

Es hora de que pongas en práctica todo lo aprendido. 🧐

Este apartado tiene el objetivo de ayudarte a seguir potenciando tus habilidades, por lo que a continuación encontrarás diferentes **desafíos** que podrás resolver de forma independiente y a tu ritmo.

Más adelante conseguirás las resoluciones para que valides tus respuestas y puedas monitorear tu progreso. 😊

### ¡Manos a la obra!

#### 1. Desafío 🎯

*Este desafío consiste en la manipulación, análisis y transformación de datos utilizando **Pandas**. Trabajarás con un conjunto de datos estructurado y aplicarás diversas operaciones para limpiarlo, organizarlo y extraer información clave.*

- **Carga y exploración de datos:**
  - Importa el conjunto de datos desde un archivo CSV([LigaEspanola2023-2024-Resultados.csv](#)).
  - Muestra las primeras y últimas 5 filas.
  - Obtén información general del DataFrame, incluyendo tipos de datos y valores nulos.
  - Calcula estadísticas descriptivas generales.
- **Manejo de valores nulos y duplicados:**
  - Identifica y cuenta los valores nulos en el conjunto de datos.
  - Imputa los valores faltantes en columnas numéricas con la **media** y en columnas categóricas con la **moda**.
  - Elimina las filas duplicadas, si existen.
- **Filtrado y selección de datos:**
  - Filtra los datos según una condición específica (ejemplo: valores mayores a un umbral en una columna numérica).
  - Selecciona solo columnas relevantes para el análisis.
  - Ordena los datos en función de una o más columnas.
- **Agrupamiento y operaciones estadísticas:**
  - Agrupa los datos en función de una variable categórica y calcula estadísticas agregadas.
  - Obtén el promedio de una variable por cada grupo.

- Aplica funciones de agregación personalizadas.
- **Creación de nuevas columnas y transformación de datos:**
  - Crea una nueva columna a partir de cálculos basados en otras columnas.
  - Convierte el tipo de datos de una columna si es necesario.
  - Renombra columnas para mejorar la legibilidad del DataFrame.

## 2. ¿Dónde se lleva a cabo? 🏠

Visual studio Code

## 3. Tiempo de dedicación ⌚

3 horas

## 4. Recursos 🛠️

Documentación de Pandas: <https://pandas.pydata.org/docs/>

## 5. Plus ➕

- Experimenta exportando los datos a otros formatos como **Excel** o **JSON**.
- Intenta automatizar algunos pasos con funciones personalizadas.

## 6. ⚠️ Condición

Esta práctica o ejercitación **no requiere ser entregada y/o evaluada** por el mentor. No obstante puedes compartir tus resultados con el resto de los bootcampers y construir conocimiento en conjunto.

# Resolución del desafío

## 1. Carga y exploración de datos en Pandas

```
import pandas as pd

# Cargar el dataset desde un archivo CSV
df = pd.read_csv('dataset.csv')

# Mostrar las primeras y últimas filas
display(df.head())
display(df.tail())

# Obtener información general
df.info()

# Calcular estadísticas descriptivas
df.describe()
```

## 2. Manejo de valores nulos y duplicados

```
# Identificar valores nulos
df.isnull().sum()

# Imputar valores faltantes
df.fillna({'columna_numerica': df['columna_numerica'].mean(),
          'columna_categorica': df['columna_categorica'].mode()[0]}, inplace=True)

# Eliminar filas duplicadas
df.drop_duplicates(inplace=True)
```

## 3. Filtrado y selección de datos

```
# Filtrar valores mayores a un umbral
df_filtrado = df[df['columna_numerica'] > 50]

# Seleccionar columnas relevantes
df_seleccionado = df[['columna1', 'columna2', 'columna3']]

# Ordenar por una columna
df_ordenado = df.sort_values(by='columna1', ascending=False)
```

#### 4. Agrupamiento y estadísticas

```
# Agrupar por una variable categórica y calcular promedio
df_grouped = df.groupby('categoria')['valor'].mean()
print(df_grouped)

# Aplicar funciones de agregación
df_agg = df.groupby('categoria').agg({'valor': ['mean', 'max', 'min']})
print(df_agg)
```

#### 5. Creación de nuevas columnas y transformación de datos

```
# Crear una nueva columna basada en cálculos
df['nueva_columna'] = df['columna1'] * df['columna2']

# Convertir tipo de datos
df['columna_fecha'] = pd.to_datetime(df['columna_fecha'])

# Renombrar columnas
df.rename(columns={'columna_antigua': 'columna_nueva'}, inplace=True)
```