



Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería Electrónica y Eléctrica
Escuela Profesional de Ingeniería de Telecomunicaciones

Beneficios de la aplicación de Big Data en el Proyecto

Core Assurance

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniera de
Telecomunicaciones

AUTOR

Elizabeth Delia MENDOZA GUTIERREZ

ASESOR

Mg. Daniel DIAZ ATAUCURI

Lima, Perú

2023



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Mendoza, E. (2023). *Beneficios de la aplicación de Big Data en el Proyecto Core Assurance*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería Electrónica y Eléctrica, Escuela Profesional de Ingeniería de Telecomunicaciones]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Elizabeth Delia Mendoza Gutierrez
Tipo de documento de identidad	DNI
Número de documento de identidad	70105021
URL de ORCID	https://orcid.org/0009-0003-9666-1128
Datos de asesor	
Nombres y apellidos	Daniel Diaz Ataucuri
Tipo de documento de identidad	DNI
Número de documento de identidad	07139361
URL de ORCID	https://orcid.org/0000-0001-5747-2795
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Milton Rios Julcapoma
Tipo de documento	DNI
Número de documento de identidad	08835538
Miembro del jurado 1	
Nombres y apellidos	Roberto Florentino Unsihuay Tovar
Tipo de documento	DNI
Número de documento de identidad	19944586
Miembro del jurado 2	
Nombres y apellidos	Jose Carlos Hurtado Rantes
Tipo de documento	DNI
Número de documento de identidad	08558088
Datos de investigación	
Línea de investigación	C.0.3.23 Administración de Base de Datos
Grupo de investigación	No Aplica
Agencia de financiamiento	No Aplica

Ubicación geográfica de la investigación	Edificio: Platinum Plaza I País: Perú Departamento: Lima Provincia: Lima Distrito: San Isidro Calle: C. Dean Valdivia 148 Latitud: -12.093480100627184 Longitud: -77.02760447264185
Año o rango de años en que se realizó la investigación	Agosto 2021 - Mayo 2023
URL de disciplinas OCDE	Ingeniería, Tecnología https://purl.org/pe-repo/ocde/ford#2.00.00 Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE INGENIERÍA ELECTRÓNICA Y ELÉCTRICA
Teléfono 619-7000 Anexo 4226
Calle Germán Amezaga 375 – Lima 1 – Perú



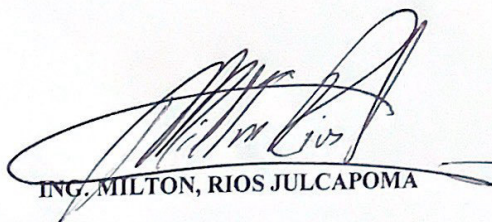
ACTA DE SUSTENTACIÓN DE TRABAJO DE SUFICIENCIA PROFESIONAL Nº 085/FIEE-CTGT/2023

Los suscritos Miembros del Jurado, docentes permanentes de las Escuelas Profesionales de la Facultad de Ingeniería Electrónica y Eléctrica, reunidos en la fecha 10 de noviembre del 2023, como Presidente de Jurado el **ING. MILTON, RIOS JULCAPOMA**, integrado por el Miembro de Jurado el, **ING. JOSE CARLOS, HURTADO RANTES**, Miembro de Jurado **ING. ROBERTO FLORENTINO, UNSIHUAY TOVAR** y Miembro Asesor el **MG. DANIEL, DIAZ ATAUCURI**.

Después de escuchar la Sustentación de Trabajo de Suficiencia Profesional del **Bach. ELIZABETH DELIA MENDOZA GUTIERREZ** con código Nº 16190266 que para optar el Título Profesional de Ingeniera de Telecomunicaciones sustentó el Trabajo de Suficiencia Profesional titulado **"BENEFICIOS DE LA APLICACIÓN DE BIG DATA EN EL PROYECTO CORE ASSURANCE"**.

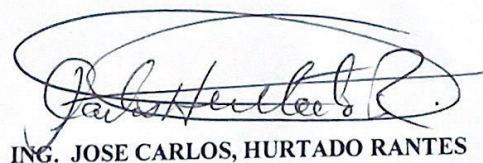
El jurado examinador procedió a formular las preguntas reglamentarias y, luego de una deliberación en privado, decidió aprobar otorgándole el calificativo *diecisiete (17)*

Ciudad Universitaria, 10 de noviembre del 2023



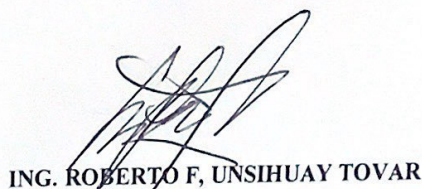
ING. MILTON, RIOS JULCAPOMA

Presidente de Jurado



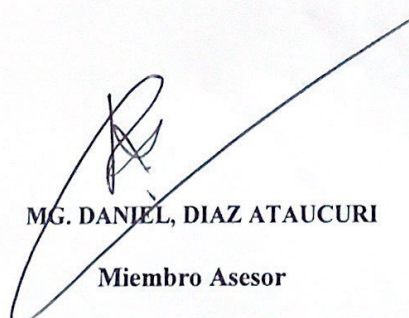
ING. JOSE CARLOS, HURTADO RANTES

Miembro Jurado



ING. ROBERTO F, UNSIHUAY TOVAR

Miembro de Jurado



MG. DANIEL, DIAZ ATAUCURI

Miembro Asesor



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Vicerrectorado de Investigación y Posgrado



CERTIFICADO DE SIMILITUD

Yo Daniel Diaz Ataucuri, en mi condición de asesor acreditado con el Acta de Sustentación de trabajo de Suficiencia Profesional N°085/CTGT-EPIE/2023 del trabajo de suficiencia profesional cuyo título es: **BENEFICIOS DE LA APLICACIÓN DE BIG DATA EN EL PROYECTO CORE ASSURANCE**, presentado por la bachiller Elizabeth Delia Mendoza Gutierrez para optar al título profesional de Ingeniero de Telecomunicaciones CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante el software de similitud textual, el documento evaluado cuenta con el porcentaje de 11% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado correspondiente.

Firma del Asesor

DNI: 07139361

Nombres y apellidos del asesor:
Mg. Daniel Diaz Ataucuri

DEDICATORIA

Este trabajo está dedicado a Dios, mis padres, mis hermanos y mis sobrinas por el gran apoyo que me han brindado en toda mi carrera profesional. Siempre recordaré los buenos consejos adquiridos por cada una de las personas que me brindo algún comentario para mejorar como persona. Seguiré con las enseñanzas de mis padres y tratare de lograr grandes cosas.

AGRADECIMIENTOS

Agradezco a mi asesor Mg. Daniel Díaz Ataucuri por el apoyo brindado en todo este proceso de realización de mi Trabajo de Suficiencia Profesional, también a mis padres por el apoyo y los buenos amigos para poder concluir con este trabajo.

RESUMEN

El Trabajo de Suficiencia Profesional titulado "Beneficios de la Aplicación de Big Data en el Proyecto Core Assurance" se enfoca en el modelo de solución de negocio implementado en un banco específico. El proyecto Core Assurance emplea el enfoque de Big Data, siguiendo lineamientos del banco para asegurar calidad y evitar duplicidad de datos. Se exploran conceptos de Big Data y se detalla el proceso paso a paso de la revisión de información de diversos orígenes.

Este trabajo destaca la importancia de seguir pautas para la gestión de fuentes de información de diversos orígenes en el contexto del Big Data bancario para obtener importantes beneficios en términos de fiabilidad, veracidad y consistencia de la información utilizada en el banco.

Por lo tanto, el trabajo resalta la aplicación exitosa del Big Data en el proyecto Core Assurance, enfatizando los lineamientos utilizados para garantizar la calidad y evitar la duplicidad de la información en el Big Data del banco. Se destaca los beneficios obtenidos al seguir de estos lineamientos y la importancia de contar con información confiable, veraz y congruente en el contexto del Big Data bancario.

Palabras clave: Big Data, Core Assurance, Lineamientos, veracidad, fiabilidad.

ABSTRACT

The Professional Sufficiency Paper entitled "Benefits of the Application of Big Data in the Core Assurance Project" focuses on the business solution model implemented in a specific bank. The Core Assurance project employs the Big Data approach, following the bank's guidelines to ensure quality and avoid data duplication. Big Data concepts are explored and the step-by-step process of reviewing information from various sources is detailed.

This paper highlights the importance of following guidelines for the management of information sources from various origins in the context of Big Data banking to obtain important benefits in terms of reliability, veracity and consistency of the information used in the bank.

Therefore, the paper highlights the successful application of Big Data in the Core Assurance project, emphasizing the guidelines used to ensure quality and avoid duplication of information in the bank's Big Data. It highlights the benefits obtained by following these guidelines and the importance of having dependable, truthful, and consistent information in the context of Big Data banking.

Keywords: Big Data, Core Assurance, guidelines, veracity, reliability.

TABLA DE CONTENIDO

RESUMEN	iii
ABSTRACT	iv
TABLA DE CONTENIDO	v
LISTA DE FIGURAS	vii
LISTA DE TABLAS	viii
CAPÍTULO I: INTRODUCCIÓN	1
CAPÍTULO II: INFORMACIÓN DEL LUGAR DONDE SE DESARROLLÓ LA ACTIVIDAD	2
2.1. Institución – Actividad que desarrolla:	2
2.2. Periodo de duración de la actividad:	2
2.3. Finalidad y objetivos de la entidad:	2
2.4. Razón social:	3
2.5. Dirección postal:	3
2.6. Correo electrónico del profesional a cargo.	3
CAPÍTULO III: DESCRIPCIÓN DE LA ACTIVIDAD	4
3.1. Organización de la actividad:	4
3.2. Finalidad y objetivos de la Actividad:	4
3.2.1. <i>Finalidad:</i>	5
3.2.2. <i>Objetivo general:</i>	5
3.2.3. <i>Objetivos específicos:</i>	5
3.3. Problemática	5
3.3.1. <i>Problema General</i>	5
3.3.2. <i>Problemas Específicos</i>	5
3.4. Metodología	6
3.4.1. <i>Bases teóricas</i>	6

3.4.2. Marco Conceptual	8
3.4.2.1. Almacenamiento de datos	8
3.4.2.2. Big Data	15
3.4.2.3. Conceptos asociados al flujo de Core Assurance	26
3.4.2.4. Alcance organizacional	28
3.4. Procedimiento	31
3.4.1. Origen de proyectos:	31
3.4.1.1. Fuente de datos	31
3.4.1.2. Tipos de proyectos (cliente)	32
3.4.2. Modelamiento de proyectos:	32
3.4.2.1. Recopilación y estructuración de proyectos	32
3.4.3. Recepción y evaluación de proyectos:	32
3.4.3.1. Ticket Entrada TDS:	33
3.4.3.2. Ticket de Procesamiento:	44
3.4.4. Aprobación y almacenamiento de proyectos	50
3.4.4.1. Aprobación de fuente:	50
3.4.4.2. Status de los tickets:	51
3.4.5. Gestión y Análisis de proyectos	52
3.4.5.1. Proceso en Datio (Big Data del banco):	52
3.4.5.2. Aprobación Core Assurance:	53
3.4.6. Resultado de la actividad	53
CAPÍTULO IV: CONCLUSIONES	60
4.1. Justificación:	60
4.1.1. Justificación práctica	60
4.1.2. Justificación tecnológica	61
4.1.3. Justificación social	61
4.1.4. Justificación económica	61
4.2. Descripción de la implementación	62
4.3. Conclusiones	62
CAPITULO V: RECOMENDACIONES	64
CAPITULO VI: BIBLIOGRAFÍA	65
CAPITULO VII: ANEXOS	69

LISTA DE FIGURAS

Figura 1: Teorema de CAP.....	11
Figura 2:Ejemplo de tablas complementarias.	20
Figura 3 :Ejemplo de tablas con estructuras iguales	21
Figura 4:Ejemplo de ficheros ad-hoc.....	22
Figura 5:Ejemplo de tablas con múltiples valores en 1 campos	22
Figura 6:Ejemplo de tablas con campos redundantes.....	23
Figura 7:Ejemplo de tablas con campos free-field.	24
Figura 8:Ejemplo partición de tablas	24
Figura 9: Base Única de Ingestas	27
Figura 10.Distribución de Core Assurance	29
Figura 11.Flujo E2E	30
Figura 12: Flujo de Entrada TDS	33
Figura 13 :Flujo de Asignación de Folio	35
Figura 14 :Flujo de Asignación de ID	36
Figura 15:Tipos de fuentes según el origen	39
Figura 16 :Flujo de Dictamen TDS	40
Figura 17: Flujo de Procesamiento.....	45
Figura 18: Flujo de Asignación de folio de Procesamiento.....	47
Figura 19: Flujo de Dictamen de Procesamiento.	50
Figura 20:Diagrama de barras # de vistas por pagina.....	55
Figura 21: Diagrama vistas a la página Site Core Data.....	56
<i>Figura 22: Diagrama de usuarios nuevos</i>	<i>56</i>
<i>Figura 23 :Diagrama de dictámenes por proyecto</i>	<i>57</i>
Figura 24: Diagrama de dictámenes por trimestre.....	58
<i>Figura 25: Diagrama de dictámenes por trimestre</i>	<i>58</i>
Figura 26 :Diagrama de resultados de valor.	61
Figura 27:Ticket creado en Jira de asignacion de Folio TDS.	69
Figura 28:Ticket creado en Jira de asignacion de ID TDS	70
Figura 29:Ticket creado en Jira de revision de dictamen TDS	70
Figura 30:Ticket creado en Jira de asignación de Folio Procesamiento	71
Figura 31:Ticket creado en Jira de revisión de procesamiento.	71
Figura 32:Ticket creado en Jira para reusos de fuentes ya existentes	72
Figura 33:Ticket creado en Jira para regularizar fuentes	72
Figura 34 :Análisis completo E2E de todas las áreas del banco.....	73

LISTA DE TABLAS

<i>Tabla 1:</i> Comparación entre las capacidades fundamentales de las bases de datos relacionales y las bases de datos NoSQL	12
Tabla 2. Correos de validación por parte del SO, DO y PO	31
Tabla 3: Status de los tickets Jira	52

CAPÍTULO I: INTRODUCCIÓN

En la actualidad el sector financiero está experimentando importantes cambios y se está acumulando de información por cada usuario que se afilia al banco ya sea adquiriendo alguna tarjeta, solicitando algún préstamo, etc.

En este contexto, la información que llega al banco puede ser por medios digitales, banca tradicional, por otras entidades que venden su información, etc. Esta información que llega al banco tiene un gran volumen de datos por lo cual esta entidad financiera genera proyectos internos para ofrecer a sus clientes las mejores soluciones bancarias, ayudarles a tomar las mejores decisiones financieras e impactar positivamente en su vida, y uno de ellos es el proyecto de **Core Assurance**, el cual tiene como objetivo analizar toda esta data que ingresa al banco mediante lineamientos para ser validados y que puedan aplicarse dentro del Big Data del banco.

El proyecto Core Assurance válida a nivel funcional cada fuente que ingresa en su flujo ya sean ingestas o procesamientos, todo mediante lineamientos generados por los mismos analistas de este proyecto. Con esto se genera que dentro del Big Data del banco solo ingrese información confiable, verídica y explotable para su uso.

En esta era digital es importante que las entidades financieras redefinen su propuesta de valor, centrándose en las necesidades reales de sus clientes, ayudándoles a tomar las mejores decisiones financieras a través de soluciones claras, transparentes e íntegras que les genere confianza y todo esto se logra gracias a la evaluación que hace el equipo de Core Assurance aplicando Big Data dentro de sus procesos.

CAPÍTULO II: INFORMACIÓN DEL LUGAR DONDE SE DESARROLLÓ LA ACTIVIDAD

2.1. Institución – Actividad que desarrolla:

En el presente trabajo de suficiencia profesional -TSP, se desarrolla la actividad de consultoría en un modelo de negocio aplicado a un banco del Perú en la cual se utiliza el Big Data en el respectivo proceso de información. Todo esto a través de la empresa Management Solutions que se trata de una firma internacional de consultoría para el asesoramiento de negocio, finanzas, etc.

2.2. Periodo de duración de la actividad:

2 año y 1 mes

2.3. Finalidad y objetivos de la entidad:

Crear propuestas de valor para sus clientes y comprometerse con su implantación efectiva, esto exige una fuerte cultura de compromiso y excelencia, y constituye la principal característica de nuestros servicios. Además de, ser eficaces en cuanto a la aplicación del Big Data, cumpliendo con los lineamientos de la entidad, gestionando y analizando de forma correcta la información con el empleo de este para el modelo de negocio.

2.4. Razón social:

MANAGEMENT SOLUTIONS PERU S.A.C.

2.5. Dirección postal:

Coronel Andrés Reyes 437 Centro Empresarial Platinum San Isidro, Calle 27,
Cercado de Lima 15046, Perú.

2.6. Correo electrónico del profesional a cargo.

juan.jose.aspilcueta@ms-peru.com
jose.maria.ruiz.cabello@ms-colombia.com

CAPÍTULO III: DESCRIPCIÓN DE LA ACTIVIDAD

3.1. Organización de la actividad:

Para lograr los objetivos y resolver la problemática identificada, en el presente TSP se analizará y evaluará el flujo de datos que se tiene en este proyecto Core Assurance aplicando Big Data en todo el proceso y siguiendo los lineamientos definidos por el banco. Las actividades para el desarrollo del trabajo se organizaron de la siguiente manera:

- Utilizando Big Data se realiza el modelamiento de la información de cada proyecto.
- Se reciben los dictámenes de los proyectos de cada cliente para su evaluación según los tipos de datos que se soliciten.
- Se analiza los proyectos en base a los lineamientos de manejo de información establecidos por la entidad financiera (banco) a través de tickets usando el software JIRA propio de la entidad financiera.
- Los tickets revisados y/o analizados son de dos tipos: ticket TDS (ingesta de datos) y procesamiento (usan los TDS como entradas para procesar información).
- Seguidamente, las áreas CA (Garantía y seguridad básica de datos) y DQA (Garantía y seguridad en la calidad de datos), aprueban los tickets analizados de todos los proyectos y se almacenan en una única base de ingesta de datos.
- Aprobados los tickets o dictámenes de los proyectos, la información tratada y contenida en estos, se lleva a la Startup Datio la cual se encarga de gestión y análisis de todos los datos o información del banco para acoplarse a lo que requieren los clientes.

3.2. Finalidad y objetivos de la Actividad:

3.2.1. Finalidad:

La finalidad de este trabajo es conocer los beneficios del Big Data dentro del Proyecto Core Assurance de una entidad financiera del Perú y ver cómo se cumple la revisión de todo el proceso siguiendo los lineamientos del banco, para que pueda llegar al Big Data.

3.2.2. Objetivo general:

Analizar los beneficios del Big Data dentro del proyecto Core Assurance para el modelado de datos.

3.2.3. Objetivos específicos:

- Determinar cómo Big Data influye en la gestión de los proyectos de ingeniería.
- Describir los lineamientos dentro del equipo de Core Assurance para el análisis de la data.
- Distinguir las fuentes del Big Data dentro de las operaciones de Core Assurance.
- Analizar los indicadores de la gestión del proyecto Core Assurance.

3.3. Problemática

3.3.1. Problema General

¿Cuáles son los beneficios del Big Data dentro del proyecto Core Assurance para el modelado de datos?

¿Cómo se desarrolla el análisis de fuentes dentro del proyecto Core Assurance utilizando Big Data?

3.3.2. Problemas Específicos

- ¿Cómo influye el Big Data dentro del equipo de Data Engineering?
- ¿Cuáles son los lineamientos dentro del equipo de Core Assurance para el análisis de la data?
- ¿Qué tipo de fuentes de dictamen se distinguen dentro del Big Data dentro de las operaciones de Core Assurance?

- ¿Cuáles son los indicadores que cuantifiquen los beneficios obtenidos por el equipo Core Assurance al usar el Big Data en sus análisis?

3.4. Metodología

3.4.1. Bases teóricas

Linares Berrocal (2019) en su trabajo de tesis llamado "Implementación de un sistema de Big Data aplicado a la migración de datos bajo la distribución cloudera con Apache Hadoop, en el Banco Interbank ", se presenta una solución por el crecimiento exponencial de los datos en Interbank que está presentando inconvenientes para la gestión y almacenamiento de datos a gran escala lo que genera un incremento en costos operativos y de infraestructura, y por esa razón es necesario contar con una plataforma tecnológica que pueda soportar las tareas de registros y procesamiento en un ecosistema escalable y de alta disponibilidad.

El documento describe el proceso de diseño e implementación del modelo, incluyendo la recopilación de datos, la limpieza y procesamiento de texto, la selección de características, la construcción del modelo y la evaluación de su rendimiento. Los resultados del modelo se presentan en forma de gráficos y tablas, y se analizan y discuten en detalle. Finalmente, se centra en dar solución a esta problemática y hacer la migración de datos entre los motores de bases de datos de fuentes Oracle hacia el Teradata utilizando herramientas como DATASTAGE v6 IMB, al final del trabajo la solución propuesta es satisfactoria (Linares Berrocal, 2019).

José Huapaya (2021) en su trabajo de tesis "Integración de entorno Big Data en la gestión financiera de un sistema bancario para optimizar productos, servicios internos y toma de decisiones" describe la implementación de un sistema de Big Data como herramienta de administración de productos y servicios de una institución financiera, optimizando los productos financieros y la toma de decisiones. Es una institución bancaria cuyas distintas ramas de negocio disponen de bases de datos y sistemas aislados, lo que conlleva un elevado consumo de los recursos informáticos, una mayor capacidad de mantenimiento y muchos retrasos en los procesos. Esta problemática es común en las entidades financieras y se hace crítica cuando la entidad es transnacional, multiplicando estas bases de datos aisladas, como es nuestro caso. Ante esto, el área de Arquitectura de Datos propuso lineamientos tales como la centralización de la información en un medio de Big Data garantizando la disponibilidad gradual de los miembros del negocio para las nuevas iniciativas de analítica financiera y desactivando así las bases de datos aisladas. El entorno de Big Data consta de la capa de ingesta de datos en un

Data Lake por medio del procesamiento distribuido de Spark; y de otra capa de consumo de información mediante sandboxes, donde los usuarios dispondrán de herramientas de Business Intelligence para efectuar analítica avanzada y elaboración de reportes. Como resultado de la implementación de la integración, existen numerosos trabajos en este entorno de Big Data, los cuales son rentables con datos gobernados, flujos definidos y un mejor tiempo de respuesta en procesos que anteriormente se desarrollaban bajo bases de datos tradicionales. Como resultado, las áreas de negocio tienen productos y servicios más eficientes (Huapaya, 2021).

Ángelo Martínez (2018) en su trabajo de tesis "Big data aplicado en el sector bancario tradicional para lograr una mayor ventaja competitiva frente a las Fintech" plantea lineamientos sobre la forma de aprovechar las grandes cantidades de datos con las que cuenta el sector bancario convencional, que debido a su gran envergadura y complejidad ha estado limitado a la hora de sacar provecho de dichas tecnologías revolucionarias. Esta situación en la actualidad ha propiciado la aparición de nuevos competidores, las fintechs, las cuales están ganando terreno en un breve periodo de tiempo mediante el empleo de la tecnología más innovadora. A partir de la implementación de un esquema de trabajo ágil para la gestión de las tecnologías Big Data en el sector financiero tradicional, se busca obtener una ventaja competitiva, centrarse en el consumidor brindándole productos novedosos y de forma personalizada que se ajusten a sus necesidades y extraer el conocimiento de los datos para aprovechar las nuevas oportunidades de negocio a partir de una toma de decisiones más inteligente basada en los datos. (Martínez, 2018).

Galeano y Domínguez (2017) en su trabajo de investigación de postgrado "Prototipo de laboratorio Hadoop para análisis Big Data en la institución universitaria politécnico grancolombiano" tienen como objetivo de proporcionar información sobre Hadoop, un framework diseñado para el almacenamiento y análisis inicial de Big Data. Se destacan las ventajas de implementar esta tecnología, así como sus características principales y los beneficios que puede ofrecer en términos de escalabilidad arquitectónica. Además, el proyecto se centra en el montaje de un prototipo de clúster Hadoop en la institución universitaria Politécnico Grancolombiano. Este prototipo se plantea como una herramienta para el análisis y almacenamiento de información Big Data, y también se presenta como una guía o referencia para aprender a configurar un entorno Hadoop. Por último, se proporciona un ejemplo básico de programación en MapReduce. (Galeano & Domínguez, 2017).

Wang, Ding, & Yu (2020), en su artículo de investigación denominado "Big data analytics on enterprise credit risk evaluation of e-Business platform" acerca de la evaluación del riesgo de crédito bancario empresarial, indica:

Como uno de los productos innovadores de los bancos comerciales, la aparición del financiamiento de la cadena de suministro refleja la idea de la innovación de productos financieros de los bancos comerciales. En la competencia feroz, los bancos comerciales necesitan innovación constante para ser más competitivos. La innovación de los productos financieros de los bancos comerciales está gradualmente cambiando al online, bajo el desarrollo de la inteligencia y procesamiento de grandes datos (Big Data).

La banca en línea hace que los datos sean más transparentes, la gestión de riesgos se vuelve más crítica e importante, y los indicadores involucrados en el proceso son más complicados. En la era de Big Data, en primer lugar, los bancos comerciales deben llevar a cabo activamente la financiación de la cadena de suministro en línea, y guiar conscientemente la transición de la financiación de compañías desde los servicios fuera de línea a servicios en tiempo real. En el entorno de la cadena de red, la gestión de riesgos de los bancos comerciales es importante, por consiguiente, es necesario integrar completamente todos los elementos. (Wang et al., 2020)

3.4.2. Marco Conceptual

3.4.2.1. Almacenamiento de datos

Tipos de datos

A. Estructurados: Son aquellos datos que poseen un formato ya definido, así como una longitud y tamaño de dato específico. Su almacenamiento se suele realizar en bases de datos de tipos relacional (conocidas como SQL) así como en tablas usando hojas de cálculo. La ventaja de trabajar con datos estructurados es su facilidad de introducción (ingesta), análisis y almacenamiento. (Martínez, 2018)

B. No estructurados: Estos datos, a diferencia de los estructurados, no tienen un formato definido (no es específico) y no poseen una estructura sólida. Además, se tiene un limitado control sobre estos ya que normalmente su estructura es no uniforme. La manipulación e interpretación de estos datos es más compleja que los estructurados debido a que la información que contienen no está representada por datos elementales. Algunos ejemplos de datos no estructurados son: fotografías, videos, audios, emails, etc. (Martínez, 2018)

C. Semiestructurados: Estos datos se presentan a menudo en formas que resultan difíciles de comprender para los consumidores y con frecuencia requieren el empleo de intrincadas herramientas para ayudar a decidir cómo interpretar cada dato individual, como los formatos HTML, XML o JSON. Además, no tienen una estructura totalmente definida como los

datos estructurados. Sin embargo, tienen una organización específica en sus metadatos donde se indican los objetos y sus conexiones. (Calvo, 2017)

Base de datos relacional

Es un tipo de base de datos que se utiliza para modificar datos basándose en relaciones predeterminadas entre tablas con columnas y filas relacionadas. Incluye una serie de objetos que pueden utilizarse para recuperar, gestionar y guardar datos. Estos objetos incluyen, entre otros, tablas, vistas, índices, funciones, disparadores y paquetes. (IBM, 2019)

Cada objeto tiene una columna la que a su vez tiene un nombre y un tipo para identificarla. De igual manera, cada objeto tiene registros o filas las que a su vez representan un dato distinto que a menudo corresponde a la información continua sobre un objeto. Además, hay una o más claves principales para cada objeto. Cada registro se identifica de forma única por sus claves principales. (Microsoft Learn, 2022)

Los objetos de las bases de datos relacional se encuentran estandarizados para de esa manera reducir la duplicación de datos, así como para proporcionar una disponibilidad e integridad garantizada de la base de datos. Con una base de datos bien diseñada se garantiza acceso total a la información necesaria. Con un diseño sólido, la construcción de la base de datos le llevará menos tiempo y obtendrá resultados más precisos con mayor rapidez. (Strohbach et al., 2017)

Base de datos NoSQL

Las bases de datos NoSQL son bases de datos no relacionales conocidas por su alto rendimiento. Son conocidas por el hecho de que son sencillas de codificar, no dependen de ninguna relación preexistente, pueden escalar horizontalmente, tienen alta disponibilidad y toleran los fallos. También se conocen como "Not Only SQL" (No solo SQL). Se desarrollaron sobre todo a raíz de la web 2.0, que hizo que las personas pasaran de ser solo consumidores de contenidos a creadores de estos. Actualmente existen en el mercado aproximadamente 225 soluciones de gestión de bases de datos NoSQL. Uno de los aspectos más importantes que se pueden contrastar entre las bases de datos NoSQL y las bases de datos relacionales tradicionales es el hecho de que NoSQL utiliza un tipo de almacenamiento no estructurado. Esto demuestra que las bases de datos NoSQL no siguen una estructura de tablas predefinida como las bases de datos relacionales. Además, SQL

(Structured Query Language) no es su lenguaje preferible de consultoría. (AichaAggoune, 2020)

Varias grandes empresas de Internet han promovido abiertamente el uso de bases de datos NoSQL, como Google y Facebook. Ante el problema de la accesibilidad, las empresas optaron primero por desplegar más máquinas, pero rápidamente reconocieron que este era un método costoso e ineficaz. Motivadas por el crecimiento explosivo de la web, que hizo necesario desarrollar aplicaciones web que pudieran atender de forma rápida y eficaz los requerimientos de un número considerable e indefinido de clientes, a la vez que procesaban cantidades masivas de datos en información utilizando estructuras horizontales más o menos similares; las empresas mencionadas anteriormente, tuvieron que optar la velocidad y la eficiencia de los requerimientos en tiempo real por encima de la consistencia de los datos., que las antiguas bases de datos tardaban mucho en manejar. (Padhy et al., 2017)

Teorema de Brewer

El teorema Brewer, llamado también como teorema CAP (*Consistency, Availability, and Partition tolerance*), describe la razón por la cual las bases de datos NoSQL no se ajustan perfectamente a ACID (*Atomic, Coherent, Isolated and Durable*), los principios ACID son características deseables en las transacciones de bases de datos: atomicidad (transacciones completas o ninguna), coherencia (sistema en estado coherente), aislamiento (transacciones aisladas) y durabilidad (cambios persistentes). Son fundamentales en bases de datos relacionales, pero en sistemas distribuidos es difícil cumplir completamente con ACID sin comprometer los criterios CAP. En una conferencia sobre los principios de la informática distribuida en el año 2000, Eric Brewer mencionó que los tres criterios de consistencia, disponibilidad y tolerancia a la división no pueden garantizarse de forma concurrente en los sistemas de datos distribuidos (Estrada & Ruiz, 2017).

A continuación, vamos a examinar estos criterios:

- **Consistencia:** Se refiere a que la información sea homogénea y consistente incluso una vez alterada, garantizando que cada usuario que acceda a los datos vea los mismos datos. Es necesario que sea coherente tanto si se introducen datos (cada nodo debe insertar los mismos datos), como si se actualizan los datos (cada nodo debe realizar la misma modificación en todos los datos) o se consultan los datos (cada nodo debe devolver los mismos datos).
- **Disponibilidad:** El sistema es capaz de mantener su operatividad y aceptar solicitudes de lecturas y escrituras, incluso si alguno de los nodos se desconecta o deja de responder. Cuando se interrumpe la transmisión

de información de uno de los nodos, el equipo tiene que ser capaz de permanecer en funcionamiento de forma automática hasta que se restablezca, y entonces tiene que sincronizarse con los demás nodos.

- **Tolerancia a la división:** El sistema debe ser accesible incluso cuando surjan dificultades de transmisión de información desde un nodo a otro, interrupciones de la señal que obstaculicen la circulación de la información o algún obstáculo que provoque la división o particionamiento del sistema.

Las siguientes combinaciones son las únicas que la teoría garantiza:

- **Consistencia y Partición:** Aunque haya dificultades de conectividad entre los nodos (particiones de la red), el sistema aplicará los cambios de forma coherente. La disponibilidad no está garantizada.
- **Disponibilidad y Partición:** El sistema siempre responderá a las consultas, incluso en caso de problemas de conectividad de los nodos (particiones de la red), aunque se producirán incoherencias temporales en los datos debido a la falta de comunicación, ya que por lo general algunos de los nodos no son idénticos.
- **Consistencia y Disponibilidad:** En todo momento, se puede acceder al sistema, el cual proporcionará información consistente en respuesta a las consultas. Como resultado, el sistema no tolera la caída de la información en los nodos.

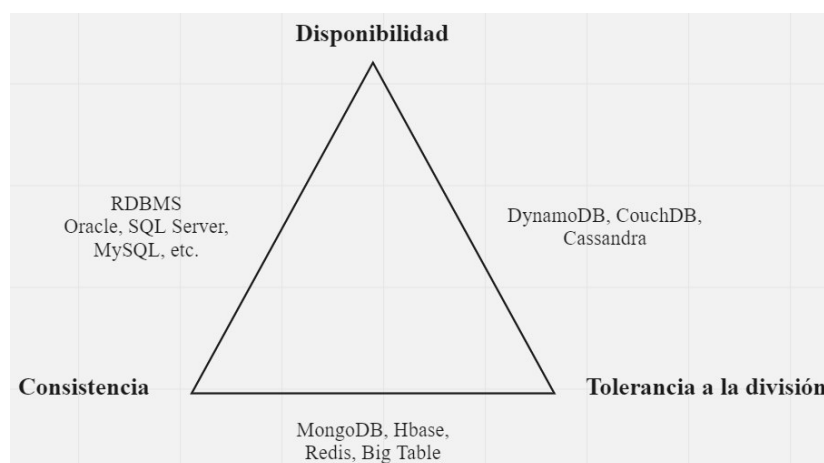


Figura 1: Teorema de CAP. Fuente: Elaboración propia.

Las características más destacadas de las bases de datos de tipo relacional y NoSQL son comparadas según la Tabla 1:

Funcionalidad	Relacional	NoSQL
Rendimiento	Bajo	Alto
Confiabilidad	Alto	Bajo
Disponibilidad	Alto	Alto
Consistencia	Alto	Pobre
Almacenamiento	Cantidades medianas y grandes de datos	Especializado para enormes cantidades de datos.
Escalabilidad	Alto, pero más costoso.	Alto

Tabla 1: Comparación entre las capacidades fundamentales de las bases de datos relacionales y las bases de datos NoSQL. Fuente: Elaboración propia

Tipo de base de datos NoSQL

Existen diferentes modelos o tipos en el campo de las bases de datos NoSQL, y algunos de ellos funcionan mejor en determinadas situaciones que otros. Existen cuatro tipos diferentes, los cuales son definidos a continuación:

- **Columnar:** Las bases de datos basadas en columnas se han diseñado para la lectura y escritura de datos en formato de columnas en sustitución a las filas. La finalidad de las bases de datos estructuradas en columnas es la lectura y escritura eficiente de los datos en el disco duro para minimizar el lapso de tiempo que requiere la respuesta a una petición. La posibilidad de comprimir mucho los datos es una de las principales ventajas de una base de datos en columnas. Los procesos de agregación pueden realizarse con relativa rapidez gracias a la compresión. (Amazon Web Services, 2017)
- **Documental:** Esta modalidad de base de datos permite almacenar datos semi-estructurados en forma de documentos en un mismo formato, como por ejemplo "JSON", "XML" o "BSON", con una clave única para cada entrada. Admite búsquedas de valor clave, además de consultas más complejas sobre el contenido de los documentos. Estas son las bases de datos NoSQL más adaptables. Numerosos proyectos, incluidos los que normalmente emplearían bases de datos de tipo relacional, pueden utilizarlas. (Amazon Web Services, 2017)
- **De Grafos:** Este tipo de bases de datos almacenan y representan datos relacionados sin necesidad de un índice, utilizan la topología de un grafo, en el que los nodos hacen de vértices y las relaciones hacen de aristas y atributos. Además, son de gran utilidad en el mantenimiento

de datos en modelos con numerosas relaciones, incluidas las redes y las comunicaciones de carácter social. En comparación con una arquitectura relacional, este tipo de base de datos permite un recorrido más eficaz entre las conexiones. (Timón et al., 2021)

- **Clave-valor:** Este tipo de base de datos NoSQL es el más utilizado y funcionalmente más sencillo. Están diseñadas para operaciones de lectura masiva o para cargas de trabajo intensivas, como las de las redes sociales, los juegos, los medios de comunicación y los portales de preguntas y respuestas (como los motores de recomendación). (Timón et al., 2021)

Data Lake

Los lagos de datos son depósitos en los que se guardan enormes volúmenes de datos, incluyendo datos estructurados, semiestructurados y no estructurados, en sus formatos nativos y sin ninguna modificación previa, con la intención de hacer un análisis sobre ellos más adelante. Como la información puede guardarse en su formato original, no es necesario transformarla en un formato estándar. En realidad, las empresas almacenan la información en un repositorio centralizado desde el que se puede recuperar cuando sea necesario. Después, la información se procesa, se clasifica y se organiza de forma que pueda ser analizada por otros departamentos (John & Misra, 2017).

El término "lago de datos" hace referencia a un reservorio de almacenaje masivamente escalable capaz de mantener grandes cantidades de información en bruto sin modificar su estructura original ("tal cual") mientras no se necesite, así como a mecanismos de procesamiento capaces de incorporar información sin afectar a la estructura de la información. Los data lakes habitualmente son contruidos con el fin de gestionar enormes y veloces cantidades de datos que no están estructurados, de los que se obtienen nuevos enfoques (a diferencia de los datos altamente estructurados de los almacenes de datos). Así pues, los data lakes usan programas de análisis de carácter dinámico (y no estáticos, como los de los almacenes de datos). (Miloslavskaya & Tolstoy, 2016)

Los datos almacenados en el lago están disponibles desde el momento en que se crean (de nuevo, en contraste con los almacenes de datos diseñados para datos que cambian lentamente). Los lagos de datos suelen incluir una base de datos de tipo semántico, un esquema de conceptos basado en los mismos estándares y tecnologías utilizados para crear hipervínculos en Internet y añaden una capa de contexto sobre los datos que define el significado de estos y sus interrelaciones con otros datos. Las estrategias de

los lagos de datos pueden combinar enfoques de bases de datos SQL y NoSQL y capacidades de procesamiento analítico en línea (OLAP) y de procesamiento de transacciones en línea (OLTP) (Miloslavskaya & Tolstoy, 2016).

Por lo general, se considera que un Data Lake es un almacén de objetos, donde los datos que posee una empresa son colocados en una plataforma específica cuyas herramientas se utilizan para la extracción y el análisis de información. No obstante, los lagos de datos pueden emplearse con éxito sin depender de una plataforma en particular, dependiendo de los requisitos y objetivos empresariales (International Business Machines, 2018).

Características de Data lake

- **El sistema no descarta datos:** Almacena y retiene la totalidad de las entradas de información y datos, independientemente de su estructura u origen. Todos los datos se conservan, aunque no se utilicen inmediatamente, porque pueden utilizarse en el futuro.
- **Modificable y de rápida adaptación:** Posibilita la adaptación a todo tipo de análisis, en todo momento, y a la vez de forma más específica. La persona encargada tiene la posibilidad de utilizar los datos almacenados como considere oportuno, en el momento y con la frecuencia que necesite.
- **Dirigido a cualquier clase de perfiles:** Es conveniente para cualquier clase de usuario, incluidos los especialistas en ciencias de la información encargados de efectuar análisis en profundidad como para los usuarios con preferencia por las vistas organizadas.

Ventajas del Data Lake

- Todos los datos existentes son fácilmente accesibles para los usuarios de la empresa a través del lago de datos.
- Los datos relacionales y transaccionales no son los únicos tipos de datos en el lago de datos.
- Mediante un data lake, no es necesario que el usuario desplace los datos.
- Los data lakes conceden autonomía a los usuarios de la empresa a fin de librarlos de los imperativos del dominio de las tecnologías de la información.
- El data lake agiliza la transmisión de información permitiendo una rápida introducción de las aplicaciones en el lago por parte de las unidades de negocio.

Desventajas del Data Lake:

- Requiere capacidades de muy alto nivel tecnológico en el tratamiento de datos.
- Amenazas al anonimato de las personas.
- La gestión de los datos históricos es difícil.
- Requiere un manejo efectivo del tiempo de vida útil de los metadatos.
- La fragmentación de los sistemas de información.
- Desafíos en la integración

3.4.2.2. Big Data

El término "Big Data" se refiere a la administración y el procesamiento de una gran cantidad de datos que exceden la capacidad y las restricciones de las plataformas informáticas utilizadas comúnmente con el fin de adquirir, administrar y procesar los datos. Esta idea incluye las infraestructuras, las tecnologías y los servicios desarrollados para proporcionar enfoques para manejar enormes cantidades de datos de carácter estructurado, desestructurado y/o semiestructurado. Los grandes volúmenes de datos, como los petabytes, los exabytes, los zettabytes y los yottabytes, son lo que denominamos Big Data (YB) (Martínez, 2018).

Entonces, la Big Data almacena grandes cantidades de información el cual es muy importante para este flujo de datos que se verá en este modelo de negocios ya que el Big Data puede recibir datos estructurados, no estructurados y semiestructurados de diferentes fuentes ya sean Oracle, Mondo DB, SQL, etc.

Dimensiones de Big Data

Además, la Big Data tiene 3 importantes aspectos como son el volumen, velocidad y variedad.

- **Volumen:** En el Big Data se procesa información en grandes cantidades lo que es muy importante en este trabajo de suficiencia ya que el flujo de trabajo almacena gran cantidad de información de todos los proyectos de este banco. Esta característica hace referencia a los enormes volúmenes de datos con los que las empresas buscan mejorar su eficacia en la elección de decisiones. Las cantidades de datos siguen creciendo a un ritmo sin precedentes. Sin embargo, un volumen verdaderamente "grande" varía según el rubro de la empresa o incluso su localización, siendo inferior a los petabytes y zettabytes que se mencionan a veces (Puyol, 2014).
- **Velocidad:** Esta característica se refiere a la agilidad de la recepción y el tratamiento de los datos. Hace referencia al ritmo de producción de datos, lo que determina la rapidez con la que se expanden programas informáticos tales como los websites, los ficheros de búsqueda, las

herramientas sociales, los forums y los e-mails (Camargo-Vega et al., 2015).

- **Variedad:** Hace referencia a la variedad de datos que hay a disposición. La base de datos relacional permite estructurar y adaptar óptimamente los diferentes grupos de datos tradicionales. Esta característica está correlacionada con el volumen porque hay varias formas de representar los datos, sean estructurados o no. Se consideran datos no estructurados los producidos a partir de sitios web, ficheros de búsqueda, plataformas sociales, fórums, mensajes de email. Por ejemplo, la conversión de 350.000 millones de registros anuales de contadores para predecir el consumo de energía (Camargo-Vega et al., 2015).

Además, de los 3 aspectos mencionados anteriormente, existen otros aspectos o características del Big Data añadidas por diferentes instituciones o empresas; algunas de estas otras características son:

- **Valor:** El valor describe los beneficios obtenidos por el empleo de Big Data, por ejemplo, la disminución de gastos, la eficiencia funcional o mejoramiento del negocio. A estas cualidades puede añadirse una más, conocida como "veracidad" (Puyol, 2014).
- **Veracidad:** Esta característica describe el grado de fiabilidad asociado a determinadas formas de datos. Incluso las mejores técnicas de depuración de datos no son capaces de evitar por completo la incertidumbre intrínseca a ciertos datos, como la meteorología, la situación económica o la conducta de compras futuras del usuario. La búsqueda de datos de alta calidad es una necesidad crucial y una dificultad básica del Big Data. El Big Data tiene una dimensión que surge cuando los directivos intentan comprender el entorno incierto en el que viven (Puyol, 2014).

Arquitectura de componentes Big Data

- **Sandbox:** Se trata de una herramienta funcional muy flexible la cual ofrece la posibilidad de realizar operaciones de análisis en las organizaciones formales y en las actividades tácticas. Además, recoge evidencias para desarrollar habilidades innovadoras de relación persona-información (Wright et al., 2006).
La cognición de "poner esto", las plantillas de modelos de procesos autónomos, los gestos para una expresión fluida del pensamiento, las afirmaciones con pruebas y los métodos de escalamiento para dar respuesta a trabajos analíticos más complejos son algunos de los puntos fuertes o capacidades de Sandbox. Utilizando una interfaz y un protocolo de servicios web, Sandbox incorpora sofisticadas características lingüísticas computacionales (Wright et al., 2006).
- **Ingesta de datos:** La ingesta de datos es un componente que debe caracterizarse por poseer la capacidad de validar, limpiar, transformar, reducir, e integrar datos en el sistema de Big Data para su posterior procesamiento. Este componente realiza la carga de información importante y limpia en la capa de almacenamiento distribuida de

Hadoop. La ingesta de datos es un proceso muy importante en un proyecto de data ya que desde aquí parte la explotación de información para los proyectos que se deseen realizar. La entrada de datos es la primera etapa en la que los datos sin procesar (tal cual vienen del origen) comienzan a tomar la forma de información utilizable. (Erraissi et al., 2018).

- **Procesamiento:** El procesamiento de datos se produce cuando se recaban datos y se traducen a información utilizable. Estos procesamientos tienen como origen a las ingestas ,pero la diferencia es que estos procesos tiene lógicas y relaciones en su creación. Se puede realizar mediante técnicas de filtrado, análisis y visualización de datos e incluso con algoritmos de aprendizaje automático, aunque el proceso en sí puede variar ligeramente dependiendo de la fuente de datos que se procesa (base de datos, redes sociales, dispositivos conectados, etc.) y su uso previsto (examen de patrones de publicidad, diagnóstico de dispositivos conectados, determinar las necesidades del cliente, etc.) (Ortega, s.f.).

Tecnologías de procesamiento y almacenamiento de Big Data

- A. Hadoop:** Hadoop es un marco de software de código abierto para el procesamiento y almacenamiento distribuido de conjuntos de datos extremadamente masivos en clústeres diseñados en dispositivos con hardware común. El objetivo de esta tecnología es crear una arquitectura de computación paralela que permita dividir el trabajo (el procesamiento de datos) entre varios dispositivos (nodos de la red) para reducir el tiempo total de procesamiento. Básicamente, el concepto de "divide y vencerás" consiste en dividir el trabajo en tareas más pequeñas y realizarlas de forma dispersa, en lugar de hacerlas todas a la vez (Rayón, 2016).

Apache Hadoop tiene numerosas partes que pueden trabajar juntas para producir una red descentralizada de propósito general. Dado que Hadoop se construyó desde el principio para llevar a cabo el procesamiento de datos en paralelo distribuido de acuerdo con los conceptos de la arquitectura "shared-nothing", su principal fuerza radica en su capacidad de escalar horizontalmente. (Baranowski et al., 2019)

Esta tecnología fue desarrollada por Google en respuesta a su necesidad de gestionar las cantidades masivas de datos que manejaba (documentos, referencias web, páginas, etc.), así como de reducir los costes de almacenamiento y mejorar la velocidad y la escalabilidad. La documentación de la empresa Google, MapReduce y GFS, sirvió de inspiración para Hadoop. En 2004, dos ingenieros publicaron un documento titulado "MapReduce: Simplified Data Processing on Large Clusters", al que se atribuye la primera mención de Hadoop. MapReduce, un nuevo enfoque de programación mencionado en el artículo, agiliza el manejo de conjuntos de datos masivos. La necesidad de Google de procesar las cantidades masivas de datos que maneja (documentos, referencias web, páginas, etc.) llevó al desarrollo de esta

nueva arquitectura, que también les ayuda a ahorrar dinero en espacio de almacenamiento y aumenta la escalabilidad y velocidad de su sistema (Jain, 2017).

El propósito de este nuevo modelo es proporcionar una estrategia de computación paralela que permita dispersar el trabajo (el procesamiento de datos) en varios ordenadores (nodos dentro de una red) de manera que el tiempo total de procesamiento pueda disminuir al dividir el trabajo en tareas más pequeñas y realizarlas de forma dispersa. Por el contrario, un enfoque centralizado consistiría en una única y enorme asignación.

Distribuciones de Hadoop

Existe una gran variedad de herramientas y distribuciones que, combinadas, permiten administrar fácilmente los clusters. A continuación, se van a detallar las tres más importantes: Cloudera, Hortonworks y MapR.

- **Cloudera:** Cloudera fue creada en 2008 por un antiguo ejecutivo de Oracle, además de tres desarrolladores de software que habían trabajado anteriormente en Google, Yahoo! y Facebook. Su software se deriva de Apache Hadoop y, además, proporciona a los principales clientes soporte, servicios y formación. Cloudera fue la primera empresa en ofrecer componentes propios de valor añadido sobre Hadoop. Estos componentes incluyen Impala, que, junto con otras herramientas como Cloudera Manager y Cloudera Navigator, son utilizados por un número importante de clientes. (Cloudera, 2018)

Cloudera ofrece una plataforma escalable, adaptable e integrada. Esta plataforma facilita el manejo de cantidades y tipos de datos en rápida expansión. Las tecnologías y soluciones que ofrece Cloudera permiten instalar y gestionar de forma segura Apache Hadoop y otros proyectos del ecosistema relacionados con la manipulación y el análisis de datos. La naturaleza del programa es de código abierto, pero también contiene software propietario y hay cuotas asociadas a la licencia y la asistencia técnica (Torres & Rincón, 2017).

- **Hortonworks:** Fundada en 2011 por un equipo de veinticuatro desarrolladores de Hadoop, Hortonworks se destaca como una opción destacada en el ecosistema de Big Data. Al ofrecer una solución completamente gratuita y de código abierto, es el único proveedor comercial que brinda Hadoop sin ningún software propietario adicional, lo que permite a los usuarios acceder a la plataforma sin costos de licencia y cobrando solo por servicios de mantenimiento. Con sede en Santa Clara, California, la empresa cuenta con más de 2.300 socios en diversos ámbitos, como ingeniería, distribución estratégica, tecnología e integración de sistemas.

La plataforma de código abierto de Hortonworks, basada en Apache Hadoop, ha sido diseñada para ayudar a las empresas a analizar, almacenar y gestionar cantidades masivas de datos de manera eficiente. Con su enfoque en el poder del Big Data, Hortonworks se ha posicionado como una opción valiosa para las organizaciones que

buscan aprovechar todo el potencial de sus datos en el actual panorama empresarial. (Ma, 2017)

- **MapR:** MapR, es una empresa con sede en Santa Clara (California), ofrece una plataforma de datos distribuidos adecuada para almacenar y analizar Big Data a nivel empresarial. Su software puede funcionar en ordenadores estándar o en la nube. Sus funcionalidades para el análisis se ven muy potenciadas por las diversas mejoras de rendimiento incluidas en su arquitectura. En una misma plataforma, formada por una única base de código, se reúnen un sistema de archivos distribuido, una base de datos NoSQL multimodelo (MapR-DB), un motor de streaming de eventos de publicación y suscripción (MapR Streams), ANSI SQL (Apache Drill™) y un conjunto completo de tecnologías de gestión y análisis de datos de código abierto. Es el distribuidor de Hadoop que más ha hecho por garantizar que las implantaciones de Hadoop con más usuarios sean fiables y eficaces. Para ello, ha construido su propio sistema de archivos nativo sobre Unix. En lugar de utilizar el componente HDFS, MapR hace uso de su propio sistema de archivos interno que recibe el nombre de MapR-FS. Ahora, Hadoop puede realizar tareas de gestión de datos con mayor eficacia, fiabilidad y facilidad de uso gracias a que MapR-FS incorpora funcionalidades de nivel empresarial. Ofrece características para la alta disponibilidad (HA), la recuperación de desastres (DR) y la recuperación de datos, todas ellas de misión crítica, con el objetivo de maximizar el tiempo de actividad y minimizar el riesgo de pérdida de datos.

Módulos del Hadoop

Hadoop se compone de varios módulos diferentes, cada uno de los cuales es responsable de una determinada función necesaria para un sistema informático construido para el análisis de grandes cantidades de datos (Apache Software Foundation, 2018):

A.1. HDFS: Es un sistema de archivos distribuido que es escalable y capaz de responder a fallos, y fue desarrollado en Java para ejecutarse en servidores de bajo coste. Gracias a la coordinación de YARN (Yet Another Resource Manager), es compatible con una amplia gama de aplicaciones de acceso a datos concurrentes y permite un acceso a datos de alto rendimiento. Cuando se utiliza HDFS, los archivos se almacenan en unidades llamadas "bloques" (128 MB). Los datos de cada bloque se duplican en muchos servidores (International Business Machines, 2021).

Capas de aprovisionamiento en HDFS:

- **Staging:** Se implementó una capa temporal de almacenamiento que mantiene los archivos en su forma original directamente desde la fuente, que pueden ser archivos csv, tablas, etc. La temporalidad de estos archivos se ha optimizado al mínimo necesario debido a la posibilidad de contener información sensible (que será encriptada en

capas posteriores). Además, la temporalidad es importante para gestionar el espacio, ya que los archivos planos ocupan mucho espacio que puede ser mejor aprovechado por otros formatos más adecuados para el análisis de Big Data, permitiendo una mayor eficiencia y consumo de recursos para los usuarios. (Vasquez, 2021)

- **Raw:** Es la capa de datos crudos en el entorno Big Data, esta capa contiene archivos con sistema de compresión a nivel de filas (formato AVRO) para posteriores transformaciones, pero no para consultas y analítica avanzada. Esta capa también nos sirve de almacenamiento histórico por su ahorro en disco y también sirve de respaldo ante posibles reprocesos de la siguiente capa. (Vasquez, 2021)
- **Master:** Capa final del Data Lake para la cual está compresada a nivel columnar con el objetivo de soportar análisis exploratorios y procesamiento de modelos analíticos, este formato de archivos es el PARQUET. Esta capa es la que tienen acceso los Data Scientist en sus respectivos Sandboxes. (Vasquez, 2021)

Modelamiento en HDFS:

- **Fuentes con Información Complementaria:**

Si dos o más fuentes están al mismo nivel de granularidad, su información es complementaria y/o los atributos que contienen están asociados a un mismo concepto, estas fuentes deberían unirse en un solo Master. Por ejemplo, en la figura 2, se observa que la Tabla Master y la Tabla auxiliar en la capa Raw se complementan ya que están al mismo nivel de granularidad por lo que se unen mediante un Primary Key (PK), esta unión genera una tabla maestra única englobando toda la información en la capa Master.

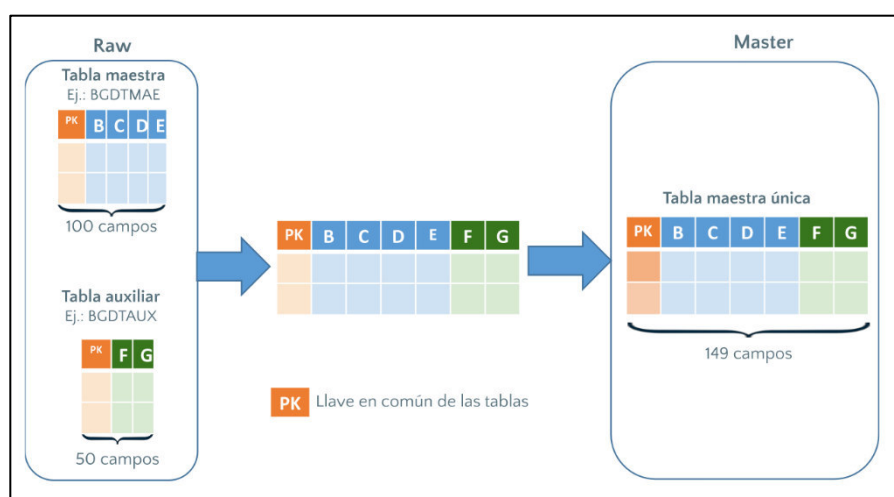


Figura 2: Ejemplo de tablas complementarias. Fuente: Site Core Data

• Fuentes con Estructuras Iguales

Existen casos puntuales, donde a nivel de estructura contienen los mismos atributos y columnas, y los universos de información son complementarios y no se cruzan, para estos casos las fuentes deberían unirse en un solo Master. Por ejemplo, en la figura 3, se observa que la Tabla de Movimientos 1 y la Tabla de Movimientos N tienen la misma cantidad de columnas y tienen los mismos atributos en la capa Raw, por lo cual se deben unir en una sola Tabla en la capa Master añadiendo un campo llamado Source que indique el origen de cada registro.

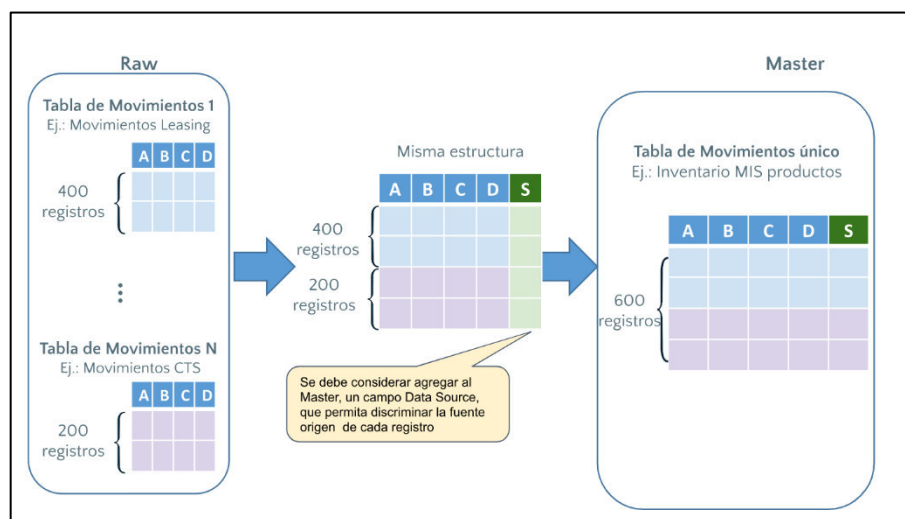


Figura 3 : Ejemplo de tablas con estructuras iguales. Fuente: Site Core Data

• Ficheros Ad-Hoc

En el caso de los Ficheros Ad-hoc, solo deben llegar al Master los campos calculados, aquellos que se generan en el Ad-hoc (y no se encuentren en otras fuentes ya ingestadas). Esto con el fin de mitigar la redundancia sobre datos naturales que puedan leerse directamente de otras fuentes ingestadas en el startup del banco. Por ejemplo, en la figura 4, observamos que el fichero ad-hoc tiene 8 campos de los cuales (5 campos calculados ,3 campos que se pueden obtener en otras fuentes), por lo que a la capa Master solo sube los campos calculados en el ad-hoc que son propios y no se puede encontrar en otras fuentes.

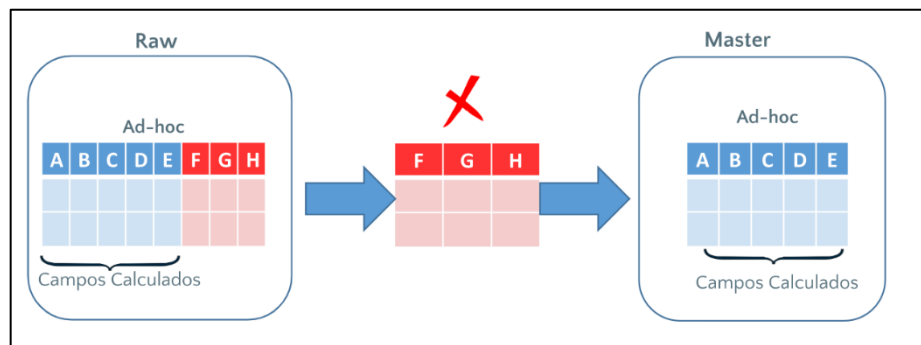


Figura 4:Ejemplo de ficheros ad-hoc. Fuente: Site Core Data

• Múltiples valores por columna

Si tenemos una fuente que contiene múltiples valores en una columna, estos deben separarse en columnas independientes en el Master para tener datos atómicos. De todas formas, pueden existir casos donde campos concatenados dan lugar a nuevos conceptos. Cada caso tiene que ser evaluado.

Por ejemplo, en la figura 5, se observa que el campo Doc. Identidad contiene el código del tipo de documento y también el número de documento en un solo campo. Lo ordenado es separar los 2 campos en la capa Máster, para tener columnas con un único concepto.

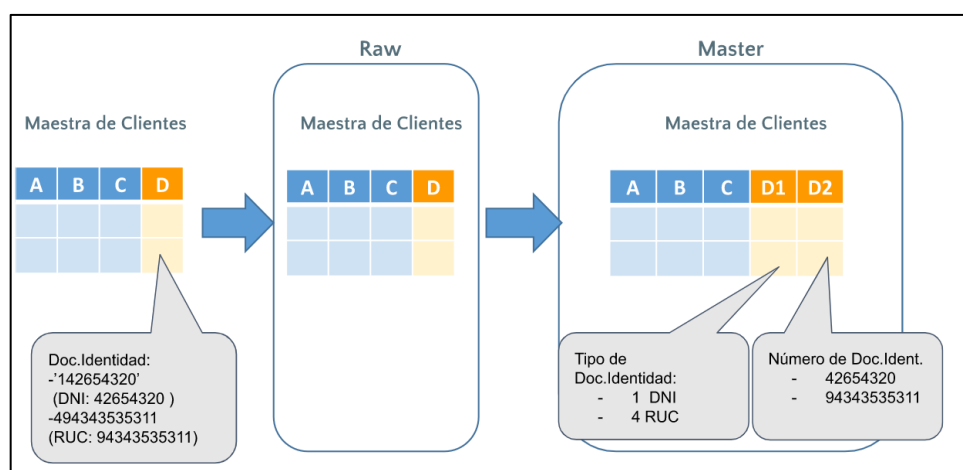


Figura 5:Ejemplo de tablas con múltiples valores en 1 campos. Fuente: Site Core Data

• Campos redundantes

Si tenemos una fuente que contiene atributos asociados a otra entidad conceptual, no deben subir al master ya que los datos no pueden estar redundantes en otra fuente. Por ejemplo, en la figura 6, vemos que en la capa Raw tiene 6 campos de los cuales los 3 últimos están redundantes ya que esa información la podemos encontrar en el campo

C, por ende, no debería subirse duplicidad de información, por eso en el ejemplo cuando sube a la capa master se elimina los campos redundantes.

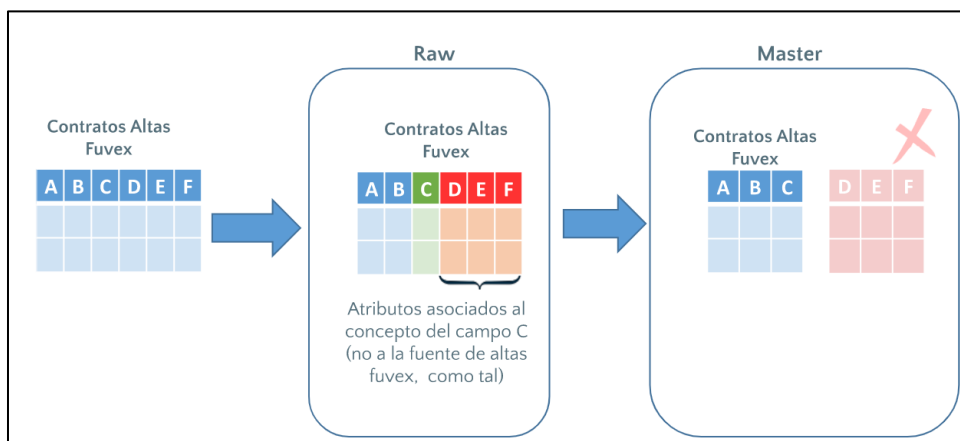


Figura 6: Ejemplo de tablas con campos redundantes. Fuente: Site Core Data

• Campos Free-Field

Si al relevar el concepto funcional de un campo con el aplicativo o dueño de la fuente, indica que dicho campo no tiene uso, no es fiable o es irrelevante o no se sabe que es, se puede hacer uso del campo `free_field`. Estos campos no deben subir al Master.

Se debe hacer el esfuerzo para obtener la descripción funcional de todos los campos con el aplicativo, usuario y/o dueño de la fuente. Los campos que no tienen descripción conocida, o que contiene información no fiable y sin uso, pueden usar como naming: `free_field`.

Por ejemplo, en la figura 7, se tiene 6 campos en la capa Raw, donde visualizamos que 1 de esos campos es un campo sin uso que no tiene información o solo viene data vacía que talvez a futuro puede que rellene información en esos campos, pero en la actualidad no tiene uso, en ese caso a la capa master solo suben los campos que se están explotando y se elimina el campo sin uso.

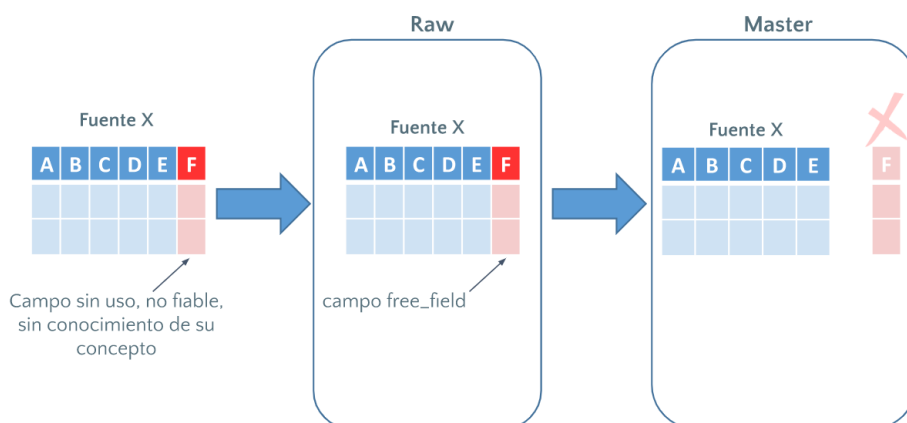


Figura 7: Ejemplo de tablas con campos free-field. Fuente: Site Core Data

• Partición

La partición del Master es importante, ya que esta capa es la que consulta el usuario; por lo tanto, se debe analizar el comportamiento de la información y el uso que se le dará a la misma.

En el caso de tablas maestras donde se requiera ver los cambios de los atributos en el tiempo, es probable que se necesite particionar por fecha para tener todas las fotos históricas.

En el caso de movimientos o transacciones, también es usual particionar el master por fecha, sin embargo, en el ejemplo de la figura 8, en este caso cada partición de la fuente de la Maestra de Clientes contiene las operaciones del día.

Hay casos en el que se requiere usar más de una partición. Estos casos se justifican siempre y cuando esto facilitará las consultas a la fuente.

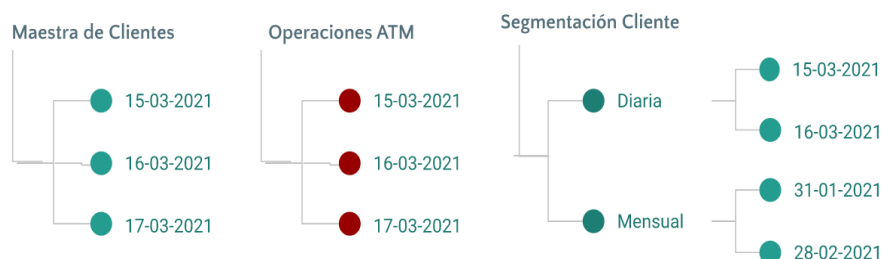


Figura 8: Ejemplo partición de tablas. Fuente: Site Core Data

A.2 YARN: El diseño de Hadoop gira en torno al sistema Yarn (Yet Another Resource Manager), que controla los recursos del clúster y permite el procesamiento de datos en paralelo.

A.3 MapReduce: Dividiendo las tareas en un "map" y un "reduce", o "MapReduce", los pares de datos de entrada (clave, valor) se mapean a un nuevo conjunto de datos basado en pares (intermedios) utilizando la función map (clave, valor). Los valores intermedios se envían a Reduce en un formato de pares (clave, valor), lo que permite un procesamiento rápido y, lo que es más importante, una fácil "reconstrucción" en el futuro. (Torres & Rincón, 2017)

A.4 Common: Incluye el código necesario para ejecutar el marco de trabajo y permite el acceso a los sistemas de archivos soportados por Hadoop. Tanto el código fuente de la herramienta como los manuales de uso están a disposición del usuario.

- B. Apache Spark:** Es un motor de procesamiento distribuido que gestiona, distribuye y supervisa aplicaciones compuestas por varios trabajos de procesamiento de datos repartidos en un clúster de ordenadores activos. Se pueden utilizar diversos lenguajes de programación, como Java, Scala, Python y R, para crear aplicaciones. Puede ser hasta 100 veces más rápido en memoria o 10 veces más rápido en disco que Hadoop MapReduce, dependiendo del software (ESIC BUSINESS & MARKETING SCHOOL, 2018).
- C. Python:** Python es una herramienta especialmente eficaz para el Big Data debido en parte a la enorme comunidad de usuarios y a las numerosas bibliotecas que han creado previamente otros usuarios. No obstante, su inconveniente es que se ejecuta con lentitud, por lo que suele utilizarse para trabajos de integración o actividades que no necesitan cálculos complejos (Huapaya, 2021).
- D. Avro:** Se trata de un formato de almacenamiento basado en filas y a veces se le denomina sistema de serialización de datos tipo Java. Avro ofrece estructuras de datos con una riqueza de atributos, caracterizado por un formato binario altamente compacto y de respuesta veloz. Para leer o escribir archivos de datos, no es necesario crear código. Además, Avro puede proporcionar un rendimiento en tiempo de ejecución mejor ajustado u optimizado (Plase et al., 2017).
- E. Parquet:** A diferencia de AVRO, Parquet tiene un formato de almacenamiento basado en columnas. Este ha sido diseñado teniendo en cuenta los conjuntos de datos de varias columnas. Trabajar con un subconjunto de esas columnas, en lugar de con todos los registros, es la forma en que se suele utilizar Parquet. El hecho de que la ordenación por columnas mejore la ejecución de consultas analíticas es una de sus ventajas más mencionadas. Los datos con baja entropía de información son ideales para las tecnologías de compresión de datos (datos con

alto valor local). Como resultado, el sistema de archivos se beneficia de las mejoras de la compresión en la eficiencia de la lectura y la escritura sin añadir carga de trabajo a la CPU durante la descompresión. Para las consultas que manejan cantidades masivas de datos, la arquitectura de archivos de datos de Parquet es muy útil (Plase et al., 2017).

3.4.2.3. Conceptos asociados al flujo de Core Assurance

TDS: La TDS es la fuente que se ingestará en el Startup del banco y debe contener datos válidos, completos y confiables. Es importante involucrar a los aplicativos ya que son los que conocen los sistemas, las tablas y procesos operativos donde nacen los datos.

Startup (Datio): El startup, resultado de la fusión entre BBVA y Stratio, se enfoca en el análisis y gestión de datos del sector bancario para adaptar productos a las necesidades de los clientes. La tecnología central que utilizan es Spark, proporcionada por Stratio. Gracias a su plataforma end-to-end, ofrecen una solución completa que cubre todas las etapas del ciclo de datos: desde la ingestión y almacenamiento hasta el análisis y la visualización.

Jira: Es una herramienta de visión general y de planificación al mismo tiempo y, por lo tanto, ayuda a optimizar el flujo de trabajo dentro de un equipo. Entre las características principales de Jira se encuentran la capacidad de crear y asignar tareas, el seguimiento del progreso del proyecto, la gestión de incidencias y errores, la personalización de flujos de trabajo y la generación de informes y estadísticas. Para ello, se crean tickets para tareas individuales, que se asignan y se procesan.

El software Jira demuestra ser especialmente idóneo para la aplicación de los métodos de gestión de proyectos Scrum y Kanban. Los proyectos pequeños, medianos e incluso muy extensos pueden llevarse a cabo de forma más eficaz con el software. Una función de información detallada también garantiza, en primer lugar, la documentación necesaria de un proyecto y, en segundo lugar, permite que los nuevos empleados se incorporen a un equipo existente de forma rápida y sencilla. (IONOS, 2022)

Folio: Es un código que asocia un proyecto y su fuente origen propuesta (registrada en el plan de gobierno y calidad de refinamiento), que es solicitado por el proyecto en la etapa de “Asignación de folio”.

BUI (Base Única de Ingestas): Es un repositorio donde se encuentra toda la información de las fuentes con sus respectivos dictámenes. Toda fuente tiene un folio que se asigna por cada ticket Jira validado,

por lo cual este repositorio estas compuesto de todas las fuentes creadas por los proyectos que deseen ir a Datio. Por ejemplo, en la figura 9, se observa todas las fuentes aprobadas por Core Assurance con sus respectivos códigos de Folio y el proyecto responsable de esas fuentes.

01. Base Única de Ingesta v0.2

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda

50%

Tecla lectura

2

</

Figura 9: Base Única de Ingestas. Fuente: Site Core Data

IDs TDS: Es un identificador que se le asigna a una fuente TDS que será ingestada a Datio o para los procesamientos que se crean en Datio.

SO (System Owner): Es el dueño del sistema/aplicativo y es clave para validar que las fuentes necesarias de información que utiliza el proyecto son confiables.

DO (Data Owner): Es el dueño del dato, encargado de garantizar la definición del dato y las reglas de calidad que son definidas por el Gobierno.

PO (Product Owner): Es el encargado de la validación de los campos y determina que campos son usados por el proyecto y si deberían subir a la capa master.

UUAA: Siglas para identificar la Unidad Aplicativa de una fuente que será ingestada a Datio. Asimismo, existe un listado de las UUAA que podemos utilizar para una fuente en el Mapa Funcional UUAA.

MSA Modelo de Solución de Arquitectura: Es un documento donde se detalla la solución técnica y de arquitectura como alcance para el funcionamiento de un proceso que será implementado por un proyecto.

Se mencionan el listado de UUAA's de entradas y salidas a utilizar en Datio en las secciones 2.2.1 y 2.1.7. Luego de haber finalizado el análisis funcional en caso no estén mencionadas todas las UUAA's que utilizará el proyecto, el Líder técnico es responsable de solicitar la actualización del MSA y seguimiento hasta su aprobación (Core Assurance no considera bloqueante esta aprobación final ya que es al final del análisis del proyecto).

Para atención de solicitudes donde se solicita el MSA debe estar como mínimo revisado por el SA (se considera las versiones draft siempre y cuando el SA lo haya revisado).

Site Core Data: Es una página donde están todos los lineamientos de como enviar ticket Jira al equipo de Core Assurance además de los requisitos que solicitamos para poder validar la información. También se podrá encontrar el concepto de cada uno de los términos que se tiene. Esta página es creada en un site de Google por cual permite saber cuántas personas ingresan a la página y si son nuevos usuarios.

3.4.2.4. Alcance organizacional

Data Engineering: Es un equipo de ingeniería a cargo de la gestión y la operatividad de la infraestructura informacional. Comprende varias áreas de ingeniería desde la generación de modelo de datos, también la revisión de la calidad de la data, etc.

Core Assurance: Es un equipo dentro de Data Engineering que se encarga de la revisión y validación de los modelados de datos que presentan los diferentes proyectos del banco. Este equipo revisa la documentación mediante lineamientos que hacen más confiable la información que se está generando dentro del banco. En general, Core Assurance se enfoca en garantizar que los sistemas de información de una empresa sean seguros y confiables, y que la información que manejan esté protegida de manera adecuada. Esto es particularmente importante en la era digital, donde las empresas dependen cada vez más de la tecnología para llevar a cabo sus operaciones diarias. Por ejemplo, en la figura 10, se observa a los encargados de los proyectos en el equipo de Core Assurance.

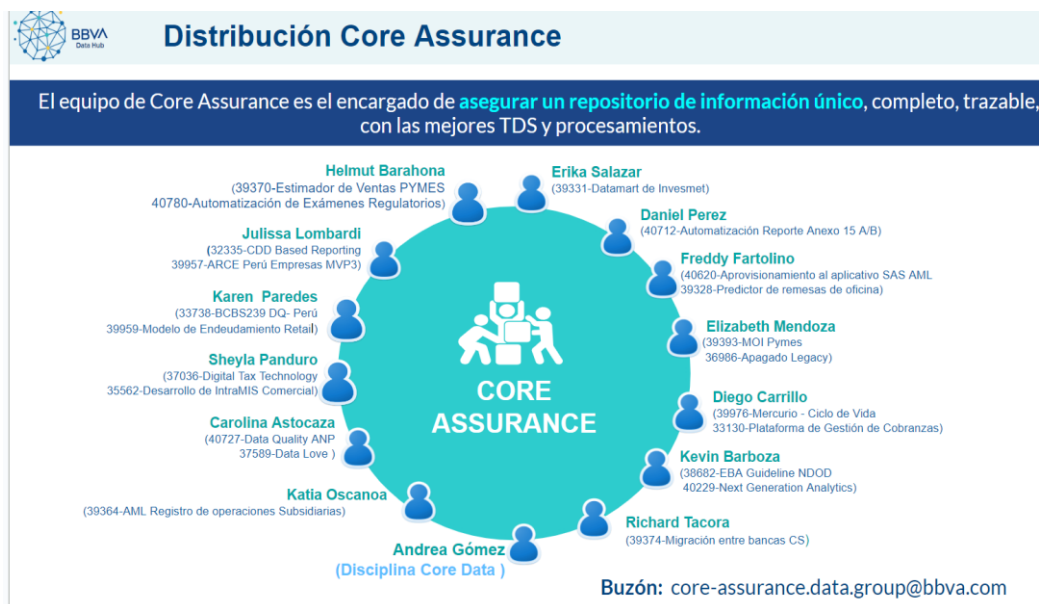


Figura 10. Distribución de Core Assurance. Fuente: Site Core Data

Flujo E2E: Es un enfoque de gestión de procesos que se centra en la visión completa de un proceso, desde el inicio hasta el final. Este enfoque está aplicado en el flujo de trabajo del proyecto de Core Assurance. El objetivo del flujo E2E es garantizar que los procesos se realicen de manera eficiente y efectiva, maximizando la satisfacción del cliente y minimizando los costos.

El flujo E2E se compone de varios pasos interconectados, que incluyen la identificación de los requisitos del proceso, el diseño y planificación del proceso, la ejecución del proceso, la supervisión y el control del proceso, y la mejora continua. Estos pasos están diseñados para garantizar que el proceso sea completo, preciso y efectivo en la entrega del producto o servicio final.

El flujo E2E se aplica en una amplia variedad de sectores, incluyendo el financiero, el de servicios, el de manufactura y el de tecnología, entre otros. Su objetivo principal es garantizar que los procesos se realicen de manera eficiente y efectiva, minimizando los costos y maximizando la satisfacción del cliente.

En resumen, el flujo E2E es un enfoque de gestión de procesos que se enfoca en la visión completa del proceso, desde el inicio hasta el final, con el objetivo de garantizar que los procesos se realicen de manera eficiente y efectiva, maximizando la satisfacción del cliente y minimizando los costos.

Por ejemplo, en la figura 11, se observa el flujo de análisis E2E que se tiene para la validación de ticket Jira que ingresan a ser resueltos por el equipo y los pasos son los siguientes:

- **Diagnóstico del dato:** Se valida el origen de la fuente.
- **Identificación de la TDS:** Se valida si aplica o no como TDS según los lineamientos del banco.
- **Relevamiento Funcional Técnico:** Se valida que información a nivel funcional contiene esa fuente.
- **Definición de Reglas de Calidad:** Se revisa que esta fuente sea nueva y la información no se esté duplicando.
- **Dictamen:** Se revisa la documentación que es llenada por los proyectos. Esta documentación se revisa según lineamientos del banco.
- **Aprobación Core Assurance:** Aprobación en el ticket Jira, además de volcar toda la información a una Base Única de Ingestas.

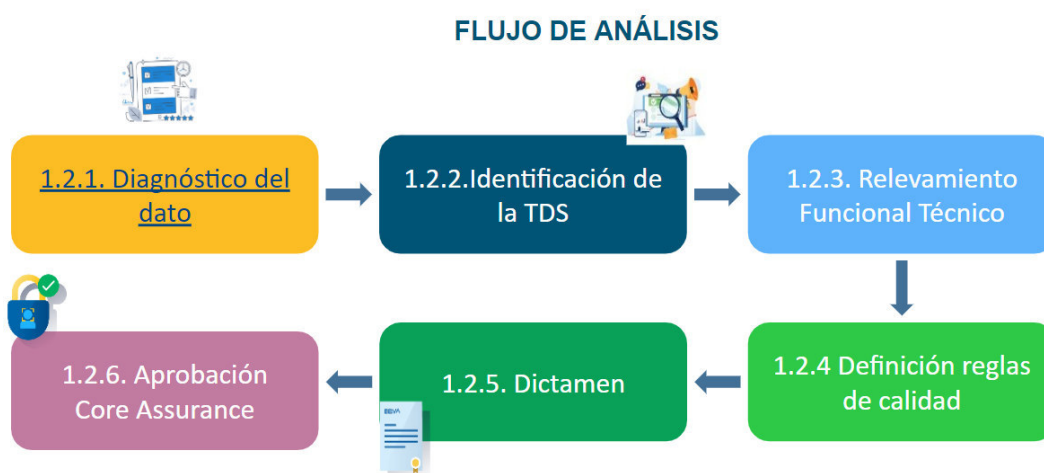


Figura 11. Flujo E2E. Fuente: Site Core Data

Correos de validación: Para cualquier validación de información de las fuentes pueden escalarlo y tener la validación del SO (System Owner), DO (Data Owner) o PO (Product Owner). Ya que por proyecto se debería tener asignado estos representantes. Por ejemplo, en la tabla 2, se observa que para una fuente nueva de origen Tabla DB2, se debe tener la conformidad del SO para los campos llave que estén en blanco, campos que suben a master y certificación de fichero histórico, además validación de PO si hay cambio en la periodicidad inicial y si hay campos que se desestimen en la subida de campos a Master.

Tipo de TDS	ROL	Confirmación TDS	Conformidad Llave	Campo Llave en blanco (5)	Campos que no Suben a Master (6)	Cambio Periodicidad /Historia - Ficha vs Dictamen (8)	Forma de Carga (9)	Certificación Fichero Histórico	Datos / Fuentes de la ficha que se desestima n
Tabla DB2 / Distribuidos	System Owner	No	No	Si	Si	No	No	Si	No
	Data Owner	No	No	No	No	No	No	No	No
	Product Owner	No	No	No	No	Si	No	No	Si
Fichero Ad Hoc	System Owner	Si	Si	Si	Si	No	No	Si	No
	Data Owner	Si	No	No	No	No	No	Si	No
	Product Owner	No	No	No	No	Si	No	No	No

Tabla 2. Correos de validación por parte del SO, DO y PO. Fuente: Site Core Data

3.4. Procedimiento

La circulación o flujo de los proyectos y la información que estas contienen o solicitan, conlleva una serie de procesos las cuales son descritas a continuación:

3.4.1. Origen de proyectos:

Inicialmente, los datos de los proyectos a analizar para su almacenamiento tienen su origen en distintas fuentes y a su vez pueden solicitar datos de diferentes tipos; esto se describe en los siguientes puntos:

3.4.1.1. Fuente de datos

La información de cada uno de los proyectos tiene un origen distinto. La fuente de estos datos debe poseer como característica primordial la confiabilidad. Las "fuentes de datos confiables" o TDS por sus siglas en inglés, son fuentes

de datos en los que puede confiar para procesar información, información utilizada para obtener conocimiento, tomar decisiones y actuar. Las TDS de la entidad financiera tratada en este trabajo incluyen: Oracle, MongoDB, SQL Server, tablas DB2 Host, Vsam, Datio Holding, Fichero Host Directo y Fichero Ad-Hoc.

3.4.1.2. Tipos de proyectos (cliente)

Existen proyectos que solicitan información, la cual puede ser de diferentes tipos de clientes como por ejemplo clientes pymes, clientes de solicitud de información de tarjeta, modelos de contactabilidad (solicitan número de teléfono, correo, etc. del cliente)

3.4.2. Modelamiento de proyectos:

El modelamiento es el proceso en el que toma la mayor presencia la tecnología del Big Data, mediante este se organiza la información implicada en los proyectos, en esta sección se lleva a cabo lo siguiente:

3.4.2.1. Recopilación y estructuración de proyectos

El Big Data en este proceso se encarga de recopilar toda la información y estructurarla de forma que sea más hacedero o ágil su posterior revisión y análisis por los analistas de Core Assurance.

3.4.3. Recepción y evaluación de proyectos:

Después del proceso de modelamiento, los proyectos estructurados por Big Data son enviados a través de tickets por medio de un software de gestión de proyectos llamado JIRA y recepcionados por el equipo de Core Assurance para su respectiva evaluación o análisis en base a los lineamientos establecidos por la entidad financiera; la evaluación a llevarse a cabo dependerá del tipo de ticket, estos pueden ser:

3.4.3.1. Ticket Entrada TDS:

Estos tickets son ingestas de datos los cuales tiene su origen en fuentes TDS. Por ejemplo, en la figura 12, se valida el flujo de revisión que se hace para una nueva fuente TDS, la evaluación de estas se realiza de la siguiente manera:

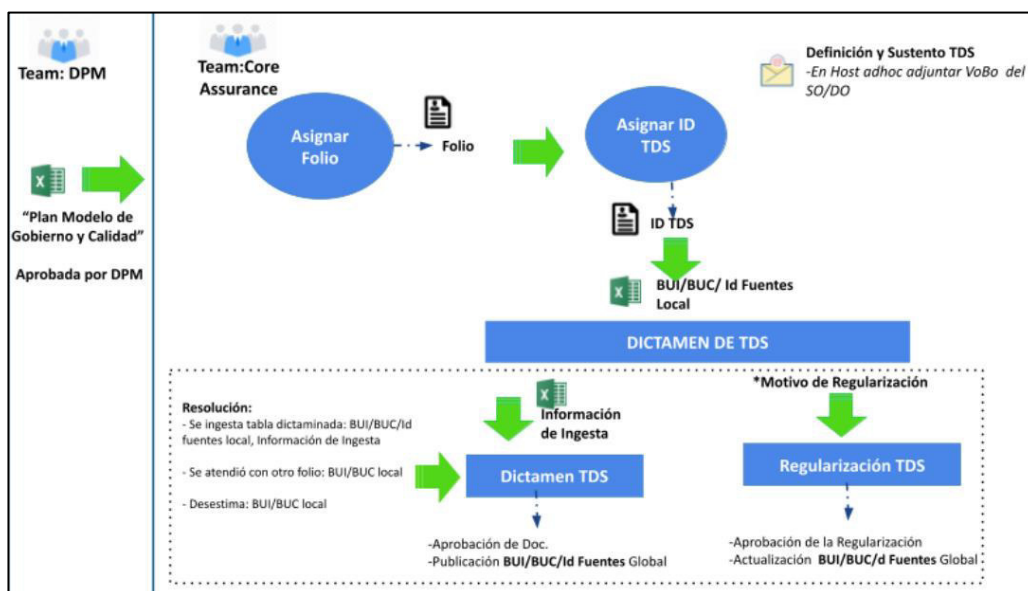


Figura 12: Flujo de Entrada TDS. Fuente: Site Core Data

a) Revisión de ficha:

La evaluación de los tickets TDS se inicia con la revisión de fichas, la cual comprende una serie de tareas o lineamientos, encargadas de revisar la descripción funcional de los datos que contiene la ficha del proyecto, entender y validar el dato y su historia.

1. Revisión de conformidad del área de DPM según la prioridad que haya solicitado revisar el proyecto.
2. Elaboración y depósito de una copia de la ficha en el repositorio de revisiones de fichas por proyecto, y se concatena el número de prioridad y la fecha de revisión. Luego se realiza la revisión y se registra en el archivo Excel de observaciones, tenga o no observaciones la ficha que se esté revisando. Igualmente, se registra la solicitud en el Jira.

3. Consideración que los criterios de validación de la lista de comprobación o verificación llamado Check List (códigos de campo únicos, en la columna de datos, observación técnica, etc) estén correctos.
4. Verificación de la no existencia de elementos duplicados respecto a los grupos y a los datos funcionales, ya que estos últimos tienen que ser precisos y no tan genéricos.
5. Revisión sobre que las descripciones funcionales se detallen adecuadamente teniendo en cuenta la especificidad, la frecuencia y el nivel en el que se están solicitando los datos.
6. Una vez finalizada la revisión, el Excel de observaciones se comparte en el drive central a fin de actualizar las columnas respectivas conforme se van cerrando las observaciones.
7. Finalizada la revisión se guarda la foto de la ficha aprobada en el drive de aprobaciones concatenando al nombre el número de prioridad y fecha, actualizando el Excel de observaciones el estado. Asimismo, se actualiza en el Jira el estado de la solicitud a aceptada.

b) Asignación de folio:

Después de la revisión de ficha, se realiza la asignación de folio. Por ejemplo, en la figura 13, se valida el flujo de revisión de la asignación de folio de la TDS nueva, en el que prácticamente se realiza una revisión poca profunda de los dictámenes de los proyectos. Los lineamientos en los que se basa la asignación de folio son:

1. Registrar las peticiones de los proyectos en Jira.
2. Validar que la 'fuente sugerida' solicitada que esté solicitando el folio, se encuentre en la más reciente versión aprobada de la ficha aprobada por el equipo de Core Assurance, según la prioridad a la que corresponde.
3. Verificar que no exista en la BUI (Base Única de Ingestas) principal la misma fuente sugerida para el proyecto, a fin de garantizar que no se produzcan duplicados en la generación de folios por errores en las solicitudes de proyectos.

4. Registrar por medio de Excel las observaciones realizadas y en caso de haberlas se llena la hoja de observaciones del proyecto y se cambia el estado del ticket de Jira a 'Bloqueado' y en el comentario se adjunta el link de la base de observaciones, para que el proyecto (origen) las corrija.
5. En caso esté todo OK, colocar en la BUI Global el folio correlativo que le corresponde, y se llenan además las columnas: 'SDATOOL-Nombre Proyecto', 'Nombre Tabla, Fichero o Concepto Original', 'Descripción Tabla, Fichero o Concepto Original', 'Histórico solicitado (meses)', 'Periodicidad solicitada', 'Fecha de Registro Folio', 'Estatus resolución' (En proceso) y 'Tipo' (Entrada o Salida).
6. Se ingresa en los comentarios del Ticket etiquetando al proyecto para informar sobre el folio asignado a la fuente. Se cierra la solicitud y el Ticket se va a 'Ready To Verify' y finalmente "Deployed". Todo esto se realiza en el Jira.

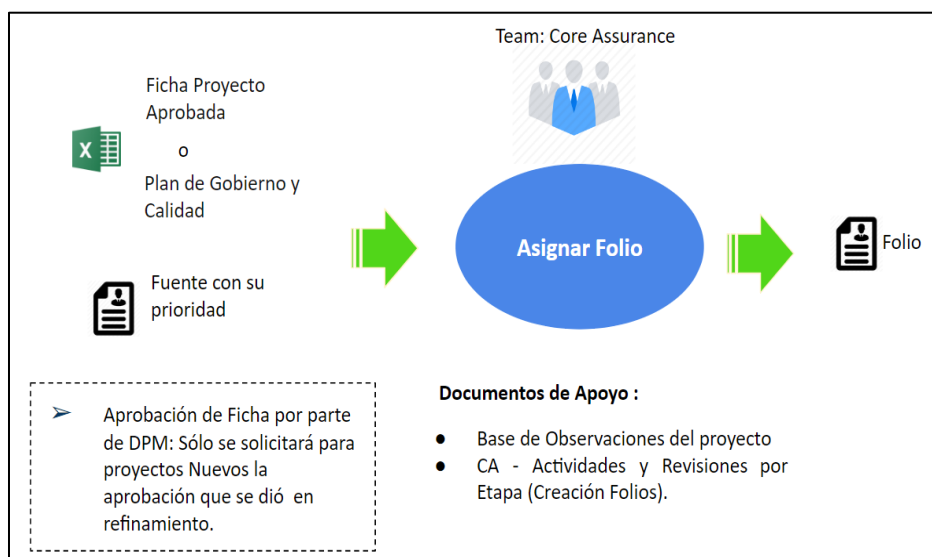


Figura 13 :Flujo de Asignación de Folio.Fuente: Site Core Data

c) Asignación de ID TDS:

Concluida la asignación de folio, se procede a realizar la asignación de un ID. Por ejemplo, en la figura 14, se visualiza el flujo de la asignación ID de una fuente nueva, la cual se realiza en base a los siguientes lineamientos:

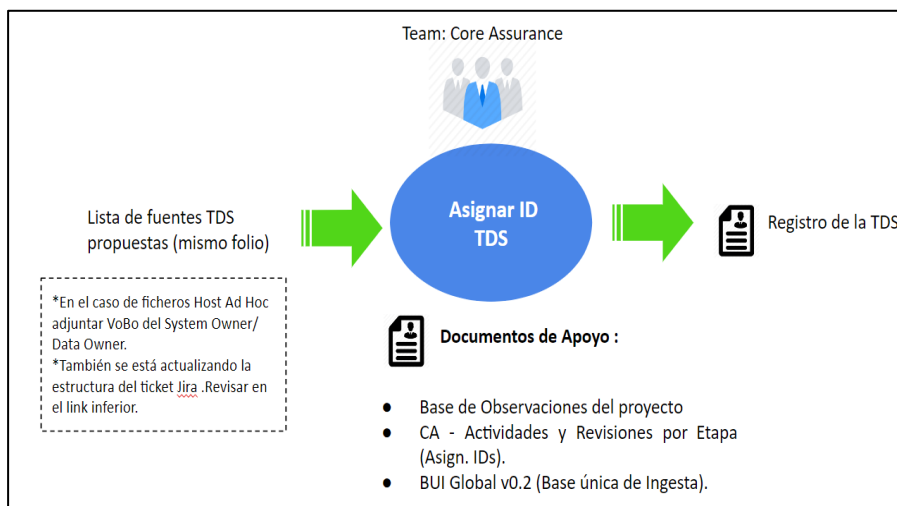


Figura 14 :Flujo de Asignación de ID. Fuente: Site Core Data

1. Registrar pedido en el Excel de observaciones del proyecto, en la pestaña 'Resumen'.
2. Validar que la solicitud contenga estos datos:
TDS propuesta | Evidencia/Sustento de TDS | Folio al que se asociará esta fuente | Descripción funcional | Origen
*En el caso de Launchpad, nos deben adjuntar en el MSA, los campos y una muestra de la fuente, para poder asignarle una UUAA.
3. Validar el folio:
 - Que el folio se encuentre registrado en la BUI: Si en el ticket en Jira no hay folio, o el folio indicado no existe en la BUI para alguna de las fuentes solicitadas, se cambia el estado del ticket de Jira a 'Bloqueado' y en el comentario se adjunta el link de la base de observaciones, para que el proyecto las corrija.
 - Que la fuente sugerida en este folio haga sentido con la TDS propuesta (podrían validarse las descripciones)
4. Validar la fuente TDS:
 - Que tenga el nombre completo (a veces se omite el PEBP).
 - Que el Fichero no contenga la palabra TMP como parte de su nomenclatura.
 - Que sea un fichero de producción (a veces ponen PEBT o PEBD).
 - Que sea solo un objeto (a veces combinan el fichero con la tabla DB2).
 - Que el fichero ya se encuentre productivo (no ficheros en proceso de creación), en este caso la TDS es la tabla.

- Que no haya sido solicitada anteriormente por otro proyecto (para evitar cruces de análisis e ingesta).
- Revisar que la información de la TDS solicitada, no se encuentre contenida o repetida en otra TDS (sin deuda) ya dictaminada. Para este análisis nos basamos en la descripción de la fuente.

5. Validar evidencia de TDS:

- Host Descargas directas de una tabla: Adjuntar pantallazo del JCL donde se pueda observar el query de descarga y el nombre del fichero. Si el fichero pertenece al módulo IL, se debe consultar al proyecto si ya está productivo y enmallado (si el proyecto dice que SI, validar con Frente único, consultándole la última fecha de ejecución del fichero en cuestión).
- Host no existe descarga: Adjuntar la estructura completa de la tabla DB2.
- Host Ficheros Ad-hoc:
 - Correo de certificación del System Owner (respaldo de engineering) y del DO (Dueño de la data) (la lógica 'funcional' para la generación de estos datos en el adhoc de host son las vigentes).
 - Adjuntar el flujo de generación de este, que permita sustentar su complejidad (procesos, rutinas, dependencias, etc.). Tener en cuenta que el flujo debe quedar claro para nosotros, los procesos o rutinas deben tener una explicación clara de lo que realiza. Asimismo, para su certificación como TDS debe tener el VoBo de los scrum de análisis (Syncro de Análisis (Site Core Data)).
- Host - VSAM: En caso exista una copia completa del mismo, adjuntar pantallazo del JCL (paso donde se realiza la copia o descarga).
- Distribuidos: Adjuntar el query para descargar la tabla y/o estructura de la tabla.
- Cartuchos, backups históricos: Correo del aplicativo indicando el nombre del objeto que se utilizará para ingestar historia.
- Launchpad: Que el fichero este en el MSA indicando que debe ir por Launchpad (Estructura de la parametría y VoBo Mesa de data).

6. Validar la descripción funcional:

- Tiene que ser clara, detallada y específica

- Validar en la medida de lo posible la fiabilidad de la descripción a nivel funcional.
- La definición de la fuente debe ser agnóstica al uso del proyecto.
- Que no tenga texto de relleno que no aporta: ""Tabla que contiene información..."", ""Fichero que contiene los datos...""

7. Validar origen:

Existen diferentes orígenes que se consideran TDS, como podemos ver en la figura 15 se observa todos los orígenes que son considerados por el equipo de Core Assurance y se siguen los siguientes lineamientos:

- Si la TDS es un fichero host que descarga una tabla DB2, el origen es la tabla DB2.
- Si la TDS es un fichero host Ad-hoc, en caso se tengan identificadas las tablas DB2 origen, estas deben listarse en el origen. Si por lo complejo, no se tiene la lista completa de tablas, colocar 'Fichero Ad-hoc'.
- Si la TDS es un Vsam (y no existe una descarga de este), en origen colocar: 'VSAM'. Si existiese un fichero que descargue el VSAM, este fichero sería la TDS y en origen se colocaría el nombre del Vsam.
- Si la TDS es una tabla DB2 (ya que no se ha encontrado una descarga directa de la misma, y debe desarrollarse). en origen colocar: 'Tabla DB2'.
- En el caso de un fichero que viene de un sistema distribuido, se debe colocar "[Nombre de la Tabla Origen]/ [Base de Datos]/[Aplicativo]".
- Si la TDS es una tabla que viene de un sistema distribuido y no existiese descarga, la TDS sería dicha tabla y en origen colocar "[Base de Datos]/[Aplicativo]".
- Si la TDS vendrá por Launchpad, en origen se debe colocar: "Launchpad".

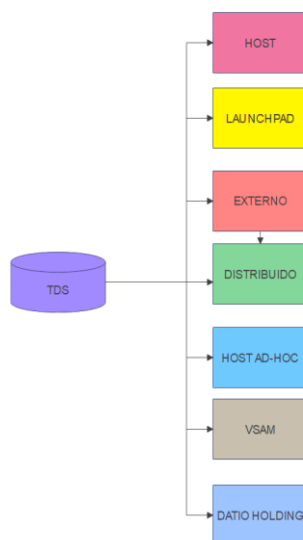


Figura 15:Tipos de fuentes según el origen. Fuente: Elaboración propia.

8. Validar Owner Modelo:

- Si la TDS corresponde a un modelo global se debe indicar el nombre del modelo: Alpha, CDD, CIB, Client 360, T&C Advanced Analytics.
- Si la TDS es un proyecto global se indica "Proyecto Global".
- Si la TDS es local se indica: "Modelo Local".

9. En caso se encuentren observaciones. Se llena la hoja de observaciones del proyecto (pestaña: "Ficha/Folio/Id") y se cambia el estado del ticket de Jira a 'Bloqueado' y en el comentario se adjunta el link de la base de observaciones, para que el proyecto las corrija.

10. En caso esté todo OK, llenar la pestaña 'Id Fuentes' de la BUI, con todos los datos, de la siguiente manera:

- Si la TDS es una fuente nueva, utilizar un 'Id libre' o en su defecto utilizar el próximo id correlativo. Se sombrea en celeste y se coloca el estado 'ID Asignado'.
- Si es una fuente que se va a reingestar (es la misma fuente), se crea el ID con 'xxx.1' debajo del registro con el id original. Se sombrea en celeste y se coloca el estado 'ID Asignado'.
- En caso esta fuente (sea nueva o reingestada), reemplace otra TDS, el registro reemplazado se sombrea en amarillo, pasa al estado 'Reemplazado' y en 'Id Reemplazo' se coloca el ID de la TDS que sustituye esta fuente (si son muchas fuentes las que

reemplazarán esta fuente, se colocan los IDs separados por coma y espacio, de esta forma: ""12, 15, 25.1, 109"".

- En caso se ingesta una fuente de holding de tablas auxiliares se debe asignar el ID con 'xxx.2', 'xxx.3', 'xxx.4' debajo del registro del id original de la tabla principal de holding. Se sombrea en celeste y se coloca el estado 'ID Asignado'
- En el caso de una fuente que ira por Launchpad, se debe registrar en la columna de 'observaciones' la UUA que se le esta asignando a la fuente.

11. El equipo de Core Assurance deberá completar las columnas de "ID Fuentes (Source)", según lo indicado en la "01. Base Única de Ingesta v0.2", pestaña "Guía V2", sección "3.- Detalle de columnas de IDs Fuentes", columna "responsable Registro", correspondiente a "Procesamiento" y "Fuentes TDS".

12. Se ingresa en los comentarios del Ticket etiquetando al proyecto para informar sobre el ID asignado a la fuente. Se cierra la solicitud, el Ticket se va a 'Ready To Verify' y finalmente "Deployed"

d) Revisión de Dictamen TDS

Finalizado la asignación de ID, se procede a revisar y evaluar los dictámenes TDS, en la figura 16 se observa el flujo de revisión de los dictámenes y aquí mencionaremos los siguientes lineamientos que deben tener los proyectos para generar el ticket:

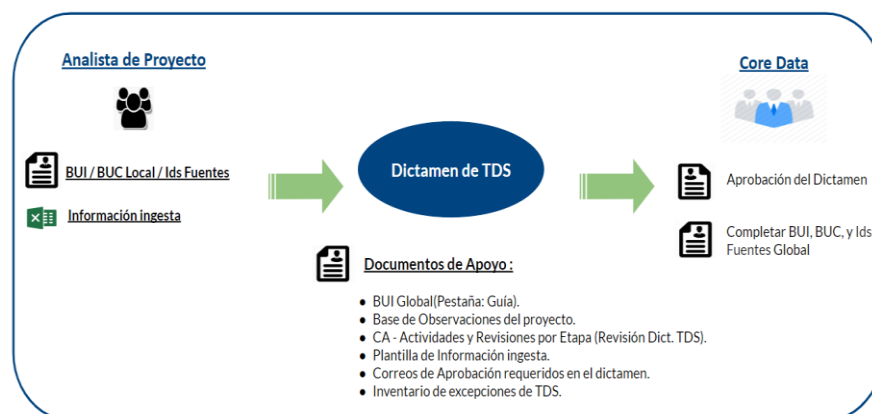


Figura 16 :Flujo de Dictamen TDS. Fuente: Site Core Data

1. Registrar pedido en el Excel de observaciones del proyecto, en la pestaña 'Resumen'.

2. Validar que la solicitud contenga estos datos: Folio | IDS | TDS propuesta y se adjunte el link de la BUI local del proyecto.
3. Guardar una copia de la BUI local en la unidad privada de CA: "02. Revisiones x Proy / [Proyecto] / BUI Local revisada" con el nombre: "YYYY-MM-DD - [Proyecto] - Base Única de Ingesta - CAXXXXXX (1ra Rev.)". En cada revisión, si hay rebotes del mismo grupo, guardar una nueva copia en la misma carpeta con la nueva fecha y revisión.
4. Revisar el correcto llenado de la BUI (BUI, BUC e ID Fuentes), Excel Información de Ingesta (Excel no aplicaría para casos de reúso); según lo indicado en la guía de la BUI y en cada plantilla de la documentación. Validar:
 - La cantidad de registros de la muestra productiva sea igual a la enviada por el equipo de host: deberá ser igual a 10000 registros; en caso de que el fichero no llegue a los 10000 registros se deberá solicitar una descarga completa.
 - En el Excel de Información de Ingesta, tenga adjunto el sustento de la llave (Link de la carpeta Anexos) y los campos que la componen estén marcados en el Excel (Estructura TDS)
 - En el caso de una fuente que no sea considerada TDS según lineamientos llamada "localito", se debe tener en cuenta que las fuentes localitos no son consideradas TDS por lo cual su uso se debe tener como sustento de creación un correo con el VoBo de la Gerencia.
 - En el caso de una regularización y nuevo versionado, en la bitácora debe estar de manera específica qué es lo que se ha actualizado y/o agregado en esta nueva versión.
5. Para validar los casos de reúso:
 - No aplica la revisión del documento (Excel Información de Ingesta).
 - Validar que la fuente a reusar, se encuentren en los siguientes Estados en los tableros de Seguimiento de ingestas: Tablero 02, columna "Estatus de la Tabla"; Siempre y cuando sea un reúso de entrada, en caso se vaya a reusar una fuente de procesamiento (Salida) recurrir al Tablero 03. columna "Estatus de la Tabla"):
 - Validación QA
 - Componentes en Live
 - Pendiente Validación PO

- Carga Histórica
 - Gestión Control M
 - Completa
 - En Reliability
6. En caso se ingesta historia desde una PDCO se debe perfilar los campos de la historia con la estructura de la tabla desde el sandbox.
 7. En caso se encuentren observaciones. Se deberá llenar la hoja de observaciones del proyecto (pestaña: "Dictamen") y se cambia el estado del ticket de Jira a 'Bloqueado' y en el comentario se adjunta el link de la base de observaciones, para que el proyecto las corrija. En el caso que sea Procesamiento en la columna tipo utilizar el label de "Doc. Procesamiento" para las observaciones en dicho documento.
 8. Cuando todo este correcto se debe volcar la información a la BUI con estas consideraciones:
 - Dejar el campo 'Estatus resolución' en finalizado.
 - Actualizar la fecha de resolución con la fecha en la que se está haciendo el volcado a la BUI central y también el campo 'Q-Año-SP de termino y/o Pase a Ingesta'.
 - Completar la columna ""Colaborador CA que aprobó el dictamen"" con el nombre del analista CA.
 - El campo 'ID Tabla' debe formatearse como entero para que pase al tablero de ingesta (si es xx.1, se debe mantener 1 decimal, sino sin decimales. Si es de miles se quita el separador de coma).
 - Verificar que la fecha de pase al tablero (columna X) tenga el formato correcto (dar doble click y si parece el calendario esta OK). Si esto está mal, no pasará al tablero.
 - Se asigna la UUAA master en la pestaña Id Fuentes, completando los 3 campos de la sección ""Mapa Funcional"". En el caso de Launchpad, la UUAA ya debe estar definida en 'Comentarios TDS' de la pestaña 'Id Fuentes'.
 - En la BUC solo actualizar: 'Estatus resolución' y 'ID Tabla' debe formatearse en entero.
 9. En el caso el dictamen se finalice como 'Se ingesta tabla dictaminada' o con otro tipo de dictamen en el que se necesite pasar el folio al tablero (pe: cuando se va a llevar más historia, o se van a implementar nuevas reglas de claridad), se deben llenar las columnas 'Pasa a Dashboard de Ingesta' con el valor 'Si', y 'Fecha Pasa a Dashboard de Scrum de

Ingesta' con la fecha en que se hace el volcado final a la BUI centralizada.

10. Si el proyecto va a terminar la Ingesta que inició otro proyecto (pero no llegó a concluirla), el folio anterior debe actualizarse. Dicho dictamen debe quedar como 'Se atendió con otro folio', el folio referenciado se debe llenar, se debe actualizar el link de documentación. En 'Comentario Resolución' explicar la actualización.
En el caso de la UUAA, si no hubo avance de diccionariado o ingesta, y vemos que la UUAA puede ser otra, se coordina para hacer el cambio en este nuevo dictamen.
11. Toda la carpeta en donde se encuentra la documentación del dictamen que el proyecto ha trabajado ubicado en: Proyectos - Dictámenes y Documentos\ 01. Dictámenes de fuentes \ [Proyecto].
Se debe mover a la carpeta congelada en donde se encuentra la documentación final del dictamen ubicada en: DH - Dictámenes y Documentos \ 01. Dictámenes de fuentes \ [Proyecto].
Esta última carpeta debería tener sólo las últimas versiones de los dos documentos de dictamen en la raíz, toda documentación adicional debe colocarse dentro de la carpeta: 'Anexos', y si hubiera versiones anteriores de los documentos de dictamen, por correcciones realizadas en la validación, deberán ubicarse dentro de la carpeta: 'Old'
12. Finalmente se ingresa en los comentarios del Ticket etiquetando al proyecto para informar la aprobación de dictamen del folio solicitado y la asignación de UUAA de la TDS. Se cierra la solicitud y el Ticket se va a 'Ready To Verify'. En la respuesta al proyecto se debe adjuntar el link de la carpeta que contiene el dictamen, y un pantallazo de los 2 documentos en la unidad congelada.
13. En la medida de lo posible, en el caso de una nueva ingesta o cuando la columna 'Pasa a Dashboard de Ingesta/Procesam.' diga SI: validar que el registro pase al tablero de ingesta. Se cierra la solicitud se pasa el Ticket a 'Ready To Verify' y finalmente "Deployed".
14. Finalmente, luego de haber aprobado el dictamen, ir a la pestaña "Id Fuentes" de la BUI y actualizar las columnas "Falta Certificación TDS", "Falta Perfilamiento de campos", "Falta o Dictamen incompleto", "Usa PDCO / Procesamiento PDCO", "Owner Modelo Global", ""Tipo Ingesta", "Nivel Deuda" y "ID Fuente Heredada" según lo revisado en el id fuente correspondiente. Asimismo, completar la columna ""Tipología"" se especifica que tipo de fuente es:

- Oro: Para TDS son todas las tablas host, distribuido, Datio Holding, descargas directas de tablas, Ad-hoc que no son remplazadas en el tiempo porque su origen de información es fiable. Para procesamientos se considera Oro en el caso que no cambien en el tiempo.
- Plata: Para TDS que con el tiempo si se va a reemplazar. Para procesamientos antiguos que aún no tienen apagado. Por ejemplo, las PDCO.
- Bronce: Para TDS que no se deben ingestar a Datio.

Tener en cuenta que Core Assurance deberá completar las columnas de "ID Fuentes (Source)", según lo indicado en la "01. Base Única de Ingesta v0.2", pestaña "Guía V2", sección "3.- Detalle de columnas de IDs Fuentes", columna "responsable Registro", correspondiente a "Fuentes TDS".

En el caso que el dictamen la resolución es ""Se ingesta tabla dictaminada"" se debe actualizar en ID Fuentes la columna Status 'Dictamen Aprobado', en el caso que se desestime y no haya ningún dictamen de que se ingeste esa fuente y su resolución sea ""Desestimado"" se debe actualizar en ID Fuentes la columna Status 'Desestimado' y en comentarios colocar el motivo.

En caso la fuente a dictaminar se haya revisado que ya se encuentra ingestada, se debe actualizar en ID Fuentes la columna Status 'Ingestado' (cuando en el tablero 02 ó 03 se encuentre el folio e Id en estado "Completa, En Reliability").

3.4.3.2. Ticket de Procesamiento:

Un procesamiento es un proceso que es generado en Datio, según la necesidad del usuario y definida por el Solutions Architect en el MSA. Estos tickets para procesamiento también son generados en el Jira para su posterior revisión. En la figura 17 podremos observar el flujo de todo el procesamiento desde la validación de la salida persistente hasta su documentación solicitada.

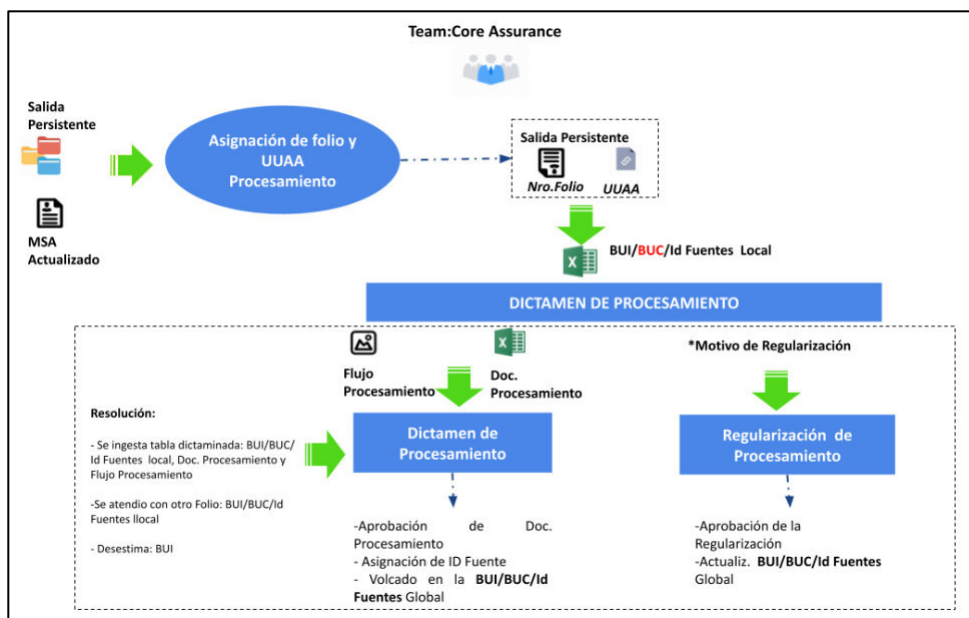


Figura 17: Flujo de Procesamiento. Fuente: Site Core Data

a) Asignación de folio de procesamiento:

La evaluación de los tickets de procesamiento se inicia con la revisión del MSA y la estructura de la fuente, por lo cual se sigue los siguientes lineamientos para asignar el folio y la UUAA correspondiente, el flujo se muestra en la figura 18:

- 1.Registrar pedido en el excel de observaciones del proyecto, en la pestaña 'Resumen'.
- 2.Validar que en la solicitud de pedido en Jira contenga todos (Nombre de salida, descripción, detalle de campos y pg. del MSA).
- 3.Se validará que las fuentes cuenten con una descripción detallada, concisa y entendible para poder brindar las UUAA's. Además, validar que la versión del MSA adjunto sea la última vigente.
- 4.Validar en la BUI que la Salida no haya sido solicitada en un pedido anterior, ya que estaríamos creando 2 folios para la misma Salida y para el mismo proyecto.
- 5.Guardar una copia del documento que contiene la lista de campos con sus definiciones en la unidad privada de CA: "02. Revisiones x Proy /

[Proyecto] / Procesamiento/ Creación de Folios" con el nombre: "YYYY-MM-DD - [Proyecto] - Estructura Procesamiento - NombreSalida (1ra Rev.)" (Si el proyecto adjunta distintos documentos, colocar cada uno en una pestaña independiente).

6. En cada revisión, si hay rebotes del mismo grupo, guardar una nueva copia en la misma carpeta con la nueva fecha y revisión. También hay que guardar una copia del MSA en la misma ruta.
7. En caso se encuentren observaciones. Se llena la hoja de observaciones del proyecto (pestaña: "Ficha/Folio/Id") y se comenta el ticket adjuntando el link que contiene las mismas, para que el proyecto las corrija.
8. En caso esté todo OK, colocar en la BUI el folio correlativo que le corresponde.
9. La UUAA se deberá asignar de acuerdo con la definición de la tabla y la descripción de los campos.
10. Se ingresa en los comentarios del Ticket etiquetando al proyecto para informar sobre el folio asignado a la fuente y su UUAA correspondiente. El Ticket se va a 'Ready To Verify' y finalmente "Deployed"
11. Se debe validar que el registro haya pasado al Tablero 03. Hay que considerar que, en este momento, en el tablero 3 las columnas 'ID Tabla', '#campos' y 'URL Carpeta con documentación Funcional /Técnica' se encontrarán vacías, pero posteriormente serán llenadas automáticamente cuando se cierre el dictamen.

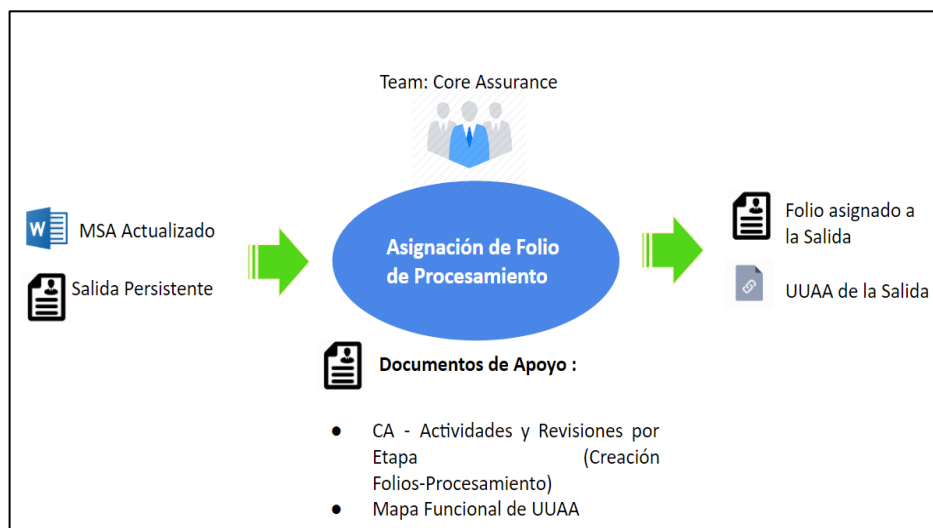


Figura 18: Flujo de Asignación de folio de Procesamiento. Fuente: Site Core Data

b) Revisión de Dictamen de procesamiento:

La evaluación de los tickets de dictamen de procesamiento es con la previa asignación del folio y de la UUAA, en la figura 19 se puede visualizar el flujo de revisión del procesamiento, en este paso se sigue los siguientes lineamientos:

1. Registrar pedido en el excel de observaciones de proyecto, en la pestaña 'Resumen'
2. Validar que el ticket de JIRA contenga estos datos: Folio, Fuente, Descripción, Diccionario en Sophia (Sophia es una página del banco donde están todos los procesamientos ya validados, esta validación la hace el área de Arquitectura del banco).
3. Guardar una copia de la BUI local en la unidad privada de CA: "02. Revisiones x Proy / [Proyecto] / Procesamiento " con el nombre: "YYYY-MM-DD - [Proyecto] - Base Única de Ingesta - CAXXXXXX (1ra Rev.)". En cada revisión, si hay rebotes del mismo grupo, guardar una nueva copia en la misma carpeta con la nueva fecha y revisión.
4. Revisar que la plantilla del documento sea la última versión, el link de acceso a la plantilla se encuentra en la BUI Local.

5. El ítem descripción (documento Excel) debe ser el misma registrada en la BUI central, cuando se asignó el folio.
6. En ítem Periodicidad indicar de manera específica en qué momento se ejecuta el procesamiento, por ejemplo, si es mensual es en fin de mes o es al cierre contable.
7. En el caso de un procesamiento que proviene de origen Datio holding y tenga lógicas encapsuladas no se tendrá la lista de campos. En caso de que si se tenga mapeo de campos cuando indican que es NULL debe indicarse el motivo en la columna observación del campo.
8. Validar que los inputs indicados en el documento de procesamiento se encuentren en los siguientes Estados en los tableros de Seguimiento de ingestas (Tablero 02, columna "Estatus de la Tabla") o de Seguimiento de Procesamientos (Tablero 03, columna "Estado de la tabla") y que estén relacionados al proyecto, porque un proyecto no puede usar un input que no haya sido mapeado previamente:
 - Validación QA
 - Componentes en Live
 - Pendiente Validación PO
 - Carga Histórica
 - Gestión Control M
 - Completa
 - En Reliability.
9. Las Relaciones Entre Fuentes del documento de procesamiento, los nombres de los campos indicados en la columna "Campos (On)", deben ser los nombres exactos, no usar abreviaturas ni otra denominación.
10. La columna Query es directo si el dato proviene de una sola tabla fuente directa, sin transformación ni lógicas de por medio.
11. Si en la lógica implica más de 2 tablas se debe especificar tabla. campo para entender que campo de x tabla está involucrado. Tener en cuenta que estos campos deben ser de Datio.
12. Las relaciones indicadas deben ser las mismas indicadas en el flujo, así como los nombres de las tablas inputs y la de salida.
13. Validar que los campos de las tablas orígenes, mencionados en la columna "Campo de la Tabla / Lógica de Carga" del documento de

procesamiento, existan exactamente con el mismo nombre. Adicional revisar que la estructura de los campos sea igual tanto en orden (Validar en el diccionario la columna "FIELD POSITION IN THE OBJECT"), nombres, llave.

Esta validación se puede hacer buscando en Sophia, el detalle del nombre físico del campo.

Asimismo, considerar en los modelos globales las abreviaciones de los namings con lo que estén mapeando debe tener relación:

Los namings que comienzan con g_ son namings globales y los que comiencen con gf_ son namings locales. La estructura de campo del código cliente global siempre debe ser PE+Entidad+0+Código cliente (código de 8). Para el caso del contrato global es PE+current_contract_id (código de 18) y según lo definido por holding el contrato local se poblará igual al global (PE+ current_contract_id).

14. Una vez esté todo correcto, se hace el volcado de la información de la BUI/ BUC global además se asigna un ID al procesamiento (en la pestaña "Id fuentes"), este ID servirá para actualizar el nombre de la documentación (inicialmente sólo se envía con el número de folio y el ID_Fuente entre corchetes, reemplazar este valor). Se completa las columnas en la pestaña ID_Fuente: "Descripcion de la TDS", "Fuente TDS", "Origen" (con el valor: Procesamiento), "Master" (con la ruta Master que se encuentra en el documento de procesamiento enviado) y Status (con el valor: 'Dictamen Aprobado')
15. Toda la carpeta en donde se encuentra la documentación del procesamiento que el proyecto ha trabajado ubicado en: Proyectos - Dictámenes y Documentos\ 02. Procesamientos \ [Proyecto]:
Se debe mover a la carpeta congelada en donde se encuentra la documentación final del procesamiento ubicada en: DH - Dictámenes y Documentos \ 02. Procesamientos \ [Proyecto]
16. Finalmente se ingresa en los comentarios del Ticket etiquetando al proyecto para informar la aprobación de dictamen. Se cierra la solicitud y el Ticket se va a 'Ready To Verify' y finalmente "Deployed". En la respuesta al proyecto se debe adjuntar el link de la carpeta que contiene el procesamiento y un pantallazo de los 2 documentos en la unidad congelada.
17. Finalmente, luego de haber aprobado el dictamen, ir a la pestaña "Id Fuentes" de la BUI y actualizar las columnas "Falta Certificación TDS", "Falta Perfilamiento de campos", "Falta o Dictamen incompleto", "Usa PDCO / Procesamiento PDCO", "Owner Modelo Global", "Tipo Ingesta",

"Nivel Deuda" y "ID Fuente Heredada" según lo revisado en el id fuente correspondiente.

*Tener en cuenta que Core Assurance deberá completar las columnas de "ID Fuentes (Source)", según lo indicado en la "01. Base Única de Ingesta v0.2", pestaña "Guía V2", sección "3.- Detalle de columnas de IDs Fuentes", columna "Responsable Registro", correspondiente a "Procesamiento".

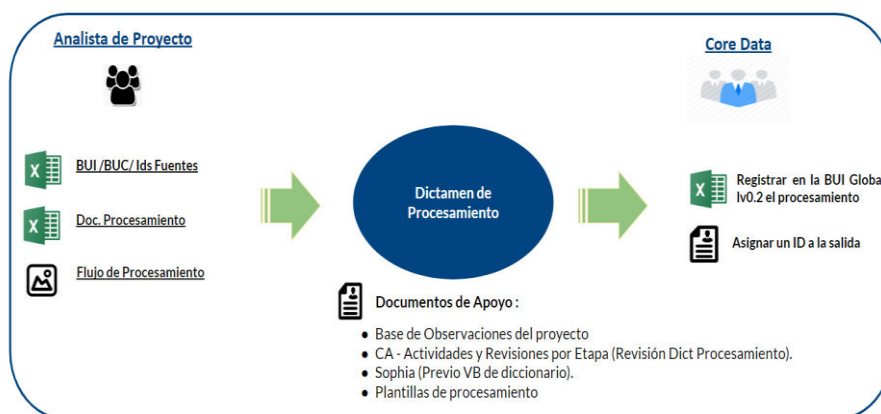


Figura 19: Flujo de Dictamen de Procesamiento. Fuente: Site Core Data

3.4.4. Aprobación y almacenamiento de proyectos

3.4.4.1. Aprobación de fuente:

Cuando una fuente ya sea procesamiento o TDS de información ha sido sometida al proceso de revisión y todos los lineamientos establecidos por la entidad han sido verificados y cumplidos de manera satisfactoria, se considera aprobada. Para formalizar este proceso, se utiliza el sistema de seguimiento de tickets Jira, donde un ticket específico es creado y asociado a cada fuente en evaluación.

Una vez que se ha verificado la conformidad con los lineamientos, el ticket en Jira es validado, y se actualiza su estado para reflejar la aprobación de la fuente. Este nuevo status indica que la fuente ha pasado por todas las etapas requeridas y cumple con los estándares de calidad y consistencia.

Con la aprobación del ticket en Jira, la información contenida en la fuente, así como toda su documentación de dictamen, es transferida a la BUI (Base Única de Información). La BUI es un repositorio centralizado que almacena y gestiona todas las fuentes de información aprobadas, asegurando que estén disponibles para su uso por parte de los usuarios autorizados.

Este proceso de aprobación y transferencia garantiza la integridad y trazabilidad de la información, permitiendo que los datos validados sean utilizados en la toma de decisiones y análisis dentro de la entidad, brindando mayor confiabilidad y seguridad en el uso de los datos almacenados en el Big Data del banco.

3.4.4.2. Status de los tickets:

En Jira, se emplean diversos estados para gestionar el progreso de las tareas o incidencias del proyecto Core Assurance. Cada estado representa una etapa específica del ciclo de vida de una tarea, desde su creación hasta su resolución final. Los estados permiten tener una visión clara y organizada del estado actual de cada elemento en el flujo de trabajo del equipo.

Los estados más comunes en Jira son "NEW" (nueva), "ANALYSING" (analizando), "READY" (listo), "IN PROGRESS" (en progreso), "BLOCKED" (bloqueado), "READY TO VERIFY" (listo para verificar) y "DEPLOYED" (implementado). Cada uno de estos estados tiene un significado específico y ayuda a entender en qué etapa se encuentra cada tarea y qué acciones se requieren para avanzar hacia la siguiente fase.

La importancia de estos estados radica en que facilitan la colaboración y coordinación del equipo de trabajo. Al tener una clara visualización de las tareas en cada estado, se puede asignar trabajo de manera eficiente, evitar duplicaciones y entender qué tareas requieren atención inmediata. También permiten identificar cuellos de botella o bloqueos en el flujo de trabajo y tomar acciones correctivas para agilizar el progreso.

Además, la configuración personalizada de los estados en Jira se adapta a las necesidades y metodologías específicas de cada proyecto como podemos ver en la tabla 3 donde se utiliza estos status para poder validar en qué fase de revisión están los tickets creados por los proyectos en Jira para la revisión que hace Core Assurance.

NEW	Estado que da el proyecto al registrar un ticket nuevo.
ANALYSING	Estado que da Core Assurance cuando comienza a revisar a nivel de formato el ticket en Jira.
READY	Estado que da Core Assurance cuando ya está listo el ticket para empezar a revisar.
IN PROGRESS	Estado que da Core Assurance cuando empieza a revisar la solicitud del ticket a nivel funcional.
BLOCKED	Estado que da Core Assurance cuando se encuentran observaciones a nivel funcional o a nivel de formato.
READY TO VERIFY	Estado que da Core Assurance cuando se finalizó la resolución del ticket.
DEPLOYED	Estado que da el proyecto al validar que lo solicitado se resolvió por Core Assurance de forma correcta.

Tabla 3: Status de los tickets Jira. Elaboración: Site Core Data

3.4.5. Gestión y Análisis de proyectos

3.4.5.1. Proceso en Datio (Big Data del banco):

El proceso de Datio es un componente esencial en el ecosistema de Big Data del banco, donde se garantiza que toda la información esté limpia, libre de duplicados y sea veraz y fidedigna. La calidad de los datos es fundamental para la precisión y eficacia de las predicciones y evaluaciones que realiza el sistema.

Con la información depurada y confiable, Datio se convierte en una herramienta poderosa para predecir comportamientos y evaluar a los clientes en profundidad. Mediante algoritmos avanzados y análisis profundos, se determina la aptitud de los clientes para obtener créditos, tarjetas, participar en campañas, entre otros. Esto permite una personalización más efectiva y una asignación más adecuada de recursos para cada segmento de clientes.

Además, Datio es fundamental para apoyar la toma de decisiones por parte del negocio del banco. La información obtenida a través del análisis de Big Data proporciona conocimientos valiosos sobre las preferencias, necesidades y comportamientos de los clientes, lo que permite tomar decisiones

informadas y estratégicas para el desarrollo de nuevos productos, la optimización de campañas de marketing y la mejora continua de la experiencia del cliente.

3.4.5.2. Aprobación Core Assurance:

El área de Core Assurance desempeña un papel de suma importancia en el flujo de datos dentro del banco. Su función principal radica en validar todos los lineamientos y garantizar la calidad de la información, lo que proporciona beneficios significativos a todos los equipos involucrados en el proceso.

La evaluación y mitigación de riesgos asociados a posibles errores de datos se convierten en su enfoque central. Al asegurarse de que la información esté limpia, precisa y libre de duplicados, Core Assurance establece una base sólida para la toma de decisiones estratégicas en toda la organización.

Este equipo actúa como guardián de la integridad y veracidad de los datos, desempeñando un papel esencial en el sistema de gestión de datos del banco. Gracias a su labor colaborativa, se mejoran la eficiencia operativa, la optimización de campañas de marketing y la experiencia general de los clientes, lo que beneficia a todas las áreas del banco.

3.4.6. Resultado de la actividad

3.4.6.1. Presentación de Resultados:

El banco reconoce la importancia de contar con información confiable para mejorar continuamente el servicio ofrecido a sus clientes. La información recopilada por los proyectos internos del banco se convierte en una valiosa base de datos, fruto de la aplicación de lineamientos establecidos para la recolección de datos de diversas fuentes y bases de datos. Esta información es utilizada en el Big Data del banco después de ser debidamente validada por el proyecto Core Assurance, asegurando su fidelidad y contribuyendo a la mejora de la experiencia del usuario.

El proyecto Core Assurance ha demostrado su eficacia al proporcionar información veraz que ha llevado a mejoras significativas en las experiencias de los usuarios. Los resultados obtenidos de este proceso son validados a través de indicadores clave de desempeño (KPI) en la página de Site Core Data.

Un ejemplo concreto de estos resultados puede observarse en el mes de mayo, donde se constató el impacto positivo del proyecto Core Assurance en la calidad de los datos utilizados en el banco. Estos beneficios han fortalecido la confianza en el enfoque implementado y han destacado la importancia de contar con información confiable para la toma de decisiones estratégicas.

- a) Primera validación:** La primera validación como se visualiza en la figura 20, consiste en analizar la interacción entre los proyectos y el Site Core Data, una página específicamente creada por el equipo de Core Assurance. Esta página juega un papel fundamental al permitir la validación de cómo los analistas de proyectos revisan los lineamientos establecidos.

El análisis de esta interacción es de gran relevancia, ya que proporciona información clave sobre la eficiencia y efectividad del proceso de revisión de lineamientos. Al examinar detalladamente las páginas que son más revisadas por los analistas, se obtiene una visión completa del comportamiento y las prioridades del equipo en cuanto a la aplicación de los lineamientos.

El acceso a esta información a través del Site Core Data ofrece una visión holística y transparente de la validación de lineamientos por parte de los proyectos. La recopilación de datos relacionados con las páginas más revisadas proporciona una valiosa retroalimentación para el equipo de Core Assurance, lo que permite identificar áreas de mejora y oportunidades para fortalecer la aplicación de los lineamientos en el banco.

Este proceso de validación permite evaluar la comprensión y cumplimiento de los lineamientos por parte de los analistas, y brinda una oportunidad para detectar posibles desafíos o áreas de riesgo en el flujo de trabajo. Además, al analizar qué páginas son más revisadas, se pueden identificar patrones de uso y tendencias que influyen en la calidad y consistencia de la información utilizada en el Big Data del banco.



Figura 20: Diagrama de barras # de vistas por página. Fuente: Site Core Data

b) Segunda validación: La segunda validación como se visualiza en la figura 21, se enfoca en obtener información clave sobre el comportamiento de los nuevos usuarios que revisan los lineamientos. Para lograrlo, se busca conocer la cantidad de nuevos usuarios que interactúan con la página Site Core Data y revisan los lineamientos en un período determinado. Esta información es fundamental para evaluar el alcance y la efectividad de la plataforma en la incorporación de nuevos usuarios.

Además, se realiza un seguimiento de la cantidad de visitas que recibe la página Site Core Data en cada mes. Este análisis mensual proporciona una visión más completa y actualizada del tráfico y la actividad de los usuarios en la plataforma. Con estos datos, se pueden identificar tendencias y patrones de uso que ayuden a mejorar la experiencia del usuario y optimizar la plataforma para satisfacer sus necesidades.

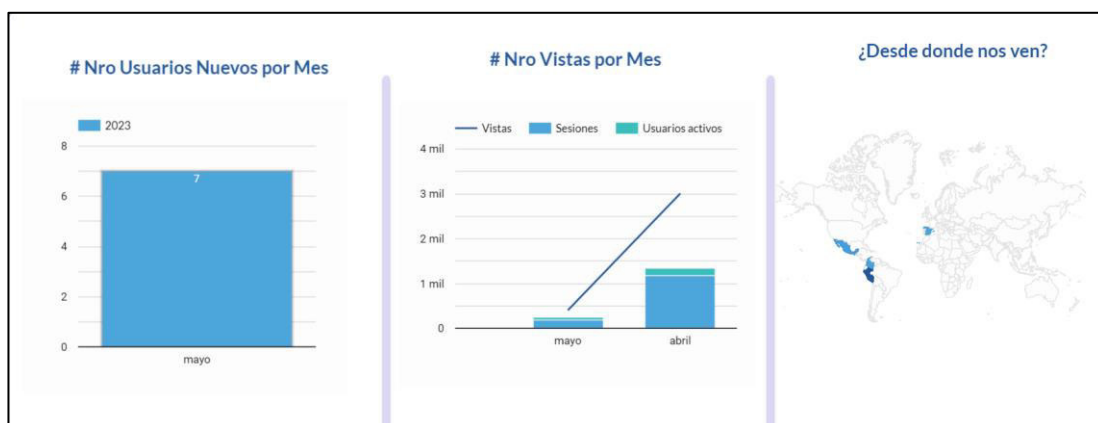


Figura 21: Diagrama vistas a la página Site Core Data. Fuente: Site Core Data

Por último, se evalúa la procedencia geográfica como se observa en la figura 22, de los usuarios que interactúan con la página Site Core Data. Conocer el país de origen de los usuarios brinda información valiosa sobre la distribución y alcance geográfico de la plataforma. Esto permite identificar regiones con mayor interés en los lineamientos y, potencialmente, adaptar la plataforma para atender mejor las necesidades específicas de cada país.

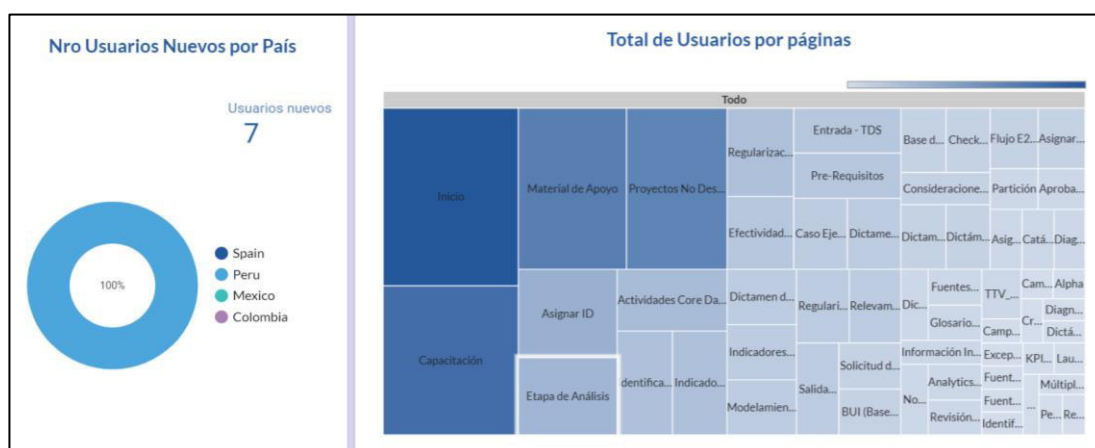


Figura 22: Diagrama de usuarios nuevos. Fuente: Site Core Data

- c) **Tercera validación:** La tercera validación como se visualiza en la figura 23, se enfoca en analizar los dictámenes que pasan por el área de Core Assurance. En primer lugar, se evalúa la cantidad total de dictámenes que se tienen, y se realiza una distinción entre aquellos que corresponden a ingestas nuevas y aquellos que son reusos de fuentes

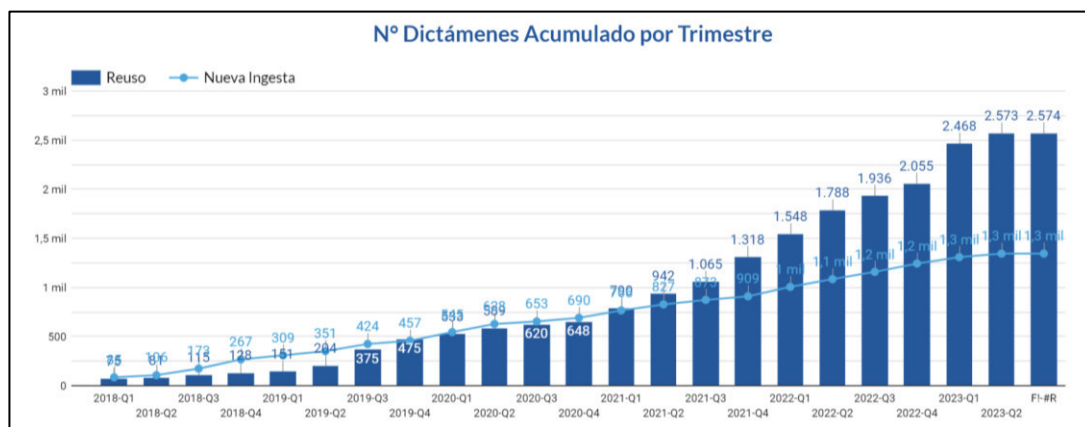


Figura 24: Diagrama de dictámenes por trimestre. Fuente: Site Core Data

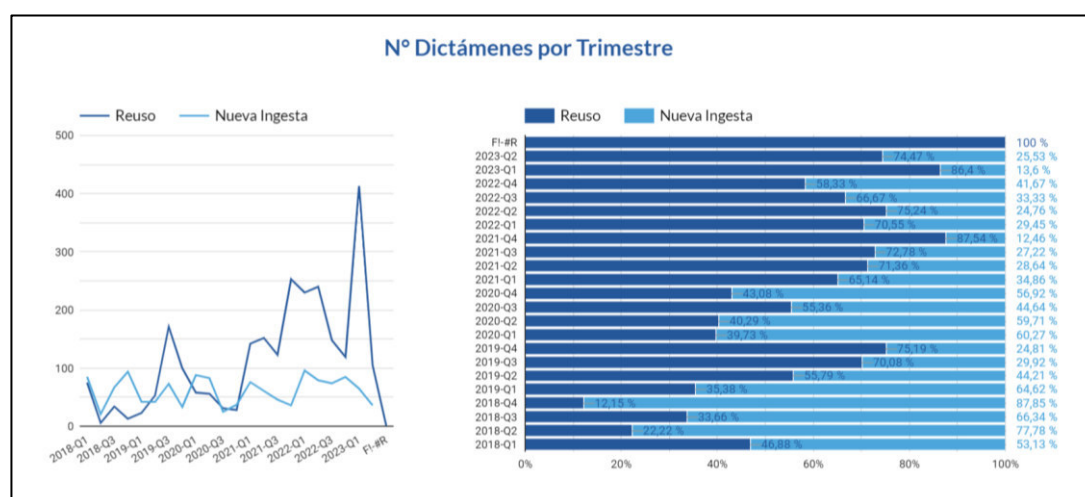


Figura 25: Diagrama de dictámenes por trimestre. Fuente: Site Core Data

La validación de los KPI y el análisis estadístico de los dictámenes en el área de Core Assurance son elementos fundamentales para demostrar la importancia y el impacto del trabajo del equipo Core Assurance en el flujo de datos del banco. Estos KPI proporcionan métricas clave que permiten evaluar la eficiencia y eficacia de todos los procesos de revisión de lineamientos.

Los gráficos y estadísticas generados brindan una visión completa de cómo ha evolucionado el número de dictámenes revisados por trimestre desde el inicio del proyecto. Estos datos son esenciales para identificar tendencias y patrones a lo largo del tiempo, lo que ayuda a tomar decisiones informadas y estratégicas.

Además, gracias a la información recopilada, se puede identificar qué fuentes son más frecuentemente revisadas y por qué se han revisado más. Esto permite comprender mejor las necesidades y prioridades de los proyectos en cuanto a la selección de las fuentes de datos, y ayuda a optimizar los procesos para una mayor eficiencia y calidad en la revisión de lineamientos.

En definitiva, la validación de KPI y el análisis estadístico son herramientas poderosas que permiten demostrar el valor y la relevancia del área de Core Assurance en el flujo de datos del banco. Con estos datos, podemos tomar decisiones informadas, identificar áreas de mejora y que se siga contribuyendo de manera significativa a la gestión de datos confiables y de alta calidad en la institución financiera.

CAPÍTULO IV: CONCLUSIONES

4.1. Justificación:

Al aplicar Big Data en los flujos internos ayuda a migrar la información tradicional en el banco a una más digital por estrategias innovadoras de enfoque al cliente y la oferta la atención personalizada. Toda esta aplicación en el banco es según lineamientos donde se revisa las fuentes que quieren ser dictaminadas y llevadas al Startup.

En el estudio de Velarde (2019) se aplica el Big Data para la migración de datos bajo una distribución Cloudera con Apache Hadoop en el banco Interbank donde desean migrar la información ya que la gran cantidad de información que genera el Banco presenta inconvenientes para su gestión y almacenamiento por lo cual aplicarán una plataforma tecnológica que pueda soportar las tareas de registros y procesamiento en un ecosistema escalable y de alta disponibilidad.

4.1.1. Justificación práctica

Las empresas de todos los sectores han constatado las ventajas de la mejora continua de los procesos internos y la mejor toma de decisiones en un entorno de transformación digital en todos los sectores. La figura 26, ilustra cómo la interacción de Big Data, IA y análisis de datos fomenta la innovación empresarial, permite ofrecer una experiencia de cliente a medida y aumenta los ingresos. (Alameda, 2018)



Figura 26 :Diagrama de resultados de valor. Fuente: Alameda

4.1.2. Justificación tecnológica

La aplicación del modelo de negocio usando Big Data permite a la institución financiera gestionar la información de sus clientes través de la planificación, control y la implementación tanto del movimiento y el almacenamiento eficaces de información relacionada, así como los bienes y servicios desde su origen hasta su destino.

4.1.3. Justificación social

El uso de Big data impacta favorablemente en el tiempo en que la entidad financiera se tarda en identificar las demandas de los clientes, provocando que se reduzca significativamente. Esto favorece que se ofrezca a los clientes un producto financiero personalizado, fomentando su satisfacción y experiencia, además de disminuir los riesgos asociados a la financiación de negocios vinculados a actividades ilegales, a la falta de liquidez o a acciones que indiquen si un prestatario reúne los requisitos para recibir un préstamo.

4.1.4. Justificación económica.

La aplicación de Big Data en el sistema de información logística de la entidad financiera facilita muchas actividades como, por ejemplo: la identificación de la necesidad del cliente, conocimiento de su comportamiento financiero, etc. Esto conlleva a que la institución financiera tenga una adecuada toma de

decisiones disminuyendo de esta manera los riesgos financieros y aumentando los ingresos. Durante el desarrollo de la presente TSP no se gastó en presupuesto.

4.2. Descripción de la implementación

4.3. Conclusiones

- **Influencia de Big Data en la Gestión de Proyectos de Ingeniería:** La implementación de Big Data impacta en la gestión de proyectos de ingeniería en los bancos de Perú. Como en el apartado 3.4.5. Gestión y Análisis de proyectos, se ha identificado que el análisis de grandes volúmenes de datos permite una toma de decisiones más informada y precisa en todas las etapas del proyecto de Core Assurance, desde la planificación hasta la ejecución y el monitoreo. El uso de herramientas de Big Data y modelado de datos proporciona una visión más completa y detallada de los procesos, mejorando la eficiencia, la eficacia y la adaptabilidad en el trabajo de equipo de un banco.
- **Lineamientos en Core Assurance para el Análisis de Datos:** A través de la descripción detallada de los lineamientos en el equipo de Core Assurance, se ha destacado la importancia de establecer reglas y criterios sólidos para la revisión y validación de datos. En el apartado 3.4.3 recepción y evaluación de Proyectos, se valida los lineamientos para fuentes TDS y procesamientos donde se explica el paso a paso de los lineamientos que se revisan en Core Assurance para cada ticket de ticket Jira enviado por el proyecto.
- **Identificación de Fuentes del Big Data en Core Assurance:** En el apartado 3.4.3 recepción y evaluación de Proyectos, mediante la revisión de los lineamientos podemos observar que se puede hacer una distinción de las fuentes del Big Data utilizadas en las operaciones de Core Assurance que ha permitido comprender la diversidad de información manejada por el equipo. Desde diversos canales digitales hasta bases de datos internas, se ha reafirmado la crucial importancia de la habilidad para elegir, autenticar y emplear fuentes de datos sólidas, a fin de salvaguardar la integridad de la información en el entramado bancario. Estas fuentes pueden englobar una amplia variedad de fuentes, tales como sistemas host, tablas Oracle, fuentes de información externa, plataformas Launchpad y muchas otras, cada una de las cuales cumple un papel esencial en garantizar la coherencia y veracidad de los datos en el sistema del banco.
- **Análisis de Indicadores de Gestión del Proyecto Core Assurance:** El análisis detallado de los indicadores de gestión del proyecto Core Assurance en el apartado 3.4.6.1. Presentación de Resultados, se ha proporcionado una visión integral sobre la eficiencia y efectividad de las

operaciones del equipo. Estos indicadores han revelado tendencias, patrones y áreas de mejora, lo que permite tomar decisiones estratégicas basadas en datos concretos. El monitoreo constante de los indicadores como el KPI brinda la oportunidad de optimizar los procesos y garantizar un flujo de datos confiable y de alta calidad.

CAPITULO V: RECOMENDACIONES

Como sugerencias de mejora para el proyecto Core Assurance, se podrían considerar las siguientes:

1. **Implementar herramientas de automatización:** La implementación de herramientas de automatización en los procesos de revisión de información podría ayudar a mejorar la eficiencia y reducir el tiempo de revisión. Se podrían considerar herramientas como procesamiento del lenguaje natural y machine learning para facilitar la revisión de grandes volúmenes de datos.
2. **Ampliar la diversidad de fuentes de información:** Si bien el proyecto Core Assurance ya considera diversas fuentes de información, se podría explorar la posibilidad de ampliar la diversidad de estas fuentes. Por ejemplo, incorporar información proveniente de redes sociales, que podrían brindar una perspectiva adicional sobre los clientes del banco.
3. **Formación continua:** Fomentar la capacitación y formación continua del equipo en temas relevantes como seguridad de datos, privacidad, nuevas tecnologías, etc. Esto mejorará la competencia del equipo y lo mantendrá actualizado en las últimas tendencias del sector.
4. **Continuar fortaleciendo los lineamientos de calidad de la información:** El equipo de Core Assurance podría seguir trabajando en el fortalecimiento de los lineamientos de calidad de la información, para garantizar la confiabilidad, veracidad y congruencia de esta.
5. **Implementar un sistema de retroalimentación:** Un sistema de retroalimentación podría ser útil para mejorar el proceso de revisión y asegurar la continuidad del proyecto. Esto permitiría que el equipo de Core Assurance reciba comentarios y sugerencias de otros departamentos o áreas del banco, y así mejorar la eficacia del proyecto.

CAPITULO VI: BIBLIOGRAFÍA

- BBVA. (2019). *BBVA*. Obtenido de BBVA: <https://extranetperu.grupobbva.pe/memoria2019/engineering.html>
- IONOS. (11 de 04 de 2022). Digital Guide IONOS. Obtenido de <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/que-es-jira/>
- Ortega, C. (s.f.). ¿Qué es la codificación de datos en una investigación? Obtenido de <https://www.questionpro.com/blog/es/codificacion-de-datos-en-una-investigacion/>
- Vasquez, J. A. (2021). Integración de entorno big data en la gestión financiera de un sistema bancario para optimizar productos, servicios internos y toma de decisiones. Obtenido de Repositorio UNMSM: https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/17040/Huapaya_vj.pdf?sequence=1&isAllowed=y
- Linares Berrocal, C. (2019). *Renati*. Obtenido de Renati: https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/1944/Carlos%20Linares_Trabajo%20de%20Suficiencia%20Profesional_Titulo%20Profesional_2019.pdf?sequence=1&isAllowed=y
- AichaAggoune. (2020). An Overview on the Mapping Techniques in NoSQL Databases. *International Journal of Informatics and Applied Mathematics*, 3(2), 53–65.
- Alameda, T. (2018, December 20). *2018: un año de logros en la transformación de BBVA | BBVA*. <https://www.bbva.com/es/2018-un-ano-de-logros-en-la-transformacion-de-bbva/>
- Amazon Web Services. (2017). *Bases de datos no relacionales*. <https://aws.amazon.com/es/nosql/>
- Apache Software Foundation. (2018). *Apache Hadoop*. <https://hadoop.apache.org/>

- Baranowski, Z., Kleszcz, E., Kothuri, P., Canali, L., Castellotti, R., Martin Marquez, M., Matos de Barros, N. G., Motesnitsalis, E., Mrowczynski, P., & Luna Duran, J. C. (2019). Evolution of the Hadoop Platform and Ecosystem for High Energy Physics. *EPJ Web Conf.*, 214. <https://doi.org/10.1051/epjconf/201921404058>
- Calvo, D. (2017, November 21). *Tipos de datos: estructurados, semiestructurados y no estructurados*. <https://www.diegocalvo.es/tipos-de-datos-estructurados-semiestructurados-y-no-estructurados/>
- Camargo-Vega, J. J., Camargo-Ortega, J. F., Joyanes-Aguilar, L., Camargo-Vega -, J. J., Felipe, J., & Joyanes-Aguilar, C.-O.-L. (2015). Conociendo Big Data Knowing the Big Data Conhecendo Big Data Fecha de Recepción: 1 de Octubre de 2014 Fecha de Aceptación: 1 de Diciembre de 2014. *Revista Facultad de Ingeniería (Fac. Ing.)*, Enero-Abril, 24(38), 63–77.
- Cloudera. (2018, April 5). *Cloudera company background information, management & board*. <https://www.cloudera.com/about/about-old.html>
- Erraissi, A., Belangour, A., & Tragha, A. (2018). *Meta-modeling of Data Sources and Ingestion Big Data layers*. <https://ssrn.com/abstract=3185342>
- ESIC BUSINESS & MARKETING SCHOOL. (2018, October). *Apache Spark: Introducción, qué es y cómo funciona | ESIC*. <https://www.esic.edu/rethink/tecnologia/apache-spark-introduccion-que-es-y-como-funciona>
- Estrada, R., & Ruiz, I. (2017). *Big Data SMACK*. <https://doi.org/10.1007/978-1-4842-2175-4>
- Galeano, L., & Domínguez, D. (2017). *Prototipo de laboratorio hadoop para análisis big data en la institución universitaria Politécnico Grancolombiano*. Politécnico Grancolombiano.
- Huapaya, J. (2021). *Integración de entorno big data en la gestión financiera de un sistema bancario para optimizar productos, servicios internos y toma de decisiones*. Universidad Nacional Mayor de San Marcos.
- IBM. (2019, August 6). *Relational Databases Explained*. <https://www.ibm.com/cloud/learn/relational-databases>
- International Business Machines. (2018). *Data Lake Solutions*. <https://www.ibm.com/data-lake>

- International Business Machines. (2021, May 27). *Hadoop vs. Spark: What's the Difference?* <https://www.ibm.com/cloud/blog/hadoop-vs-spark>
- Jain, V. K. (2017). *Big Data and Hadoop* (Khanna Publishing, Ed.; reimpressa). Khanna Book Publishing. <https://books.google.com.pe/books?id=i6NODQAAQBAJ>
- John, T., & Misra, P. (2017). *Data Lake for Enterprises* (Packt Publishing).
- Ma, Z. (2017). *Managing big data in cloud computing environments* (IGI Global, Ed.). Information Science Reference.
- Martínez, A. (2018). *BIG DATA APLICADO EN EL SECTOR BANCARIO TRADICIONAL PARA LOGRAR UNA MAYOR VENTAJA COMPETITIVA FRENTE A LAS FINTECH*. Universidad de Palermo.
- Microsoft Learn. (2022, September 26). *Restricciones entre claves principales y claves externas*. <https://learn.microsoft.com/es-es/sql/relational-databases/tables/primary-and-foreign-key-constraints?view=sql-server-ver16>
- Miloslavskaya, N., & Tolstoy, A. (2016). Big Data, Fast Data and Data Lake Concepts. *Procedia Computer Science*, 88, 300–305. <https://doi.org/10.1016/j.procs.2016.07.439>
- Padhy, R., Ranjan, M., Suresh, P., & Satapathy, C. (2017). RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Database's. *INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES*, 11(1). <http://www.ijaest.iserp.org>.
- Plase, D., Niedrite, L., & Taranovs, R. (2017). A Comparison of HDFS Compact Data Formats: Avro Versus Parquet. *Mokslas - Lietuvos Ateitis*, 9(3), 267–276. <https://doi.org/10.3846/mla.2017.1033>
- Puyol, J. (2014). UNA APROXIMACIÓN A BIG DATA AN APPROACH TO BIG DATA 1. In *Revista de derecho UNED* (Vol. 14). <http://www.rizomatica.net/big-data-la-fuente-para-el-conocimiento-en-el-siglo-xxi/>
- Rayón, A. (2016, August 19). *Cuando empieza esta era del Big Data: MapReduce - Deusto Data*. <https://blogs.deusto.es/bigdata/cuando-empieza-esta-era-del-big-data-mapreduce/#comment-2026>

- Strohbach, M., Daubert, J., Ravkin, H., & Lischka, M. (2017). Big Data Storage. In J. M. Cavanillas, E. Curry, & W. Wahlster (Eds.), *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe* (pp. 119–141). Springer International Publishing. https://doi.org/10.1007/978-3-319-21569-3_7
- Timón, S., Rincon, M., & Martínez, R. (2021). An overview of graph databases and their applications in the biomedical domain. *Database*, 2021. <https://doi.org/10.1093/database/baab026>
- Torres, E., & Rincón, W. (2017). *IMPLEMENTACIÓN DE UNA HERRAMIENTA DE AUTOGESTIÓN Y AUTOCONFIGURACIÓN PARA LA IMPLEMENTACIÓN DE SERVICIOS EN PROYECTOS DE BIG DATA*. Universidad Católica de Colombia.
- Wang, F., Ding, L., Yu, H., & Zhao, Y. (2020). Big data analytics on enterprise credit risk evaluation of e-Business platform. *Information Systems and E-Business Management*, 18(3), 311–350. <https://doi.org/10.1007/s10257-019-00414-x>
- Wright, W., Schroh, D., Proulx, P., Skaburskis, A., & Cort, B. (2006). *The Sandbox for analysis*. 801–810. <https://doi.org/10.1145/1124772.1124890>

CAPITULO VII: ANEXOS

Anexo A:

Se presentará cada uno de los tickets generados en Jira revisados por el área de Core Assurance, todos los ejemplos se presentarán con el status “Deployed” porque son ticket ya cerrados:

1. Ticket de asignación de folio TDS:

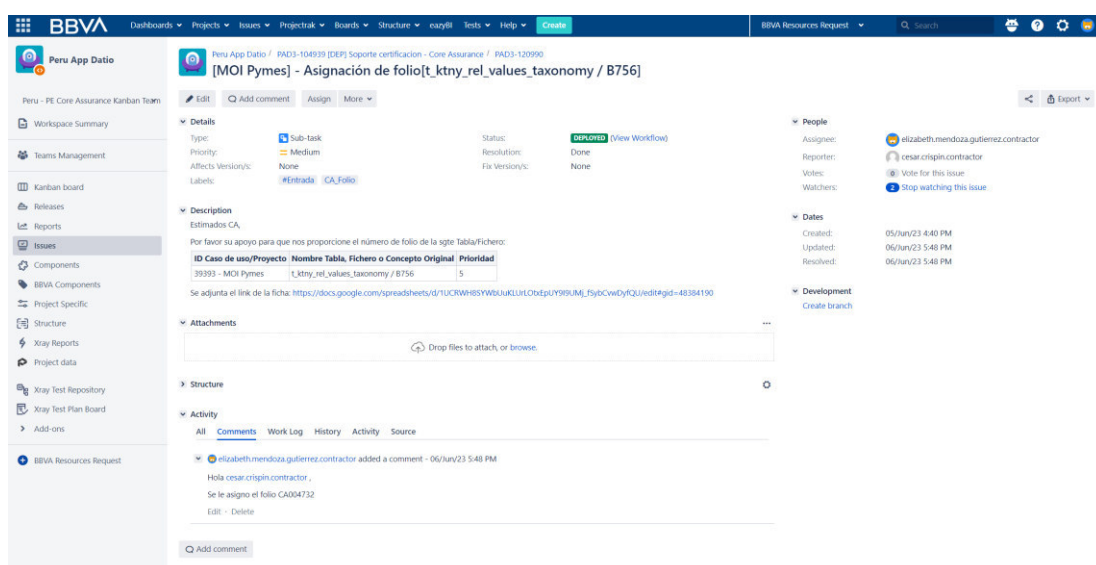


Figura 27: Ticket creado en Jira de asignación de Folio TDS. Fuente: Jira

2. Ticket de asignación de ID TDS:

[MOI PYMES] Asignación de ID del Folio CA004593: RC2_DATOS_GENERALES

Details

- Type: Sub-task
- Priority: Medium
- Affects Version/s: None
- Labels: CA_ID
- Status: **DEPLOYED** (View Workflow)
- Resolution: Done
- Fix Version/s: None

Description

Por favor, su apoyo para que nos proporcione el numero de ID de la siguiente TDS

TDS Propuesto	Evidencia/Sustento TDS	Folio	Descripción Funcional	Origen	Owner
RC2_DATOS_GENERALES	https://docs.google.com/document/d/1NaZ09imBpYMTvUlvLrIesf9KXKS2s2mEVZOLME16o/edit	CA004593	Contiene la información mensual de los datos generales como el DNI, RUC, TIPO DE DOCUMENTO, EDAD, entre otros de los accionistas titulares asociados a los clientes Pymes registrados en el aplicativo Reporte de crédito web.	ORACLE/REPORTE DE CREDITO WEB	Modelo local

Attachments

Drop files to attach, or browse.

Structure

People

- Assignee: elizabeth.mendoza.gutierrez.contractor
- Reporter: jonathan.barrientos.contractor
- Votes: 0
- Watchers: 2

Dates

- Created: 15/May/23 5:49 PM
- Updated: 18/May/23 5:19 PM
- Resolved: 18/May/23 5:19 PM

Development

Create branch

Figura 28: Ticket creado en Jira de asignación de ID TDS. Fuente: Jira

3. Ticket de revisión de dictamen TDS

[MOI PYMES] - Revision Dictamen de folio CA003919

Details

- Type: Sub-task
- Priority: Medium
- Affects Version/s: None
- Labels: #ingestaTDS, CA_Dictamen
- Status: **DEPLOYED** (View Workflow)
- Resolution: Done
- Fix Version/s: None

Description

Estimados CA,

Por favor su apoyo con la revisión del dictamen:

Folio	ID	DSF
CA003919	1828	PEBP.SBFD.FIX.UNILO.SBW00502.SBDTDSU

Attachments

Drop files to attach, or browse.

Structure

Activity

People

- Assignee: elizabeth.mendoza.gutierrez.contractor
- Reporter: enzo.cardenas
- Votes: 0
- Watchers: 2

Dates

- Created: 12/Jun/23 7:53 PM
- Updated: 22/Jun/23 12:17 AM
- Resolved: 22/Jun/23 12:17 AM

Development

Create branch

Figura 29: Ticket creado en Jira de revisión de dictamen TDS. Fuente: Jira

4. Ticket de asignación de folio de procesamiento:

[Desarrollo IntraMIS Comercial]- Asignación de folio de Procesamiento - Tabla de catálogo de epígrafes con jerarquización

Type: Sub-task
Priority: Medium
Status: **DEPLOYED** (View Workflow)
Resolution: Done
Fix Version/s: None

Labels: #Salida CA_Folio

Description
Por favor su apoyo en brindamos el folio para la salida descrita:
UUAA Propuesta PMFI

Nombre de Salida	Descripción	Campos de la Salida	MSA
Tabla de catálogo de epígrafes con jerarquización	Catálogo que contiene información de detalle de epígrafes con sus niveles de jerarquización y agregación de la misma. Dicha información obtiene los estados de epígrafes T2 (resultados espejo) y 88 (web). Siendo un epígrafe el concepto corporativo que contienen las cuentas contables que expresan un valor de los estados financieros.	Campo s	pagina 13 https://docs.google.com/document/u/0/11x7bn9ZxohRfjw8GkuhFv4s4z2XUdEC0H9gh0Ug/edit?usp=share_link

Attachments
Drop files to attach, or browse.

Structure

People
Assignee: elizabeth.mendoza.gutierrez.contractor
Reporter: amador.galeano.contractor
Votes: 0
Watchers: 2

Dates
Created: 25/May/23 4:11 PM
Updated: 02/Jun/23 1:00 AM
Resolved: 02/Jun/23 1:00 AM

Development
Create branch

Figura 30: Ticket creado en Jira de asignación de Folio Procesamiento. Fuente: Jira

5. Ticket de revisión de dictamen de procesamiento:

[FX Dynamic Pricing V2] - Revisión Dictamen Procesamiento Folio CA004702

Type: Sub-task
Priority: Medium
Status: **DEPLOYED** (View Workflow)
Resolution: Done
Fix Version/s: None

Labels: #Procesamiento CA_Dictamen

Description
Estimados CA:
Favor su apoyo el número de folio para la salida descrita:

Folio	Fuente	Descripción	Diccionario Sophia
CA 004702	Reporte interno rep_Reporte_DL410 que genera el área Global Market de las tablas Spectrum en relación a las transacciones vencidas de Spot, Non deliverable forward y Forward y que servirán mejor para mejorar el tipo cambio de divisas a través del aplicativo Star y el motor Global FX.		http://datadictionary.bbva.com/data/dictionary_next_gen/field/?country=11&data_source=1240&obje=45329

Link BUI Local:
https://docs.google.com/spreadsheets/d/1g_VB2d5R3T0s0oPbXHD3NNjkuNc4s/edit#gid=1607497751

Attachments
Drop files to attach, or browse.

People
Assignee: kevin.barboza.contractor
Reporter: PIERRE PINEDA
Votes: 0
Watchers: 2

Dates
Created: 19/Jun/23 6:14 PM
Updated: 22/Jun/23 7:01 AM
Resolved: 22/Jun/23 7:01 AM

Development
Create branch

Figura 31: Ticket creado en Jira de revisión de procesamiento. Fuente: Jira

6. Ticket de reúso: Estos tickets solo aplican cuando un proyecto desea reusar fuentes ya existentes y solo utilizar los campos que otros proyectos dictaminaron.

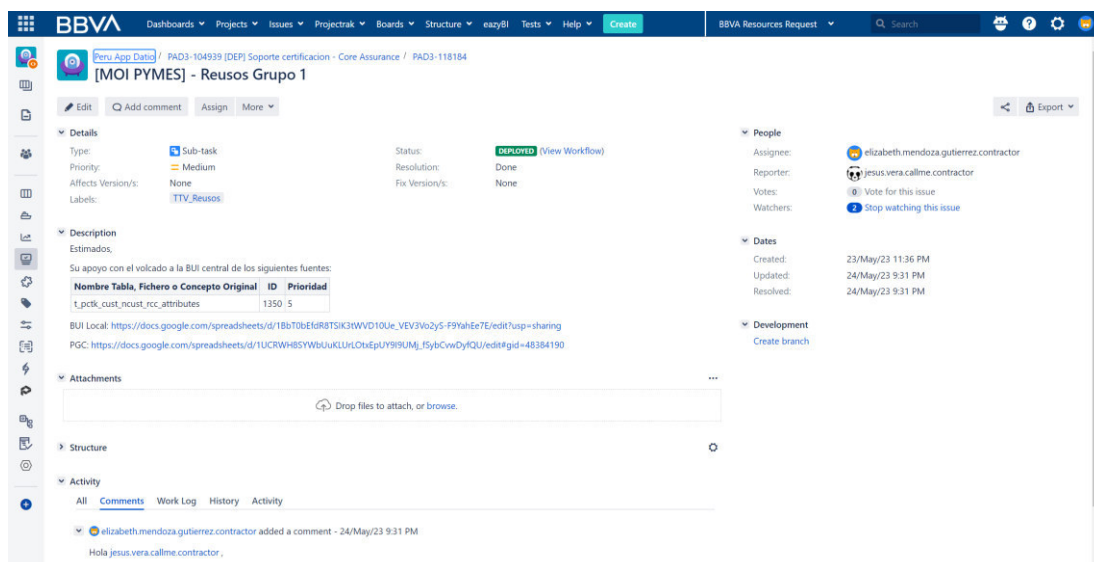


Figura 32: Ticket creado en Jira para reúsos de fuentes ya existentes. Fuente: Jira

- 7. Ticket de regularización:** Estos tickets son creados cuando se desea regularizar algo que ya se haya revisado con anterioridad, pero hubo alguna modificación posterior a su aprobación.

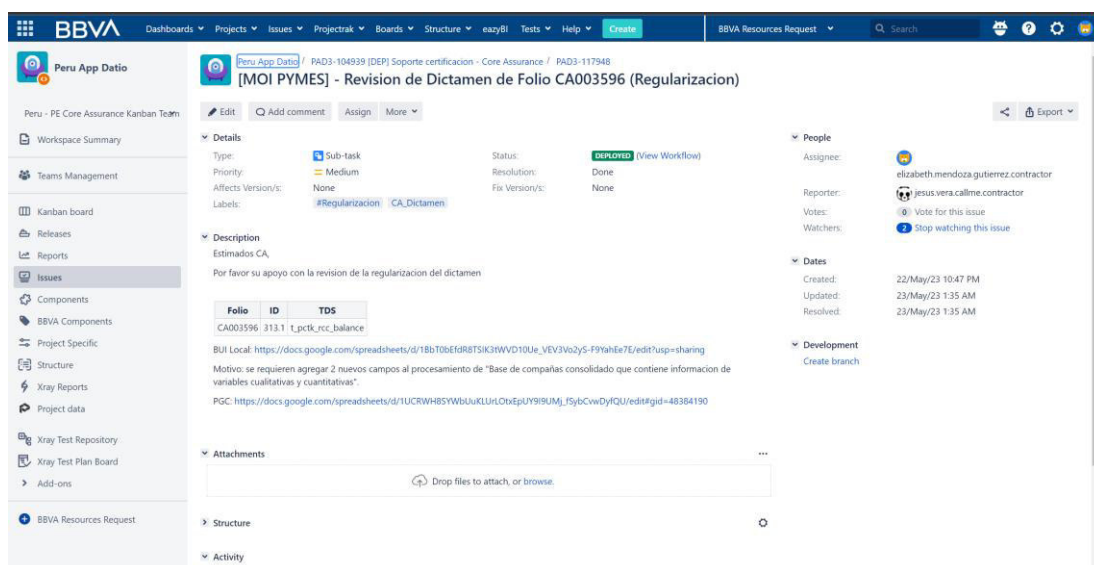


Figura 33: Ticket creado en Jira para regularizar fuentes. Fuente: Jira

Anexo B:

Etapas de análisis en el flujo E2E



Figura 34 :Análisis completo E2E de todas las áreas del banco. Fuente: Site Core Data

Anexo C:

Los KPI's una herramienta utilizada en la gestión de proyectos bancarios para medir el éxito o el rendimiento de una actividad en función de objetivos específicos. Estos indicadores son seleccionados cuidadosamente para medir el progreso hacia metas estratégicas y asegurar que los esfuerzos se están dirigiendo en la dirección correcta. Los KPI's pueden variar dependiendo del proyecto y pueden incluir medidas como la rentabilidad, el rendimiento de los activos, la satisfacción del cliente, el riesgo crediticio, entre otros. En resumen, los KPI's son una herramienta crítica para medir el desempeño y el éxito de un proyecto bancario y ayudar a tomar decisiones informadas para lograr los objetivos estratégicos.