Optimización del tratamiento y la recuperación de datos transaccionales en entornos Big Data a partir de archivos XML

Cristhian Esteban Hernández Gómez

Asesor

Andrés Felipe Solís Pino

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Especialización en Ciencia de Datos y Analítica

2025

_____

Nombre Director de Trabajo de Grado

_____ _____

Jurado Jurado

Resumen

El presente proyecto tiene como objetivo optimizar la accesibilidad y la consulta de información transaccional en entornos con grandes volúmenes de datos, con un enfoque particular en el tratamiento de datos no relacionados en formato XML. En un contexto donde las organizaciones, especialmente las empresas del sector financiero generan y almacenan cantidades masivas de información, se vuelve necesario implementar metodologías que permitan estructurar, recuperar y analizar datos de manera eficiente para el aprovechamiento.

El proyecto se orienta al desarrollo e implementación de técnicas que mejoren el rendimiento de los sistemas de almacenamiento y consulta. Esto incluye la organización lógica de los datos, el uso de modelos de recuperación optimizados y herramientas tecnológicas que favorezcan la velocidad de acceso y la precisión de los resultados. Debido a que el formato XML es común en la transmisión de información estructurada, pero no se adapta fácilmente a modelos relacionales, se abordarán los desafíos de procesamiento y explotación efectiva, con énfasis en su aplicación en bases de datos transaccionales del sector financiero.

Esta propuesta busco aportar soluciones que permitan manejar con eficiencia grandes volúmenes de información transaccional en XML, garantizando tiempos de respuesta óptimos, escalabilidad y consistencia de los datos, contribuyendo así a una toma de decisiones sólida y basada en evidencia.

Palabras clave: *Python, SQL Server, XML, Extracción y Datos.*

Abstract

This project aims to optimize the accessibility and query of transactional information in environments with large volumes of data, with a particular focus on the processing of unrelated data in XML format. In a context where organizations, especially those in the financial sector, generate and store massive amounts of information, it becomes necessary to implement methodologies that allow for efficient data structuring, retrieval, and analysis for its utilization.

The project focuses on the development and implementation of techniques that improve the performance of storage and query systems. This includes the logical organization of data, the use of optimized retrieval models, and technological tools that improve access speed and the accuracy of results. Because the XML format is common for transmitting structured information but does not easily adapt to relational models, the challenges of processing and effective exploitation will be addressed, with an emphasis on its application in transactional databases in the financial sector. This proposal seeks to provide solutions that efficiently manage large volumes of transactional information in XML, ensuring optimal response times, scalability, and data consistency, thus contributing to sound, evidence-based decision-making.

Keywords: *Python, SQL Server, XML, Extraction and Data*

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Introducción

En el contexto actual de transformación digital, las organizaciones enfrentan el reto creciente de gestionar volúmenes masivos de información transaccional generada por diversos sistemas informáticos. Esta información, que constituye un recurso estratégico, suele almacenarse en formatos semiestructurados como XML, los cuales ofrecen flexibilidad y compatibilidad entre plataformas, pero al mismo tiempo presentan importantes desafíos para su tratamiento, consulta y explotación eficiente.

A pesar de que muchas empresas, especialmente del sector financiero, cuentan con grandes cantidades de datos almacenados en este formato, una parte significativa de dicha información no es aprovechada más allá de su uso técnico u operativo. Ejemplo de ello son los archivos log, frecuentemente utilizados para detectar errores en la transmisión de datos, pero con un alto potencial analítico si son tratados, clasificados y estructurados de manera adecuada. Esta falta de explotación se debe, en gran medida, a las dificultades de acceso, ausencia de categorización lógica y la carencia de herramientas y metodologías especializadas para trabajar con datos semiestructurados en entornos de gran escala.

Este proyecto de grado tiene como objetivo optimizar la accesibilidad, consulta y categorización de la información transaccional contenida en archivos XML, a través del desarrollo de un modelo lógico para el tratamiento de datos. El enfoque se centra en transformar datos poco aprovechados en información útil para áreas que brindan respuesta a consumidores financieros, mejorando los tiempos de respuesta, la claridad de la información y la eficiencia operativa. Para ello, se adoptó la metodología CRISP-DM, que guía todas las etapas del proceso desde la comprensión del negocio hasta el despliegue del modelo implementado.

La propuesta busca sentar las bases para una explotación más eficiente de los datos semiestructurados, abriendo el camino para futuros desarrollos que fortalezcan la analítica institucional y la toma de decisiones basada en datos.

Justificación

En la era digital actual, las organizaciones, especialmente aquellas del sector financiero, generan y almacenan grandes volúmenes de datos transaccionales como resultado de sus operaciones cotidianas. Esta información constituye un recurso estratégico clave, ya que permite identificar patrones, evaluar comportamientos del cliente, y apoyar decisiones orientadas a la mejora del servicio y la eficiencia operativa. Sin embargo, su valor depende directamente de la forma en que estos datos son almacenados, organizados, consultados y presentados (Inmon, Strauss & Neushloss, 2008).

En particular, el uso del formato XML para almacenar datos semiestructurados representa un reto importante, ya que este tipo de archivo, aunque flexible y ampliamente adoptado por su capacidad para representar estructuras jerárquicas, no está optimizado para consultas complejas ni para análisis en tiempo real. Esta situación se agrava cuando las consultas deben realizarse con rapidez, claridad y precisión, en entornos donde los datos deben ser comprensibles no solo para perfiles técnicos, sino también para usuarios sin formación especializada (Bose, 2012).

El presente proyecto se justifica por la necesidad de desarrollar un modelo de estructuración y tratamiento de datos XML que mejore significativamente la accesibilidad, la categorización lógica y la presentación de la información transaccional. Esto permitirá no solo reducir los tiempos de consulta y eliminar inconsistencias, sino también transformar la información técnica en reportes claros, legibles y orientados a públicos no técnicos, facilitando su interpretación y uso en distintos niveles organizacionales.

Este enfoque es especialmente relevante para el sector financiero, donde la capacidad de acceder de forma clara y rápida a los datos transaccionales es vital para la prevención de fraudes, la personalización de servicios, y el cumplimiento de regulaciones. Implementar tecnologías

como XPath, XQuery, DOM, SAX, XSD y XSLT dentro de un modelo lógico de categorización permitirá transformar el manejo tradicional de archivos XML en un proceso inteligente y adaptable a las necesidades de negocio.

Objetivos

Objetivo General

Optimizar la accesibilidad, consulta y categorización de la información transaccional contenida en archivos XML en empresas del sector financiero, mediante el desarrollo de un sistema de estructuración y tratamiento de datos que permita mejorar la claridad y calidad de consultas y facilitar la comprensión de la información, incluso para usuarios no técnicos.

Objetivos Específicos

Proponer técnicas para la extracción de información clara y concisa desde archivos en formato XML, en contextos de alto volumen de datos, con el fin de identificar oportunidades de mejora en su estructuración, acceso y recuperación.

Desarrollar e implementar un modelo lógico y técnicas especializadas para optimizar la categorización, consulta y recuperación de archivos XML, reduciendo inconsistencias y mejorando la eficiencia

Descripción del Problema

Planteamiento del Problema

En el contexto actual de transformación digital, las organizaciones manejan un creciente volumen de datos transaccionales generados por plataformas digitales, canales financieros, y sistemas administrativos. Estos datos, en muchos casos, son almacenados en formatos semiestructurados como XML, debido a su flexibilidad para representar estructuras jerárquicas y su alta interoperabilidad entre sistemas heterogéneos (Bose, 2012). Sin embargo, este tipo de formato presenta importantes limitaciones cuando se trata de acceder, consultar, transformar y analizar la información de forma rápida, clara y comprensible, especialmente en entornos que no implementan un modelo relacional tradicional.

Uno de los principales problemas es la baja accesibilidad y lentitud en la consulta de información contenida en archivos XML, especialmente cuando estos archivos son de gran tamaño o no han sido organizados de forma lógica. Esta situación genera retrasos en la entrega de información clave para la toma de decisiones, afectando directamente los tiempos de respuesta operativa y la capacidad de análisis oportuno. Asimismo, la ausencia de categorización adecuada y procesos de estandarización de datos provoca redundancias, inconsistencias y errores interpretativos (Fan & Poulovassilis, 2011).

Otro desafío crítico es que la información almacenada en XML no suele estar diseñada para ser comprendida por personas sin conocimientos técnicos. Esto dificulta su lectura, interpretación y aprovechamiento por parte de profesionales de áreas como la gerencia, el servicio al cliente, que necesitan acceder a reportes claros, legibles y orientados a la claridad de los datos. La necesidad de transformar los datos técnicos en formatos comprensibles y relevantes

se vuelve indispensable en estos entornos, donde se requiere traducir el contenido de XML a un lenguaje más accesible y contextualizado (Hernandez & Kambhampati, 2004).

A esto se suma la falta de implementación de tecnologías como XPath, XQuery, DOM, SAX, XSD y XSLT, las cuales permitirían realizar operaciones más eficientes sobre los datos. Sin un enfoque especializado en el tratamiento de datos XML, muchas organizaciones se enfrentan a cuellos de botella informativos que afectan su productividad y competitividad.

En este escenario, surge la necesidad de diseñar un modelo lógico y técnico que permita estructurar, transformar y presentar la información contenida en archivos XML de forma más accesible, categorizada y útil tanto para usuarios técnicos como no técnicos. Este modelo debe estar orientado a mejorar la recuperación de información, reducir los tiempos de consulta y permitir una comprensión más clara de los datos transaccionales, facilitando así los procesos analíticos y estratégicos en entornos de alta demanda informativa.

Sistematización del Problema

Para comprender de forma estructurada los retos que enfrenta el tratamiento de datos transaccionales en formato XML dentro de entornos Big Data, se plantean las siguientes preguntas de investigación, orientadas a guiar el desarrollo del proyecto:

¿Cómo afecta el uso de archivos XML no categorizados ni estructurados a la eficiencia en la consulta y recuperación de datos transaccionales en empresas del sector financiero?

¿Qué técnicas y herramientas pueden ser implementadas para optimizar la estructuración, transformación y consulta de archivos XML con grandes volúmenes de datos?

¿Cómo puede garantizarse que la información extraída de archivos XML sea comprensible y útil para usuarios no técnicos en contextos de toma de decisiones?

¿Cómo medir el impacto de la optimización en la recuperación de datos XML en términos de tiempos de respuesta, claridad y escalabilidad?

Estas preguntas permiten descomponer el problema general en aspectos más específicos que orientan el análisis, el diseño del modelo lógico y la implementación técnica, asegurando que la solución propuesta responda tanto a los desafíos técnicos como a las necesidades de negocio.

Marco Conceptual y Teórico

El tratamiento de grandes volúmenes de información transaccional se ha convertido en un reto fundamental en el contexto actual de la gestión de datos. En particular, los archivos en formato XML (eXtensible Markup Language) son ampliamente utilizados para almacenar y compartir información estructurada entre aplicaciones, pero presentan desafíos considerables cuando se trata de organizar, acceder y recuperar eficientemente la información que contienen.

XML es un formato flexible y auto-descriptivo, lo que lo convierte en una excelente opción para el intercambio de datos heterogéneos entre sistemas. Sin embargo, su naturaleza jerárquica y su falta de estructura relacional convencional pueden dificultar su manipulación cuando se trata de realizar consultas complejas, análisis automatizados o integración con herramientas de análisis de datos tradicionales.

Aunque formatos como JSON también se utilizan ampliamente para el intercambio de datos semiestructurados en aplicaciones modernas, el presente proyecto se enfoca exclusivamente en XML, dado que este es el formato original utilizado para almacenar los datos transaccionales analizados. Esta decisión responde a las condiciones reales de los entornos financieros estudiados, donde aún no se ha migrado hacia tecnologías más recientes como JSON.

Esto hace necesario el diseño de modelos lógicos que permitan categorizar, estructurar y procesar la información de forma más eficiente.

La categorización de datos se convierte así en un proceso clave dentro del proyecto, ya que permite organizar la información XML en clases, grupos o secciones semánticas que faciliten su interpretación y explotación posterior. Al categorizar los datos correctamente, se puede mejorar significativamente el rendimiento de los sistemas de recuperación, reducir la

redundancia, y optimizar el acceso a la información más relevante para los usuarios o sistemas que la consumen.

Además, los principios de gestión de grandes volúmenes de datos (Big Data) también son relevantes en este contexto, ya que se requiere aplicar técnicas que permitan escalar el procesamiento, mantener la integridad de los datos y reducir los tiempos de respuesta en consultas, incluso cuando se trabaja con millones de registros o archivos de gran tamaño.

En este marco, el presente proyecto se apoya en fundamentos de la gestión de datos no relacionales, el modelado lógico de estructuras XML, y las estrategias de optimización para acceso y consulta eficiente. Estas bases teóricas permiten establecer una metodología que responda a los desafíos del manejo de información transaccional en entornos exigentes, orientando las soluciones hacia una mayor agilidad, precisión y control sobre los datos tratados.

Metodología

El presente proyecto adopta la metodología CRISP-DM (Cross Industry Standard Process for Data Mining), por ser un modelo estructurado, iterativo y ampliamente reconocido en proyectos de analítica de datos. Esta metodología permite organizar de forma coherente las fases de desarrollo, desde la comprensión del problema hasta la implementación del modelo, favoreciendo la trazabilidad y replicabilidad del proceso.

Comprensión del Negocio

Se analizará el contexto del manejo de información transaccional en archivos XML dentro del sector financiero, con énfasis en los problemas de accesibilidad, consulta, redundancia y legibilidad de los datos. Se establecerán los objetivos del proyecto alineados con las necesidades de mejorar el tratamiento de datos semiestructurados para la toma de decisiones oportunas y comprensibles.

Comprensión de los Datos

Se realizará una exploración de los archivos XML disponibles, que simulan transacciones financieras. En esta fase se identificarán etiquetas clave, estructuras jerárquicas, atributos, nodos repetitivos y posibles inconsistencias. Esta revisión permitirá definir los criterios para su estructuración lógica.

Preparación de los Datos

Se llevará a cabo el análisis, transformación y validación de archivos XML utilizando herramientas disponibles en el entorno de trabajo:

DOM y SAX mediante módulos estándar de Python para el análisis sintáctico.

Validación estructural mediante reglas de negocio programadas en Python, o mediante el uso de la librería xml.

Transformaciones orientadas a facilitar la consulta se realizarán con scripts Python o directamente en SQL Server, aprovechando sus funciones nativas de manejo XML (nodes(), value(), query() y exist()).

Se aplicarán técnicas de limpieza y normalización para eliminar redundancias, estandarizar etiquetas y asegurar consistencia entre documentos XML.

Modelado

Se diseñará e implementará un modelo lógico de categorización que permita mejorar la consulta, comprensión y eficiencia del acceso a los datos XML. Este modelo incorporará técnicas como:

- XPath y XQuery para búsquedas optimizadas
- Consultas sobre SQL Server XML usando nodes(), query(), value() y exist(),
- Indexación y compresión de etiquetas clave.

Evaluación

Se medirán indicadores como:

Mejorar los tiempos de entregas de información a los usuarios,

Reducción de redundancia estructural,

Claridad de los resultados para usuarios no técnicos.

Esto permitirá valorar la efectividad del sistema propuesto y su aplicabilidad a escenarios reales.

Despliegue

Se documentará el sistema implementado con ejemplos de uso, formatos de salida amigables y orientados a la interpretación de datos por parte de usuarios técnicos y no técnicos. Se establecerán recomendaciones para su integración en entornos reales de trabajo.

Herramientas Tecnológicas

Python (bibliotecas: xml.etree.ElementTree, lxml, pandas)

SQL Server (funciones para XML)

Editores XML: Visual Studio Code

Justificación Metodológica

El uso de CRISP-DM permite alinear el proceso con estándares internacionales en ciencia de datos, proporcionando una estructura robusta y adaptable al enfoque aplicado del proyecto. Además, esta metodología es compatible con las recomendaciones metodológicas propuestas por autores como Baena (2014) y García (2003), y responde a los retos actuales de gestión de grandes volúmenes de datos en entornos Big Data (Parra Méndez et al., 2021).

Cronograma

Tabla 1

Actividades

Actividad	M1	M2	M3	M4	M5	M6
Levantamiento del requerimiento o necesidad	x					
Levantamiento de historias de usuario		x				
Codificación de proyecto - Creación de ambientes			x	x	x	
Pruebas del producto e implementación						x

Recursos y Presupuesto

Tabla 2

Recursos

Recurso	Descripción	Presupuesto
Recurso Humano	Gestor de la demanda	$12.500.000
	Líder técnico	$30.000.000
	Desarrollador	$13.500.000
	Persona de pruebas	$4.500.000
Equipos y Software	Sqlite3	Software Free
	Python	Software Free
Materiales	4 equipos PC	8.000.000
Total		$60.500.000

Análisis y Desarrollo de Soluciones para el Tratamiento de Archivos XML

Diseño e Implementación de la Solución de Transformación de Datos XML

Durante el desarrollo del proyecto se diseñó e implementó un modelo lógico que permite transformar archivos XML con datos transaccionales en estructuras relacionales optimizadas, utilizando funciones avanzadas de SQL Server y procesamiento con Python. Como parte clave de la solución, se aplicó la función value() de SQL Server para acceder de manera precisa a los nodos y atributos del XML, lo que facilitó una extracción estructurada y exacta de la información requerida. Esta automatización en la extracción, categorización y consolidación de los datos permitió reducir significativamente los tiempos de ejecución: el proceso que antes tomaba entre 30 y 40 minutos, ahora se realiza en un rango de 3 a 5 minutos. Además de la mejora en eficiencia, el sistema garantiza mayor calidad en los datos, minimizando errores humanos y asegurando resultados consistentes para el análisis y la toma de decisiones.

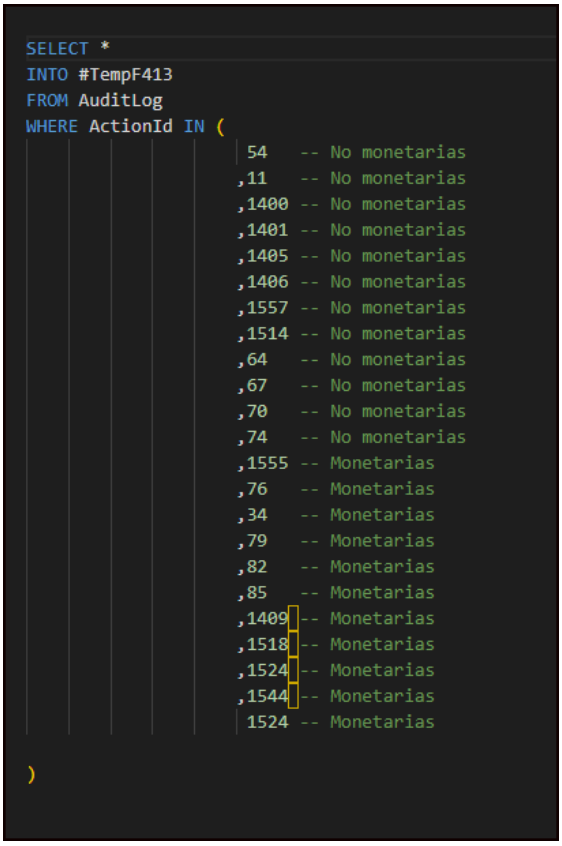
Para la elaboración de este proyecto, se llevó a cabo un análisis detallado de los datos contenidos en los archivos XML, los cuales se relacionaron con los datos estructurados almacenados en la base de datos, específicamente con las acciones que un usuario puede realizar en la plataforma financiera. Se tomó cada una de las acciones posibles y se buscó su correspondencia en la base de datos. A partir de esta operación, se identificaron las acciones efectivamente utilizadas en la plataforma y se descartaron aquellas que no presentaban registros relevantes. En este paso se llevó a cabo la segunda etapa de la metodología CRISP-DM que corresponde a recopilar los datos teniendo en cuenta la categorización.

Clasificación de los Datos

En esta parte se realiza la clasificación de los datos como paso inicial, los datos que tenemos como materia prima se dividen en acciones monetarias y no monetarias.

Figura 1

Clasificación de Datos para Filtrar la Información Requerida en el Proceso



```
SELECT *
INTO #TempF413
FROM AuditLog
WHERE ActionId IN (
                    54   -- No monetarias
                   ,11   -- No monetarias
                   ,1400 -- No monetarias
                   ,1401 -- No monetarias
                   ,1405 -- No monetarias
                   ,1406 -- No monetarias
                   ,1557 -- No monetarias
                   ,1514 -- No monetarias
                   ,64   -- No monetarias
                   ,67   -- No monetarias
                   ,70   -- No monetarias
                   ,74   -- No monetarias
                   ,1555 -- Monetarias
                   ,76   -- Monetarias
                   ,34   -- Monetarias
                   ,79   -- Monetarias
                   ,82   -- Monetarias
                   ,85   -- Monetarias
                   ,1409 -- Monetarias
                   ,1518 -- Monetarias
                   ,1524 -- Monetarias
                   ,1544 -- Monetarias
                    1524 -- Monetarias

)
```

Posteriormente, se identificaron los identificadores (ID) correspondientes a cada acción. Cabe destacar que cada acción genera un archivo XML individual con toda la información asociada. Con base en esto, se seleccionaron aquellas acciones que aportaban mayor valor al proceso de extracción.

Entendimiento de los Archivos XML

Se inició un estudio de la estructura de los archivos XML, con el objetivo de mapear los campos clave que serían recolectados durante el análisis. Los archivos presentan diferentes estructuras las cuales fueron analizada y por medio de un proceso manual se identificó los tags que contenían la información que el proceso requería recuperar.

Figura 2

Segmento Documento XML que se Tomó como Ejemplo

```
<Name xmlns="urn:*********.administration.businessentities.users">CAR******</Name>
<LastName xmlns="urn:*********.administration.businessentities.users">AVIL***</LastName>
<Mail xmlns="urn:*********.administration.businessentities.users">ca****@***rcer.com</Mail>
<DocumentTypeId xmlns="urn:*********.administration.businessentities.users">101</DocumentTypeId>
<DocumentNumber xmlns="urn:*********.administration.businessentities.users">416***</DocumentNumber>
<CountryId xmlns="urn:*********.administration.businessentities.users" i:nil="true" />
<CellPhone xmlns="urn:*********.administration.businessentities.users">3142****</CellPhone>
<WorkPhone xmlns="urn:*********.administration.businessentities.users" />
<CreatedDate xmlns="urn:*********.administration.businessentities.users">2025-02-06T15:27:14.617</CreatedDate>
<ModifiedDate xmlns="urn:*********.administration.businessentities.users">0001-01-01T00:00:00</ModifiedDate>
<CanSaveDocuments xmlns="urn:*********.administration.businessentities.users">false</CanSaveDocuments>
<LoginDevice xmlns="urn:*********.administration.businessentities.users" xmlns:e="urn:*********.framework.businessentities.security">
  <Value xmlns="urn:*********.framework.businessentities.common">1000</Value>
```

Recuperación de Datos no Estructurados

Una vez comprendida la estructura de los archivos, se procedió al parser utilizando SQL Server y la función value(), la cual permitió extraer la información contenida en las etiquetas (tags) del archivo XML. Este paso fue fundamental para comenzar a transformar los datos no estructurados en información organizada y útil.

En esta parte del proceso de recuperación de información se eligió trabajar con la función value() por su presión para encontrar los datos, la facilidad de asignar el tipo de dato con el cual se quiere guarda la información, cuenta con un mayor rendimiento que query() cuando se tiene que buscar un solo dato. Cuenta con la colaboración de otras funciones como exist() para que las búsquedas presenten mayor precisión.

Figura 3

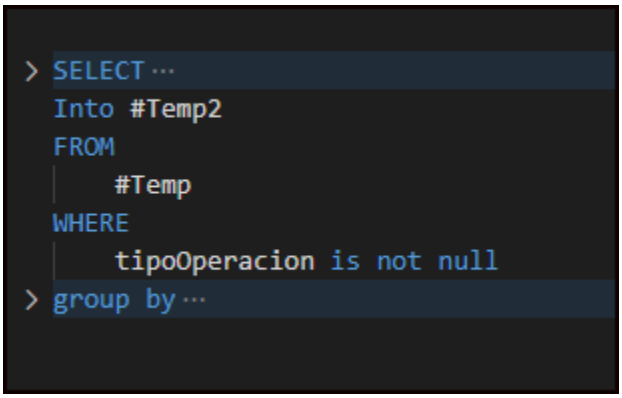
Uso de la Función Value() en la Preparación de los Datos

```
  a.RequestMessage.value('(./*[local-name() = ''Envelope'']/*[local-name() = ''Body'']/*
  [local-name() = ''ExecuteTransferMessageIn'']/*[local-name() = ''Transfer'']
/*[local-name() = ''DebitProduct'']/*[local-name() = ''ProductType'']/*
[local-name() = ''Value'']/text()) [1]', 'nvarchar(max)') = 1
```

Los datos extraídos fueron almacenados temporalmente en tablas, mientras se recorrían todos los archivos disponibles dentro de un rango de fechas previamente definido por el proceso.

Figura 4

Tablas Temporales Donde se Almacenan los Datos después de la Extracción

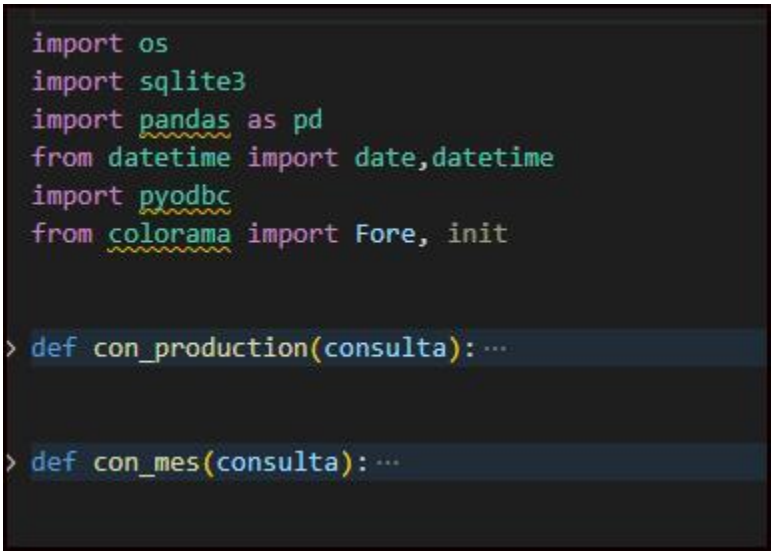


Preparación de Datos

Adicionalmente, se utilizó el lenguaje de programación Python para realizar tareas de clasificación y transformación de los datos. Este proceso inició con la importación de las librerías necesarias para la extracción y manipulación de datos. Asimismo, se definieron las funciones encargadas de conectarse a las bases de datos y obtener la información requerida para su posterior tratamiento.

Figura 5

Librerías Python que se Requieren para el Procesamiento de la Información



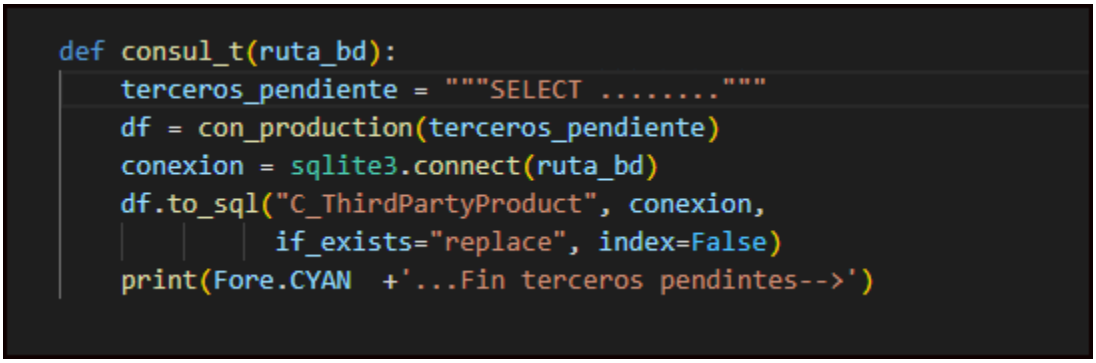
Una vez definidos los campos relevantes, se procedió a establecer una conexión con el motor de base de datos SQL Server, desde donde se extrajo la información necesaria para el proceso de análisis. Esta información fue recuperada mediante consultas específicas que permitieron acceder a los registros asociados a las acciones previamente identificadas.

Centralizar Datos

Posteriormente, los datos obtenidos fueron migrados a una base de datos **SQLite**, con el fin de centralizar la información y facilitar su manipulación en las siguientes etapas del proyecto. Este entorno más ligero y portátil permitió realizar pruebas de transformación, limpieza y análisis sin afectar los sistemas de producción ni requerir una infraestructura robusta.

Figura 6

Código de Extracción que se Utiliza para Conexión a BD

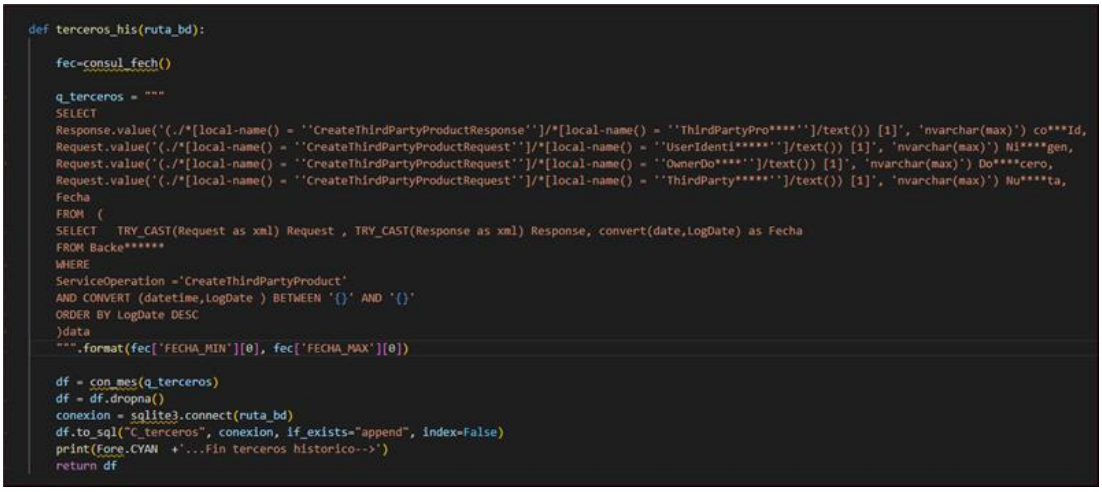


```
def consul_t(ruta_bd):
    terceros_pendiente = """SELECT ........"""
    df = con_production(terceros_pendiente)
    conexion = sqlite3.connect(ruta_bd)
    df.to_sql("C_ThirdPartyProduct", conexion,
              if_exists="replace", index=False)
    print(Fore.CYAN  +'...Fin terceros pendintes-->')
```

Este proceso se realizó en varias fuentes de información.

Figura 7

Extracción de Datos Lanzado Desde Python



```
def terceros_his(ruta_bd):

    fec=consul_fech()

    q_terceros = """
    SELECT
    Response.value('(./*[local-name() = ''CreateThirdPartyProductResponse'']/*[local-name() = ''ThirdPartyPro****'']/text()) [1]', 'nvarchar(max)') co***Id,
    Request.value('(./*[local-name() = ''CreateThirdPartyProductRequest'']/*[local-name() = ''UserIdenti*****'']/text()) [1]', 'nvarchar(max)') Ni****gen,
    Request.value('(./*[local-name() = ''CreateThirdPartyProductRequest'']/*[local-name() = ''OwnerDo****'']/text()) [1]', 'nvarchar(max)') Do***cero,
    Request.value('(./*[local-name() = ''CreateThirdPartyProductRequest'']/*[local-name() = ''ThirdParty*****'']/text()) [1]', 'nvarchar(max)') Nu****ta,
    Fecha
    FROM  (
    SELECT   TRY_CAST(Request as xml) Request , TRY_CAST(Response as xml) Response, convert(date,LogDate) as Fecha
    FROM Backe******
    WHERE
    ServiceOperation ='CreateThirdPartyProduct'
    AND CONVERT (datetime,LogDate ) BETWEEN '{}' AND '{}'
    ORDER BY LogDate DESC
    )data
    """.format(fec['FECHA_MIN'][0], fec['FECHA_MAX'][0])

    df = con_mes(q_terceros)
    df = df.dropna()
    conexion = sqlite3.connect(ruta_bd)
    df.to_sql("C_terceros", conexion, if_exists="append", index=False)
    print(Fore.CYAN  +'...Fin terceros historico-->')
    return df
```

Complementando los Datos

Otra de las fuentes de información utilizadas en el proyecto correspondió a archivos de Excel, los cuales contenían datos complementarios relevantes para el análisis. Estos archivos

fueron cargados a la base de datos SQLite mediante una función desarrollada en Python, que permitió automatizar el proceso de lectura y almacenamiento.

La función implementada se encargó de leer cada archivo Excel, validar su estructura y transformar los datos en un formato compatible con la base de datos destino. Una vez procesada la información, esta fue insertada en las tablas correspondientes dentro del entorno SQLite, permitiendo así su integración con los demás datos recolectados y centralizando la información en un único repositorio para facilitar el análisis posterior.

Figura 8

Cargue de Información de Excel Utilizando Fuentes Adicionales para Completar la Información

```
def cargar_ach(ruta_archivo):
    data = pd.read_csv(ruta_archivo, sep=';', encoding='latin-1',
                       converters={'CuentaDestino': str, 'nitOriginador': str, 'IdDestinatario': str}, usecols=[
                       'nitOri**', 'IdDestin****', 'CuentaDes****', 'produ****',
                       'EstadoTrans***', 'caus****','Fech*****'])
    data = data[data['producto'] == 'Prenota credito']
    conexion = sqlite3.connect(ruta_bd)
    data.to_sql("C_spRep_Movi****", conexion,
                if_exists="replace", index=False)
    print(Fore.CYAN  +'...Fin cargar archivo Nue-->')
```

Con el ingreso de esta información a la base de datos, se completó el proceso de transformación de datos no relacionados a una estructura relacional, permitiendo una organización más eficiente y coherente de los registros. Esta transformación facilitó la integración de múltiples fuentes de datos en un modelo unificado, orientado a su posterior explotación analítica.

Procesando Datos

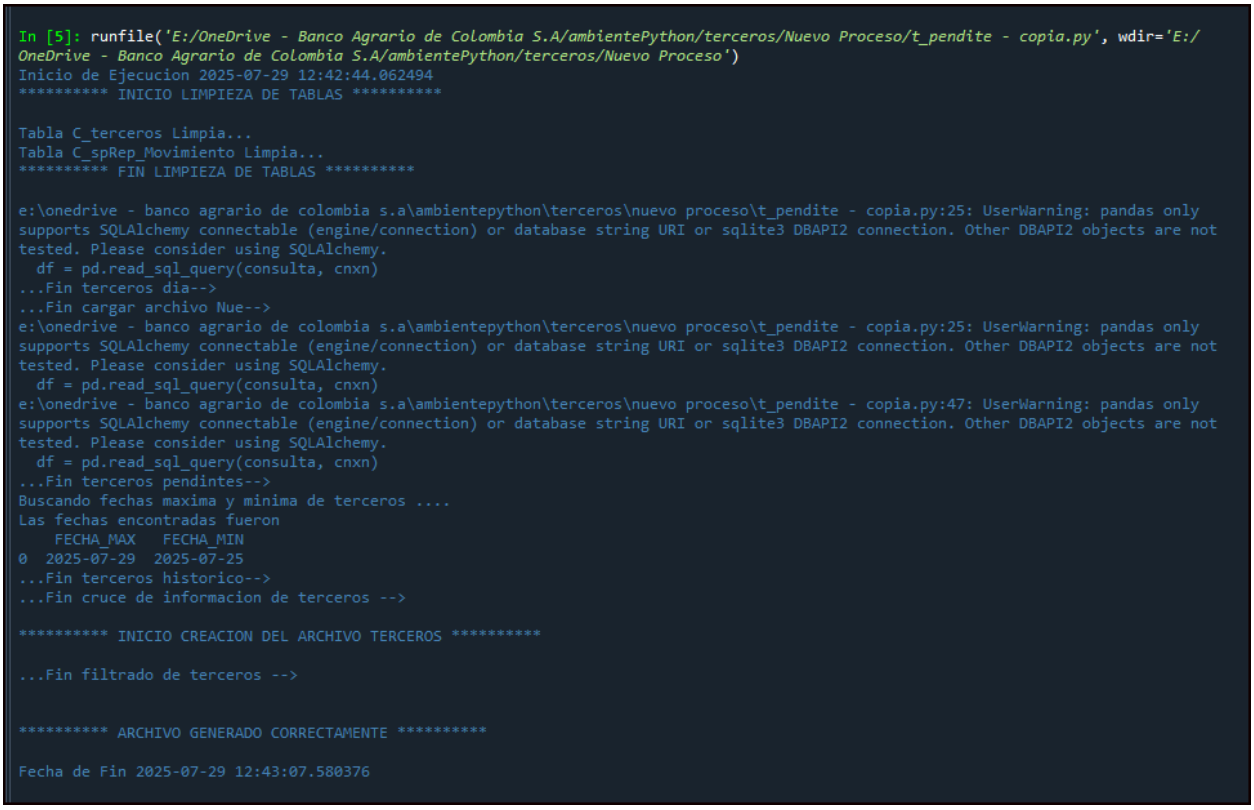
En la etapa de procesamiento de datos se complementa con la inclusión de un log de actividad para detectar que todos los procesos implementados se ejecuten sin problemas y con todas las condiciones requeridas, también podamos medir el tiempo de ejecución por proceso.

En la imagen que se muestra a continuación, se evidencian claramente cada uno de los pasos del proceso. Si estos se ejecutan correctamente, se notifica al usuario que el script está en funcionamiento. Esto permite llevar un control detallado de cada etapa y evita errores en la escritura de la información. Este control de errores hace que el código sea más amigable para el usuario final, ya que, en caso de presentarse un fallo, podrá identificarlo fácilmente.

También se logró mejorar significativamente los tiempos de generación de la información. Con el proceso propuesto, que anteriormente requería entre 30 y 40 minutos para extraer datos desde diversas fuentes, ahora se obtiene el mismo resultado de forma automatizada. Es importante destacar que el proceso anterior era manual, lo que implicaba un alto riesgo de errores humanos. Al delegar estas tareas a sistemas automatizados, se alcanzó una efectividad del 100%, con una mejora del 87% en rendimiento y del 92% en eficiencia. Estos porcentajes se incrementan aún más cuando las fechas de procesamiento son inferiores a dos días.

Figura 9

Procesando Datos Paso a Paso y Realizando Medición de Tiempos



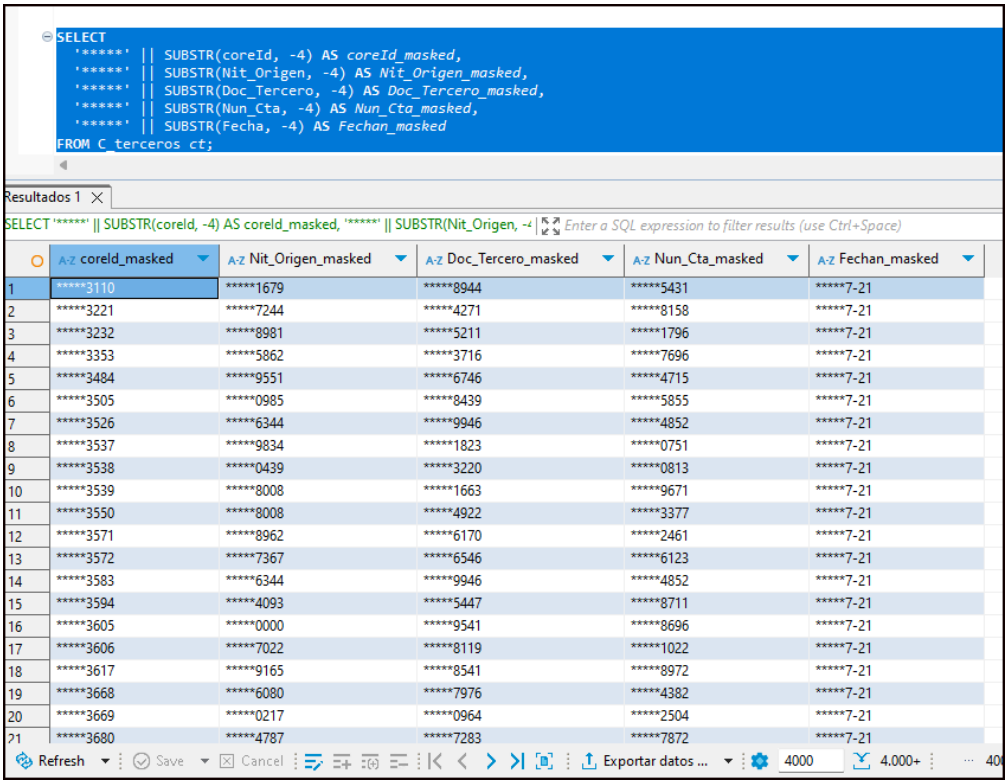
Resultados

La información centralizada y estructurada fue preparada para ser utilizada en procesos de actualización de registros y en la generación de reportes financieros, permitiendo así mejorar la disponibilidad, precisión y oportunidad de los datos requeridos por las diferentes áreas de la organización.

Los resultados obtenidos a partir de las transformaciones realizadas se presentan a continuación

Figura 10

Resultados Proceso Terceros y Enmascaramiento de Información



```
SELECT
  '*****' || SUBSTR(coreId, -4) AS coreId_masked,
  '*****' || SUBSTR(Nit_Origen, -4) AS Nit_Origen_masked,
  '*****' || SUBSTR(Doc_Tercero, -4) AS Doc_Tercero_masked,
  '*****' || SUBSTR(Nun_Cta, -4) AS Nun_Cta_masked,
  '*****' || SUBSTR(Fecha, -4) AS Fechan_masked
FROM C_terceros ct;
```

Resultados 1

	coreId_masked	Nit_Origen_masked	Doc_Tercero_masked	Nun_Cta_masked	Fechan_masked
1	*****3110	*****1679	*****8944	*****5431	*****7-21
2	*****3221	*****7244	*****4271	*****8158	*****7-21
3	*****3232	*****8981	*****5211	*****1796	*****7-21
4	*****3353	*****5862	*****3716	*****7696	*****7-21
5	*****3484	*****9551	*****6746	*****4715	*****7-21
6	*****3505	*****0985	*****8439	*****5855	*****7-21
7	*****3526	*****6344	*****9946	*****4852	*****7-21
8	*****3537	*****9834	*****1823	*****0751	*****7-21
9	*****3538	*****0439	*****3220	*****0813	*****7-21
10	*****3539	*****8008	*****1663	*****9671	*****7-21
11	*****3550	*****8008	*****4922	*****3377	*****7-21
12	*****3571	*****8962	*****6170	*****2461	*****7-21
13	*****3572	*****7367	*****6546	*****6123	*****7-21
14	*****3583	*****6344	*****9946	*****4852	*****7-21
15	*****3594	*****4093	*****5447	*****8711	*****7-21
16	*****3605	*****0000	*****9541	*****8696	*****7-21
17	*****3606	*****7022	*****8119	*****1022	*****7-21
18	*****3617	*****9165	*****8541	*****8972	*****7-21
19	*****3668	*****6080	*****7976	*****4382	*****7-21
20	*****3669	*****0217	*****0964	*****2504	*****7-21
21	*****3680	*****4787	*****7283	*****7872	*****7-21

Refresh | Save | Cancel | Exportar datos ... | 4000 | 4.000+

Se hace uso de la técnica de enmascarar los datos para proteger la información de los clientes de la entidad financiera.

Nuestro objetivo general fue optimizar la accesibilidad de la información para personas no técnicas, ya que esta se encontraba resguardada en archivos XML, como se puede ver en la ilustración, esta información que no se puede analizar de una forma sencilla.

Figura 11

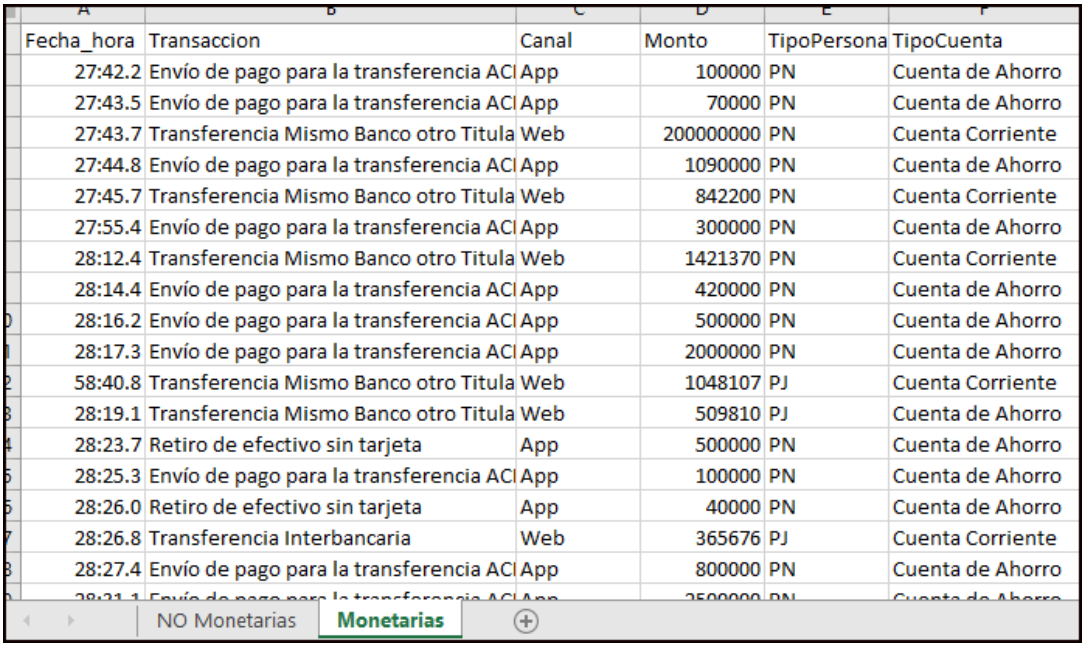
Colección de Archivos XML sin ser Transformados

RequestMessage	ResponseMessage
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...
<Envelope><Body><GetNewsMessageIn><ApprovalConte...	<Envelope><Body><GetNewsMessageOut><ApprovalCont...

Gracias al resultado que se muestra a continuación, podemos afirmar que se cumplió completamente dicho objetivo. Además de transformar los datos, se logró relacionar la información y facilitar su acceso y lectura mediante archivos Excel. De esta manera, datos que anteriormente no se utilizaban adquirieron valor y propósito dentro de la organización.

Figura 12

Resultado Proceso Tx Monetarias



	Fecha_hora	Transaccion	Canal	Monto	TipoPersona	TipoCuenta
	27:42.2	Envío de pago para la transferencia ACI	App	100000	PN	Cuenta de Ahorro
	27:43.5	Envío de pago para la transferencia ACI	App	70000	PN	Cuenta de Ahorro
	27:43.7	Transferencia Mismo Banco otro Titula	Web	200000000	PN	Cuenta Corriente
	27:44.8	Envío de pago para la transferencia ACI	App	1090000	PN	Cuenta de Ahorro
	27:45.7	Transferencia Mismo Banco otro Titula	Web	842200	PN	Cuenta Corriente
	27:55.4	Envío de pago para la transferencia ACI	App	300000	PN	Cuenta de Ahorro
	28:12.4	Transferencia Mismo Banco otro Titula	Web	1421370	PN	Cuenta Corriente
	28:14.4	Envío de pago para la transferencia ACI	App	420000	PN	Cuenta de Ahorro
0	28:16.2	Envío de pago para la transferencia ACI	App	500000	PN	Cuenta de Ahorro
1	28:17.3	Envío de pago para la transferencia ACI	App	2000000	PN	Cuenta de Ahorro
2	58:40.8	Transferencia Mismo Banco otro Titula	Web	1048107	PJ	Cuenta Corriente
3	28:19.1	Transferencia Mismo Banco otro Titula	Web	509810	PJ	Cuenta de Ahorro
4	28:23.7	Retiro de efectivo sin tarjeta	App	500000	PN	Cuenta de Ahorro
5	28:25.3	Envío de pago para la transferencia ACI	App	100000	PN	Cuenta de Ahorro
6	28:26.0	Retiro de efectivo sin tarjeta	App	40000	PN	Cuenta de Ahorro
7	28:26.8	Transferencia Interbancaria	Web	365676	PJ	Cuenta Corriente
8	28:27.4	Envío de pago para la transferencia ACI	App	800000	PN	Cuenta de Ahorro

NO Monetarias | Monetarias

En el desarrollo de los objetivos específicos, se planteó la implementación de técnicas para la extracción de información clara y concisa desde archivos en formato XML. Con la ejecución de este proceso, se logró cumplir dicho objetivo, ya que se transformó un archivo XML en una tabla relacional, facilitando el acceso a los datos. Esto permite que los usuarios, mediante herramientas de escritorio, puedan procesar la información de manera sencilla.

Figura 13

Ejemplo Archivo XML Semiestructurado


```xml
<d:ProductType xmlns:f="urn:ba.********.businessentities.framework.common" i:type="f:ProductTypeExtended">
  <d:Value>2</d:Value>
</d:ProductType>
<d:ProductOwnerName>ASOCIACION DE PADRES *******</d:ProductOwnerName>
<d:ProductBranchName i:nil="true" />
<d:ClientId>1103228</d:ClientId>
<d:Currency>...</d:Currency>
<d:Features...>...</d:Features>
<d:ProductStatus xmlns:f="urn:ba.********.businessentities.framework.common" i:type="f:ProductStatusExtended">
  <d:Value>1</d:Value>
</d:ProductStatus>
<d:ExtendedProperties xmlns:f="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>...</f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>...</f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
    <f:Key>accountOffice</f:Key>
    <f:Value i:type="d:ExtendedPropertyValueString">
      <d:ExtendedPropertyValue>BALBOA****</d:ExtendedPropertyValue>
    </f:Value>
  </f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
    <Name xmlns="urn:********p.administration.businessentities.users">MAIRA LOR***</Name>
    <LastName xmlns="urn:********p.administration.businessentities.users">IBARRA****</LastName>
    <Mail xmlns="urn:********p.administration.businessentities.users">NOTIFICACI***@AS****CBFOLAYA.COM</Mail>
    <DocumentTypeId xmlns="urn:********p.administration.businessentities.users">10***1</DocumentTypeId>
    <DocumentNumber xmlns="urn:********p.administration.businessentities.users">2527****</DocumentNumber>
    <CountryId xmlns="urn:********p.administration.businessentities.users" i:nil="true" />
    <CellPhone xmlns="urn:********p.administration.businessentities.users">3127918****</CellPhone>
    <WorkPhone xmlns="urn:********p.administration.businessentities.users" />
    <CreatedDate xmlns="urn:********p.administration.businessentities.users">2024-03-26T1*****</CreatedDate>
    <ModifiedDate xmlns="urn:********p.administration.businessentities.users">0001-01-01T00:00:00</ModifiedDate>
    <CanSaveDocuments xmlns="urn:********p.administration.businessentities.users">false</CanSaveDocuments>
    <LoginDevice xmlns="urn:********p.administration.businessentities.users" xmlns:e="urn:********p.framework.business
        <Value xmlns="urn:********p.framework.businessentities.common">10*****</Value>
    </LoginDevice>
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>...</f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>...</f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>...</f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>...</f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>...</f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>...</f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>...</f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
  <f:KeyValueOfstringExtendedPropertyValueEPwN05B4>...</f:KeyValueOfstringExtendedPropertyValueEPwN05B4>
</d:ExtendedProperties>
<d:CanTransact>false</d:CanTransact>
```


Los resultados obtenidos en este proyecto representan victorias tempranas para la organización, al evidenciar el valor oculto en datos que anteriormente no se explotaban. Información que antes era vista como una carga, ahora se convierte en un recurso estratégico gracias al procesamiento de datos en bruto.

Figura 14

Resultado Proceso Tx no Monetarias

Canal_Id	departamen	ciudad	tipoOperacic	Tipo_person	codigo_ciiu	mecanismos	CantidadTX	MontoOperacion
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0
13			58	1		0	1557	0

NO Monetarias | Monetarias

Diagramas de los Procesos Expuestos

Finalizado el desarrollo del proyecto, se documentaron los flujos de proceso implementados en cada etapa, con el propósito de proporcionar al lector una visión clara y contextualizada del enfoque adoptado.

Los datos se encuentran alojados en una base de datos SQL. A través de un script, se realiza la extracción y clasificación de la información. Una vez clasificados, los datos son transformados para obtener únicamente la información relevante, como los montos de las transacciones, el canal y el tipo de transacción, entre otros. Durante este proceso, se descartan los nombres de los tags y los encabezados del documento XML, ya que no aportan valor en esta etapa.

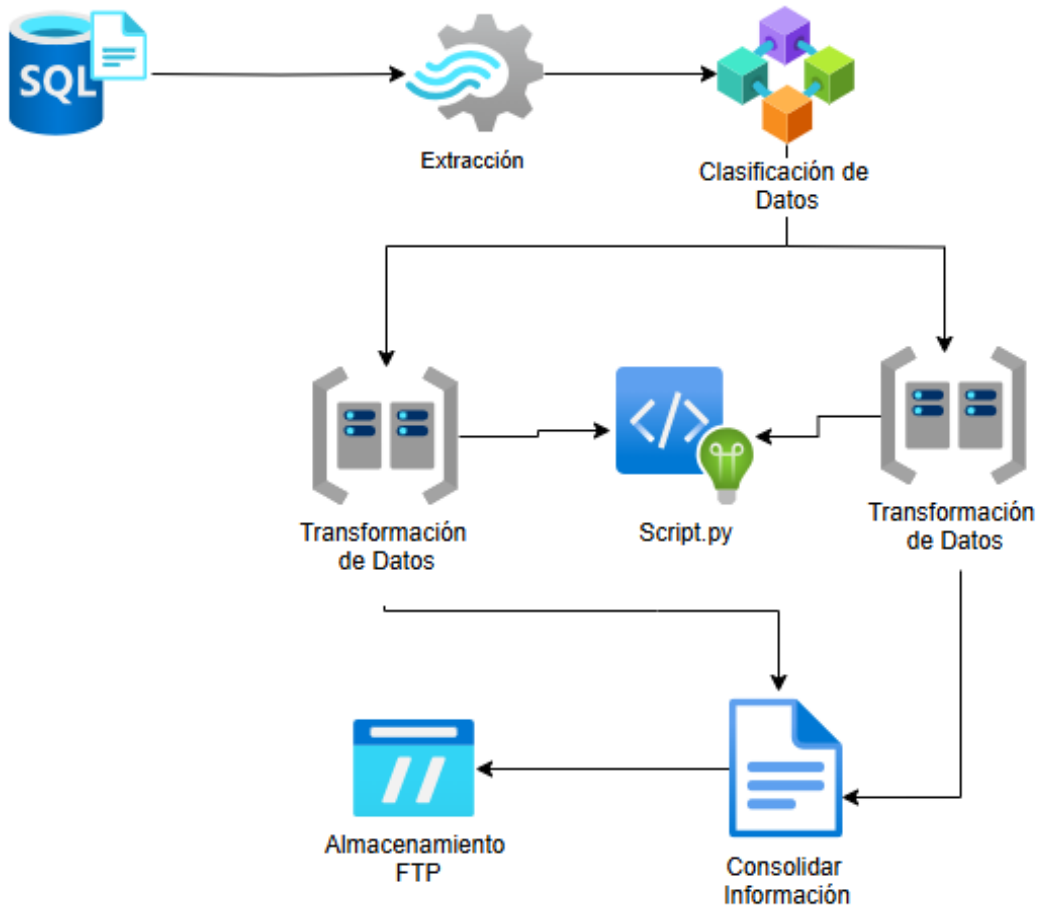
Finalmente, se genera un archivo en formato TXT o Excel, el cual se deposita en un servidor FTP. Esta información puede ser utilizada por otros sistemas o empleada para la

generación de reportes, facilitando así su integración y aprovechamiento dentro de la organización.

En el siguiente diagrama se presentan los componentes y fases del proceso necesario para transformar una colección de archivos XML en un archivo Excel. Este flujo está diseñado para facilitar el uso de la información por parte de personal no técnico, que no trabaja directamente con bases de datos, pero que domina con gran destreza las herramientas del paquete Office. Gracias a esta transformación, los usuarios pueden aprovechar los datos procesados de manera eficiente y sencilla.

Figura 15

Flujo de Transformación y Consolidación de Datos Desde SQL Server

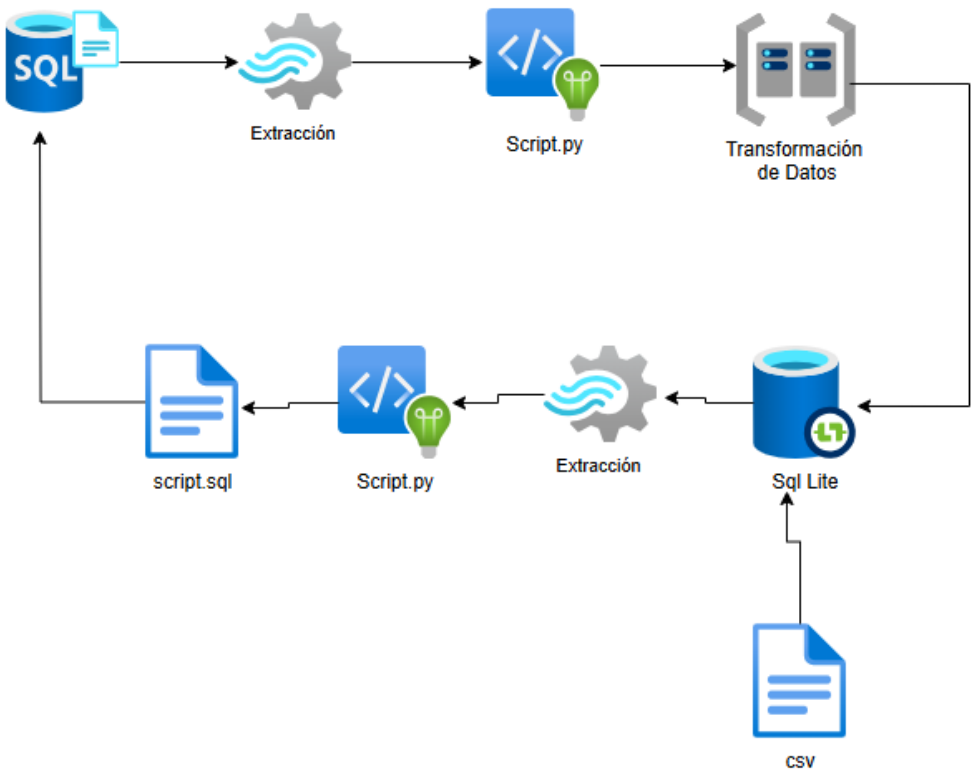


También se quiere mostrar en los diagramas el proceso de extracción de información de los terceros que tiene la entidad financiera, en este proceso se extrae el log que genera el consumo de servicios que se encuentran en Soap y generan archivos XML, que posteriormente los alojamos en la base de datos SQL, por medio de un proceso de extracción apoyado del lenguaje Python y haciendo uso de las librerías pandas y PYODBC.

La información se carga en un entorno Python, donde se realizan transformaciones y posteriormente se centraliza en una base de datos SQLite. Esta base se complementa con una segunda fuente de información: un archivo Excel. Finalmente, se realiza un cruce de datos entre ambas fuentes y se genera un archivo .sql, el cual se utiliza para cargar la información consolidada en una base de datos.

Figura 16

Esquema de Extracción y Conversión Hacia Sqlite y CSV



Conclusiones

En el desarrollo de este proyecto se identificaron diversas técnicas modernas orientadas a la explotación de información en entornos con altos volúmenes de datos, como es el caso de los archivos XML almacenados en los sistemas informáticos de las organizaciones. Uno de los principales hallazgos fue el reconocimiento del valor oculto de los archivos log, los cuales tradicionalmente son utilizados únicamente para la detección de errores en la transmisión de datos entre sistemas. Sin embargo, el análisis realizado permitió evidenciar que estos registros pueden convertirse en una fuente estratégica de información cuando son adecuadamente procesados, clasificados y presentados, especialmente para áreas comerciales y financieras.

Uno de los beneficios más significativos identificados tras el desarrollo del proyecto fue la reducción del tiempo de procesamiento de los datos XML. Previamente, el proceso de consulta y cruce de información podía tomar entre 30 y 40 minutos, ya que requería la extracción manual de registros desde múltiples fuentes y su posterior tratamiento en hojas de cálculo de Excel. Con la implementación del modelo automatizado mediante Python y funciones nativas de SQL para el manejo de XML, el tiempo promedio de procesamiento se redujo a un rango de 3 a 5 minutos, lo cual representa una mejora de aproximadamente entre el 87 % y el 92 % en eficiencia operativa.

Adicionalmente, se incrementó de manera significativa la fiabilidad y calidad de los datos procesados. La automatización del flujo de trabajo permitió minimizar errores humanos recurrentes en el copiado, filtrado y consolidación de datos, lo cual se traducía anteriormente en inconsistencias o pérdidas de información crítica. El nuevo modelo garantiza resultados consistentes, trazables y aptos para el análisis por parte de áreas no técnicas, mejorando la calidad del soporte a la toma de decisiones.

El proyecto aportó a la organización una visión renovada sobre la importancia de los datos no relacionados o poco estructurados, demostrando que su valor no radica únicamente en su almacenamiento, sino en su capacidad de ser explotados analíticamente. Esta iniciativa permitió abrir nuevas posibilidades para el desarrollo de sistemas de explotación de datos, generando una ruta de acceso hacia un ecosistema informacional más eficiente. Se evidenció que, mediante el uso de tecnologías como XPath, XQuery, DOM, SAX, XSD y XSLT, es posible facilitar el acceso y comprensión de datos tradicionalmente inaccesibles, reduciendo las barreras técnicas y mejorando los tiempos de consulta.

Entre las principales limitaciones encontradas durante la ejecución del proyecto se destaca la ausencia de una infraestructura tecnológica robusta para el procesamiento de datos a gran escala. La falta de un entorno dedicado a la explotación de grandes volúmenes de información dificultó la ejecución de ciertos procedimientos y simulaciones. Sin embargo, esto también permitió identificar áreas de mejora y reforzó la necesidad de futuras inversiones en capacidades tecnológicas que soporten iniciativas de análisis avanzado en la organización.

Recomendaciones

Implementar una infraestructura tecnológica adecuada para el tratamiento de datos en gran volumen, que permita escalar los procesos de extracción, transformación y consulta de información no estructurada, especialmente en formatos como XML. Esto facilitará la automatización de procesos y mejorará la eficiencia operativa.

Crear una unidad o equipo de análisis de datos semiestructurados, con personal capacitado en técnicas de procesamiento y modelado de datos no relacionales, orientadas a mejorar la accesibilidad, eficiencia y aprovechamiento de este tipo de información.

Establecer procesos de categorización y estandarización de datos desde la etapa de captura, lo cual permitirá reducir redundancias, mejorar la calidad de la información y facilitar su uso posterior por parte de distintas áreas funcionales dentro de la organización.

Aprovechar los resultados obtenidos en este proyecto como base para futuros desarrollos, priorizando la exploración y análisis de fuentes de datos semiestructurados que no han sido tratadas tradicionalmente, y extendiendo el modelo propuesto a otros sistemas de información de la entidad.

Referencias Bibliográficas

Artunduaga, L. (2019). *Elementos conceptuales del diseño y evaluación de proyectos* [Objeto virtual de información (OVI)]. Repositorio Institucional UNAD. https://repository.unad.edu.co/handle/10596/28361

Baena, G. (2014). *Metodología de la investigación* (pp. 43–117). eLibro. https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/40362

Bose, R. (2012). Advanced analytics: opportunities and challenges. *Industrial Management & Data Systems*, *112*(2), 156–170.

Fan, W., & Poulovassilis, A. (2011). Information extraction and integration from semi-structured data. *ACM Computing Surveys*, *43*(4), 1–47.

García, J. (2003). *Cómo elaborar un proyecto de investigación* (pp. 23–79). EBSCOhost. https://research-ebsco-com.bibliotecavirtual.unad.edu.co/c/qcagk4/ebook-viewer/pdf/3ioip6femz

González Farran, X., Rodríguez, J. R., & Guitart, I. (2016). *¿Cómo planificar un proyecto de inteligencia de negocio?* Editorial UOC. https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/58548?page=27

Hernandez, M. A., & Kambhampati, S. (2004). Integration of XML and relational data with full query expressiveness. *IEEE Data Engineering Bulletin*, *27*(4), 39–45.

Ollé, C., & Cerezuela, B. (2018). *Gestión de proyectos paso a paso*. Editorial UOC. https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/116314?page=41

Parra Méndez, C. A., Santos Méndez, D. J., & Pineda Romero, M. M. (2021). Big data, educación y post-acuerdo. *Cultura de paz en redes sociales. Publicaciones e Investigación*, *14*(3). https://doi.org/10.22490/25394088.4486

Sánchez-Torres, J. A., Molina Arévalo, N., Tovar Perilla, N. J., & Sánchez Echeverri, L. A. (2022). *Proceso de formulación y gestión de proyectos de investigación, desarrollo e innovación (i+d+i) de acuerdo con los requisitos de la norma técnica colombiana 5802 del 2008*. Sello Editorial UNAD. https://doi.org/10.22490/9789586518581