DATAXPERIENCE

PROYECTO DE CIENCIA DE DATOS EN GESTIÓN DE INVENTARIOS

INTEGRANTES

Adriana Maria Melo Portilla

Paula Fernanda Munevar Cantor

Diego Alejandro Aroca Cataño

TUTOR

Camila Silva Gomez

AÑO 2025

INTRODUCCIÓN

La gestión de inventarios es uno de los procesos más críticos dentro de cualquier organización ya que influye directamente en la disponibilidad de los productos, costos operativos y satisfacción del cliente.

Un inventario desbalanceado puede generar tanto quiebres de inventario en algunos puntos como sobre inventario en otros, lo que distorsiona los pronósticos y afecta el aprovisionamiento de manera importante. Este desajuste provoca situaciones en las que no hay inventario suficiente para atender la demanda, impactando el nivel de servicio, o bien escenarios en los que el exceso de productos permanece almacenado por largos periodos.

Como resultado, se incrementan los costos de almacenamiento, se inmoviliza capital de trabajo y aumenta el riesgo de obsolescencia o vencimiento de la mercancía.

Actualmente, se presentan excesos de inventario en diferentes tiendas, los cuales no solo generan sobrecostos y limitaciones de espacio, sino que también distorsiona los cálculos de aprovisionamiento y compras futuras. Esta situación hace que los sistemas de planificación interpreten una falsa necesidad o demanda, provocando decisiones ineficientes en la reposición de mercancía y dificultando mantener un equilibrio real entre la oferta y la demanda, lo que impacta tanto la rotación del inventario como la precisión de las proyecciones de compra.

Este proyecto aplica un ciclo de ciencia de datos para analizar y gestionar inventarios a partir de tres bases de datos iniciales con el fin de unificar, limpiar y procesar la información y finalmente calcular las necesidades y excesos que permitan una toma de decisiones más acertada.

OBJETIVO GENERAL

Implementar un proceso de análisis de datos para la gestión de inventarios que permita diseñar un proceso de nivelación de inventarios que identifique y gestione los excesos en tiendas, con base en criterios objetivos, de manera que se logre un aprovisionamiento más preciso y alineado con las necesidades reales de la empresa.

OBJETIVOS ESPECÍFICOS

1. Unificar y depurar las bases de datos de inventario para formar una base de datos general consolidada.
2. Identificar los excesos de inventario aplicando datos y fórmulas con condiciones lógicas para las tiendas bajo criterios cuantitativos como: costo, rotación, días de inventario y necesidades reales de la operación.
3. Analizar el impacto de estos excesos en la planificación de compras, evidenciando la distorsión que generan en los cálculos de aprovisionamiento.
4. Reducir el impacto de los excesos en los cálculos de reposición, asegurando que las proyecciones de compra reflejen la demanda real.
5. Generar indicadores de control y seguimiento que permitan evaluar periódicamente el nivel de inventarios y ajustar oportunamente la estrategia de aprovisionamiento.

MARCO CONCEPTUAL

El análisis de datos es una estructura sistemática que permite transformar grandes volúmenes de información en conocimiento útil para la toma de decisiones. Esta estructura ayuda a comprender de manera más detallada los datos y mejorar la eficacia en su análisis.

El ciclo de análisis abarca desde la identificación del problema y la recolección de información, pasando por una limpieza o preprocesamiento y el análisis exploratorio para lograr desarrollar la ejecución de los modelos predictivos hasta la presentación y comunicación de resultados, garantizando que los datos sean confiables, procesables y generen valor estratégico.

Su propósito es comprender el estado real de los procesos de una organización, al igual que detectar tendencias, anomalías o patrones en la información base y así apoyar la toma de decisiones de una manera más acertada y lograr optimizar procesos y recursos en las organizaciones.

Aspectos a tener en cuenta en un análisis de datos:

- **Calidad de los datos**: Si los datos están incompletos o incorrectos, el análisis perderá valor.
- **Consistencia**: Es importante alinear nombres de variables, formatos de fechas y unidades de medida para lograr establecer el análisis de manera efectiva.
- **Contexto de la organización:** Es importante conocer el entorno y realidad de la organización para comprender y lograr unos resultados más precisos.
- **Flexibilidad**: Diseñar procesos que permitan actualizar los datos con facilidad.
- **Documentación**: Cada paso debe estar documentado para garantizar transparencia.

HERRAMIENTAS UTILIZADAS EN EL ANÁLISIS DE DATOS

El ciclo de análisis de datos se apoya en un conjunto de herramientas tecnológicas que facilitan las fases de recolección, transformación, análisis y visualización de la información. Estas herramientas permiten, además, automatizar tareas repetitivas y garantizar precisión, trazabilidad y escalabilidad en los proyectos.

En el presente trabajo se utilizaron tanto herramientas específicas del proyecto como un conjunto de herramientas ampliamente utilizadas en el campo de la ciencia de datos.

Herramientas utilizadas en el proceso:

Google colab: Es una herramienta de entorno de programación en la nube, basado en Python, esta permite ejecutar códigos sin necesidad de instalar un aplicativo en computadoras, facilita el acceso colaborativo y la integración con Google drive.

Google drive: Se utiliza como repositorio de almacenamiento de las bases de datos iniciales, permite el acceso compartido entre usuarios y la sincronización en la nube.

Python: Es el lenguaje principal de programación utilizado en el análisis de datos por su gran flexibilidad y disponibilidad de librerías.

Pandas: Es una librería de Python cuyo propósito es la manipulación de datos en formato tabular, es decir dataframes, en el proyecto su uso esta aplicado a la lectura de archivos, la limpieza de las columnas, la unificación de la base de datos, la creación de variables y la agrupación y resumen de la información.

GitHub: Es una plataforma en la nube para alojar repositorios de Git, lo cual permite la colaboración de proyectos, controlar versiones y gestionar cambios en el código de manera eficiente.

METODOLOGÍA

El desarrollo del análisis de datos se llevó a cabo en un ciclo de cuatro etapas, las cuales permitieron transformar las diferentes bases de datos en una información clara para la toma de decisiones.

1. Recolección, preparación de datos, procesamiento y limpieza.
2. Análisis de necesidades y excesos de inventarios.
3. Definición de criterios y parámetros de gestión.
4. Obtención de resultados finales.

ETAPA 1: CREACIÓN DE BASE DE DATOS GENERAL.

En esta etapa se integran las fuentes de información iniciales y se construye la base principal de trabajo, su propósito es unificar la información proveniente de las diferentes bases de datos en una única base consolidada que asegure la consistencia de la información.

Esta etapa consta de:

Autorización de acceso y carga de bases de datos: Se cargaron 3 archivos en el entorno de Google Colab en la cual se encuentran la base de datos de atributos, la base de parámetros y la base de inventarios.

```
"""Etapa 1. Crear base de datos general"""

"""Autoriazacion acceso Colab a google Drive"""
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd
"""Cargar bases de datos"""
df_base_Inv = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/Base EAN Inventarios.xlsx')
df_base_Atri = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/Base EAN Atributos.xlsx')
df_base_Parm = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/Base EAN Parametros.xlsx')
```

Ilustración 1 Autorización de acceso y carga de bases de datos. Elaboración propia, Python, 2025.

Visualización inicial y corrección de títulos: Se revisaron los títulos de las columnas, corrigiendo inconsistencias que podrían efectuar el análisis.

```
"""Visualización inicial de las Bases de datos DataFrame"""
display(df_base_Inv.head(2))
display(df_base_Atri.head(2))
display(df_base_Parm.head(2))
```

Ilustración 2 Visualización inicial y corrección de títulos. Elaboración propia, Python, 2025.

	CODIGO SAP	NOMBRE	NOMBRE PROVEEDOR	COSTO UND	DEPARTAMENTO	MUNDO	CATEGORIA	TIPO PAGO	ALTO	ANCHO	PROFUNDO
0	80467294	APOSITO NEXCARE TEGADERM 10X12CM X6	3M COLOMBIA S A	20486	PROD MEDICOS ESPECIALIZADOS	SOLUCIONES DE SALUD	ASEPSIA Y CUIDADO HERIDAS	FIRME	21.10	14.10	3.10
1	80469882	ZOLOF TAB 50MG X10	3M COLOMBIA S A	69094	FARMACEUTICOS	ETICOS	N_SISTEMA NERVIOSO	FIRME	6.00	8.00	10.00

	CentroMaterial	Material	Centro	Stat.mat.especif.ce.	Indicador ABC	Aprovis.especial	Punto de pedido	Stock máximo	Rotacion Mes
0	A3180460077	80460077	A301	NaN	A	ZA	15	20	8
1	B4020460077	80460077	B402	NaN	A	ZA	8	12	10

0

Ilustración 3 Visualización inicial y corrección de títulos. Elaboración propia, Python, 2025

Homologación de formatos: Se verificaron tipos de datos, alineando formatos para permitir concatenación y cruces mediante llaves, donde se busca preparar los datos de las tres bases de datos (Inventarios, atributos y parámetros) para cruzarlos y combinarlos en una sola base de bastos maestra.

```
"""Codigos o llaves para el cruce (Centro + Material)"""
df_base_Inv["Llave"] = df_base_Inv["Centro"].astype(str) + "_" + df_base_Inv["Material"].astype(str)
df_base_Parm["Llave"] = df_base_Parm["Centro"].astype(str) + "_" + df_base_Parm["Material"].astype(str)
```

Ilustración 4 Homologación de formatos- Elaboración propia, Python, 2025

Limpieza y filtros iniciales: Se aplicó la limpieza de columnas a la base de inventarios para depuración y eliminar datos no funcionales de tal manera que se facilite el análisis; Posterior a ello se realiza el filtro del data Frame para mantener solo las filas donde la columna 'Almacen' esté disponible, garantizando que solo se considere el inventario disponible actualmente.

```
"""Limpieza de datos inventarios (Titulos y filtros)"""
"""Limpiar titulos"""
df_base_Inv = df_base_Inv.drop(columnas=['Trans./Trasl.', 'ALTO', 'ANCHO', 'PROFUNDO'], eje=1)
"""Filtrar texto inventario disponible"""
df_base_Inv = df_base_Inv[df_base_Inv["Almacen"].isin(["Disponible"])]
mostrar(df_base_Inv.head(2))
```

Ilustración 5 Limpieza y filtros iniciales. Elaboración propia, Python, 2025

Unificación de bases de datos: Este código, consolida la información del inventario con sus parámetros y atributos correspondientes, uniendo los datos de las tres data frames en un único data frame completo.

Este paso es fundamental para el procesamiento de datos cuando la información está distribuida en diferentes fuentes y se necesita

consolidarla para realizar análisis más complejos como calcular las necesidades y excesos.

Normalización de títulos: Se renombraron las columnas de la nueva base de datos para evitar duplicados y errores de interpretación.

Duplicación de la base general: Se genera una copia de la base general y se renombra para realizar pruebas sin alterar la base principal.

ETAPA 2: CÁLCULOS DE NECESIDADES Y EXCESOS DE INVENTARIO.

En esta etapa se busca detectar materiales con riesgo de faltantes o necesidad y sobre stock o exceso.

2.1. Necesidades de inventario: Se calcula la necesidad de inventarios con una fórmula que considera la demanda, los niveles actuales de stock y los límites máximos de stock y se aplica un condicional para descartar valores negativos, evitando interpretaciones erróneas.

En este código se ejecuta la resta de la necesidad con la libre utilización del stock máximo, se utiliza el código. clip(lower=0) para garantizar que el resultado no sea negativo, es decir, si el uso gratuito ya supera la existencia máxima la necesidad es considerada cero.

Ilustración 6 Necesidades de inventario. Elaboración propia, Python, 2025

Agrupación de la necesidad por material: Se realiza un agrupamiento por material para consolidar los requerimientos, de esta manera se toma la información detallada de la necesidad de inventario calculada en la línea anterior para cada combinación específica de centro y material y así lograr consolidarla.

El resultado de este código es un resumen donde se exprese la necesidad total para cada material individual sin importar el centro en el que se encuentre esa necesidad particular.

De igual manera se realiza una eliminación de columnas que aunque estaba presentes en los datos originales, estos ya no tendrían un significado interpretativo o relevante en el resultado consolidado y se filtra el data frame resultante para mantener solo los materiales donde la necesidad total es mayor que cero.

```
"""Agrupar la necesidad por material"""
df_base_Necesidad_Consol= df_base_Necesidad.groupby('Material_x', as_index=False).sum()
"""Limpiar titulos"""
df_base_Necesidad_Consol= df_base_Necesidad_Consol.drop(columns=['Centro_x','Almacen','Libre utilización','Stock máximo',
```

Ilustración 7 Necesidades de inventario. Elaboración propia, Python, 2025

	Material_x	Necesidad
0	80450077	19.0
1	80450084	6.0

Ilustración 8 Necesidades de inventario. Elaboración propia, Python, 2025.

2.2. Excesos de inventario: En esta línea de código se identifica y cuantifica cuánto exceso de inventario hay para cada combinación específica del centro y material.

El cálculo se realiza mediante una transformación de columnas a nivel de filas, no implica agrupar y resumir datos iniciales como el cálculo de la necesidad, solo se toman los valores de las dos columnas existentes en cada fila y se realiza la operación matemática de resta y nuevamente se aplica .clip(lower=0) para asegurar que el resultado no sea negativo y se realiza un filtrado del data Frame para mostrar solo las filas donde el exceso es mayor que cero.

```
"""Filtrar necesidad mayor que cero"""
df_base_Necesidad_Consol = df_base_Necesidad_Consol[df_base_Necesidad["Necesidad"]> 0]
display(df_base_Necesidad_Consol.head(2))
```

Ilustración 9 Excesos de inventario. Elaboración propia, Python, 2025.

Centro_x	Almacen	Libre utilización	Stock máximo	Rotacion Mes	NOMBRE	NOMBRE PROVEEDOR	COSTO UND	MUNDO	Excesos	Dias Inv Excesos
M301	Disponible	11.0	9	9	CRE DEN COLGATE TOTAL CLEANMINT X75G	LABORATORIOS MINERALIN S A S	6233	CUIDADO PERSONAL	2.0	6.7
M301	Disponible	8.0	6	1	CRE DEN COLGATE TOTAL12 CLEAN MINT X195G	LABORATORIOS NATURAL FRESHLY SAS	12284	CUIDADO PERSONAL	2.0	60.0
M608	Disponible	6.0	4	6	CRE DEN COLGATE TOTAL12 CLEAN MINT X195G	LABORATORIOS NATURAL FRESHLY SAS	12284	CUIDADO PERSONAL	1.0	6.0
M301	Disponible	10.0	9	8	CRE DEN COLGATE SENSITIVE BLANQ X100G	LABORATORIOS CERO S A	6362	CUIDADO PERSONAL	1.0	3.8

Ilustración 10 Excesos de inventario. Elaboración propia, Python, 2025.

Cálculo del exceso de inventario: En este código se determina la diferencia entre lo que se tiene actualmente y lo que se desea idealmente como máximo, de esta manera si se tiene más que el máximo esa diferencia es un excedente o si se tienen menos o igual que el máximo su excedente es cero.

El cálculo se realiza mediante una comparación directa entre lo que hay y lo que debería haber como máximo y aísla solo la cantidad que exceda ese máximo almacenando en una nueva columna de excesos.

ETAPA 3: DEFINICIÓN DE CRITERIOS Y PARÁMETROS

En esta fase se establecen las reglas de análisis para definir qué materiales requieren atención prioritaria y se definen los siguientes parámetros.

1. Exceso Reubicable: Corresponde al inventario excedente en una tienda que puede ser utilizado para cubrir necesidades detectadas en otras tiendas. Este criterio filtra los excesos considerando únicamente aquellos materiales que presentan demanda activa en otros puntos de venta, permitiendo optimizar la redistribución del inventario, reducir riesgos de vencimiento u obsolescencia y mejorar la disponibilidad global de productos.

```
"""Criterio 1: EXCESO REUBICABLE"""
df_base_Excesos_C1= df_base_Excesos.copy()
df_base_Excesos_C1 = pd.merge(df_base_Excesos_C1, df_base_Necesidad_Consol, on=['Material_x'], how='left').fillna(0)
df_base_Excesos_C1 = df_base_Excesos_C1[df_base_Excesos_C1['Necesidad']> 0]
display(df_base_Excesos_C1.head(2))
```

Ilustración 11 Exceso reubicable. Elaboración propia, Python, 2025.

2. Exceso prolongado: Corresponde al inventario excedente que supera los 15 días de cobertura, umbral definido con base en la frecuencia quincenal de compra.
Este criterio identifica los excesos que no se corregirán de manera natural mediante la rotación regular ni con el siguiente ciclo de abastecimiento.

```
"""Criterio 2: EXCESO PROLONGADO (15 DIAS)"""
df_base_Excesos_C2= df_base_Excesos_C1.copy()
df_base_Excesos_C2 = df_base_Excesos_C2[df_base_Excesos_C2["Dias Inv Excesos"]> 15]
display(df_base_Excesos_C2.head(2))
```

Ilustración 12 Exceso prolongado. Elaboración propia, Python, 2025.

3. Clasificación de Prioridad: Se segmenta el exceso a partir de un ranking final ponderado, asignando niveles de prioridad que permiten enfocar esfuerzos en los materiales de mayor impacto para la operación. Se utiliza un análisis de tipo Pareto (porcentaje acumulado) para agrupar los artículos en categorías de prioridad y facilitando la toma de decisiones estratégicas de la siguiente manera:
3.1. Ranking individual para Exceso y necesidad
3.2. Ranking final combinando los rankings de exceso y necesidad.
3.3. Clasificar exceso por nivel de prioridad:
Prioridad 1: Del 0 al 25% acumulado -> máxima atención.
Prioridad 2: Del 25% al 50% -> atención alta.
Prioridad 3: Del 50% al 75% -> atención media.
Prioridad 4: Del 75% al 100% -> menor prioridad

```
"""Criterio 3: CLASIFICACION DE EXCESO Y NECESIDAD"""
"""ranking individual"""
df_base_Excesos_C3= df_base_Excesos_C2.copy()
df_base_Excesos_C3['Rank_Exceso'] = df_base_Excesos_C3['Exceso'].rank(method='min', ascending=False)
df_base_Excesos_C3['Rank_Necesidad'] = df_base_Excesos_C3['Necesidad'].rank(method='min', ascending=False)
"""ranking final"""
df_base_Excesos_C3['Rank_Final'] = df_base_Excesos_C3['Rank_Exceso'] + df_base_Excesos_C3['Rank_Necesidad']
df_base_Excesos_C3['Rank_Final'] = df_base_Excesos_C3['Rank_Final'].rank(method='min', ascending=True)
df_base_Excesos_C3= df_base_Excesos_C3.drop(columns=['Rank_Exceso','Rank_Necesidad','Almacen','Dias Inv Exceso'], axis=1)
"""Clasificacion del exceso"""
df_base_Excesos_C3 = df_base_Excesos_C3.sort_values(by='Rank_Final', ascending=True)
df_base_Excesos_C3['%_Acumulado'] = (df_base_Excesos_C3['Rank_Final'] / df_base_Excesos_C3['Rank_Final'].sum()).cumsum() * 100
def clasificar_Exceso(af):
    if af <= 25:
        return "Prioridad 1"
    elif af <= 50:
        return "Prioridad 2"
    elif af <= 75:
        return "Prioridad 3"
    else:
        return "Prioridad 4"
```

Ilustración 13 Clasificación de prioridad. Elaboración propia, Python, 2025.

	NOMBRE	NOMBRE PROVEEDOR	COSTO UND	MUNDO	Excesos	Necesidad	Rank_Final	%_Acumulado	Clasificacion_Exceso
	BOLSA COMPRA MEDIANA LOCATEL 30x45	SAJUFOOD COLOMBIA SAS	253	HOGAR	961.0	4116.0	1.0	0.000964	Prioridad 1
	GUJA P/PEN ALFASAFE 51G X5MM CJA X100	CASA IBANEZ COLOMBIA S.A.S	28169	CONSULTORIO E INSTRUMENTAL MEDICO	467.0	758.0	2.0	0.000150	Prioridad 1
	DEO ELEMENTAL MUJER ROLL ON X70ML	POLIUMED SAS	2104	CUIDADO PERSONAL	663.0	476.0	3.0	0.000396	Prioridad 1
	BOLSA COMPRA MEDIANA LOCATEL 30x45	SAJUFOOD COLOMBIA SAS	253	HOGAR	164.0	4116.0	4.0	0.000643	Prioridad 1
	ERSOD ELEMENTAL MUJER 48H +VIT E BRS X50G	NATURAL CARE S.A.	3864	CUIDADO PERSONAL	275.0	450.0	5.0	0.000964	Prioridad 1
	...	...	...	...	...	...	...	...	...
	LEVOFLOXACINA SANDOZ TAB 500MG X7	PROQUIDENT S.A.	15488	ETICOS	1.0	1.0	1719.0	99.888987	Prioridad 4

Ilustración 14 Clasificación de prioridad. Elaboración propia, Python, 2025

ETAPA 4: PLAN DE EJECUCIÓN DE REUBICACIÓN DE EXCESOS

Como resultado del análisis de gestión de inventarios y la segmentación de excesos por prioridad, se diseña el siguiente plan de trabajo para la

ejecución de las acciones de reubicación, buscando optimizar el uso del inventario y mejorar la disponibilidad en los puntos de venta.

Metodología de Ejecución

Envío de listados de excesos:
Se remite a cada tienda el listado de productos en exceso clasificados por prioridad, junto con las instrucciones de alistamiento y empaque.

Alistamiento y empaque:
Las tiendas verifican cantidades, alistan los productos identificados y los embalan de acuerdo con el estándar definido para garantizar su correcta manipulación.

Traslado al CEDI:
Se generan los documentos de traslado y se programa el transporte hacia el CEDI, asegurando trazabilidad en cada movimiento.

Recepción y almacenamiento:
En el CEDI se realiza el conteo, conciliación y registro en el sistema, para luego ubicar los productos en el inventario disponible.

Redistribución hacia tiendas con necesidad:
Con base en el análisis de necesidades, se programan los despachos desde el CEDI hacia las tiendas deficitarias, cerrando así el ciclo de balanceo de inventario.

RESULTADOS

Aplicación de los criterios de limpieza de la base de datos:

Después de aplicar los criterios de limpieza, la base de datos pasó de 8492 registros a 1771 registros, lo que demuestra que una parte considerable de los datos originales no cumplía con los estándares de calidad definidos. Esta depuración asegura que los análisis posteriores se realicen con información más confiable, precisa y relevante.

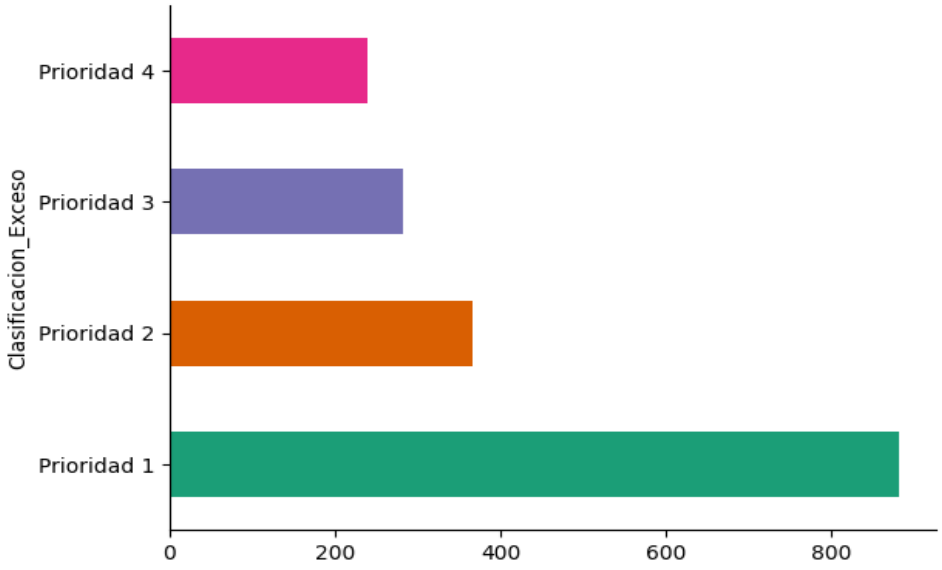
	Libre utilización	Stock máximo	Rotacion Mes	COSTO UND	Excesos	Dias Inv Excesos
count	8492.000000	8492.000000	8492.000000	8.492000e+03	8492.000000	8492.000000
mean	10.293028	5.272138	4.976095	3.614949e+04	5.020890	138.132595
std	41.938298	26.336435	3.182328	1.089080e+05	26.074179	558.281895
min	1.000000	0.000000	0.000000	1.050000e+02	0.002000	0.000000
25%	3.000000	2.000000	2.000000	1.041175e+04	1.000000	6.000000
50%	5.000000	3.000000	5.000000	2.212100e+04	2.000000	15.000000
75%	9.000000	4.000000	8.000000	4.086600e+04	4.000000	45.000000
max	2364.000000	2200.000000	10.000000	8.332500e+06	1380.000000	41400.000000

Ilustración 15 Excesos antes de aplicar los criterios. Elaboración propia, Python, 2025.

	Libre utilización	Stock máximo	Rotacion Mes	COSTO UND	Excesos	Necesidad	Rank_Final	%_Acumulado
count	1771.000000	1771.000000	1771.000000	1771.000000	1771.000000	1771.000000	1771.000000	1771.000000
mean	19.045970	9.546019	3.184077	28876.517222	9.499950	27.459905	878.578769	33.473181
std	74.364373	55.632841	2.904700	39720.830671	34.191009	156.300299	504.053752	29.877695
min	1.000000	0.000000	0.000000	120.000000	0.090000	1.000000	1.000000	0.000064
25%	5.000000	3.000000	1.000000	8663.000000	2.000000	2.000000	440.000000	6.323544
50%	9.000000	4.000000	2.000000	19125.000000	4.000000	5.000000	884.000000	25.177655
75%	15.000000	8.000000	5.000000	34841.000000	8.000000	15.000000	1327.000000	56.482738
max	2364.000000	2200.000000	10.000000	728235.000000	961.000000	4116.000000	1719.000000	100.000000

Ilustración 16 Excesos antes de aplicar los criterios. Elaboración propia, Python, 2025

Generación de gráfica de excesos por clasificación de prioridades:



Gráfica 1 Excesos por clasificación de prioridad. Elaboración propia, Python,2025.

La gráfica muestra la distribución de referencias con exceso de inventario agrupadas en cuatro prioridades.

Prioridad 1 concentra el 49,8 % de las referencias (882 materiales), representando prácticamente la mitad de los excesos detectados.

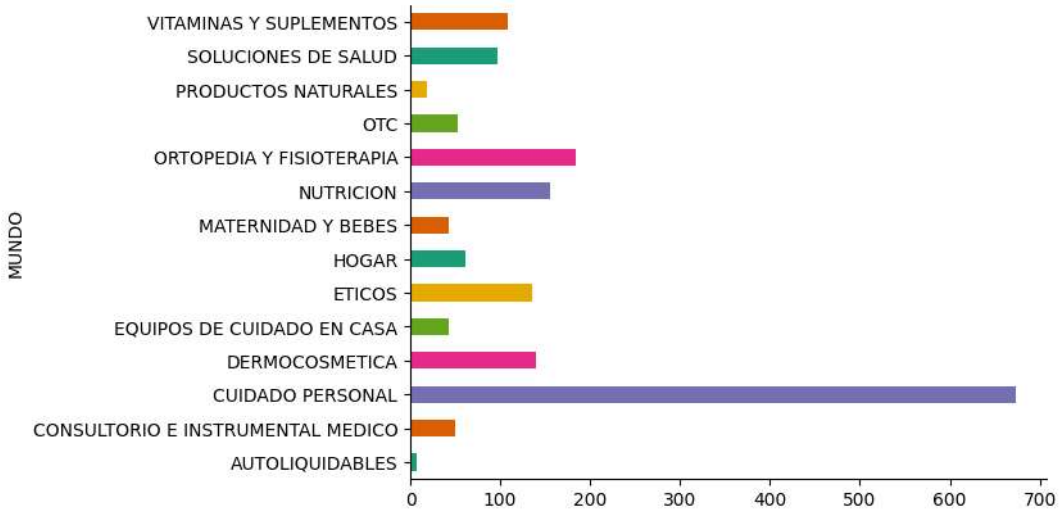
Prioridad 2 representa un 20,7 % de los materiales, lo que indica que cerca de 7 de cada 10 referencias con exceso se encuentran en las dos primeras prioridades.

Prioridades 3 y 4, aunque menos representativas (16 % y 13,5 %), suman en conjunto casi el 30 % restante, por lo que no deben ser descartadas en el plan de acción, aunque su urgencia es menor.

Por lo anterior podemos concluir que la alta concentración en prioridad 1 justifica que las acciones de reubicación y corrección se enfoquen inicialmente en este grupo, ya que representa la mayor oportunidad de optimización de inventario. De manera adicional la combinación de Prioridad 1 y 2 (70,5 %) confirma que la mayor parte del exceso es accionable en el corto plazo, y que una correcta ejecución sobre estos dos grupos tendría un impacto significativo en la reducción de sobrecostos y en la mejora de rotación.

En el caso de las prioridades 3 y 4 pueden ser gestionadas en fases posteriores, o mantenerse bajo monitoreo, evitando generar sobrecarga operativa innecesaria en la primera etapa.

Generación de gráfica de excesos por mundo:

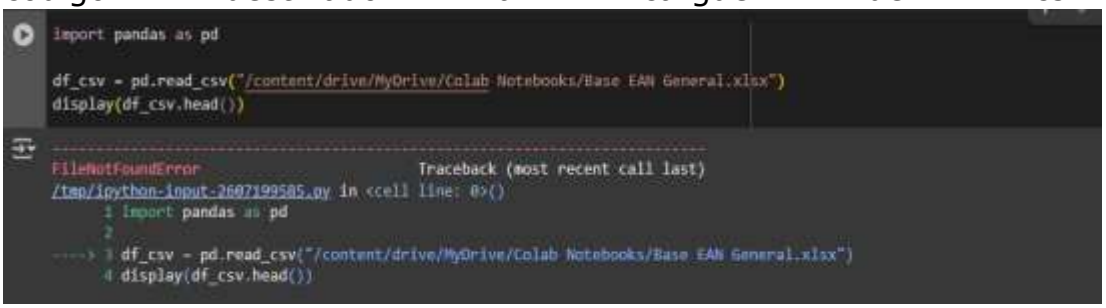


Gráfica 2 Excesos por mundo. Elaboración propia, Python,2025

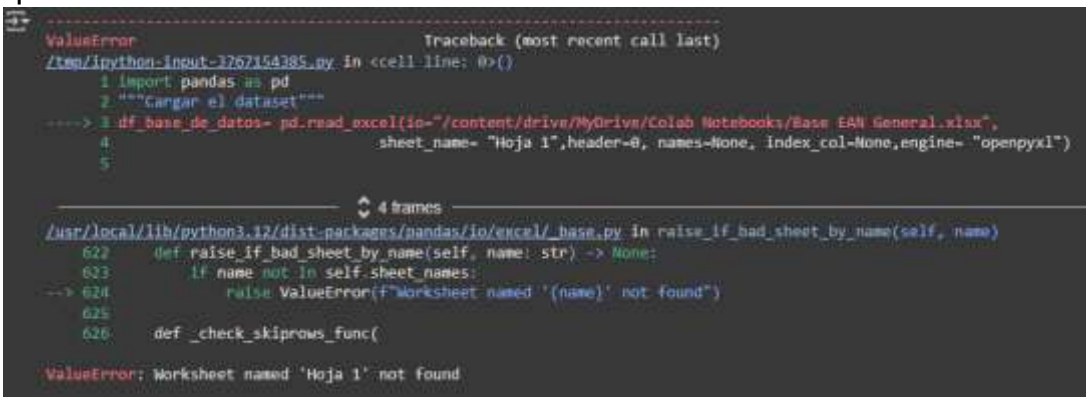
La gráfica muestra la distribución de excesos por categoría de producto denominada “mundo”. Muestra que el mundo con mayor exceso es la de cuidado personal con 674 materiales, equivalente al 38% y la de menor exceso es autoliquidables con 7 materiales que equivale al 0,4%

PROBLEMAS DETECTADOS DURANTE LA EJECUCIÓN DE ANÁLISIS.

1. Error en el cargue de dataframe: Se detectó un error “File Not Found Error” debido a un error en el cargue de la base de datos .xlsx con código destinado al cargue de .csv



2. Error en definición del sheet name: Se detectó un error “Value error” debido a un error en la definición del sheet name, se debe tener precaución de llamar correctamente la hoja del documento que necesitemos.



3. Error de escritura: Se presentó un error “Name error” debido a la incorrecta escritura, por esto, no se logra encontrar el data frame.

4. Error de escritura: Se presentó un error “KeyError” debido a la escritura de una variable usando espacios que no estaban en el

nombre original.

```
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in drop(self, labels, errors)
   7068         if mask.any():
   7069             if errors != "ignore":
-> 7070                 raise KeyError(f"{labels[mask].tolist()} not found in axis")
   7071             indexer = indexer[~mask]
   7072         return self.delete(indexer)

KeyError: "[' CATEGORIA '] not found in axis"
```

5. Renombramiento de la base de datos: Al renombrar una base de datos no se genera una nueva copia independiente, sino que se crea una referencia al mismo objeto en memoria, las modificaciones realizadas en una de las variables afectan directamente a la otra, ya que ambas apuntan al mismo objeto.

```
"""Duplicar base general"""
df_base_Necesidad= df_base_Tnv_Parm_Atri
df_base_Excesos= df_base_Tnv_Parm_Atri
display(df_base_Necesidad.head(2))
display(df_base_Excesos.head(2))
```

CONCLUSIÓN

El ciclo de análisis de datos permitió transformar archivos dispersos y heterogéneos en información clara, organizada y útil para la gestión de inventarios. Se logró transformar información fragmentada en un conocimiento claro y estratégico, evidenciando tanto las necesidades críticas como los excesos que afectan la eficiencia operativa, donde la organización de cada etapa iniciando desde la integración y limpieza de datos, hasta la definición de criterios y priorización de resultados nos mostró que el análisis sistemático no solo nos representa riesgos y oportunidades ocultas, sino que también facilita la toma de decisiones más acertadas.

Se obtuvieron 2 gráficas donde se identifica información clara sobre la clasificación de excesos por prioridades y por mundos, esto facilita la identificación de comportamiento de excesos y facilita la toma de decisiones frente el tratamiento de los mismos.

Con el trabajo se demuestra que el estudio estructurado de los datos es una herramienta clave para optimizar la gestión, anticipar problemas y orientar acciones que fortalezcan la sostenibilidad y el equilibrio de los recursos disponibles en las empresas o cualquier organización.

REFERENCIAS

Astera Marketing. (2025). Blog Las 15 mejores herramientas de análisis de datos en 2025.https://www.astera.com/es/type/blog/data-analysis-tools/

Datacamp. (2024). ¿Qué es el análisis de datos? Una guía experta con ejemplos. https://www.datacamp.com/es/blog/what-is-data-analysis-expert-guide?utm_source=chatgpt.com