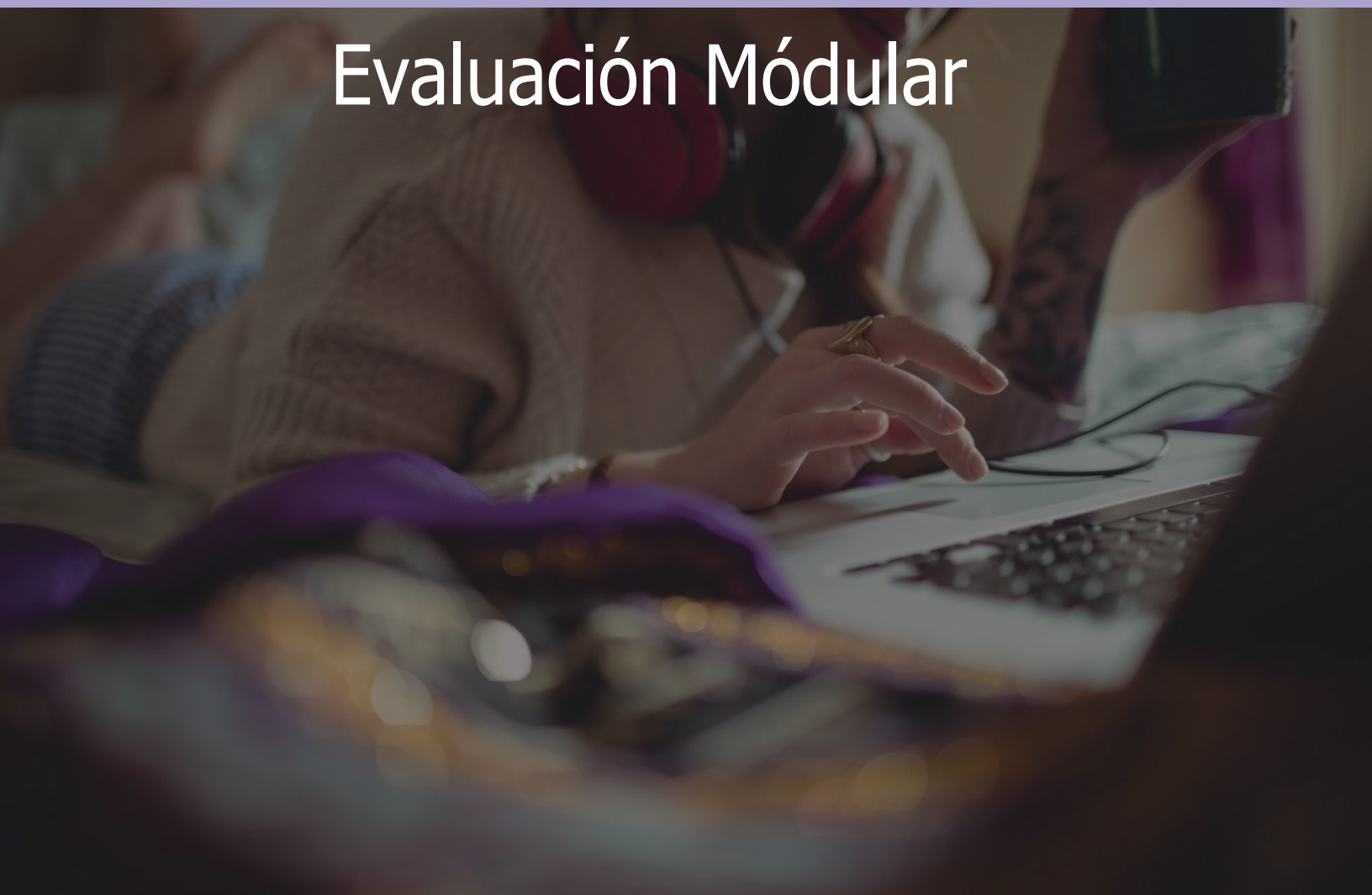
IT Academy
by KIBERNUM

Módulo 3

Evaluación Módular



ACTIVIDAD:

Del Archivo Crudo al Reporte Analítico: Pipeline de datos Realista

Objetivo:
Manipular y analizar datos en Python utilizando NumPy y Pandas, aplicando técnicas de exploración, limpieza y tratamiento de valores perdidos y atípicos, realizando procesos de *data wrangling* avanzado, operaciones de agrupamiento y pivoteo, y generando productos exportables y visualizaciones útiles para la toma de decisiones empresariales.

Contexto:
Imagina que trabajas como analista de datos para una empresa que recibe semanalmente información de clientes y ventas en archivos planos, muchas veces desordenados, incompletos o con errores. El equipo necesita construir un pipeline en Python que automatice la limpieza, integración y generación de reportes confiables, listos para la toma de decisiones del negocio. Para simular este escenario real, primero generarás los datos crudos de manera aleatoria, y luego diseñarás todo el proceso de data wrangling y análisis.

1. **Generación automática de datos**:
    - Crea dos archivos sintéticos:
        - clientes.csv con información de clientes.
        - ventas.xlsx con información de ventas.
    - Asegúrate de incluir **errores intencionales**, valores faltantes, datos atípicos y formatos inconsistentes.
2. **Construcción del pipeline**:
    - Desde la carga hasta la exportación, tu objetivo es **simular el trabajo profesional de un ingeniero/a de datos**:
        - Explora, limpia, transforma, integra y resume la información.
        - Documenta cada paso con comentarios y justifica las decisiones relevantes.
        - Entrega el código y un breve informe justificando tu proceso.

Instrucciones:

1. Generación de archivos de datos sintéticos

a) Archivo clientes.csv

Debe contener al menos 100 registros y las siguientes columnas:

- id_cliente (único, numérico)
- nombre (ficticio)
- edad (entre 18 y 70 años, con algunos nulos)
- ciudad (de una lista, agregando errores de escritura y algunos nulos)
- ingreso (valores numéricos, incluye outliers y nulos)

b) Archivo ventas.xlsx

Debe tener al menos 500 registros:

- id_venta (único)
- id_cliente (relacionado con clientes, algunos nulos o inexistentes)
- fecha (dos años de rango, incluye fechas en varios formatos y algunos nulos)
- producto (aleatorio de una lista)
- monto (valores numéricos, incluye atípicos, ceros y nulos)

```python
import pandas as pd
import numpy as np
from faker import Faker

# Inicialización de Faker
fake = Faker('es_ES')

# CLIENTES
n_clientes = 100
clientes = pd.DataFrame({
    'id_cliente': range(1, n_clientes + 1),
    'nombre': [fake.name() for _ in range(n_clientes)],
    'edad': np.random.choice(list(range(18, 70)) + [np.nan], n_clientes, p=[.01]*52 + [.48]),
    'ciudad': np.random.choice(
        ['Santiago', 'Concepción', 'Valparaíso', 'Antofagasta', 'Sntiago', 'Vina del Mr', np.nan],
n_clientes),
    'ingreso': np.round(np.random.normal(1800, 900, n_clientes), 0)
})

# Insertar valores extremos y nulos en ingreso
clientes.loc[np.random.choice(clientes.index, 4), 'ingreso'] = [10000, 0, 30000, np.nan]

clientes.to_csv('clientes.csv', index=False)

# VENTAS
n_ventas = 500
ventas = pd.DataFrame({
    'id_venta': range(1, n_ventas + 1),
    'id_cliente': np.random.choice(clientes['id_cliente'].tolist() + [np.nan], n_ventas),
    'fecha': pd.to_datetime(np.random.choice(
        pd.date_range('2023-01-01', '2024-12-31', freq='D'), n_ventas)),
    'producto': np.random.choice(['A', 'B', 'C', 'D', 'X'], n_ventas),
    'monto': np.round(np.random.uniform(500, 8000, n_ventas), 0)
})

# Añadir errores de formato y nulos en fecha y monto
ventas.loc[np.random.choice(ventas.index, 7), 'fecha'] = [
    '2023/05/20', '31-12-2024', None, '2024-01-02', 'julio 2023', np.nan, '2024/03/15']
ventas.loc[np.random.choice(ventas.index, 5), 'monto'] = [np.nan, 50000, 0, None, -100]

ventas.to_excel('ventas.xlsx', index=False)
```

2. Extracción y exploración de los datos
   - Lee ambos archivos usando Pandas (read_csv y read_excel).
   - Muestra las primeras filas y la información general de los DataFrames (columnas, tipos de datos, valores únicos).
   - Busca duplicados y analiza la estructura de los datos.

3. Limpieza inicial y transformación de tipos
   - Normaliza nombres de columnas si es necesario.
   - Convierte tipos de datos (por ejemplo, fechas, ingresos).
   - Corrige errores de escritura en ciudades usando replace.
   - Elimina o marca duplicados y filas irreparables.

4. Detección y tratamiento de valores perdidos
   - Identifica y reporta los valores nulos en cada columna.
   - Decide, justifica y ejecuta: eliminación o imputación (media, mediana, moda, categoría especial) según el caso.
   - Explica tus criterios en el informe.

5. Detección y tratamiento de outliers
   - Identifica outliers en variables numéricas (edad, ingreso, monto) usando IQR o z-score.
   - Decide cómo tratar los outliers (eliminación, winsorización, transformación, mantenerlos si tienen sentido de negocio).
   - Documenta las decisiones.

6. Data wrangling avanzado
   - Elimina duplicados.
   - Corrige formatos (por ejemplo, unifica fechas).
   - Crea nuevas columnas útiles:
     - grupo_edad usando bins (por ejemplo: <30, 30-45, 46-60, >60)
     - nivel_ingreso usando percentiles (ej: bajo, medio, alto)
   - Cambia tipos de dato si corresponde.
   - Enriquecer la tabla con nuevas variables (por ejemplo, ciudad limpia, día de la semana de la venta, etc.).

7. Integración y agrupamiento
   - Une ambas fuentes usando id_cliente.
   - Agrupa y resume por variables clave:
     - Por ciudad, grupo de edad y nivel de ingreso.
     - Calcula estadísticas de ventas (suma, media, conteo).
   - Usa groupby, pivot, pivot_table o melt según la necesidad.

8. Exportación y visualización
   - Exporta el DataFrame resumen a CSV y/o Excel.
   - Realiza al menos una visualización relevante (por ejemplo, ventas promedio por ciudad o grupo de edad), usando Pandas o Matplotlib.
   - Guarda la figura si es posible.

9. Informe final
   - Explica en máximo 1 página (o como markdown dentro del notebook):
     - Las decisiones de limpieza y transformación,
     - La estrategia para nulos y outliers,
     - Limitaciones del pipeline y posibles mejoras futuras.

RÚBRICA

Indicador de logro / criterio	Insuficiente (0%-20%)	Por lograrlo (21%-40%)	Medianamente logrado (41%-60%)	Logrado (61%-80%)	Sobresaliente (81%-100%)
1. Generación de archivos de datos sintéticos	No genera los archivos o contiene menos del 25% de los requisitos.	Genera los archivos con estructuras mínimas, sin variedad o errores intencionales.	Cubre parcialmente los requisitos (entre 50%-70%), con errores controlados poco representativos.	Genera ambos archivos con la mayoría de los requisitos cumplidos, incluyendo variedad y errores intencionales.	Genera los dos archivos cumpliendo el 100% de los requisitos, con datos variados, realistas y errores representativos.
2. Extracción y exploración de los datos	No se extraen los archivos o se omite la exploración.	Lectura incompleta o limitada a una sola fuente, con mínima exploración.	Lee ambos archivos, pero muestra solo información básica o incompleta.	Lee ambos archivos correctamente y realiza exploración adecuada.	Lee ambos archivos, realiza exploración completa con comentarios y análisis preliminar.
3. Limpieza inicial y transformación de tipos	No transforma tipos ni corrige columnas o errores.	Realiza algunos cambios de tipo o corrige parcialmente columnas.	Aplica transformaciones básicas y algunas correcciones en ciudades.	Aplica transformaciones y correcciones adecuadas y justificadas.	Aplica todas las transformaciones necesarias, justifica y documenta claramente los cambios.
4. Detección y tratamiento de valores perdidos	No detecta ni trata valores nulos.	Detecta algunos nulos, pero no los trata adecuadamente.	Identifica nulos en la mayoría de las columnas y aplica un tratamiento parcial.	Identifica y trata nulos con lógica adecuada, explicando criterios.	Aplica tratamiento apropiado y justificado a todos los nulos, con documentación clara.
5. Detección y tratamiento de outliers	No identifica ni trata outliers.	Detecta pocos outliers sin tratarlos correctamente.	Identifica correctamente los outliers, pero con tratamiento limitado.	Aplica detección (IQR/z-score) y tratamiento adecuado, con explicación.	Identifica, trata y documenta completamente los outliers con análisis crítico.
6. Data wrangling avanzado	No realiza operaciones de wrangling ni genera nuevas columnas.	Elimina duplicados, pero no genera nuevas variables ni corrige formatos.	Realiza parte del wrangling y crea una o dos nuevas columnas útiles.	Realiza wrangling completo y crea nuevas columnas significativas.	Realiza wrangling completo, transforma tipos, corrige formatos y enriquece significativamente el dataset.
7. Integración y agrupamiento	No une las fuentes ni agrupa los datos.	Une parcialmente las fuentes, sin aplicar agrupamientos.	Realiza unión correcta, pero con agrupamientos limitados.	Une, agrupa y resume adecuadamente por variables clave.	Integra, agrupa, resume con profundidad analítica y estructura clara.
8. Exportación y visualización	No exporta resultados ni realiza visualización.	Exporta sin estructura clara o visualiza con errores.	Exporta correctamente y genera al menos una visualización simple.	Exporta datos limpios y genera visualización relevante con Pandas o Matplotlib.	Exporta correctamente y genera visualización clara, útil y documentada. Guarda la figura.
9. Informe final	No entrega informe o es irrelevante.	Informe incompleto, sin justificar decisiones.	Informe presenta decisiones básicas sin profundidad.	Informe completo, con justificación clara de decisiones y limitaciones.	Informe claro, conciso y reflexivo. Justifica decisiones, discute limitaciones y sugiere mejoras.