Desarrollo de una herramienta analítica para el diagnóstico de catálogos de inventario

Andrés Felipe Rodgers Calderón

Asesor

María Alejandra Varona Taborda

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas Tecnología e Ingeniería ECBTI

Especialización en Ciencia de Datos y Analítica

2025

Resumen

El presente proyecto tiene por objetivo diseñar una herramienta analítica automatizada que diagnostique la calidad de los catálogos de inventario, integrando un proceso ETL para la extracción, transformación y carga de datos, validaciones estructurales basadas en el estándar UNSPSC y técnicas de procesamiento de lenguaje natural para detectar inconsistencias semánticas. A partir de una base de datos maestra consolidada, aplicando algoritmos de clustering y detección de duplicados para identificar patrones de error y clasificar registros según su nivel de calidad.Se fundamenta en teorías de calidad de datos y gobernanza informacional, incorporando buenas prácticas de normalización y trazabilidad. Metodológicamente, se adoptó un enfoque cuantitativo y experimental con un diseño iterativo en cuatro fases: construcción de la base de conocimiento, desarrollo del módulo de diagnóstico semántico, validación con catálogos reales y mejora continua del sistema. Se prevé que el prototipo reduzca en al menos un 80 % el tiempo de validación manual, mejore la codificación y garantice la auditabilidad de los procesos. Como resultado, se generan reportes técnicos automatizados, visualizaciones y un mecanismo histórico de ejecuciones que facilitan el hallazgo por usuarios no técnicos. Además, contempla la migración del prototipo a un servicio web, para facilitar su despliegue y colaboración remota en entornos corporativos. Este enfoque fortalece la disponibilidad continua y la mantenibilidad del sistema, cumpliendo con las recomendaciones de gobernanza de datos internacionales. Finalmente, la propuesta ofrece una solución escalable, accesible y replicable que optimiza los procesos de catalogación, aporta eficiencia operativa y eleva la calidad de la información en la cadena de abastecimiento.

Palabras claves: abastecimiento, ETL, inventario, semántica, trazabilidad.

Abstract

The objective of this project is to design an automated analytical tool that diagnoses the quality of inventory catalogs, integrating an ETL process for data extraction, transformation, and loading, structural validations based on the UNSPSC standard, and natural language processing techniques to detect semantic inconsistencies. Based on a consolidated master database, clustering and duplicate detection algorithms are applied to identify error patterns and classify records according to their quality level. It is based on data quality and information governance theories, incorporating good practices in standardization and traceability. Methodologically, a quantitative and experimental approach was adopted with an iterative design in four phases: construction of the knowledge base, development of the semantic diagnosis module, validation with real catalogs, and continuous improvement of the system. The prototype is expected to reduce manual validation time by at least 80%, improve coding, and ensure the auditability of processes. As a result, automated technical reports, visualizations, and a historical execution mechanism are generated, facilitating discovery by non-technical users. In addition, it contemplates the migration of the prototype to a web service to facilitate its deployment and remote collaboration in corporate environments. This approach strengthens the continuous availability and maintainability of the system, complying with international data governance recommendations. Finally, the proposal offers a scalable, accessible, and replicable solution that optimizes cataloging processes, provides operational efficiency, and raises the quality of information in the supply chain.

***Keywords*:** supply, ETL, inventory, semantics, traceability.

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Lista de Apéndices

Introducción

En la actualidad, la gestión eficiente de inventarios es un factor crítico para la competitividad de las empresas en la cadena de abastecimiento. La creciente diversidad de proveedores y formatos de catálogo, junto con errores semánticos y duplicidades, dificulta la trazabilidad y la interoperabilidad con sistemas ERP, lo que genera retrasos operativos y pérdida de precisión en la codificación de materiales. Esta problemática demanda soluciones automatizadas que permitan unificar y estandarizar la información de manera sistemática.

Con el fin de atender esta necesidad, el presente proyecto propone el desarrollo de una herramienta analítica fundamentada en una base de datos maestra semánticamente normalizada, complementada con módulos de diagnóstico para validar la estructura y jerarquía de cada registro. La aplicación integrará procesamiento de lenguaje natural, validación contra el diccionario UNSPSC y generación automática de reportes y visualizaciones, todo accesible desde una interfaz asistida. La finalidad del trabajo es optimizar los procesos de catalogación, mejorar la calidad de los datos y reducir los tiempos de revisión manual, contribuyendo a una toma de decisiones más ágil y precisa en entornos industriales.

Justificación

La propuesta de desarrollar una herramienta analítica para el diagnóstico automatizado de catálogos de inventario surge como una respuesta directa a una necesidad operativa crítica: mejorar la calidad de los datos de inventario que son la base de decisiones estratégicas en compras, logística, mantenimiento y control patrimonial. En contextos empresariales, la mala calidad de los datos puede generar sobrecostos, reprocesos, errores en la trazabilidad de materiales y pérdidas financieras evitables (Turquie, 2024).

Desde la perspectiva académica, este proyecto se alinea con las tendencias emergentes de la ciencia de datos, donde se prioriza la calidad del input antes del desarrollo de modelos predictivos. Tal como señalan Data Innovation, (2025) y IEBS (2029, los flujos de trabajo actuales requieren estructuras limpias, normalizadas y auditables para garantizar que los sistemas analíticos sean confiables. La integración de procesos ETL, validación estructurada, análisis de lenguaje natural y visualización de datos representa un enfoque disciplinar moderno, con aplicación directa en entornos reales.

En el plano personal, el proyecto representa una oportunidad de aplicar de forma práctica los conocimientos adquiridos durante la especialización, generando un producto útil, replicable y con potencial de escalabilidad. Finalmente, desde una dimensión social y organizacional, la herramienta propuesta podrá ser implementada en sectores como salud, infraestructura o educación, donde los errores en la catalogación afectan directamente la eficiencia operativa y la transparencia institucional (Tuduri, 2025)

Descripción del Problema

La heterogeneidad de formatos y la falta de un repositorio único de datos de inventario generan brechas críticas en la calidad de la información. Solo el 53 % de los líderes de la cadena de suministro califican la calidad de sus datos maestros como adecuada, lo que evidencia deficiencias en la integridad y visibilidad de los registros históricos de materiales (Overvest, 2025).

Esta falta de consistencia se traduce en bajos índices de precisión: en algunas empresas la exactitud de inventario llega apenas al 65 %, muy por debajo del estándar de 90–95 % considerado de clase mundial (Zeiger, 2024) Tales imprecisiones provocan retrasos operativos, sobrecostos en reposición y dificultades en la planificación logística, estimándose pérdidas globales superiores a USD 1,77 billones anuales en sectores minoristas y manufactura (Retail Insight, 2024).

La ausencia de procesos automatizados para detectar y corregir inconsistencias semánticas incrementa el riesgo de duplicados y codificación errónea, afectando directamente la trazabilidad y la interoperabilidad con sistemas ERP. Este problema cobra especial relevancia en industrias de alto volumen como hidrocarburos y minería, donde cada error en el catálogo puede detener líneas de producción y generar pérdidas millonarias.

Planteamiento del Problema

En la cadena de abastecimiento, la calidad de los catálogos de inventario es un factor crítico para la visibilidad y eficiencia operativa. Sin embargo, continúan gestionándose de forma manual y descentralizada por catalogadores o almacenistas, lo que introduce alto grado de subjetividad, baja trazabilidad y falta de control sistemático de los datos (Andrade Clavijo & Guerrero Cepeda, 2023; Rodríguez et al., 2021).

Los archivos de catálogo presentan errores estructurales frecuentes, descripciones duplicadas con variaciones mínimas, mezcla de idiomas y códigos UNSPSC mal asignados que afectan la precisión de la codificación y retrasan procesos de abastecimiento. Cada lote de 3 000 registros requiere entre 8 y 12 horas de revisión manual, acumulando semanas de trabajo improductivo y desaprovechando más de 200 000 registros históricos validados, sin aprovecharse como referencia para la detección de patrones o anomalías (Flores Castillo, 2024; Samaniego, 2019).

A pesar de la disponibilidad de flujos ETL estandarizados y técnicas de ciencia de datos, detección de duplicados, NLP, clustering y visualización inteligente el uso operativo real se limita a hojas de cálculo básicas (Betancourt, 2025). Esta brecha evidencia la necesidad de una solución automatizada que garantice eficiencia, trazabilidad y calidad uniforme de los catálogos.

Sistematización del Problema

- ¿Cómo unificar y normalizar registros históricos de inventario procedentes de distintas fuentes para construir una base de datos maestra semánticamente consistente?
- ¿Qué técnicas de procesamiento de lenguaje natural y validación semántica permiten detectar de manera automatizada duplicados e inconsistencias en las descripciones de los catálogos?
- ¿Cómo puede un proceso de carga y limpieza estructural reducir el tiempo y mejorar la calidad de los datos en comparación con el método manual actual?
- ¿De qué forma la generación automática de reportes y visualizaciones diagnósticas contribuye a la toma de decisiones en entornos logísticos y de compras?
- ¿Cómo garantizar la trazabilidad y la auditabilidad de los análisis mediante el registro histórico de ejecuciones?
- ¿Qué requisitos debe cumplir la interfaz gráfica para facilitar la ejecución asistida del diagnóstico por usuarios con perfiles no técnicos?

Objetivos

Objetivo General

Diseñar una herramienta analítica automatizada que permita diagnosticar la calidad de catálogos de inventario mediante procesos ETL, validaciones estructuradas, procesamiento de lenguaje natural y visualización inteligente.

Objetivos Específicos

Desarrollar un proceso automatizado de integración y limpieza de registros de inventario mediante técnicas ETL y normalización semántica, con el fin de crear una base de datos maestra que permita comparar y estandarizar nuevos catálogos recibidos de distintas fuentes.

Diseñar un módulo de diagnóstico inteligente que, mediante procesamiento de lenguaje natural, detección de duplicidad estructurada y agrupamiento automático, identifique errores recurrentes en los catálogos y evalúe objetivamente la calidad de cada registro.

Construir un sistema de visualización y generación automática de reportes técnicos que resuma los hallazgos del diagnóstico mediante gráficos interpretables y puntuaciones de calidad, con el fin de facilitar la toma de decisiones, mejorar la trazabilidad de errores y optimizar el tiempo operativo.

Marco de Referencia

Estado del Arte

En los últimos años, el uso de herramientas de ciencia de datos para mejorar la calidad de los catálogos de inventario ha cobrado relevancia, especialmente en sectores que buscan estandarizar información y reducir errores operativos. Una de las líneas más activas de investigación ha sido la aplicación de procesamiento de lenguaje natural (NLP) y técnicas de aprendizaje automático para analizar descripciones de productos y asignar códigos UNSPSC de forma automatizada (CertiDevs, 2022). Sin embargo, estos enfoques suelen probarse en conjuntos de datos limitados y rara vez se integran con flujos ETL completos ni validaciones jerárquicas sobre diccionarios oficiales.

Por ejemplo, mediante bibliotecas como spaCy, combinadas con limpieza semántica y embeddings, se ha logrado automatizar parcialmente el mapeo de productos a códigos estándar, alcanzando tasas de clasificación superiores al 29 % en bases de datos reales (CertiDevs, 2022).

Esta estrategia reduce el trabajo manual y mejora la trazabilidad, pero carece de mecanismos para incorporar validaciones estructuradas y reportes automáticos dentro de un mismo pipeline.

También se ha documentado el uso de modelos de lenguaje de gran escala (LLMs) para la clasificación de catálogos masivos, con mejoras en precisión y eficiencia operativa (Parada Torralba, 2024). No obstante, el elevado costo computacional y la opacidad de estos modelos dificultan su adopción en entornos industriales donde la auditabilidad y el control de versiones son críticos.

Por otra parte, las técnicas de clustering aplicadas a representaciones vectoriales de texto han permitido agrupar descripciones semánticamente similares que no coinciden textualmente,

mejorando la detección de duplicados no evidentes (Betancourt, 2025; Garrit, 2025). Estas soluciones, sin embargo, dependen de parámetros manuales y apenas se han integrado con validaciones jerárquicas o generación de dashboards.

En entornos de alto volumen, frameworks como Spark NLP sobre PySpark han demostrado ser adecuados para ejecutar flujos de diagnóstico distribuido sin comprometer el rendimiento (Microsoft Fabric, 2025). Aun así, la mayoría de las implementaciones se enfocan en etapas aisladas, dejando de lado la creación de reportes técnicos automatizados y la provisión de interfaces accesibles para usuarios no técnicos.

No obstante, a pesar de estos avances, persiste el vacío de soluciones integrales que combinen procesos ETL, validaciones semánticas y jerárquicas, generación automática de reportes y visualizaciones interactivas, todo en un único pipeline que garantice trazabilidad, escalabilidad y usabilidad para diferentes perfiles de usuario.

Marco Contextual

El presente proyecto se desarrolla en el contexto de Stock Gestión Integral S.A.S., una empresa privada del sector industrial dedicada a la consultoría de la cadena de abastecimiento, con énfasis en inventario, catalogación técnica y control de activos. Con operaciones a nivel nacional en Colombia—en sectores como hidrocarburos, minería, infraestructura y manufactura—atiende clientes como Masa Stork, Ocensa, Canacol, Cenit y Continental Gold, cuyos entornos exigen alta precisión en la clasificación y estandarización de materiales.

En este escenario, los procesos de inventario y catalogación se inician con levantamientos manuales en bodega, seguidos de correcciones y estandarizaciones por catalogadores según la codificación UNSPSC. Cuando se reciben catálogos antiguos, incompletos o mal estructurados,

la validación se efectúa mediante hojas de cálculo y criterios individuales, lo que genera dependencia de la experiencia personal, baja trazabilidad de errores y baja eficiencia operativa.

Frente a estas limitaciones, el proyecto propone una herramienta analítica automatizada capaz de diagnosticar la calidad estructural y semántica de catálogos de inventario, facilitando la validación masiva de registros, la detección de errores recurrentes y la generación de reportes estandarizados.

Marco Teórico

En la última década, la gestión de inventarios ha pasado de ser un proceso netamente operativo para convertirse en un componente estratégico dentro de las cadenas de suministro modernas. Este cambio ha sido impulsado por la digitalización, la trazabilidad y la necesidad de tomar decisiones basadas en datos limpios, confiables y estructurados (Rodríguez et al., 2021). Sin embargo, como señalan (Andrade Clavijo & Guerrero Cepeda, 2023) y (Cortez Vásquez & García Conde, 2024), muchas organizaciones aún enfrentan serios desafíos relacionados con la calidad de los datos en sus catálogos de inventario.

Diversos estudios han demostrado que la automatización del diagnóstico y la validación de datos puede reducir drásticamente el tiempo de análisis y aumentar la precisión del control de calidad. Por ejemplo, (Maestre, 2024) y (Garrit, 2025) evidencian cómo el uso de sistemas inteligentes permite pasar de procesos manuales de horas a validaciones automáticas en minutos, integrando reglas de negocio, puntuaciones de calidad y análisis predictivo.

Desde el enfoque metodológico, la implementación de procesos ETL se reconoce como paso obligatorio para cualquier tratamiento estructurado de datos. Este proceso permite extraer información de múltiples fuentes, transformarla mediante reglas específicas (normalización,

limpieza, mapeo) y cargarla en un repositorio base que se utilizará para análisis comparativos (Flores Castillo, 2024; Pratt, 2025).

Asimismo, la aplicación de procesamiento de lenguaje natural (NLP) ha tomado relevancia en entornos industriales, no solo para clasificación de texto, sino también para validar coherencia semántica en descripciones técnicas de productos (Universidad de Alcalá, 2025). Esta tecnología facilita la identificación de errores invisibles al análisis estructural clásico, como frases mal formadas, uso de sinónimos no controlados o mezcla de idiomas.

Otra técnica relevante es el clustering o agrupamiento automático, que permite identificar descripciones similares, aunque estén escritas de forma distinta. Esta técnica es especialmente útil para la detección de duplicados no textuales, es decir, aquellos que no coinciden exactamente, pero refieren al mismo ítem (Betancourt, 2025).

Finalmente, la incorporación de visualizaciones interactivas y dashboards como medio de entrega de resultados ha sido ampliamente validada como práctica recomendada para facilitar la interpretación de los datos y la toma de decisiones (INARQ, 2024; Turquie, 2024). Estas herramientas permiten a usuarios no técnicos comprender los hallazgos del sistema de diagnóstico y priorizar acciones de mejora en la calidad del catálogo.

Con base en esta revisión, el presente proyecto se ubica en la intersección entre ingeniería de datos, análisis automatizado y gestión de inventario, contribuyendo con una solución que no solo responde a una necesidad operativa, sino que incorpora prácticas reconocidas y tendencias actuales en ciencia de datos aplicada.

Marco Conceptual

Según Vélez Vélez y Pazmiño Linares (2022), la calidad de los datos es un pilar fundamental para la toma de decisiones en entornos organizacionales modernos. En el contexto de inventarios, esto se traduce en catálogos estructurados que describen ítems mediante atributos como código, unidad de medida, descripción corta, descripción larga y código UNSPSC. Un catálogo de inventario debe garantizar coherencia sintáctica, integridad semántica y normalización de formatos para ser confiable en procesos de abastecimiento y control patrimonial.

La validación estructurada se basa en reglas predefinidas que evalúan cada registro según criterios como duplicidad, longitud mínima o codificación estándar. Este enfoque requiere un proceso ETL (extract, transform, load) que unifique datos de distintas fuentes y los convierta en insumos comparables y auditables (Cabrera Cruz et al., 2022; DataCamp, 2025).

La automatización del diagnóstico mobiliza técnicas de procesamiento de lenguaje natural (NLP) para analizar las descripciones textuales, detectar errores semánticos, inconsistencias y mezclas de idioma. Según (Universidad de Alcalá, 2025), la aplicación de NLP en entornos industriales permite identificar fallos invisibles al análisis estructural clásico.

Para la detección de duplicados no textuales, se emplea clustering sobre vectores de texto, agrupando descripciones similares que no coinciden literalmente. Microsoft Fabric, (2025) y Quiza (2018) documentan cómo el clustering mejora la precisión en la identificación de registros redundantes, aunque su eficacia depende de la configuración de los parámetros de agrupamiento.

Finalmente, la inteligencia visual operativa busca transformar grandes volúmenes de datos en insights accionables mediante dashboards e informes automáticos. Este enfoque facilita

que usuarios no técnicos interpreten los resultados del diagnóstico, prioricen acciones de mejora y mantengan la trazabilidad de los hallazgos (INARQ, 2024; Universidad de Alcalá, 2025).

Marco Normativo

El desarrollo de una herramienta analítica para el diagnóstico de catálogos de inventario se fundamenta en el cumplimiento del estándar internacional UNSPSC (United Nations Standard Products and Services Code), un sistema de clasificación que permite categorizar productos y servicios mediante códigos jerárquicos de ocho dígitos (UNSPSC, 2023). Este sistema está organizado en cuatro niveles: segmento, familia, clase y producto, y ha sido adoptado globalmente para normalizar procesos de compras, logística, control de inventarios y contratación pública.

La implementación del UNSPSC en catálogos empresariales permite establecer criterios de coherencia, trazabilidad y análisis estructurado, lo cual es esencial en entornos industriales donde el volumen de referencias y la variabilidad de descripciones generan errores frecuentes de codificación. Su uso garantiza una identificación precisa de los ítems, facilita la auditoría y mejora la interoperabilidad con sistemas ERP (United Nations Development Programme & Dun & Bradstreet, 2022)

Además, el proyecto se enmarca en las recomendaciones técnicas establecidas por los marcos de gobierno de datos, los cuales promueven principios como la calidad, integridad, trazabilidad y auditabilidad de la información (DAMA International, 2017). Estas buenas prácticas fomentan la automatización de tareas repetitivas, la validación estructurada mediante reglas de negocio, y el diseño de estructuras orientadas a la eficiencia operativa y el control técnico.

Aunque no existe una regulación nacional específica para el diagnóstico automatizado de catálogos, las empresas del sector industrial en Colombia han adoptado de forma voluntaria estos estándares internacionales, en especial en sectores como hidrocarburos, minería y manufactura, donde los errores de codificación impactan directamente la cadena de abastecimiento y la gestión de activos.

Metodología

Método

El Este estudio se enmarca en un enfoque de investigación aplicada con método inductivo-analítico de carácter cuantitativo. Se adopta un diseño iterativo de desarrollo tecnológico, cuya estrategia integra flujos ETL (extracción, transformación y carga) (Cabrera Cruz et al., 2022; DataCamp, 2025), validación estructural jerárquica, procesamiento de lenguaje natural (Universidad de Alcalá, 2025), detección de duplicidad mediante clustering (Microsoft Fabric, 2025) y generación automatizada de visualizaciones y reportes técnicos.

La metodología se estructura en cuatro fases:

- Construcción de la base de conocimiento: procesamiento y normalización de registros históricos de inventario para crear la base de datos maestra.
- Diseño e implementación de la herramienta diagnóstica: desarrollo modular de los componentes de carga de datos, análisis semántico y exportación de resultados.
- Validación experimental: ejecución de pruebas con catálogos reales, medición de métricas de rendimiento (tiempo de proceso, precisión de detección) y comparación con el método manual.
- Mejora continua: retroalimentación de los hallazgos en la base de datos maestra y ajuste de algoritmos para garantizar la escalabilidad y confiabilidad del sistema.

Cada fase se documenta mediante código reproducible, estructuras de datos estandarizadas y mecanismos de auditoría interna, garantizando trazabilidad, replicabilidad y alineación con buenas prácticas de ingeniería de datos y desarrollo analítico.

Tipo de Estudio

Este estudio experimental de carácter aplicado y cuantitativo se orienta al desarrollo de una solución tecnológica funcional para automatizar el diagnóstico de calidad en catálogos de inventario. El enfoque del trabajo responde a una necesidad específica del entorno empresarial, combinando conocimientos técnicos en ciencia de datos con principios de ingeniería de software para construir una herramienta práctica y replicable.

La solución planteada se considera un proyecto de innovación tecnológica, ya que integra módulos de procesamiento de lenguaje natural, validación estructural, visualización analítica y mecanismos de retroalimentación continua. Su diseño responde a un contexto real y dinámico, donde la calidad del dato es un factor crítico para la trazabilidad y la eficiencia operativa.

Además, el enfoque metodológico es cuantitativo y experimental, dado que contempla la medición de resultados mediante métricas específicas: reducción de tiempos operativos, frecuencia de errores detectados, precisión del análisis y consistencia con estructuras validadas previamente. Estas evidencias permiten evaluar el impacto y la aplicabilidad del sistema en otros escenarios similares del sector industrial.

Recolección de Datos

La recolección de datos se realizó mediante análisis documental de cinco archivos históricos de inventario suministrados por la organización, en formato Excel y comprendidos entre 2018 y 2024. Cada archivo fue seleccionado con base en los siguientes criterios: inclusión de campos críticos (código del material, descripción corta y larga, unidad de medida, tipo de artículo y código UNSPSC) y procedencia de entornos operativos distintos. Se excluyeron archivos incompletos o con menos de 1 000 registros.

La información recopilada se sometió a un proceso ETL (extracción, transformación y carga), que incluyó validación de columnas clave, eliminación de nulos, detección de registros incompletos, corrección de formatos y homologación de unidades. Esta etapa permitió construir una base de datos maestra que sirve de referencia para evaluar la calidad de nuevos catálogos.

Para la fase experimental, se utilizaron catálogos de prueba independientes de la base histórica. Los diagnósticos manuales previos fueron registrados en informes técnicos elaborados por expertos y sirvieron como instrumento de comparación para medir la precisión, el nivel de coincidencia y la reducción de tiempos operativos.

Todos los datos recolectados se trataron exclusivamente con fines analíticos y no contienen información sensible o confidencial. El proceso está documentado mediante estructuras reproducibles que garantizan trazabilidad y replicabilidad del experimento.

Resultados

Primer Resultado

Desarrollar un proceso automatizado de integración y limpieza de registros de inventario mediante técnicas ETL y normalización semántica, con el fin de crear una base de datos maestra que permita comparar y estandarizar nuevos catálogos recibidos de distintas fuentes.

Construcción de la Base de Datos Maestra (BDM) y Normalización de Registros Históricos

El primer resultado obtenido en el desarrollo del proyecto corresponde a la creación de una Base de Datos Maestra (BDM) consolidada, que almacena y organiza los registros históricos de inventario provenientes de cinco archivos suministrados por la organización. Esta base actúa como núcleo referencial para el diagnóstico automatizado, permitiendo validar nuevos catálogos contra una estructura limpia, jerárquica y libre de duplicados.

Para ello, se implementó un flujo tipo ETL (extracción, transformación y carga) mediante scripts desarrollados en Python, que permitieron leer cada archivo Excel, validar la existencia de columnas clave, eliminar registros nulos, homologar formatos y unificar separadores textuales. Posteriormente, se aplicaron reglas de transformación para estandarizar las descripciones y extraer pares clave:valor desde las descripciones largas, los cuales fueron normalizados semánticamente.

La estructura de la BDM se implementó sobre MySQL 8.0, siguiendo principios de normalización estricta. Se crearon tablas especializadas para códigos UNSPSC (unspsc_codes), características por producto (characteristics), valores válidos (characteristic_values), materiales únicos (validated_materials), y mapeos de atributos (characteristic_value_mappings). Cada material histórico quedó representado por un hash único calculado a partir de sus atributos

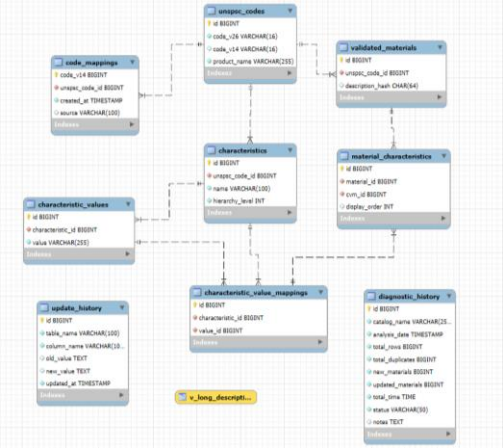
(SHA-256), lo que garantiza la unicidad de cada combinación de características, evitando la duplicación.

El resultado final de esta fase fue una BDM con registros limpios, categorizados por tipo de producto, con sus respectivas características jerarquizadas y listos para ser utilizados como referencia en el análisis automatizado. Esta base estructurada no solo permite comparaciones precisas, sino que también facilita auditorías y aprendizaje continuo al integrarse con nuevos catálogos procesados.

Apéndice A contiene el fragmento del script SQL que define la estructura relacional completa de la base de datos maestra.

Figura 1

Esquema Lógico de la Base de Datos Maestra.



Nota. Muestra la relación entre las tablas claves que conforman la BDM, incluyendo códigos UNSPSC, características por producto, valores, materiales validados, historial de diagnóstico y auditoría de cambios.

Tabla 1

Script para la Creación de la Base de Datos Maestra.

Tabla	Estructura principal (SQL simplificado)
unspsc_codes	id BIGINT, code_v26 VARCHAR(16), code_v14 VARCHAR(16), product_name VARCHAR(255)
characteristics	id BIGINT, unspsc_code_id BIGINT, name VARCHAR(100), hierarchy_level INT
validated_materials	id BIGINT, unspsc_code_id BIGINT, description_hash CHAR(64)

Nota. La tabla unspsc_codes almacena los códigos y nombres genéricos de productos. characteristics define atributos por tipo de producto, y validated_materials registra cada material único usando un hash semántico.

Carga y Limpieza Estructural del Catálogo Ingresado

El segundo componente desarrollado en la herramienta corresponde al módulo de carga y validación estructural del catálogo suministrado por el usuario. Esta parte del sistema representa el punto de entrada para el diagnóstico automatizado, y fue construida en lenguaje Python, empleando PySpark para la manipulación distribuida de datos y openpyxl para compatibilidad con archivos Excel.

El proceso inicia con la lectura del archivo .xlsx, verificando la existencia de las columnas críticas necesarias para el análisis, tales como: Descripción corta, Descripción larga, Unidad de medida, Código UNSPSC, Tipo de artículo, entre otras. Si alguna está ausente, el sistema emite una alerta inmediata e impide continuar con el diagnóstico.

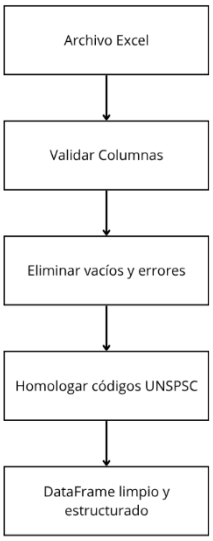
Una vez validado el archivo, se ejecuta la eliminación de filas vacías o incompletas, el ajuste de formatos y la conversión de separadores inconsistentes (como ;, /, -) a un formato estandarizado basado en comas. Este paso garantiza que los textos puedan ser analizados estructuralmente de forma homogénea.

Adicionalmente, se implementó un mecanismo de homologación automática de códigos UNSPSC versión 1.4 a versión 2.6, usando una tabla de equivalencias incluida en el sistema. Esto asegura que todos los registros trabajen con un mismo estándar jerárquico, evitando ambigüedades en la validación.

Todo el flujo fue encapsulado dentro del script cargar_datos.py (ver Apéndice B), y se integra con la barra de progreso visual de la interfaz gráfica, que indica al usuario el avance de esta etapa. El resultado de esta fase es un DataFrame limpio, estructurado y consistente, listo para ser procesado por los módulos de diagnóstico posteriores.

Figura 2

Flujo de Carga, Validación y Limpieza Estructural del Catálogo de Inventario



Segundo Resultado

Diseñar un módulo de diagnóstico inteligente que, mediante procesamiento de lenguaje natural, detección de duplicidad estructurada y agrupamiento automático, identifique errores recurrentes en los catálogos y evalúe objetivamente la calidad de cada registro.

Desarrollo del Módulo de Diagnóstico Semántico y Validación Estructural

El tercer componente estructural de la herramienta corresponde al módulo de diagnóstico semántico y validación estructural de los catálogos cargados. Este módulo permite identificar errores frecuentes como duplicados, claves no válidas, mezclas de idiomas, y registros con estructura incorrecta, alineándose con los lineamientos propuestos para el diagnóstico automatizado de calidad.

El módulo fue desarrollado en Python, incorporando bibliotecas como fuzzywuzzy, langdetect, spaCy, sklearn y re, e incluye múltiples subprocesos encadenados. En primer lugar, se implementó la detección de duplicados a través de dos enfoques: coincidencia exacta y similitud difusa, permitiendo identificar descripciones repetidas o muy similares, tanto en texto corto como en texto largo.

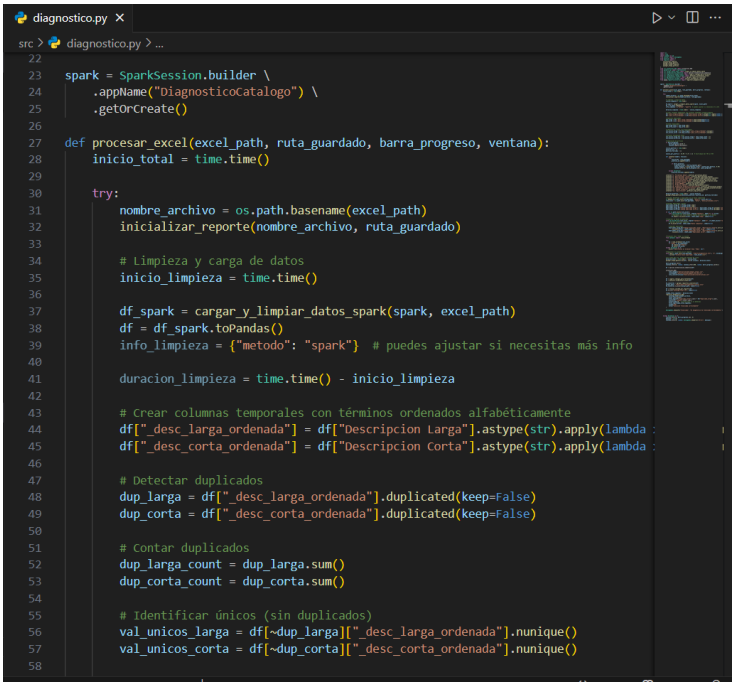
Luego, se integró un sistema de detección de idioma mediante análisis de tokens, el cual permite identificar si los textos están en español, inglés o mixto, y así señalar incoherencias lingüísticas que puedan afectar la clasificación automática.

Posteriormente, se desarrolló el componente de extracción y validación de pares clave:valor, el cual identifica atributos técnicos mediante expresiones regulares y valida que las claves extraídas correspondan a las jerarquías definidas por el código UNSPSC asociado. También se verifica si el orden jerárquico es correcto y si existen claves faltantes o fuera de lugar.

La validación estructural está respaldada por un diccionario jerárquico oficial, implementado como estructura tipo OrderedDict, el cual guía la comparación entre la estructura esperada y la real de cada registro. Esta lógica se encuentra encapsulada dentro del script diagnostico.py (ver Apéndice C), y forma el núcleo técnico del análisis.

Figura 3

Vista del Módulo de Diagnóstico Semántico Implementado en Python



Nota. Fragmento funcional del archivo diagnostico.py, en el cual se desarrollaron los procesos de carga, limpieza, detección de duplicados y conversión entre entornos Spark y Pandas. Este módulo forma parte central del flujo automatizado de análisis estructural y semántico de catálogos.

Extracción y Normalización Semántica de Características Clave: Valor

Este componente técnico desarrollado corresponde al módulo de extracción estructurada de características tipo clave: valor, presente en las descripciones largas de los catálogos. Este proceso es esencial para interpretar semánticamente cada ítem, facilitando su posterior validación contra diccionarios técnicos y estructuras jerárquicas normalizadas.

El módulo fue implementado en Python utilizando expresiones regulares personalizadas y estructuras tipo OrderedDict, que permiten mantener el orden original en que aparecen las

características. El sistema detecta automáticamente patrones del tipo Clave: Valor, considerando separadores múltiples como punto y coma, guiones o barras. Además, realiza una limpieza previa para homogeneizar estos separadores a un formato único y eliminar guiones residuales.

Una vez extraídos, los pares clave: valor son estandarizados: se eliminan guiones bajos o medios al inicio o final de cada componente, y se convierten en espacios si se ubican entre palabras. Esta normalización semántica permite reducir errores y facilitar comparaciones estructuradas posteriores.

Adicionalmente, se implementó una lógica para corregir estructuras desordenadas y detectar claves incompletas, repetidas o fuera de contexto, permitiendo marcar estos registros para revisión o exclusión. El resultado final de este módulo es una lista ordenada y limpia de características técnicas por registro, que luego se utiliza para validación jerárquica con el código UNSPSC asociado.

Todo el módulo fue encapsulado dentro del archivo normalizador.py (ver Apéndice E), y se integró con el flujo principal de diagnóstico.

Figura 4

Ejemplo de Limpieza y Normalización Semántica de Descripciones Técnicas

Nota. La imagen muestra el resultado del módulo de normalización semántica aplicado sobre una descripción con múltiples separadores. A la izquierda, la entrada original con formato inconsistente; a la derecha, la salida limpia con los pares clave:valor extraídos correctamente. Este proceso permite evaluar cada ítem contra estructuras técnicas jerárquicas.

Validación Jerárquica con Diccionario Oficial UNSPSC

El desarrollado corresponde al módulo de validación estructural, encargado de comparar las características clave:valor extraídas desde cada descripción con una jerarquía oficial de atributos válidos definidos por código UNSPSC. Este paso asegura que los registros cumplan con un estándar técnico alineado a normativas internacionales, evitando combinaciones arbitrarias o errores de clasificación.

El sistema implementa un diccionario técnico UNSPSC, cargado en memoria como un objeto tipo OrderedDict, que contiene para cada código de producto la lista ordenada de claves válidas junto con su nivel jerárquico. Esta jerarquía fue construida previamente a partir de catálogos maestros limpios y complementada con reglas de negocio internas.

Durante la validación, se verifica que:

- Todas las claves extraídas estén presentes en el diccionario del código UNSPSC correspondiente.

- Las claves estén en el orden correcto según su jerarquía.
- No haya claves repetidas ni fuera de contexto.
- Los valores asociados cumplan con patrones esperados (número, texto, unidades).

Los registros que no cumplen con estos criterios son marcados automáticamente para revisión, y el sistema los clasifica como estructuralmente inconsistentes. Además, se genera una lista de claves faltantes o excedentes, útil para retroalimentar al usuario.

La validación se implementó dentro del script validacion_unspsc.py (ver Apéndice F), el cual se conecta con el módulo anterior de normalización y es ejecutado para cada fila del catálogo.

Figura 5

Esquema de Validación Estructural Jerárquico UNSPSC

Nota. La figura muestra cómo el sistema compara las claves extraídas desde una descripción técnica con la jerarquía de claves válidas definida en el diccionario UNSPSC. Se identifican inconsistencias por claves no reconocidas, repetidas o fuera de orden, permitiendo calificar automáticamente cada registro como válido o inválido.

Evaluación Automatizada de Calidad por Registro

Este módulo desarrollado corresponde al sistema de evaluación de calidad individual por registro. Este componente tiene como objetivo calificar cada ítem del catálogo en función de su

integridad estructural, limpieza semántica y cumplimiento técnico, generando una etiqueta interpretativa que permite identificar registros completos, dudosos o inconsistentes.

La evaluación considera múltiples variables previamente diagnosticadas:

- Presencia de duplicados exactos o difusos.
- Detección del idioma (y mezcla de idiomas).
- Existencia de valores nulos en columnas críticas.
- Resultados de la validación con el diccionario UNSPSC.
- Longitud mínima de las descripciones.
- Presencia y formato correcto de pares clave:valor.

Cada uno de estos criterios es ponderado internamente, y el sistema genera una calificación tipo etiqueta para cada registro, por ejemplo: "Alta calidad", "Incompleto", "Estructura inválida" o "Revisión manual sugerida". Este resultado se agrega automáticamente como una nueva columna en el DataFrame final.

Este módulo fue implementado como función independiente dentro del script evaluador_calidad.py (ver Apéndice F), y puede activarse como paso adicional tras finalizar el diagnóstico.

Tabla 2

Etiquetas de Calidad Generadas Automáticamente por Registro Procesado

Descripción corta	Idioma	Duplicado	Validación UNSPSC	Etiqueta de calidad
Válvula de compuerta 2”	es	False	Estructura válida	Alta calidad
Filtro 3M con válvula	unknown	False	Clave inválida	Estructura inválida
Sello mecánico ACERO	es	True	Estructura válida	Duplicado
Tornillo hexagonal inox	en	False	Descripción insuficiente	Descripción insuficiente
Motor trifásico 1.5 HP	es	False	Sin características	Sin características

Nota. La tabla muestra ejemplos reales del sistema de evaluación de calidad. La etiqueta final se asigna con base en duplicación, idioma, validación estructural, longitud y existencia de características técnicas. Estas etiquetas permiten segmentar automáticamente los registros según su confiabilidad.

Tercer Resultado

Construir un sistema de visualización y generación automática de reportes técnicos que resuma los hallazgos del diagnóstico mediante gráficos interpretables y puntuaciones de calidad, con el fin de facilitar la toma de decisiones, mejorar la trazabilidad de errores y optimizar el tiempo operativo.

Generación Automatizada de Reportes Técnicos

Este séptimo módulo desarrollado corresponde al sistema de generación de reportes técnicos automáticos, encargado de consolidar los resultados obtenidos tras el diagnóstico en formatos exportables, legibles y organizados por tipo de hallazgo. Esta funcionalidad fue diseñada para facilitar la trazabilidad y análisis posterior por parte del usuario, sin necesidad de procesamiento adicional.

El sistema genera, de forma automática, los siguientes archivos estructurados por cada ejecución:

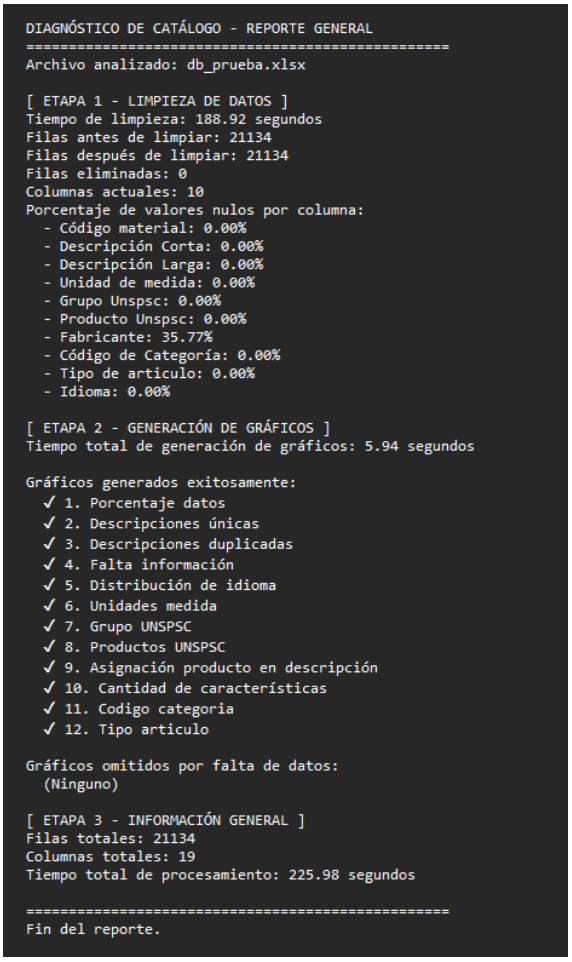
- Archivo .csv: contiene el catálogo procesado completo, incluyendo columnas adicionales como idioma detectado, claves extraídas, validación UNSPSC y etiqueta de calidad por registro.
- Archivo .xlsx: agrupa los hallazgos en hojas separadas, incluyendo duplicados de descripciones cortas y largas, registros con claves inválidas, elementos sin estructura reconocida, y aquellos con problemas de idioma o falta de características.
- Archivo .txt: actúa como reporte técnico final, presentando un resumen por etapas del flujo, tiempos de ejecución y cantidad de registros afectados por cada validación.

Estos archivos se almacenan automáticamente en una carpeta generada con nombre dinámico basado en la fecha y hora del análisis, lo que permite mantener un historial limpio y ordenado de cada ejecución sin intervención del usuario.

Todo el sistema de reportes fue encapsulado dentro del módulo reporte.py (ver Apéndice G), el cual recibe como entrada el DataFrame diagnosticado y los indicadores registrados durante cada etapa del flujo.

Figura 6

Fragmento del Reporte Técnico Generado Automáticamente



```
DIAGNÓSTICO DE CATÁLOGO - REPORTE GENERAL
==================================================
Archivo analizado: db_prueba.xlsx

[ ETAPA 1 - LIMPIEZA DE DATOS ]
Tiempo de limpieza: 188.92 segundos
Filas antes de limpiar: 21134
Filas después de limpiar: 21134
Filas eliminadas: 0
Columnas actuales: 10
Porcentaje de valores nulos por columna:
  - Código material: 0.00%
  - Descripción Corta: 0.00%
  - Descripción Larga: 0.00%
  - Unidad de medida: 0.00%
  - Grupo Unspsc: 0.00%
  - Producto Unspsc: 0.00%
  - Fabricante: 35.77%
  - Código de Categoría: 0.00%
  - Tipo de articulo: 0.00%
  - Idioma: 0.00%

[ ETAPA 2 - GENERACIÓN DE GRÁFICOS ]
Tiempo total de generación de gráficos: 5.94 segundos

Gráficos generados exitosamente:
  ✓ 1. Porcentaje datos
  ✓ 2. Descripciones únicas
  ✓ 3. Descripciones duplicadas
  ✓ 4. Falta información
  ✓ 5. Distribución de idioma
  ✓ 6. Unidades medida
  ✓ 7. Grupo UNSPSC
  ✓ 8. Productos UNSPSC
  ✓ 9. Asignación producto en descripción
  ✓ 10. Cantidad de características
  ✓ 11. Codigo categoria
  ✓ 12. Tipo articulo

Gráficos omitidos por falta de datos:
  (Ninguno)

[ ETAPA 3 - INFORMACIÓN GENERAL ]
Filas totales: 21134
Columnas totales: 19
Tiempo total de procesamiento: 225.98 segundos

==================================================
Fin del reporte.
```

Nota. La figura muestra un reporte técnico generado tras la ejecución de diagnóstico. En él se resumen el tiempo invertido por etapa, el porcentaje de datos faltantes por columna crítica, los gráficos generados y las estadísticas globales del catálogo procesado.

Generación automática de visualizaciones diagnósticas

Este El octavo módulo implementado corresponde al sistema de visualización automática, encargado de generar representaciones gráficas de los principales hallazgos del diagnóstico. Este componente tiene como propósito facilitar la interpretación visual de los datos

procesados y permitir una evaluación rápida de tendencias, errores y patrones dentro del catálogo.

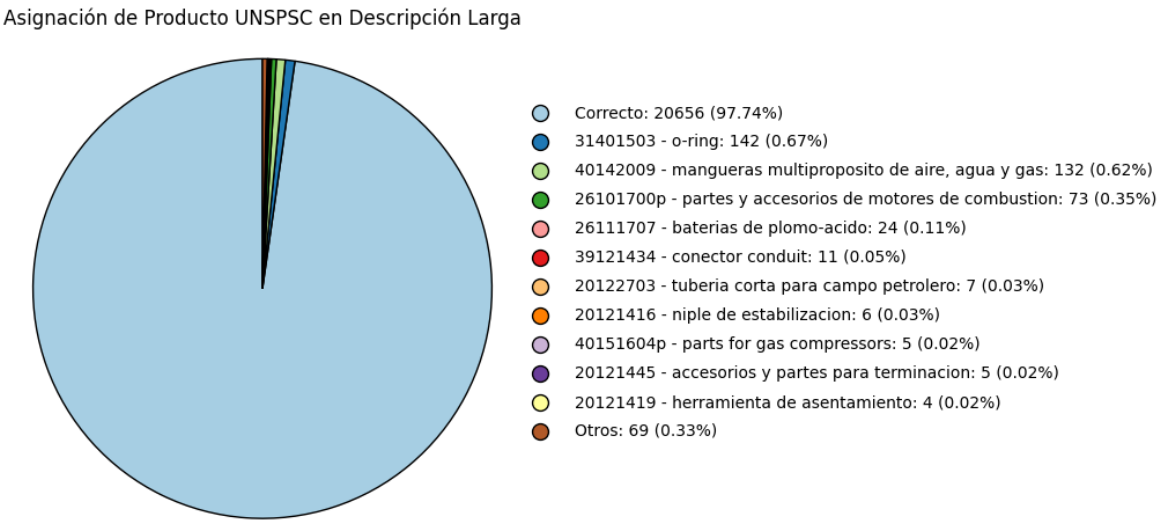
El sistema genera gráficos de forma automática utilizando la biblioteca matplotlib, sin requerir intervención del usuario. Los gráficos se exportan en formato .png y se almacenan dentro de la carpeta del análisis correspondiente. Las visualizaciones implementadas abarcan:

- Gráficos de pastel sobre:
  - Porcentaje de datos válidos/nulos por columna.
  - Proporción de registros por idioma detectado.
  - Distribución por tipo de artículo.
  - Porcentaje de descripciones duplicadas.
- Gráficos de barras que muestran:
  - Frecuencia de unidades de medida.
  - Distribución por grupo y producto UNSPSC.
  - Códigos de categoría más comunes.
  - Cantidad de características extraídas por registro.
- Gráfico de líneas (si se activa histórico): evolución del tiempo total de ejecución por lote procesado.

El módulo fue desarrollado dentro del script graficos.py (ver Apéndice H), y está diseñado para adaptarse automáticamente a la disponibilidad de datos. Si algún gráfico no puede generarse por falta de información, se omite sin afectar el flujo.

Figura 7

Asignación de Producto UNSPSC en Descripción Larga



Nota. La figura muestra un gráfico de pastel generado por el módulo graficos.py. Se representa la proporción de registros con asignación correcta del producto UNSPSC a partir de la descripción, en comparación con asignaciones erradas o ausentes. Este tipo de visualización permite detectar desviaciones en el uso de códigos estándar.

Registro de Histórico de Ejecuciones y Trazabilidad de Análisis

Este sistema de trazabilidad histórica de diagnósticos realizados por la herramienta. Este componente permite registrar automáticamente cada ejecución en una base de datos persistente, con el fin de mantener control sobre el volumen de datos procesados, tiempos de análisis, cantidad de duplicados, y evolución de nuevos materiales.

Para ello, se creó la tabla diagnostic_history dentro de la base de datos principal, la cual almacena por cada ejecución:

- Fecha y hora exacta del análisis.

- Nombre del catálogo procesado.
- Cantidad total de filas y columnas.
- Número de duplicados detectados
- Nuevos materiales identificados.
- Tiempo total de ejecución.
- Estado del análisis y notas opcionales.

Esta información se consulta posteriormente para realizar comparativas temporales, auditorías o seguimiento de mejoras en la calidad de los datos.

Tabla 3

Ejemplo de Registros en la Tabla Diagnostic_history

ID	Catálogo procesado	Fecha análisis	Filas	Duplicados	Nuevos materiales	Tiempo	Estado
1	catalogo_2025_06_24.xlsx	2025-06-24 10:48:12	21134	154	92	225.98	Completado
2	catalogo_2025_06_18.xlsx	2025-06-18 08:31:45	19547	98	61	201.44	Completado
3	catalogo_2025_06_10.xlsx	2025-06-10 14:22:19	20321	120	78	213.65	Completado

Nota. La tabla muestra ejemplos de ejecuciones registradas por el sistema. Cada fila representa un diagnóstico realizado, permitiendo mantener trazabilidad de volúmenes procesados, hallazgos detectados y eficiencia en el tiempo de análisis.

Interfaz Gráfica de Usuario para Ejecución Asistida del Diagnóstico

Este sistema corresponde a la implementación de una interfaz gráfica de usuario (GUI), desarrollada con la biblioteca customtkinter. Este componente tiene como finalidad facilitar la interacción con la herramienta para usuarios no técnicos, permitiendo ejecutar el diagnóstico de forma guiada y visual.

La interfaz permite:

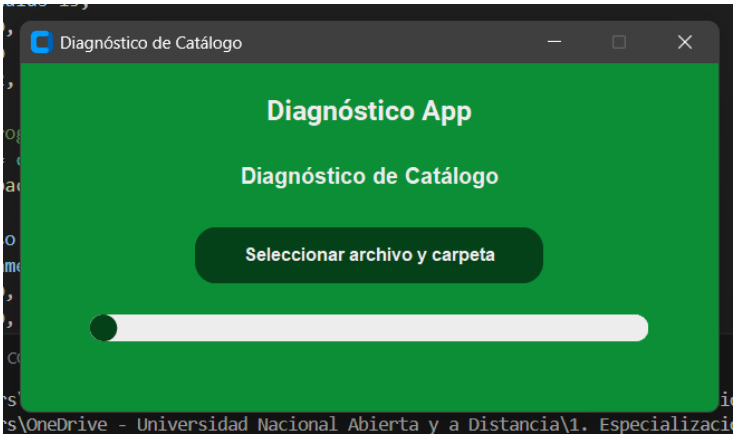
- Seleccionar el archivo Excel de entrada mediante explorador.
- Visualizar una barra de progreso que refleja el avance de cada etapa (limpieza, diagnóstico, visualización).
- Mostrar mensajes informativos, errores y advertencias durante el proceso.
- Generar automáticamente una carpeta de resultados, cuyo nombre incluye fecha, hora y nombre del archivo analizado.
- Ejecutar todo el flujo desde un botón central, sin necesidad de código.

Además, se integraron detalles visuales como el logotipo de la empresa, estilos personalizados en los botones, bordes redondeados y manejo de ventanas emergentes para reportes.

Este módulo se encuentra centralizado en el archivo ui.py (ver Apéndice H) y se enlaza directamente con el archivo main.py, que actúa como punto de entrada de la aplicación.

Figura 8

Interfaz Gráfica con Barra de Progreso Integrada



Nota. La figura muestra la interfaz desarrollada con la biblioteca customtkinter, donde el usuario puede seleccionar un archivo Excel para análisis, iniciar el proceso de diagnóstico y visualizar el avance mediante una barra de progreso. La interfaz está optimizada para facilitar la ejecución del sistema sin necesidad de conocimientos técnicos, integrando elementos visuales personalizados y mensajes informativos.

Conclusiones

La implementación de la base de datos maestra centralizada concentró un diccionario exhaustivo de materiales y garantizó la unicidad de cada registro mediante hashing y normalización, sentando las bases para futuras iniciativas de catalogación.

El módulo de normalización semántica y validación jerárquica corrigió errores descriptivos, unificó formatos y confirmó la conformidad con el estándar UNSPSC, automatizando la detección de duplicados e inconsistencias y reduciendo sustancialmente la revisión manual.

La generación automática de reportes técnicos, visualizaciones diagnósticas y el registro histórico de ejecuciones mejoró la toma de decisiones y disminuyó la dependencia de procesos manuales, aunque la complejidad de combinaciones de datos sigue siendo un reto de interpretabilidad.

Recomendaciones

Desarrollar e integrar dashboards interactivos con filtros por fecha, categoría UNSPSC y tipo de error, que muestren gráficos en tiempo real de indicadores de calidad, volumen de registros y duplicados para facilitar la exploración dinámica de los resultados.

Convertir la aplicación en un servicio web accesible desde cualquier navegador, con acceso protegido y actualizaciones automáticas, para que los usuarios puedan utilizarla sin instalar nada en sus equipos.

Referencias Bibliográficas

Andrade Clavijo, B. P., & Guerrero Cepeda, M. J. (2023). *Evaluación de la gestión de inventarios y facturación en el almacén Peregrine Falcón en la ciudad de Ambato*. https://ciencialatina.org/index.php/cienciala/article/view/5576

Betancourt, D. (2025). *Análisis de datos con PySpark: Data science nivel 1*. Betancourt Editorial.

Cabrera Cruz, S. A., Aguilar López, J. L., & Villegas Patraca, R. (2022). *Complementando inventarios biológicos con datos abiertos a través de protocolos de limpieza de datos*. https://ecoevorxiv.org/repository/view/6411/

CertiDevs. (2022). *PySpark para Apache Spark: API de Python para big data*. https://certidevs.com/curso-pyspark-apache-spark

Cortez Vásquez, A., & García Conde, R. U. (2024). *La Inteligencia artificial y sus implicancias en el Control Interno y la Gestión de los Inventarios*. https://www.researchgate.net/publication/388871147_La_Inteligencia_artificial_y_sus_implicancias_en_el_Control_Interno_y_la_Gestion_de_los_InventariosArtificial_Intelligence_and_its_implications_in_Internal_Control_and_Inventory_Management

DAMA International. (2017). *The DAMA Guide to the Data Management Body of Knowledge*. Technics Publications.

DataCamp. (2025). *Data & AI Trends & Predictions 2025*. https://www.datacamp.com/report/data-trends-predictions-2025

Flores Castillo, C. (2024). *Sistema de Control de Inventarios – 8 Mejores Opciones*. https://www.leafio.ai/es/blog/sistema-de-control-de-inventarios-8-mejores-opciones/

Garrit, R. R. (2025). *Las 5 tendencias clave en la gestión de procesos para 2025 que transformarán tu negocio* [LinkedIn]. https://www.linkedin.com/pulse/las-5-tendencias-clave-en-la-gesti%C3%B3n-de-procesos-para-garri-s--ozqze/

INARQ. (2024). *Diseño y construcción de dashboards específicos para operaciones logísticas con Power BI*. https://inarq.edu.pe/logistica/diseno-y-construccion-de-dashboards-especificos-para-operaciones-logisticas-con-power-bi

Maestre, R. (2024, noviembre 15). *Cómo los modelos predictivos revolucionan la gestión de inventarios*. https://www.rubenmaestre.com/descubre-como-los-modelos-predictivos-revolucionan-la-gestion-de-inventarios

Microsoft Fabric. (2025). *Análisis de datos con Apache Spark y Python*. https://learn.microsoft.com/en-us/fabric/data-science/python-guide/python-visualizations

Overvest, M. (2025). *Supply Chain Statistics—70 Key Figures of 2025* [Procurement Tactics]. https://procurementtactics.com/supply-chain-statistics/

Parada Torralba, P. (2024). *Tendencias y previsiones en Big Data y Data Analytics*. https://www.iebschool.com/hub/tendencias-en-big-data-e-inteligencia-artificial/

Pratt, M. K. (2025). *18 herramientas de ciencia de datos para utilizar en 2025*. https://www.techtarget.com/searchbusinessanalytics/feature/15-data-science-tools-to-consider-using

Quiza, J. (2018). *Exploración y Preprocesamiento de datos usando PySpark*. https://medium.com/datos-y-ciencia/exploraci%C3%B3n-y-pre-procesamiento-de-datos-credit-card-pyspark-d5afa9d30320

Retail Insight. (2024). *Unveiling the true cost of inventory inaccuracy*. https://www.retailinsight.io/blog/unveiling-the-true-cost-of-inventory-inaccuracy

Rodríguez, A. M., Fuentes, E., & Sabogal Cáceres, T. (2021). *Sistema de Gestión de Inventarios para Compañías de Hardware—Caso de Estudio*. https://ojs.urepublicana.edu.co/index.php/ingenieria/article/view/748

Samaniego, H. (2019). *Un modelo para el control de inventarios utilizando dinámica de sistemas*. https://www.redalyc.org/pdf/7198/719877735007.pdf

Tuduri, A. (2025). *Data Science y Big Data en 2025*. https://blogs.salleurl.edu/es/data-science-y-big-data-en-2025-tendencias-clave

Turquie, J. (2024). *Conoce todas las ventajas que brinda un dashboard de inventarios* [Intelligent Business Solutions]. https://ibso.mx/blog/ventajas-que-brinda-un-dashboard-de-inventarios

United Nations Development Programme, & Dun & Bradstreet. (2022). *United Nations Standard Products and Services Code (UNSPSC)*. https://www.unspsc.org/

Universidad de Alcalá. (2025). *Tendencias Emergentes en Ciencia de Datos para 2025*. https://www.master-data-scientist.com/tendencias-emergentes-ciencia-datos-2025/

Vélez Vélez, S. M., & Pazmiño Linares, S. A. (2022). *Importancia de los sistemas de inventarios en las organizaciones a través de una revisión bibliográfica*. https://www.alfapublicaciones.com/index.php/alfapublicaciones/article/view/163

Zeiger, D. (2024). *The Monthly Metric: Inventory Accuracy Rate*. https://www.ismworld.org/supply-management-news-and-reports/news-publications/inside-supply-management-magazine/blog/2024/2024-03/the-monthly-metric-inventory-accuracy-rate/

Apéndices

Apéndice A

Script SQL para la Creación de la Base de Datos Maestra

```
-- Tabla de códigos UNSPSC
CREATE TABLE IF NOT EXISTS unspsc_codes (
  id           BIGINT PRIMARY KEY AUTO_INCREMENT,
  code_v26     VARCHAR(16) NOT NULL UNIQUE,
  code_v14     VARCHAR(16) UNIQUE,
  product_name VARCHAR(255) NOT NULL
);


-- Tabla de características por código
CREATE TABLE IF NOT EXISTS characteristics (
  id               BIGINT PRIMARY KEY AUTO_INCREMENT,
  unspsc_code_id   BIGINT NOT NULL,
  name             VARCHAR(100) NOT NULL,
  hierarchy_level  INT NOT NULL,
  UNIQUE (unspsc_code_id, name),
  FOREIGN KEY (unspsc_code_id) REFERENCES unspsc_codes(id)
);


-- Tabla de materiales validados
CREATE TABLE IF NOT EXISTS validated_materials (
  id               BIGINT PRIMARY KEY AUTO_INCREMENT,
  unspsc_code_id   BIGINT NOT NULL,
  description_hash CHAR(64) UNIQUE,
  FOREIGN KEY (unspsc_code_id) REFERENCES unspsc_codes(id)
);
```

Apéndice B

Módulo de Carga y Validación Distribuida

Se presenta un fragmento representativo del script principal de carga, validación y limpieza de catálogos de inventario. El módulo fue desarrollado en PySpark, permitiendo la lectura

distribuida de archivos Excel, validación de columnas críticas, eliminación de registros nulos y transformación de descripciones para su posterior diagnóstico estructurado.

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, trim

# Crear sesión Spark
spark = SparkSession.builder \
    .appName("DiagnosticoCatalogo") \
    .getOrCreate()

# Cargar archivo Excel
df = spark.read.format("com.crealytics.spark.excel") \
    .option("header", "true") \
    .option("inferSchema", "true") \
    .load("archivo_catalogo.xlsx")

# Validar columnas requeridas
columnas_requeridas = ["Descripcion corta", "Descripcion larga",
"Unidad de medida", "Codigo UNSPSC"]
for col_name in columnas_requeridas:
    if col_name not in df.columns:
        raise ValueError(f"Falta la columna requerida: {col_name}")

# Limpiar separadores y espacios
df = df.withColumn("Descripcion larga", trim(col("Descripcion
larga")))
df = df.na.drop(subset=columnas_requeridas)
```

Apéndice C

Diagnóstico Semántico y Validación Estructural

Se presenta un fragmento representativo del módulo encargado de realizar el diagnóstico semántico y estructural de los catálogos. Este componente aplica técnicas de procesamiento de lenguaje natural, validación jerárquica y análisis de duplicados, basándose en reglas personalizadas y estructuras jerárquicas externas.

```
from fuzzywuzzy import fuzz
from langdetect import detect
from collections import OrderedDict
import re

# Detección de duplicados por similitud
def detectar_duplicados(df, columna):
    duplicados = []
    for i, desc in enumerate(df[columna]):
        for j, otra in enumerate(df[columna]):
            if i != j and fuzz.ratio(desc, otra) > 90:
                duplicados.append((i, j))
    return duplicados

# Detección de idioma
def detectar_idioma(texto):
    try:
        return detect(texto)
    except:
        return "unknown"

# Extracción de pares clave:valor
def extraer_pares(texto):
    pattern = r"([\w\s\-]+):([\w\s\-\./]+)"
    return re.findall(pattern, texto)
```

Apéndice D

Extracción y Limpieza de Características Técnicas Tipo Clave: Valor desde Descripciones Largas de Inventario

Esta lógica asegura una interpretación semántica coherente, facilitando la validación posterior contra jerarquías UNSPSC.

```
import re

# Normaliza separadores y limpia claves/valores
def normalizar_pares(texto):
    if not isinstance(texto, str):
        return []
```

```
    texto = texto.replace(";", ",").replace("/", ",").replace("-",
",")
    texto = re.sub(r",+", ",", texto)
    pares = re.findall(r"([^,:]+):([^,:]+)", texto)
    resultado = []
    for clave, valor in pares:
        clave = clave.strip().replace("_", " ").strip("-").strip()
        valor = valor.strip().replace("_", " ").strip("-").strip()
        resultado.append((clave, valor))
    return resultado
```

Apéndice E

Validación Jerárquica con Diccionario UNSPSC

Se presenta un fragmento del módulo encargado de validar las características clave: valor extraídas desde las descripciones, comparándolas contra un diccionario oficial UNSPSC cargado en memoria. Esta validación asegura que cada ítem cumpla con el conjunto correcto de atributos y el orden jerárquico correspondiente a su categoría UNSPSC.

```
# Diccionario jerárquico de claves válidas por código
jerarquia_unspsc = {
    "40141616": ["Tamaño", "Material", "Espesor"],
    "26131602": ["Voltaje", "Frecuencia", "Potencia"]
}

# Validar claves extraídas para un producto
def validar_caracteristicas(codigo_unspsc, claves_extraidas):
    claves_validas = jerarquia_unspsc.get(codigo_unspsc, [])
    inconsistencias = []

    for i, clave in enumerate(claves_extraidas):
        if clave not in claves_validas:
            inconsistencias.append(f"Clave inválida: {clave}")
        elif claves_validas.index(clave) != i:
            inconsistencias.append(f"Orden incorrecto: {clave}")

    return inconsistencias if inconsistencias else ["Estructura
válida"]
```

Apéndice F

Calificación Registros del Catálogo con Base en Múltiples Criterios

Es responsable de calificar cada registro del catálogo con base en múltiples criterios: duplicación, limpieza de texto, validación jerárquica, longitud mínima y formato estructurado. Este análisis genera una etiqueta interpretativa que se añade como nueva columna en el catálogo procesado.

```
def evaluar_calidad_registro(fila):
    if fila["Duplicado"]:
        return "Duplicado"
    if fila["Idioma"] not in ["es", "en"]:
        return "Idioma desconocido"
    if fila["Validacion UNSPSC"] != "Estructura válida":
        return "Estructura inválida"
    if len(fila["Descripcion larga"]) < 30:
        return "Descripción insuficiente"
    if pd.isnull(fila["Pares clave:valor"]) or len(fila["Pares
clave:valor"]) == 0:
        return "Sin características"
    return "Alta calidad"
```

Apéndice G

Generación de Reportes Técnicos

Se presenta un fragmento representativo del módulo reporte.py, encargado de generar automáticamente los archivos de salida en formato .csv, .xlsx y .txt. Estos reportes consolidan los hallazgos y métricas obtenidas en cada etapa del diagnóstico para su posterior trazabilidad y revisión técnica.

```
import pandas as pd
import os
from datetime import datetime

def inicializar_reporte(nombre_archivo, ruta_guardado):
```

```
    with open(os.path.join(ruta_guardado, "reporte_diagnostico.txt"),
"w") as f:
        f.write(f"Reporte técnico generado: {datetime.now()}\n")
        f.write(f"Archivo analizado: {nombre_archivo}\n")
        f.write("-" * 50 + "\n")

def agregar_etapa_general(filas, columnas, tiempo_total,
ruta_guardado):
    with open(os.path.join(ruta_guardado, "reporte_diagnostico.txt"),
"a") as f:
        f.write("Resumen general del archivo:\n")
        f.write(f"Filas procesadas: {filas}\n")
        f.write(f"Columnas: {columnas}\n")
        f.write(f"Tiempo total: {tiempo_total:.2f} segundos\n\n")
```

Apéndice H

Generación de Gráficos e Interfaz Gráfica

Este apéndice presenta fragmentos clave de los módulos encargados de generar visualizaciones automáticas y de implementar la interfaz gráfica de usuario. El archivo graficos.py produce las figuras de diagnóstico almacenadas como .png, mientras que ui.py construye la interfaz con customtkinter, facilitando la interacción visual y la ejecución asistida del análisis.

import matplotlib.pyplot as plt

```
def generar_grafico_pastel(datos, etiquetas, titulo, ruta_guardado):
    fig, ax = plt.subplots()
    ax.pie(datos, labels=etiquetas, autopct='%1.2f%%')
    ax.set_title(titulo)
    plt.savefig(ruta_guardado)
    plt.close()

import customtkinter as ctk
from tkinter import filedialog
from diagnostico import procesar_excel

def iniciar_app():
    app = ctk.CTk()
    app.title("Diagnóstico de Catálogo")
```

```
    app.geometry("500x300")

    def seleccionar_archivo():
        path = filedialog.askopenfilename(filetypes=[("Excel files",
"*.xlsx")])
        if path:
            procesar_excel(path, barra_progreso=barra)

    boton = ctk.CTkButton(app, text="Seleccionar archivo",
command=seleccionar_archivo)
    boton.pack(pady=20)

    barra = ctk.CTkProgressBar(app)
    barra.pack(pady=10)
    barra.set(0)

    app.mainloop()
```