                                        withColumn('FLGBT', lit(None)).\
                                        withColumn('FLGTARJETACREDITOPER', lit(None))


dfPortafolioTarjetas = spark.read.format("parquet").load(PRM_SPARK_FILE_INPUT_LOCATION + "/" + PRM_SPARK_FILE_PARQUET_TP_HD_PORTAFOLIOCREDITOTARJETAS)
dfPortafolioTarjetasUnion = dfPortafolioTarjetas.withColumn('FECDESEMBOLSO', lit(None)).\
                                        withColumn('CODPRODUCTONIVEL0PYME', lit(None)).\
                                        withColumn('CODPRODUCTONIVEL1PYME', lit(None)).\
                                        withColumn('CODPRODUCTONIVEL2PYME', lit(None)).\
                                        withColumn('CTDPAGOPENDIENTE', lit(None)).\
                                        withColumn('CTDPAGOREALIZADO', lit(None)).\
                                        withColumn('MTOVALORBIEN', lit(None)).\
                                        withColumn('MTOORIGINALCREDITO', lit(None)).\
                                        withColumn('MTODESEMBOLSADO', lit(None)).\
                                        withColumn('MTOCUOTAENVIO', lit(None)).\
                                        withColumn('MTOVCDA', lit(None)).\
                                        withColumn('MTOTOTALVCDA', lit(None)).\
                                        withColumn('MTOTOTALINTERESSEGURO', lit(None)).\
                                        withColumn('MTOTOTALINTERESSEGUROSOL', lit(None)).\
                                        withColumn('MTOTOTALINTERESSEGURODOL', lit(None)).\
                                        withColumn('MTOINTERESMORATORIO', lit(None)).\
                                        withColumn('MTOINTERESMORATORIOSOL', lit(None)).\
                                        withColumn('MTOINTERESMORATORIODOL', lit(None)).\
                                        withColumn('MTOTOTALGASTOCOBRANZA', lit(None)).\
                                        withColumn('MTOTOTALGASTOCOBRANZASOL', lit(None)).\
```

Figura 27. Script Pyspark de integración PV/LSA
Nota: Elaboración propia

Utilizando las buenas prácticas de Big Data se realizó la creación del proceso, tomando en cuenta almacenamientos en formato parquet snappy y una tabla final que permita guardar la historia diaria de los procesos de PV y LSA logrando el objetivo de refactorizar sus procesos, unificando la información otorgándole valor al negocio en los tiempos requeridos pues el proceso de la experiencia al cliente tendrá información diaria disponible en un tiempo razonable.

Sprint 5

Planning sprint 5

Se expuso que, teniendo ya las fuentes disponibles antes mencionadas, se tiene la necesidad de crear un tablón de clientes que me permita integrar a todos los clientes siguiendo unas reglas de negocio, las cuales me permiten identificar a los clientes por fuente de información y así poder tomar decisiones para el negocio y disponibilizar esta información para la toma de decisiones.

Este proceso actualmente se encuentra en Oracle y genera mensualmente aproximadamente 22 millones de registros y esta que presenta problemas que han impactado al negocio y se requiere brindar una solución óptima que permita corregir esta situación. Por lo cual se planeó realizar la siguiente actividad de integración

Tabla 12

Actividades del sprint 5

Ticket	Actividad	Responsable
TAREA-816	Realizar Integración del modelo para el control de clientes y no clientes integrando las fuentes de RCC, SUNAT, Portafolio y Riesgos	Data Engineer

Nota. Elaboración propia

Inicio sprint 5

a) Diseño de la solución

La finalidad de este proyecto es poder brindar una solución óptima al proceso actual que presenta el negocio, por lo cual, haciendo un diagnóstico de todas las características antes expuestas, nos vemos en la necesidad de migrar este proceso utilizando tecnologías de Big data, la cual nos brindara los siguientes beneficios:

- Procesamiento en Spark permitiendo alta capacidad de procesamiento distribuido
- Almacenamiento distribuido en HDFS brindando seguridad
- Brindar una solución escalable y modular, la cual permita poder integrar más procesos posteriormente y cuando se requieran
- Suficiente capacidad de almacenamiento que me permita analizar la información histórica de la información

Se planteo un modelo modularizado que permitirá la integración de nuevos componentes, ofreciendo escalabilidad y mantenimiento. Utilizando un patrón de desarrollo en Spark. El patrón checkpoint el cual permitirá ir incrementando progresivamente los

procesos y no sufrir problemas de memoria, que usualmente se presentan en proyectos utilizando tecnología Spark.

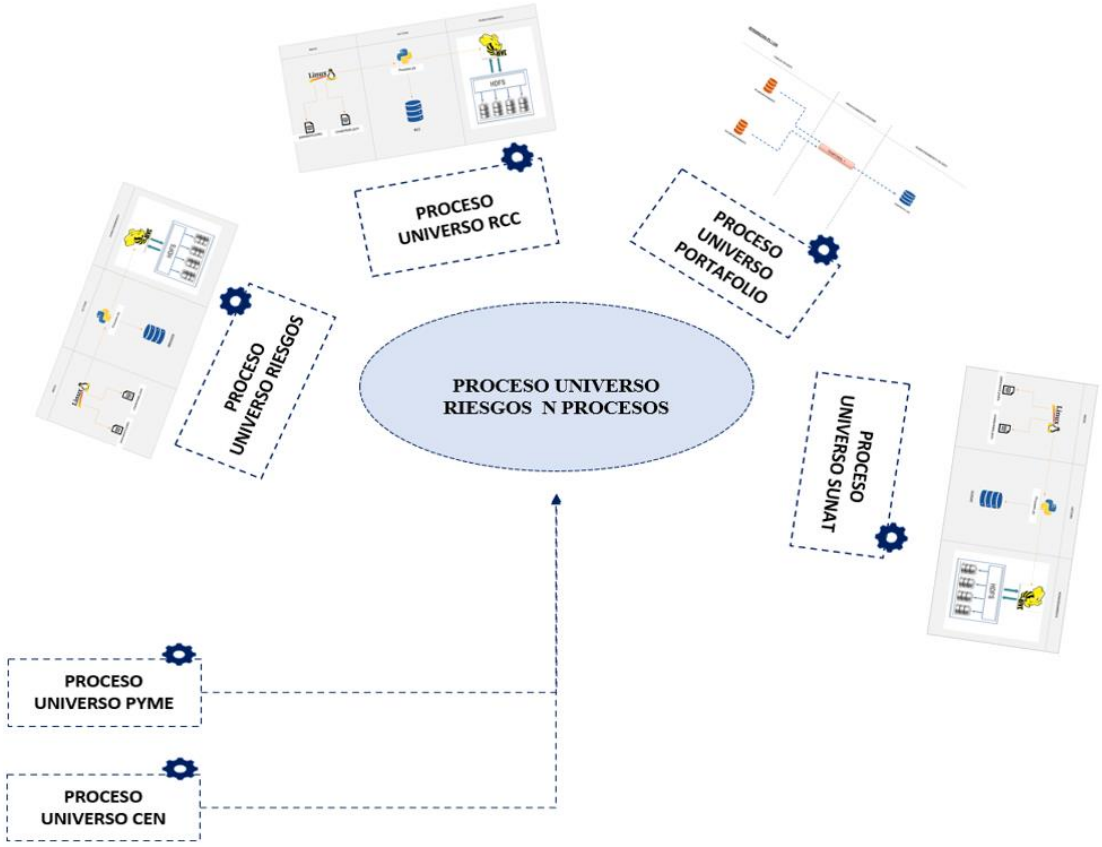


Figura 28. Propuesto de solución
Nota: Elaboración propia

Lo que se buscara en este proceso es poder modularizarlo con la finalidad de que pueda crecer el modelo y soportar más fuentes de información, de tal manera que puedan ser integradas al proceso, además de ser mantenible y escalable.

La representación final de la solución con todo el proceso integrados se presenta a continuación:

INTEGRACION DEL MODELO

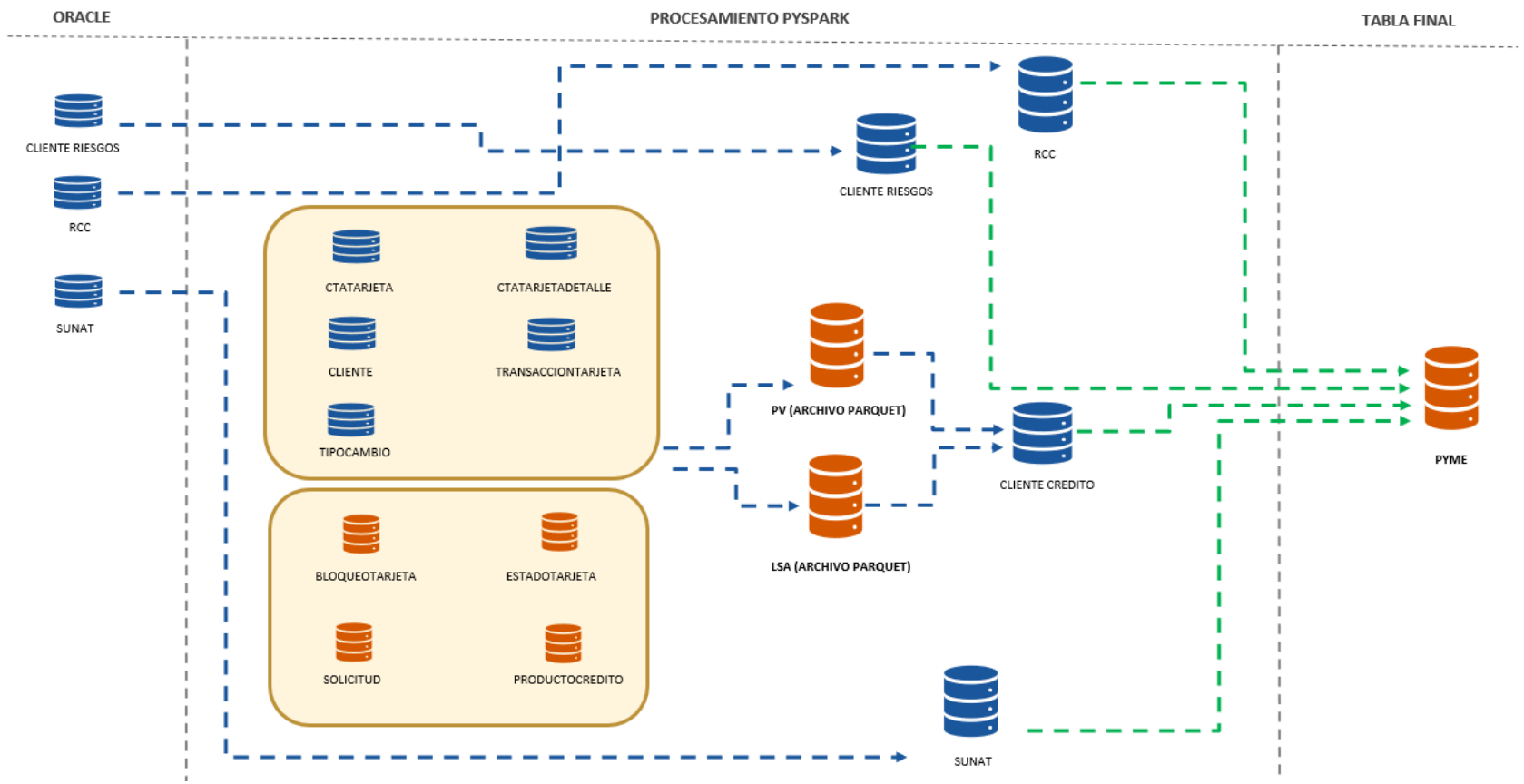


Figura 29. Diagrama de integración del cliente pyme
Nota: Elaboración propia

En la Figura 29 se muestra cómo se realiza la integración de todos los procesos antes construidos. Se muestran las fuentes RCC, SUNAT y Cliente Riesgos las cuales serán ingestadas mediante un proceso pyspark desde Oracle a Data Lake. Además, se muestra el proceso para la construcción de la fuente Cliente crédito, las cual se formará a partir de la integración de las fuentes PV y LSA las cuales están almacenada en formato Parquet. Finalmente se muestra como estas cuatro fuentes mencionadas RCC, SUNAT, Cliente Riesgos y Cliente Crédito las cuales se integrarán generando la fuente Cliente Pyme.

b) Estructura de la tabla final del modelo

Se realizo la creación de la tabla en HIVE de tipo external para permitirle integridad y persistencia a los datos

```
DROP TABLE IF EXISTS  ${hiveconf:AMBIENTE}_clientepyme.pyme;
CREATE EXTERNAL TABLE ${hiveconf:AMBIENTE}_clientepyme.pyme
(
     CODCLAVE                       VARCHAR(128),
     CODPARTY                       VARCHAR(128),
     CODCOMPU                       VARCHAR(30),
     TIPIDENTIFICACION              CHAR(20),
     DESIDENTIFICACION              VARCHAR(256),
     CODSEG                         VARCHAR(30),
     DESSEG                         VARCHAR(256),
     MESESDEUDA                     INT,
     FLGRCC                         CHAR(1),
     FLGCLI                         CHAR(1),
     FLGCLIPYME                     CHAR(1),
     FLGCREDITO                     CHAR(1),
     FLGSUNAT                       CHAR(1),
     FLGPROSPECTOPYME               CHAR(1),
     FECRUTINA                      TIMESTAMP,
     FECACTUALIZACIONREGISTRO       TIMESTAMP
)
PARTITIONED BY (CODMES INT)
STORED AS PARQUET
LOCATION '/${hiveconf:AMBIENTE}/empresa/clientepyme/pyme'
TBLPROPERTIES ('parquet.compression'='SNAPPY');
```

Figura 30. Creación de la tabla Pyme
Nota: Elaboración propia

c) Script final

Parte del proceso de desarrollo final del proceso de integración Pyspark se presenta a continuación.

```
dfDetalleRCC = dfDetalleRCCAux.select(col('CODCLAVEUNICOCLI').alias('CODCLAVEUNICOCLI_RCC'), col('CTD'))
dfDetalleRCCTp = dfDetalleRCCAux.select(col('CODCLAVEUNICOCLI').alias('CODCLAVEUNICOCLI_RCC'), col('TIPPARTYIDENTIFICACION_RCC'))
dfClienteRiesgos = dfClienteRiesgosAux.select(col('CODCLAVEUNICOCLI').alias('CODCLAVEUNICOCLI_CLI'), col('CODSUBSEGMENTO'))
dfClienteRiesgosTp = dfClienteRiesgosAux.select(col('CODCLAVEUNICOCLI').alias('CODCLAVEUNICOCLI_CLI'),col('CODCLAVEPARTYCLI'), col('CODSUBSEGMENTO'), col('DESSUBSEGMENTO'), col('CODINTERNOCOMPUTACIONAL'), col('TIPPARTYIDENTIFICACION_CLI'))
dfUniversoPymePortaFolio = dfUniversoPymePortaFolioAux.select(col('CODCLAVEUNICOCLI').alias('CODCLAVEUNICOCLI_PORT'), col('CODMES'))
dfUniversoPymePortafolioTp = dfUniversoPymePortaFolioAux.select(col('CODCLAVEUNICOCLI').alias('CODCLAVEUNICOCLI_PORT'),col('TIPPARTYIDENTIFICACION_PORT'))
dfEmpresaSunat = dfEmpresaSunatAux.select(col('CODCLAVEUNICOCLI').alias('CODCLAVEUNICOCLI_SUN'))
dfEmpresaSunatTp = dfEmpresaSunatAux.select(col('CODCLAVEUNICOCLI').alias('CODCLAVEUNICOCLI_SUN'), col('TIPPARTYIDENTIFICACION_SNT'))


dfUnion = dfDetalleRCC.join(dfClienteRiesgos, dfDetalleRCC.CODCLAVEUNICOCLI_RCC == dfClienteRiesgos.CODCLAVEUNICOCLI_CLI, how='full')\
                      .join(dfUniversoPymePortaFolio, dfDetalleRCC.CODCLAVEUNICOCLI_RCC == dfUniversoPymePortaFolio.CODCLAVEUNICOCLI_PORT, how='full')\
                      .join(dfEmpresaSunat, dfDetalleRCC.CODCLAVEUNICOCLI_RCC == dfEmpresaSunat.CODCLAVEUNICOCLI_SUN, how='full')


dfFlagFuente = dfUnion.withColumn('CODCLAVEUNICOCLI', coalesce(coalesce(coalesce(col('CODCLAVEUNICOCLI_RCC'), col('CODCLAVEUNICOCLI_CLI')), col('CODCLAVEUNICOCLI_PORT')), col('CODCLAVEUNICOCLI_SUN')))\
    .withColumn('CTDMESDEUDALINEAPYMEU12', coalesce(col('CTD'), lit(0)))\
    .withColumn('FLGRCCDEUDALINEAPYMEU12', when(col('CODCLAVEUNICOCLI_RCC').isNotNull(), lit(1)).otherwise(lit(0)))\
    .withColumn('FLGCLI', when(col('CODCLAVEUNICOCLI_CLI').isNotNull(), lit(1)).otherwise(lit(0)))\
    .withColumn('FLGPORTAFOLIOCREDITOPYME', when(col('CODCLAVEUNICOCLI_PORT').isNotNull(), lit(1)).otherwise(lit(0)))\
    .withColumn('FLGCONTRIBUYENTESUNAT', when(col('CODCLAVEUNICOCLI_SUN').isNotNull(), lit(1)).otherwise(lit(0)))


dfFuentes = dfFlagFuente.select(col('CODCLAVEUNICOCLI'), col('CTDMESDEUDALINEAPYMEU12'), col('FLGRCCDEUDALINEAPYMEU12'), col('FLGCLI'), col('FLGPORTAFOLIOCREDITOPYME'), col('FLGCONTRIBUYENTESUNAT'))\
    .groupBy(col('CODCLAVEUNICOCLI')).agg(max('CTDMESDEUDALINEAPYMEU12').alias('CTDMESDEUDALINEAPYMEU12'),
                                          max('FLGRCCDEUDALINEAPYMEU12').alias('FLGRCCDEUDALINEAPYMEU12'),
                                          max('FLGCLI').alias('FLGCLI'),
                                          max('FLGPORTAFOLIOCREDITOPYME').alias('FLGPORTAFOLIOCREDITOPYME'),
                                          max('FLGCONTRIBUYENTESUNAT').alias('FLGCONTRIBUYENTESUNAT'))


dfaddSnt = dfFuentes.join(dfEmpresaSunatTp, dfFuentes.CODCLAVEUNICOCLI == dfEmpresaSunatTp.CODCLAVEUNICOCLI_SUN, how='left')
dfaddSntPort = dfaddSnt.join(dfUniversoPymePortafolioTp, dfaddSnt.CODCLAVEUNICOCLI == dfUniversoPymePortafolioTp.CODCLAVEUNICOCLI_PORT, how='left')
dfaddSntPortRcc = dfaddSntPort.join(dfDetalleRCCTp, dfaddSntPort.CODCLAVEUNICOCLI == dfDetalleRCCTp.CODCLAVEUNICOCLI_RCC, how='left')
dfaddSntPortRcc = dfaddSntPortRcc.drop('CODCLAVEUNICOCLI_SUN','CODCLAVEUNICOCLI_PORT','CODCLAVEUNICOCLI_RCC')


dfAgregacion = dfaddSntPortRcc.join(dfClienteRiesgosTp, dfaddSntPortRcc.CODCLAVEUNICOCLI == dfClienteRiesgosTp.CODCLAVEUNICOCLI_CLI, how='left')
dfAgregacion = dfAgregacion.drop('CODCLAVEUNICOCLI_CLI')


dfPymeUniverso = dfAgregacion.withColumn('TIPPARTYIDENTIFICACION', coalesce(coalesce(coalesce(col('TIPPARTYIDENTIFICACION_SNT'), col('TIPPARTYIDENTIFICACION_PORT')), col('TIPPARTYIDENTIFICACION_CLI')), col('TIPPARTYIDENTIFICACION_RCC')))
```

Figura 31. Script Pyspark Pyme
Nota: Elaboración propia

d) Resultados del modelo de clientes pyme

La tabla final del proceso con la información de los Flags identificadores por tipo de fuente.

Tabla 13

Resultados del modelo Pyme

	codclave	codparty	codcompu	tipidentificacion	desidentificacion	codseg	desseg		mesesdeuda	flgrcc	flgcli	flgclipyme	flgcredito	flgsunat	flgprospectopyme	fecrutina	fecactualizacionregistro
1	796572258ab	a60d6e9974	5523	6	REGISTRO UNICO CONTRIBUYENTE	TRE	NO BANCO	NORMAL	2	1	1	0	0	1	NULL	2021-09-08 00:00:00.0	2021-10-12 00:00:00.0
2	79701483b1	1e7863c3ae	0939	6	REGISTRO UNICO CONTRIBUYENTE	P4K	PEQ.EMPRESA	NORMAL	12	1	1	1	0	1	NULL	2021-09-08 00:00:00.0	2021-10-12 00:00:00.0
3	797bd65999	81bb9e2bce	0364	6	REGISTRO UNICO CONTRIBUYENTE	P4K	PEQ.EMPRESA	NORMAL	12	1	1	1	0	1	NULL	2021-09-08 00:00:00.0	2021-10-12 00:00:00.0
4	799b08c3e9	a51a23b5fe	0947	6	REGISTRO UNICO CONTRIBUYENTE	P4K	PEQ.EMPRESA	NORMAL	3	1	1	1	0	1	NULL	2021-09-08 00:00:00.0	2021-10-12 00:00:00.0
5	79a6d8eb54	fac73afda6	0941	6	REGISTRO UNICO CONTRIBUYENTE	TRE	NO BANCO	NORMAL	0	0	1	0	0	1	NULL	2021-09-08 00:00:00.0	2021-10-12 00:00:00.0
6	79b18dfc3c	fb081cd6a9	3758	6	REGISTRO UNICO CONTRIBUYENTE	TRE	NO BANCO	NORMAL	0	0	1	0	0	1	NULL	2021-09-08 00:00:00.0	2021-10-12 00:00:00.0
7	79b38f933f	dbb3780811	4617	6	REGISTRO UNICO CONTRIBUYENTE	TRE	NO BANCO	NORMAL	0	0	1	0	0	1	NULL	2021-09-08 00:00:00.0	2021-10-12 00:00:00.0
8	79c1871d81	2b61279920	0942	6	REGISTRO UNICO CONTRIBUYENTE	P4K	PEQ.EMPRESA	NORMAL	0	0	1	1	0	1	NULL	2021-09-08 00:00:00.0	2021-10-12 00:00:00.0
9	79c5b83241	e370a79ff3	0941	6	REGISTRO UNICO CONTRIBUYENTE	TRE	NO BANCO	NORMAL	0	0	1	0	0	1	NULL	2021-09-08 00:00:00.0	2021-10-12 00:00:00.0

Nota. Elaboración propia

Mediante estos flags presentados el cliente final tomara la decisión de si el cliente que interactúa en los cuatro módulos es apto o no para recibir una línea de crédito.

Sprint 6

Planning sprint 6

El negocio tiene la necesidad de productivizar los procesos actuales, por lo cual se realizará el despliegue de toda la solución utilizando las herramientas que dispone la organización.

Tabla 14
Actividades del sprint 6

Ticket	Actividad	Responsable
TAREA-816	Despliegue y pase a producción	Data Engineer

Nota. Elaboración propia

Inicio sprint 6

a) Puesta en producción

Actualmente se cuenta con un proceso de Integración continua que permite realizar el despliegue de todos los componentes creados, y los que permitirán disponibilizar en corto tiempo la solución, los principales elementos utilizados son los siguientes:

Tabla 15
Herramientas de integración continua

Bitbucket	**Jenkins**	**Jira**	**Datastage**
			
La cual dispondrá de tres ramas. Desarrollo, certificación y producción, las cuales se desplegarán en los tres ambientes	Es el encargado de recrear el proceso y el despliegue de los archivos Hive y Python en el entorno data Lake	Es el encargado de crear un ticket en el cual se detallarán las ordenes que debe ejecutar un operador para poder en producción una solución	Es el encargado de ejecutar el proceso Python en el entorno de desarrollo, certificación o producción

Nota. Elaboración propia

Etapa de validación

Durante estos últimos tres meses se ha llevado a cabo una marcha blanca en las cuales no han ocurrido caídas por ejecución del proceso ni por temas de espacio de memoria, también

desde el punto de vista del cliente final se ha percibido que los tiempos de ejecución se han reducido significativamente en comparación con los tiempos proporcionados por el procesamiento en Oracle.

Etapa de adaptación

Con la puesta en producción, se ha capacitado al usuario para el uso del entorno Data Lake. Muchas veces el usuario es resistente al cambio, pero mostrándole los beneficios de explotar la información en el Data Lake el usuario ha ido mostrando interés y obteniendo buenos resultados, lo cual ha beneficiado al área y empresa, proporcionándole datos con valor y calidad.

Expansión de procesos

Esta iniciativa ha permitido que el área en mención, que dispone actualmente de muchos procesos analíticos, apueste por una transformación y refactorización de muchos procesos en cartera, mediante esta metodología de trabajo se ha podido dar solución a muchos problemas y modelos actuales que se trabajan en el área, esto ha permitido la disminución de tiempos de procesamiento y brindar información con valor en beneficio del área. Por último, esto ha dado la iniciativa de poder integrar a dos áreas del negocio como son Finanzas y For Analytics quienes actualmente están en el proceso de aprendizaje de los procesos que maneja el área de Migración.

Metodología de trabajo

La migración de esta solución ha permitido que nuevas áreas se vean interesadas en el modo de trabajo, por lo que se ha requerido realizar un trabajo en conjunto con otros equipos y poder poner en marcha nuevas migraciones de analítica avanzada. Cabe resaltar que la solución planteada en este proyecto tiene un homólogo en el área de negocios, por lo que la solución modularizada permitirá ser adoptada para el área de negocios cambiando las fuentes de información y respetando el proceso actual.

Gobierno de los datos

Disponibilizar la información de tablas y campos, así como documentación de reglas de negocio a los custodios de la información. Quienes estarán a cargo de consolidar la

información siguiendo los lineamientos del negocio. Esto garantiza la reutilización de tablas y campos a procesos futuros que serán migrados y que requieran de esta información, lo cual permitirá evitar la redundancia de datos.

Iniciativas de framework de calidad

Como parte de poder entregar datos con valor se ha formado un equipo en el área el cual realizará un motor de calidad siguiendo la metodología utilizada y el cual permitirá poder controlar y monitorear la calidad de los datos en el plan de migración.

3.3 EVALUACIÓN ECONOMICA

Para realizar este análisis de esta solución de Big Data, lo enfocaremos desde el punto costo-beneficio

3.3.1 EVALUACIÓN COSTO

En el siguiente grafico se realiza un resumen de los costos e inversión en capital humano que ha demandado el proyecto de migración para el área de riesgos durante los 10 meses que lleva el proyecto.

Tabla 16
Inversión en capital humano

Area de Riesgos - 2021 Analisis/Implementacion/Pruebas/Pase a Produccion													
Rol	Puesto	N°	Marzo	Abril	Mayo	Junio	Julio	Agosto	Setiembre	Octubre	Noviembre	Diciembre	Total
Gestion	Producto Owner	1	12000	12000	12000	12000	12000	12000	12000	12000	12000	12000	120000
Gestion	Chapter Leader	1	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	100000
Gestion	Scrum master	1	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	50000
Analisis	Modeladores	2	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	120000
Ingesta	Data Engineer	4	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	280000
Total													670000

Los participantes en el proyecto actualmente son 9 personas los cuales han participado continuamente en el proyecto desde sus inicios de marzo del 2021 hasta la actualidad, generando una inversión de 670 mil soles en estos primeros 10 meses del proyecto.

Como se puede apreciar el equipo está formado por el Producto Owner quien es el encargado del proyecto, el chapter leader quien facilita las herramientas y guía al equipo, el Scrum máster quien tiene el rol de ser un facilitador, así como los modeladores quienes y Data Engineer encargados de todo el proceso de análisis e ingesta de datos.

3.3.1 BENEFICIO PARA LA ORGANIZACION

Este proyecto que inicio como un piloto en el área de riesgos ha permitido que dos áreas externas se encuentren interesadas en el proceso de migración y es un punto de inicio para poder realizar la migración de todos los procesos de la entidad, los cuales requieran de procesamiento con grandes volúmenes de información.

La información se encuentra centralizada en el Data Lake por lo cual muchos usuarios del negocio tendrán acceso a esta información centralizada, apoyándolos en sus tareas diarias y en una correcta toma de decisiones y gobierno de datos.

El proyecto de migración ha permitido utilizar las tecnologías de Hadoop y el almacenamiento distribuido en HDFS lo cual ha permitido el acceso simultaneo de muchos usuarios brindándole beneficios a la organización.

El procesamiento con tecnologías Big data y el uso de Spark desde el punto de vista del usuario final ha permitido obtener soluciones de manera correcta, ordenada y rápida en comparación a los procesos tradicionales realizados actualmente en sus entornos de Bases de datos Oracle.

Fue muy importante el aporte y apoyo por parte de los expertos en los modelos desarrollados actualmente por el negocio, pues han permitido brindar apoyo y aprobación a la refactorización de los procesos y a la mejora continua.

El proceso actual desde el punto de vista del cliente otorga gran valor al negocio, pues se obtiene la información de manera fácil, ordenada y con datos confiables otorgándole valor al negocio.

El proyecto en mención a permitido resolver un problema que se presentaba todos los cierres de mes y ha permitido aportar mediante una solución mejoras sustanciales en el proceso de entrega de datos a la organización.

Se puede afirmar que este proceso de migración ha otorgado a los procesos actuales del área de riesgos velocidad, mantenibilidad, escalabilidad y refactorización de estos. Además, ha permitido despertar el interés de otras áreas para migrar sus procesos que actualmente padecen de esta problemática, ya que, si bien las bases de datos han permitido resolver muchos problemas actuales en la organización, existen problemas y enfoques que requieren de mucha cantidad de datos y un alto grado de procesamiento.

CAPÍTULO IV
REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

Si bien esta solución le permitió al expositor de este trabajo poder consolidar sus conocimientos utilizando el framework Hadoop en Data Lake OnPremise, existe una resistencia al cambio por parte del usuario final lo cual no ha permitido integrar otros procesos que puedan aportar un mayor valor al proceso actual de los modelos desarrollados.

Por otra parte, se realizaron la revisión de los scripts los cuales en muchos casos no tenían al autor del proceso en la organización por lo cual fue más complicado poder mapear los procesos, pero a pesar de ellos se pudieron superar en corto tiempo.

Por otra parte, al trabajar bajo una metodología ágil (scrum) y poder brindar una solución en el corto plazo solo ha permitido generar documentaciones graficas del proceso mas no ha permitido documentar las buenas prácticas, estándares, experiencias y lecciones aprendidas, los cuales han quedado en la mente de los miembros del equipo lo cual podría impactar en los futuros desarrollos.

Hubo fases durante el proceso de migración donde la infraestructura del negocio sufrió caídas por la concurrencia, lo cual generaba retrasos en los desarrollos, pues los Data Enginner dependían de esta infraestructura. A futuro se podría resolver este impedimento migrando a la nube.

La curva de aprendizaje de los nuevos integrantes del equipo impacto en el desarrollo del proyecto, esto se podría mejorar con un plan de capacitación para elevar el nivel de productividad.

CAPITULO V
CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Como parte de la solución se logró realizar la migración aplicando las buenas prácticas del Marco de trabajo Big Data, las cuales han permitido mejorar los problemas del área brindándole con esta solución los siguientes beneficios: procesamiento (SPARK) y almacenamiento (HDFS) distribuido, Alta disponibilidad, encriptación de datos, Gobierno, patrones de diseño, escalabilidad, persistencia y paralelización en sus procesos utilizando herramientas tecnológicas como Hadoop, Hive y Spark.
2. Se logro diseñar una solución para los clientes Pyme que permita poder ir integrando progresivamente nuevas fuentes de información lo cual no era viable en una base de datos convencional. Además, que esta propuesta ha permitido apostar por implementar una nueva solución idéntica a la expuesta para el grupo de clientes de negocio
3. Se logro refactorizar las tablas de Oracle brindándole un modelo de datos en data lake el cual ha permitió a los usuarios finales tomar buenas decisiones.
4. El uso de la metodología scrum ha permitido poder obtener resultados y otorgar valor al negocio en el corto plazo. Las reuniones y retroalimentación han sido fundamentales para el aprendizaje continuo y el trabajo en equipo. Logrando crear el modelo en 6 iteraciones.

RECOMENDACIONES

1. Dentro del Sprint se debería tomar en cuenta la documentación de los desarrollos realizados, pues estos servirán como conocimiento a los nuevos integrantes del equipo y al Líder del proyecto.

2. La organización se encuentra en una etapa de maduración por lo cual considero que es una buena estrategia el de tener sus procesos OnPremise, una vez alcanzado el grado de maduración que implica tener buenas prácticas, lineamientos bien definidos y un correcto gobierno de los datos, pueda tentar la posibilidad de realizar la migración a la nube, la cual sin duda causara grandes beneficios a la organización para el desarrollo de sus modelos como se ve actualmente en muchas organizaciones.

3. Los usuarios finales deberían involucrarse progresivamente en el uso y adopción de sus tareas interactuando con el Data Lake dejando de lado sus dependencias con Oracle.

REFERENCIAS BIBLIOGRAFICAS

Cravero, A., Sepúlveda, S., & Muñoz, L. (2020). Big Data Architectures for the Climate Change Analysis: A Systematic Mapping Study. *IEEE LATIN AMERICA TRANSACTIONS*, 1793-1794.

Fernandez, Y., Gutierrez, M., & Palomo, R. (2019). *¿Cómo percibe la banca cooperativa el impacto de la transformación digital?* Madrid: CIRIEC.

Honar, H., & Rashid, M. (2021). IoT Big Data provenance scheme using blockchain on Hadoop ecosystem. *Journal of Big Data*, 7.

INDRA. (2021). *Compañia Global de Tecnologia y Consultoria*. Obtenido de https://www.indracompany.com/: https://www.indracompany.com/es/indra

Meehan, J., Tatbul, N., & Du, J. (2017). Data Ingestion for the Connected World. *Creative Commons*, 1.

Mision y Vision. (2021). Obtenido de https://logingroup.wordpress.com/indra/.

Nexla. (2018). *An Introduction to Big Data Formats.* Millbrae: Nexla.

Peñaloza, M. (2010). *Teoria de las decisiones.* Cochabamba: Perspectivas.

Perez, W. (2020). Propuesta de una metodología para el proceso de Migración de Datos en. *Universidad de las Fueras Armadas ESPE*, 2.

Pwint, P., & Zhao, S. (2018). Data lake: a new ideology in big data era. *ITM*, 1-2.

Rodriguez, Y., & Pinto, M. (2017). Modelo de uso de información para la toma de decisiones estratégicas en organizaciones de información. 60.

Roman, N. U. (2021). Big Data. *RISI*, 13.

Seliya, N., Abdollah, A., & Taghi, K. (2021). A literature review on one-class classification and its potential applications in big data. *Journal of Big Data*, 1-2.

Shaikh, E., Mohiuddin, I., Alufaisan, Y., & Nahvi, I. (2019). Apache Spark: A Big Data Processing Engine. *MENACOMM*, 1-3.

GLOSARIO DE TERMINOS

- **Big Data**: Es un marco de trabajo que permite el procesamiento de grandes volúmenes de datos, las cuales cuentan con diferentes tipos de estructuras (estructurado, semiestructurada y no estructurado) que puedan variar con el tiempo, que puedan generar a grandes velocidades y puedan generar valor al negocio.

- **Entidad Bancaria**: Es una institución del rubro financiero que se encarga de la administración de dinero de personas naturales y jurídicas.

- **Procesos de migración**: Transferir los datos de un sistema o software a otro.

- **Toma de decisiones**: Es un proceso que atraviesan los seres humanos cuando se presentan conflictos y deben elegir entre distintas opciones para buscar la mejor solución.

- **Data Lake**: En español conocido como lago de datos, es un repositorio centralizado que nos permite almacenar datos de tipo estructurado, no estructurado y semi estructurado a cualquier escala.

- **Refactorizar:** Es una técnica de ingeniería que consiste en reestructurar un código alterando su estructura interna sin cambiar el comportamiento de este.

- **Hadoop:** Hadoop es una estructura para almacenar datos, proporciona procesamiento y almacenamiento de manera distribuida, procesando trabajos concurrentes ilimitadamente.

- **Hive:** Forma parte del ecosistema de Hadoop y es utilizado para gestionar los datos mediante consultas de tipo HQL las cuales son muy similares a las consultas SQL tradicionales.

- **Spark:** Es un framework que permite realizar procesamiento de manera distribuida. Esta diseñado para soportar grandes cantidades de trabajo y puede ser utilizado por diferentes lenguajes de programación como Python, Scala y R.

- **HDFS:** Es el componente principal del ecosistema Hadoop conocido como el sistema de archivos distribuidos de Hadoop. Es el encargado de almacenar los datos de tipo estructurado, semi estructurad y no estructurado como pueden ser imágenes, videos y datos de sensores.

- **Datastage:** Forma parte de la suite de herramientas de IBM y es utilizado para realizar procesos ETL el cual es usado por las empresas para sus procesos de extracción, transformación y carga de datos

- **Procesamiento distribuido:** Es la forma de conectar varias computadoras en un tipo de red de comunicaciones logrando que una tarea pueda ser ejecutada dividiendo en pequeñas tareas que puedan ser resueltas por cada una de las computadoras que forman el conjunto.

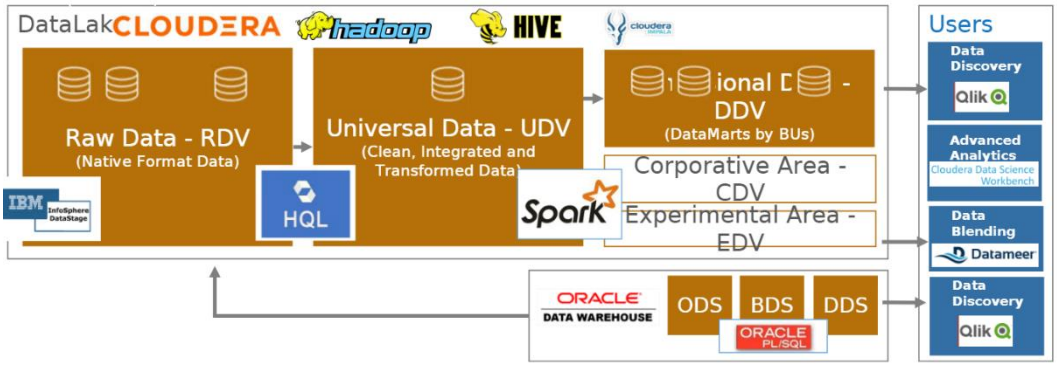
- **Disponibilizar:** Es un término muy utilizado en el ámbito profesional de tecnología y viene de la palabra disponible y se refiere al acto de poner a disposición algún elemento o objeto a alguna persona, grupo o sistema.

ANEXOS

Anexo 1: Arquitectura de Datos Unificado de Datos – Tecnologías

Anexo 2: Procesos de migración de Oracle a Data Lake

Anexo 1: Arquitectura de Datos Unificado de Datos – Tecnologías

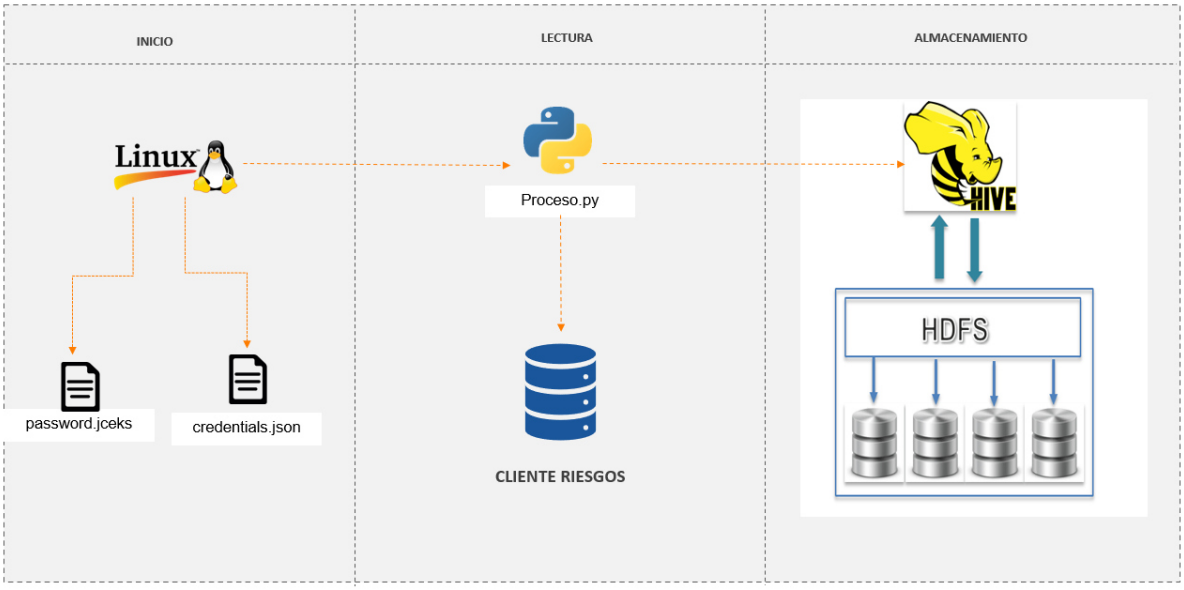


Descripción de la arquitectura de Datos:

- La infraestructura Data Lake está bajo Cloudera contando con cuatros capas las cuales son RDV, UDV, DDV, y EDV
- RDV: Raw Data Vault, es aquel lugar donde se almacenará los datos de manera cruda es un formato nativo y los datos serán llevados desde Oracle a Data Lake de dos maneras: Utilizando la herramienta Datastage, actualmente se están realizando las cargas utilizando Spark
- UDV: Universal Data Vault, es aquella capa donde se almacenarán los datos limpios, íntegros y transformados, la tecnología utilizada será la de HIVE, la información será llevada de la capa RDV a UDV mediante Scripts de tipo hql.
- DDV: Dimensional Data Vault, es aquella capa donde se realizará los modelos y Datamarts que requiera la organización para generar reportes y presentaciones que otorguen valor a la organización, todas estas soluciones se realizaran utilizando scripts Pyspark y Scala.
- CDV/EDV: Corporative y Experimental Data Vault, es aquella capa donde trabajaran modelos experimentales, los data scientist utilizan esta zona para experimentar y generar nuevos modelos que puedan otorgar valor al negocio. Una vez consolidados estos modelos pueden ser migrados a la capa DDV para su explotación a nivel organizacional. Los desarrollos de estos modelos se realizan utilizando scripts Pyspark y Scala

- Visualización: Es la capa donde el cliente final genera sus reportes y análisis de datos. Estas visualizaciones y cálculos de KPIs se realizan utilizando la herramienta Qliksense.

Anexo 2: Procesos de migración de Oracle a Data Lake



El siguiente grafico representa el flujo que realiza la organización para la carga de datos de Oracle a Datalake, como caso de ejemplo se tomó la tabla Cliente Riesgos.

Los pasos para realizar el proceso de migración son los siguientes:

- Password.jceks: Se crea este archivo con la finalidad de poder encryptar la contraseña Oracle del usuario
- Credentials.json: Se utiliza este archivo para almacenar las cadenas de conexión de Oracle, IP y la matricula del usuario
- Tabla Oracle: Es la tabla que se encuentra disponible para ser migrada y la cual será consumida por el proceso Pyspark
- Proceso.py: Archivo de procesamiento Pyspark el cual tendrá como objetivo conectarse a Oracle mediante una consulta SQL
- Hive/HDFS: La información de la tabla Oracle migrada se almacenará en una tabla HIVE y los datos físicos de la tabla en formato parquet en HDFS.
- Ejecución: La ejecución de este proceso se realizará por consola llamando al archivo pyspark mediante una sentencia spark2-submit, la cual ejecutará el archivo en el entorno Lake y realizará el proceso de Ingesta de datos.

Mediante estos pasos detallados en el grafico se realizará el proceso de ingesta de datos de Oracle a Data Lake.