## Evaluación del módulo 3

Consigna del proyecto



## Evaluación del módulo

Proyecto: Preparación de datos

## Situación inicial 📍



Unidad solicitante: Equipo de Analítica de Datos de una empresa de e-commerce

📌 El equipo de analítica de datos ha recibido la solicitud de preparar y estructurar un conjunto de datos provenientes de diversas fuentes (archivos CSV, Excel y páginas web) para su posterior análisis. Actualmente, los datos presentan múltiples problemas: valores perdidos, registros duplicados, formatos inconsistentes y presencia de outliers que distorsionan los resultados.

La problemática a resolver consiste en transformar y limpiar estos datos, garantizando su calidad y estructuración para que puedan ser utilizados en futuros modelos predictivos y reportes de negocio. La gerencia de datos ya ha definido las herramientas a utilizar y solicita la implementación técnica de un flujo de trabajo integral para el tratamiento de los datos.

## Nuestro objetivo 📋

El objetivo principal del proyecto es desarrollar un proceso automatizado y eficiente para la obtención, limpieza, transformación, análisis y estructuración de datos utilizando Python y las librerías NumPy y Pandas.

Al finalizar el proyecto, se espera contar con un dataset limpio, confiable y estructurado, listo para ser utilizado en procesos de análisis y toma de decisiones en la organización.

Este objetivo responde a la necesidad de la empresa de disponer de datos de calidad para la elaboración de reportes estratégicos y la implementación de modelos de machine learning.



## Requerimientos 🤝

## Requerimientos generales:

- → Implementar un flujo de trabajo que incluya las etapas de carga, limpieza, transformación y estructuración de datos.
- → Utilizar exclusivamente las librerías NumPy y Pandas para la manipulación de los datos.
- → Documentar cada paso del proceso para asegurar la trazabilidad y comprensión del trabajo realizado.

## Requerimientos técnicos específicos:

- → Leer datos desde archivos CSV, Excel y páginas web.
- → Identificar y gestionar valores nulos mediante imputación o eliminación.
- → Detectar y tratar outliers aplicando técnicas estadísticas.
- → Realizar tareas de Data Wrangling: ordenamiento, eliminación de duplicados, reemplazo y transformación de valores.
- → Aplicar agrupamiento y pivotado de datos utilizando funciones como groupby(), pivot() y melt().
- → Combinar múltiples fuentes de datos mediante merge() y concat().
- → Exportar el dataset final en formato CSV y Excel.
- → Incluir un script Python funcional y modularizado para ejecutar todo el proceso.

## Paso a paso 👣

Este proyecto refiere exclusivamente al **módulo 3**: Obtención y preparación de datos, y se compone de **6 etapas (lecciones)**, las cuales podrás avanzar de forma progresiva y escalonada con la ayuda de los manuales teóricos y los contenidos desarrollados en las clases en vivo.

Ten en cuenta de invertir **tiempo asincrónicos** para el desarrollo de cada etapa a modo de poder finalizar el módulo y realizar la entrega formal de tu propuesta.



Cualquier consulta que surja compártela en los espacios sincrónicos para resolver las dudas en equipo.

A continuación encontrarás las consignas y tareas a desarrollar:

- Lección 1 La librería numpy
  - **Objetivo:** Crear un conjunto de datos ficticio utilizando NumPy, aplicando operaciones básicas para la preparación inicial.

### Tareas a desarrollar:

- 1. Crear un archivo .py o un Notebook .ipynb.
- 2. Generar datos ficticios de clientes y transacciones utilizando arrays de NumPy.
- 3. Aplicar operaciones matemáticas básicas (suma, media, conteo, etc.).
- 4. Guardar los datos generados en un archivo .npy o convertirlos a listas para usarlos luego en Pandas.
- 5. Explicar en un breve documento por qué NumPy es eficiente para el manejo de datos numéricos.
- → **Nota**: Este archivo servirá de insumo para la siguiente lección, donde estos datos serán cargados y explorados con Pandas.
- Lección 2 La librería pandas
  - **Objetivo:** Explorar y transformar los datos generados en la Lección 1, utilizando la estructura de DataFrame de Pandas.

#### Tareas a desarrollar:

- 1. Leer los datos preparados en NumPy y convertirlos en un DataFrame.
- 2. Realizar una exploración inicial:
  - Visualizar primeras y últimas filas.
  - Obtener estadísticas descriptivas.
  - Aplicar filtros condicionales.
- 3. Guardar el DataFrame preliminar en un archivo CSV para ser utilizado en la siguiente lección.



- 4. Redactar un documento breve describiendo los hallazgos y la utilidad de Pandas para la manipulación de datos.
- Lección 3 Obtención de datos desde archivos
  - **Objetivo:** Integrar datos de diversas fuentes y unificarlos en un solo DataFrame para su posterior limpieza.

#### Tareas a desarrollar:

- 1. Cargar el archivo CSV generado en la Lección 2.
- 2. Incorporar nuevas fuentes de datos:
  - Leer un archivo Excel con información complementaria.
  - Extraer datos de una tabla web usando read\_html().
- 3. Unificar las diferentes fuentes de datos en un único DataFrame.
- 4. Guardar el DataFrame consolidado y documentar los desafíos encontrados al obtener datos de distintos formatos.
- → **Nota**: Este DataFrame unificado será la base para realizar la limpieza y transformación en las siguientes etapas.
- Lección 4 Manejo de valores perdidos y outliers
  - **Objetivo:** Aplicar técnicas de limpieza de datos, resolviendo problemas de valores nulos y datos atípicos.

## Tareas a desarrollar:

- 1. Identificar valores nulos en el DataFrame consolidado.
- 2. Aplicar técnicas de imputación, eliminación o categorización para gestionar los valores nulos.
- 3. Detectar outliers utilizando técnicas como IQR y Z-score.
- 4. Documentar las decisiones tomadas y cómo impactan en la calidad del dataset.
- 5. Guardar el DataFrame limpio para ser usado en la siguiente etapa.
- Lección 5 DATA WRANGLING
  - **© Objetivo:** Transformar y enriquecer los datos mediante técnicas de manipulación avanzada.



#### Tareas a desarrollar:

- 1. Tomar el DataFrame limpio de la Lección 4.
- 2. Aplicar técnicas de Data Wrangling:
  - Eliminar registros duplicados.
  - Transformar tipos de datos.
  - Crear nuevas columnas calculadas.
  - Aplicar funciones personalizadas (apply(), map(), lambda).
  - Normalizar o discretizar columnas según sea necesario.
- 3. Guardar la nueva versión del DataFrame optimizado.
- Lección 6 Agrupamiento y pivoteo de datos
  - 🎯 **Objetivo:** Organizar y estructurar los datos para el análisis utilizando técnicas de agrupamiento y pivotado.

## 📍 Tareas a desarrollar:

- 1. Tomar el DataFrame final de la Lección 5.
- 2. Aplicar técnicas de agrupamiento (groupby()) para obtener métricas resumidas.
- 3. Reestructurar los datos utilizando pivot() y melt().
- 4. Combinar nuevas fuentes de ser necesario con merge() y concat().
- 5. Exportar el DataFrame final listo para análisis en formatos CSV y Excel.
- 6. Elaborar un documento resumen explicando todo el flujo de trabajo realizado, desde la Lección 1 hasta la Lección 6.

## ¿Qué vamos a validar? 🔍



#### Aspectos técnicos:

- → Correcto uso de las librerías NumPy y Pandas.
- → Legibilidad, modularización y organización del código.
- → Aplicación correcta de las técnicas vistas: imputación, detección de outliers, wrangling, agrupamiento y combinación de datos.
- → Exportación del dataset limpio y transformado.



#### **Aspectos estructurales:**

- → Cumplimiento de todos los requerimientos generales y específicos.
- → Documentación clara y detallada que explique cada etapa del proceso.
- → Calidad de la estructura final del dataset.

#### Aspectos de performance:

- → Correcta gestión del tiempo en la resolución del proyecto.
- → Claridad en la presentación y entrega de los resultados.
- → Capacidad de resolución de problemas encontrados durante el tratamiento de los datos.

## Referencias 🦺



- → Documentación oficial de NumPy: <a href="https://numpy.org/doc/">https://numpy.org/doc/</a>
- → Documentación oficial de Pandas: <a href="https://pandas.pydata.org/docs/">https://pandas.pydata.org/docs/</a>

## Recursos 🎁



- → <a href="https://github.com/pandas-dev/pandas">https://github.com/pandas-dev/pandas</a>
- → clientes\_ecommerce.csv
- → clientes\_ecommerce.xlsx

## Entregables 🗸

Al finalizar el proyecto "Preparación de Datos con Python", se espera un consolidado integral final de lo planteado en cada una de las etapas (lecciones), como evidencia concreta del trabajo realizado:

## Entregable Final (Proyecto Integrador):

#### 1. Script o Notebook Python (.py o .ipynb)

- Código funcional y modularizado que integre todo el flujo de trabajo, desde la generación de datos con NumPy hasta la exportación final del dataset procesado.
- Puede estar dividido por secciones/lecciones o en bloques bien documentados dentro de un único archivo.

#### 2. Dataset final estructurado

- Archivos exportados en formato **CSV** y **Excel** que contengan el dataset limpio, transformado y preparado para análisis o uso en modelos.
- Debe reflejar los pasos de imputación, detección de outliers, wrangling, agrupamientos y combinaciones realizados.

#### 3. Documento resumen del flujo de trabajo (PDF o Markdown)

- Un documento claro y conciso que explique el proceso completo seguido en el proyecto:
  - o Justificación del uso de NumPy y Pandas.
  - Descripción del dataset generado y de las fuentes externas integradas.
  - o Técnicas aplicadas para la limpieza y transformación.
  - o Principales decisiones tomadas y desafíos encontrados.
  - o Resultados obtenidos y estado final del dataset.

## Portafolio

El proyecto "**Preparación de datos**" podrá ser incorporado en tu portafolio profesional como un caso práctico de manipulación y preparación de datos reales. Te recomendamos presentar el proyecto destacando:

- Las técnicas aplicadas y librerías utilizadas.
- Las principales problemáticas encontradas y cómo fueron resueltas.
- Capturas del código, el dataset limpio y las visualizaciones de los resultados.

Esto demostrará tu capacidad para abordar proyectos de preparación de datos, una de las habilidades más demandadas en la industria de datos y tecnología.



# ¡Éxitos!

Nos vemos más adelante

