

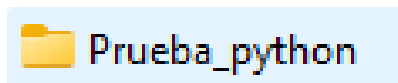
# Documentación Script de Procesamiento de Datos en Python




Este proyecto consiste en un script en Python diseñado para procesar un archivo CSV y generar estadísticas básicas junto con un filtrado específico de datos.

El script realiza las siguientes operaciones:

- Lectura de un archivo CSV.
- Conteo total de registros.
- Cálculo del promedio de columnas numéricas.
- Filtrado de registros donde la columna **Salary** sea mayor a **6500**.
- Generación de un archivo de salida en formato JSON con los resultados obtenidos.

## Estructura de proyecto:



-  resultado.json
-  Script de Procesamiento de Datos.csv
-  script.py

## Requisitos del Sistema:

- Python 3.x
- Librerías:
  - pandas
  - json (incluida en Python)
  - os (incluida en Python)

## Instalación de dependencias:

*pip install pandas*

## Funcionamiento del Script:

El flujo de ejecución es el siguiente:

1. Obtiene dinámicamente la ruta del archivo CSV.
2. Carga los datos en un DataFrame usando pandas.
3. Calcula estadísticas generales.
4. Aplica un filtro sobre la columna Salary con umbral 6500.
5. Exporta los resultados en un archivo JSON.

## Descripción de Funciones:

### ***cargar\_datos(ruta\_archivo)***

Lee el archivo CSV y lo convierte en un DataFrame.

Parámetros:

- ruta\_archivo (str): Ruta del archivo CSV.

### ***calcular\_estadisticas(df)***

Calcula estadísticas básicas del DataFrame.

#### **Operaciones realizadas:**

- Conteo total de registros.
- Promedio de todas las columnas numéricas.

### ***filtrar\_datos(df, columna, umbral)***

Filtra registros donde el valor de una columna numérica sea mayor a un umbral definido.

#### **Parámetros:**

- df: DataFrame
- columna: Nombre de la columna a evaluar (Salary)
- umbral: Valor mínimo permitido (6500)

#### **Validaciones:**

- Verifica que la columna exista.
- Verifica que la columna sea numérica.

### ***exportar\_json(resultado, ruta\_salida)***

Genera un archivo JSON con los resultados procesados.

#### **Parámetros:**

- resultado: Diccionario con estadísticas y datos filtrados.
- ruta\_salida: Ruta donde se guardará el archivo JSON

### **Formato del Archivo de Salida:**

El archivo resultado.json contiene:

```
{
  "estadisticas": {
    "total_registros": 100,
    "promedios_columnas_numericas": {
      "Age": 35.4,
      "Salary": 7200
    }
  },
  "filtro_aplicado": {
    "columna": "Salary",
    "umbral": 6500
  },
  "datos_filtrados": [
    {
      "Name": "John",
      "Salary": 7000
    }
  ]
}
```

### **Manejo de Errores**

El script incluye:

- Validación de existencia del archivo.

- Validación de existencia de columna.
- Validación de tipo numérico.
- Manejo de excepciones al exportar JSON.

### Resultado Ejercicio:

```
PS E:\Documentos\Proyectos\Prueba_python> python script.py
Archivo cargado correctamente.
Se encontraron 4 registros con 'Salary' > 65000.
Archivo JSON generado correctamente.
Proceso completado exitosamente.
PS E:\Documentos\Proyectos\Prueba_python>
```

### Archivo Json:

```
{
  "estadisticas": {
    "total_registros": 10,
    "promedios_columnas_numericas": {
      "ID": 5.5,
      "Age": 34.1,
      "Salary": 62000.0
    }
  },
  "filtro_aplicado": {
    "columna": "Salary",
    "umbral": 65000
  },
  "datos_filtrados": [
    {
      "ID": 3,
      "Name": "Bob Johnson",
      "Age": 45,
      "Salary": 75000,
      "Department": "Engineering"
    },
    {
      "ID": 5,
      "Name": "Tom Clark",
      "Age": 37,
      "Salary": 67000,
      "Department": "Marketing"
    }
  ]
}
```

```
    {
      "ID": 6,
      "Name": "Mary Green",
      "Age": 41,
      "Salary": 73000,
      "Department": "HR"
    },
    {
      "ID": 8,
      "Name": "Nancy Wilson",
      "Age": 36,
      "Salary": 80000,
      "Department": "Marketing"
    }
  ]
}
```

