



Universidad  
Tecnológica  
del Perú

**FACULTAD DE  
INGENIERÍA**

**Domina  
Python y  
Conquista el  
Mundo de los  
Datos.**



# PYTHON Y EL MUNDO DE LOS DATOS



## SESIÓN 05

### Integración en un caso práctico de análisis de datos.

#### Logros de Aprendizaje

#### General

Desarrollar competencias en programación con Python, desde fundamentos básicos hasta el manejo avanzado de datos con NumPy y Pandas. Los participantes aprenderán a implementar estructuras, funciones, conceptos de POO (herencia y encapsulamiento) y librerías de análisis, integrando estos conocimientos en un caso práctico para resolver problemas reales y presentar resultados de manera efectiva.

#### Sesión

Al finalizar la sesión, el estudiante elabora soluciones de análisis de datos con las bibliotecas NumPy y Pandas mediante el desarrollo de un caso práctico.

# PYTHON Y EL MUNDO DE LOS DATOS



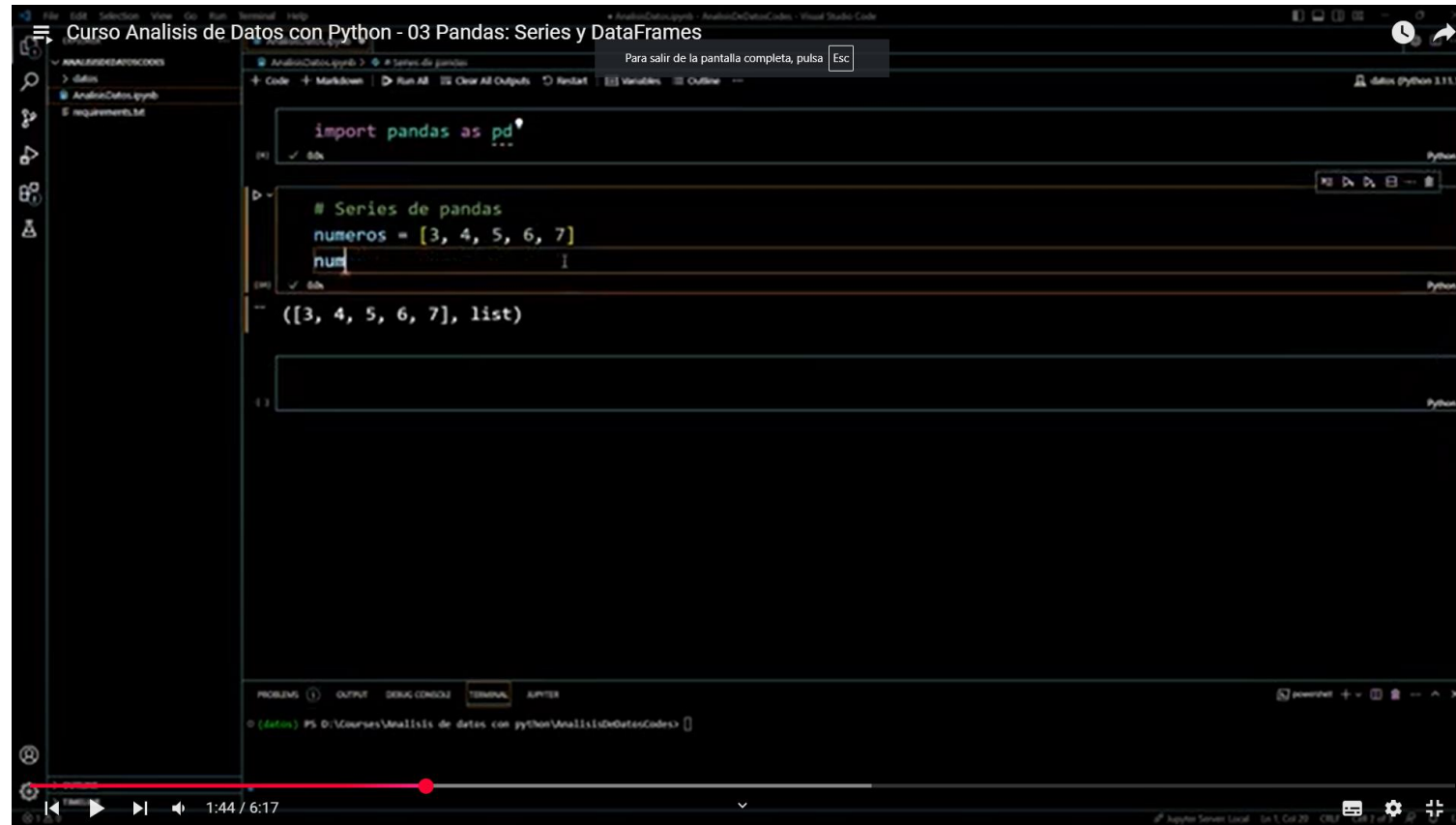
## SESIÓN 05

### Integración de los conceptos en un caso práctico.

- Uso conjunto de Python, NumPy y Pandas para resolver problemas reales.
- Limpieza de datos y análisis básicos: operaciones como media, máximo y mínimo.

# INICIO - Conocimientos Previos

## Caso práctico de análisis de datos en Python



The screenshot shows a video player interface with a dark theme. The main content is a Visual Studio Code window titled 'Curso Analisis de Datos con Python - 03 Pandas: Series y DataFrames'. The code editor displays the following Python code:

```
import pandas as pd

# Series de pandas
numeros = [3, 4, 5, 6, 7]
num = ...
```

Below the code editor, the output console shows the result of the code execution: `(([3, 4, 5, 6, 7], list))`. The video player controls at the bottom indicate the video is at 1:44 / 6:17.



<https://www.youtube.com/watch?v=0eTCNICKYnY>

¿Cuál es la utilidad de integrar los conceptos de Python en un caso práctico de análisis de datos?

- Integrar conceptos de **Python** en un caso práctico de análisis de datos permite automatizar la recolección, limpieza, análisis e interpretación de grandes volúmenes de datos.
- Combinar estructuras como listas, diccionarios, bucles y condicionales, con bibliotecas como **NumPy** y **Pandas**, facilita la manipulación de datos.
- Otras herramientas, como **Matplotlib**, permiten la visualización de datos para identificar patrones y tendencias. Además, el uso de funciones y módulos ayuda a modularizar el código, haciéndolo más organizado y reutilizable.

## Objetivos de la Clase

- Comprender la integración de conceptos clave de Python en un caso práctico.
- Aplicar bibliotecas como **Pandas**, **NumPy** y **Matplotlib**.
- Realizar un análisis de datos estructurado y visualización de información.
- Interpretar resultados para la toma de decisiones.





# TRANSFORMACIÓN

Herramientas y conceptos clave:

- **Estructuras de Datos** (Listas, Diccionarios, DataFrames)
- **Cálculos Numéricos** con **NumPy**.
- **Manejo de Datos** con **Pandas**.
- **Visualización de Datos** con **Matplotlib**.
- **Análisis y Conclusiones**.



## Planteamiento del Caso

**Escenario:** El Fondo Mi Vivienda desea saber cómo se ha comportado el otorgamiento de créditos para adquisición de viviendas en el Perú entre los años 2020 y 2024.



## Datos a analizar:

- Créditos Mi Vivienda otorgados entre los años 2020 a 2024 a nivel nacional, contenidos en el archivo [mivivienda.csv](#)

**Objetivo:** Obtener estadísticas clave e identificar las ciudades de mayor desarrollo del fondo.

Datos obtenidos de la Plataforma Nacional de Datos Abiertos (<https://datosabiertos.gob.pe/>).



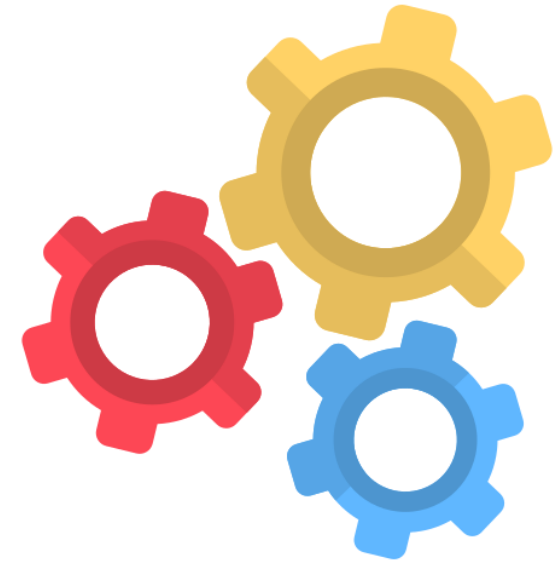
# TRANSFORMACIÓN

## Dataset

Variable	Descripción	Tipo de dato
FECHA_DESEMBOLSO	Es la fecha en la que se ejecuta el desembolso del crédito	Numérico
PRODUCTO	Nuevo Crédito MIVIVIENDA (NCMV), Mi Terreno (MT), Financiamiento Complementario Techo Propio (FCTP) y Servicio CRC (S-CRC) 1/	Numérico
DEPARTAMENTO	Nombre del departamento	Texto
PROVINCIA	Nombre de la provincia	Texto
DISTRITO	Nombre del distrito	Texto
UBIGEO	Codificación de la ubicación geográfica donde se encuentra ubicada la vivienda. Está conformada por 6 dígitos, siendo su estructura, recorriendo el valor de izquierda a derecha, la siguiente: - 1er y 2do dígito identifican el departamento (02) - 3er y 4to dígito identifican la provincia (02) - 5to y 6to dígito identifican el distrito (02)	Alfanumérico
IFI	Institución Financiera Intermediaria, que opera bajo el ámbito de supervisión de la SBS y es considerada como apta por el FMV para el otorgamiento de los créditos MIVIVIENDA.	Texto
TIPO_IFI	Las IFI se dividen en siete tipos: Banca Múltiple (Banco), Financiera, Cooperativa de Ahorro y Crédito (CAC), Caja Municipal de Ahorro y Crédito (CMAC), Caja Rural de Ahorro y Crédito (CRAC), Empresa de Créditos (Edpyme) y Empresas Administradoras Hipotecarias (Hipotecaria)	Texto
MONTO_CREDITO	Monto del crédito colocado	Numérico
MONTO_CUOTA_INICIAL	Monto de la cuota inicial	Numérico
PLAZOS	Es el número de meses del crédito	Numérico
TASA	Tasa de interés del crédito	Numérico
MONTO_VALOR_VIVIENDA	Monto del valor de la vivienda.	Numérico
FECHA_CORTE	Fecha de corte de información.	Numérico

## Proceso:

1. Carga de datos de archivo *mivivienda.csv*
2. Limpieza de los datos.
3. Análisis de los datos.
  - a. Monto de crédito promedio por producto.
  - b. Distribución de los créditos por departamento.
  - c. Número de créditos otorgados por cada año.
  - d. Cantidad de dinero otorgado como crédito por departamento.
4. Presentación de resultados.



# TRANSFORMACIÓN

## 1. Carga de datos de archivo *mivivienda.csv*

```
FECHA_DESEMBOLSO,PRODUCTO,DEPARTAMENTO,PROVINCIA,DISTRITO,UBIGEO,IFI,TIPO_IFI,MONTO_CREDITO,MONTO_CUOTA_INICIAL,PLAZOS,TASA,MONTO_VALOR_VIVIENDA,FECHA_CORTE
20200107,NCMV,LIMA,LIMA,CHORRILLOS,150108,CREDITO,BANCO,300357.96,34374.30,228,7.10,343743.0,20230811
20200107,NCMV,LAMBAYEQUE,LAMBAYEQUE,LAMBAYEQUE,140301,CREDITO,BANCO,79800.00,24398.00,180,8.20,121990.0,20230811
20200107,NCMV,LIMA,LIMA,RIMAC,150128,CREDITO,BANCO,62798.08,19990.00,120,8.30,99900.0,20230811
20200107,NCMV,LIMA,LIMA,ATE,150103,CREDITO,BANCO,162062.14,19992.00,240,7.50,199916.0,20230811
20200107,NCMV,LAMBAYEQUE,CHICLAYO,CHICLAYO,140101,CREDITO,BANCO,97500.00,19500.00,240,10.45,130000.0,20230811
20200107,NCMV,LAMBAYEQUE,CHICLAYO,JOSE LEONARDO ORTIZ,140105,CREDITO,BANCO,99384.62,121990.0,20230811
20200107,NCMV,CAJAMARCA,JAEN,JAEN,060801,CREDITO,BANCO,100000.00,42000.00,180,9.70,150000.0,20230811
20200107,NCMV,LAMBAYEQUE,LAMBAYEQUE,LAMBAYEQUE,140301,CREDITO,BANCO,133536.54,26802.00,240,7.50,268020.0,20230811
20200107,NCMV,LAMBAYEQUE,CHICLAYO,JOSE LEONARDO ORTIZ,140105,CREDITO,BANCO,138213.59,26802.00,240,7.50,268020.0,20230811
20200107,NCMV,LIMA,LIMA,RIMAC,150128,CREDITO,BANCO,54461.54,34000.00,120,10.90,105240.0,20230811
20200107,NCMV,LA LIBERTAD,TRUJILLO,TRUJILLO,130101,CREDITO,BANCO,123350.00,15150.00,180,8.20,151500.0,20230811
20200107,NCMV,AREQUIPA,AREQUIPA,CERRO COLORADO,040104,CREDITO,BANCO,173570.00,20730.00,240,7.50,207300.0,20230811
20200107,NCMV,LIMA,LIMA,COMAS,150110,CONTINENTAL,BANCO,50375.00,54000.00,60,8.50,120900.0,20230811
20200107,NCMV,LIMA,LIMA,COMAS,150110,CONTINENTAL,BANCO,101038.46,13120.00,240,8.00,131200.0,20230811
20200107,NCMV,LIMA,LIMA,SURQUILLO,150141,CREDITO,BANCO,307155.34,55830.00,240,6.90,372985.34,20230811
20200107,NCMV,AREQUIPA,AREQUIPA,CERRO COLORADO,040104,CREDITO,BANCO,172349.51,45980.00,240,7.50,172349.51,20230811
20200107,NCMV,LIMA,LIMA,LINCE,150116,CREDITO,BANCO,243951.46,28630.00,240,7.50,286300.0,20230811
20200107,NCMV,LIMA,LIMA,JESUS MARIA,150113,CREDITO,BANCO,226796.12,60000.00,240,8.52,286796.12,20230811
20200107,NCMV,LIMA,LIMA,RIMAC,150128,CREDITO,BANCO,190912.62,50760.00,120,9.00,253800.0,20230811
20200107,NCMV,LIMA,LIMA,SAN MIGUEL,150136,CREDITO,BANCO,332980.16,60524.04,204,7.02,403504.20,20230811
20200107,NCMV,LIMA,LIMA,JESUS MARIA,150113,CREDITO,BANCO,310718.45,35560.00,240,7.10,355600.0,20230811
```

```
import pandas as pd
```

```
ruta_archivo = "<ruta>\\mivivienda.csv"
df = pd.read_csv(ruta_archivo)
print(df.head())
```

	FECHA_DESEMBOLSO	PRODUCTO	DEPARTAMENTO	PROVINCIA	DISTRITO	UBIGEO	IFI	TIPO_IFI	MONTO_CREDITO	MONTO_CUOTA_INICIAL	PLAZOS	TASA	MONTO_VALOR_VIVIENDA	FECHA_CORTE
0	20200107	NCMV	LIMA	LIMA	CHORRILLOS	150108	CREDITO	BANCO	300357.96	34374.3	228.0	7.10	343743.0	20230811
1	20200107	NCMV	LAMBAYEQUE	LAMBAYEQUE	LAMBAYEQUE	140301	CREDITO	BANCO	79800.00	24398.0	180.0	8.20	121990.0	20230811
2	20200107	NCMV	LIMA	LIMA	RIMAC	150128	CREDITO	BANCO	62798.08	19990.0	120.0	8.30	99900.0	20230811
3	20200107	NCMV	LIMA	LIMA	ATE	150103	CREDITO	BANCO	162062.14	19992.0	240.0	7.50	199916.0	20230811
4	20200107	NCMV	LAMBAYEQUE	CHICLAYO	CHICLAYO	140101	CREDITO	BANCO	97500.00	19500.0	240.0	10.45	130000.0	20230811

## 2. Limpieza de los datos

```
# registros con alguna columna nula
nro_filas_invalidas = df.isna().all(axis=1).sum()
print("Filas no válidas: ", nro_filas_invalidas)
if nro_filas_invalidas > 0:
    df = df.dropna()
    print("Filas actuales: ", df.shape[0])

# registros duplicados
nro_filas_duplicadas = df.duplicated().sum()
print("Filas duplicadas: ", nro_filas_duplicadas)
if nro_filas_duplicadas > 0:
    df = df.drop_duplicates()
    print("Filas actuales: ", df.shape[0])

# removemos la columna FECHA_CORTE
df = df.drop("FECHA_CORTE", axis = 1)
print("Filas válidas finales: ", df.shape[0])
```

```
Filas no válidas:  0
Filas duplicadas:  435
Filas actuales:   55133
Filas válidas finales:  55133
```

# TRANSFORMACIÓN

## 3. Análisis de los datos » a. Monto de crédito promedio por producto.

i. Agrupamos los registros por producto

ii. Calculamos el promedio (*mean*) de los montos de crédito otorgados

```
df_PromCredito_x_Producto = df.groupby(["PRODUCTO"])["MONTO_CREDITO"].mean()  
print(df_PromCredito_x_Producto.reset_index(name = "PROMEDIO_CREDITO"))
```

	PRODUCTO	PROMEDIO_CREDITO
0	FCTP	34974.239788
1	NCMV	164067.375289
2	NMIV	188610.460900
3	S-CRC	343921.417201

iii. Genera un nuevo DataFrame con el index reiniciado

# TRANSFORMACIÓN

## 3. Análisis de los datos » b. Distribución de los créditos por departamento.

```
df_creditos_departamento = df.groupby(["DEPARTAMENTO"]).size()  
print(df_creditos_departamento.reset_index(name = 'NUMERO_CREDITOS'))
```

	DEPARTAMENTO	NUMERO_CREDITOS
0	AMAZONAS	67
1	ANCASH	266
2	APURIMAC	15
3	AREQUIPA	1451
4	AYACUCHO	99
5	CAJAMARCA	467
6	CALLAO	2495
7	CUSCO	203
8	HUANCAVELICA	30
9	HUANUCO	54
10	ICA	3677
11	JUNIN	1261
12	LA LIBERTAD	2979

	DEPARTAMENTO	NUMERO_CREDITOS
13	LAMBAYEQUE	4292
14	LIMA	32047
15	LORETO	80
16	MADRE DE DIOS	4
17	MOQUEGUA	156
18	PASCO	53
19	PIURA	4067
20	PUNO	314
21	SAN MARTIN	430
22	TACNA	485
23	TUMBES	12
24	UCAYALI	129



# TRANSFORMACIÓN

## 3. Análisis de los datos » c. Número de créditos otorgados por cada año.

Creando una nueva columna para almacenar el año de desembolso del crédito

```
df["AÑO_DESEMBOLSO"] = pd.to_datetime(arg = df["FECHA_DESEMBOLSO"], \
                                         format = "%Y%m%d").dt.year
df_creditos_x_año = df.groupby(["AÑO_DESEMBOLSO"]).size()
print(df_creditos_x_año.reset_index(name = "NUMERO_CREDITOS"))
```

	AÑO_DESEMBOLSO	NUMERO_CREDITOS
0	2020	8285
1	2021	12710
2	2022	13378
3	2023	11424
4	2024	9336

# TRANSFORMACIÓN

## 3. Análisis de los datos » d. Cantidad de dinero otorgado como crédito por departamento.

Establecemos el formato de los valores con decimales

```
pd.set_option('display.float_format', '{:.2f}'.format)
df_TotalCredito_x_Depto = df.groupby(["DEPARTAMENTO"])[ "MONTO_CREDITO"].sum()
df_TotalCredito_x_Depto = df_TotalCredito_x_Depto.reset_index(name = "TOTAL_CREDITO")
df_TotalCredito_x_Depto = df_TotalCredito_x_Depto.sort_values(by = "TOTAL_CREDITO", \
                                                                ascending = False)

print(df_TotalCredito_x_Depto)
```

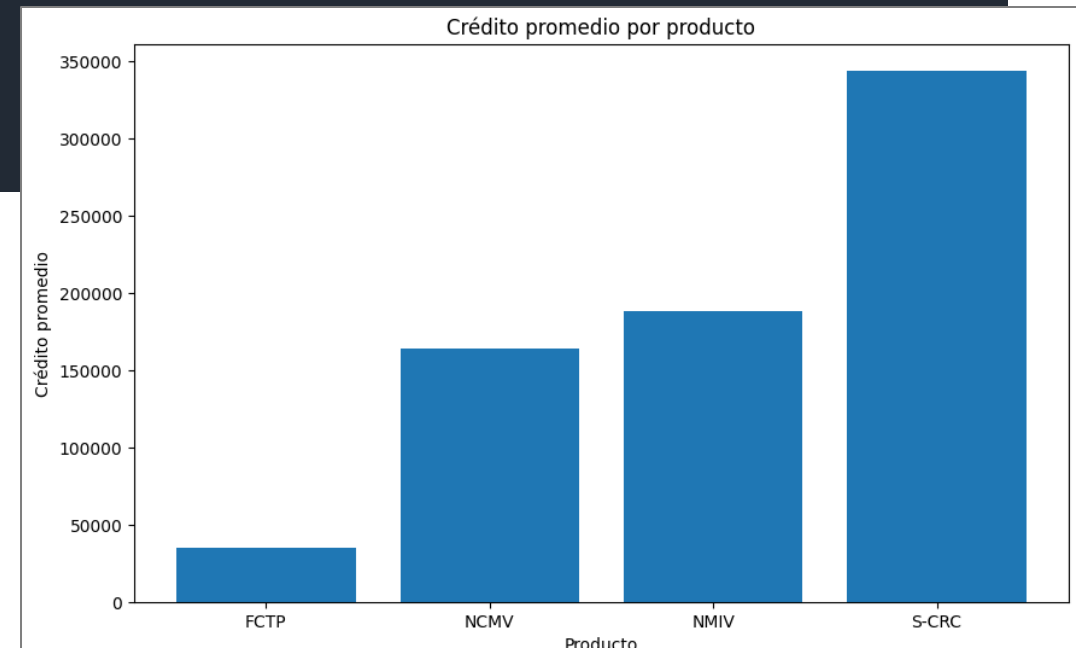
	DEPARTAMENTO	TOTAL_CREDITO
14	LIMA	6408217000.83
13	LAMBAYEQUE	490105591.05
6	CALLAO	468893158.65
19	PIURA	378416641.97
12	LA LIBERTAD	320357226.00
10	ICA	275511693.46
3	AREQUIPA	259639140.88
11	JUNIN	131201780.32

	DEPARTAMENTO	TOTAL_CREDITO
5	CAJAMARCA	43645677.19
22	TACNA	41585926.74
7	CUSCO	36503423.02
20	PUNO	32997778.70
21	SAN MARTIN	28365972.98
1	ANCASH	22643802.34
17	MOQUEGUA	15482009.40
4	AYACUCHO	10179602.59
15	LORETO	7663921.14

	DEPARTAMENTO	TOTAL_CREDITO
9	HUANUCO	5933777.90
24	UCAYALI	5283552.50
0	AMAZONAS	4433386.43
18	PASCO	4100216.00
8	HUANCAVELICA	2934178.35
2	APURIMAC	2214019.46
23	TUMBES	1098119.30
16	MADRE DE DIOS	495900.00

## 4. Presentación de resultados:

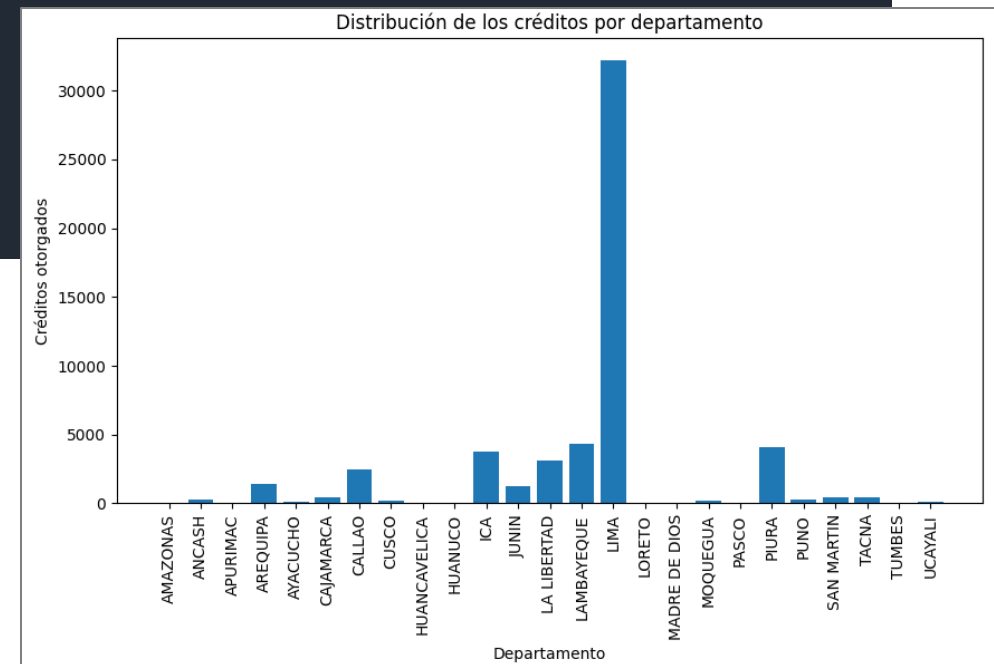
```
plt.figure(figsize=(10, 6))  
plt.bar(df_PromCredito_x_Producto.index, df_PromCredito_x_Producto.values)  
plt.title('Crédito promedio por producto')  
plt.xlabel('Producto')  
plt.ylabel('Crédito promedio')  
plt.show()
```



# TRANSFORMACIÓN

## 4. Presentación de resultados:

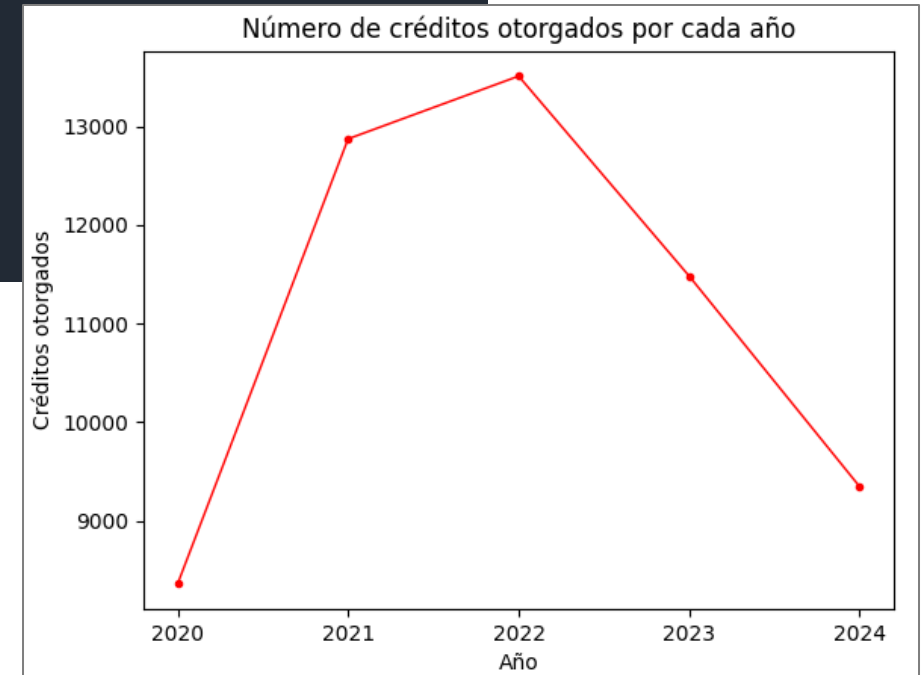
```
plt.figure(figsize=(10, 6))  
plt.bar(df_creditos_departamento.index, df_creditos_departamento.values)  
plt.title('Distribución de los créditos por departamento')  
plt.xlabel('Departamento')  
plt.ylabel('Créditos otorgados')  
plt.xticks(rotation=90)  
plt.show()
```



## 4. Presentación de resultados:

```
arr_años = np.array(df_creditos_x_año["AÑO_DESEMBOLSO"]).astype(str)
arr_frecuencias = np.array(df_creditos_x_año["NUMERO_CREDITOS"])

plt.title('Número de créditos otorgados por cada año')
plt.xlabel('Año')
plt.ylabel('Créditos otorgados')
plt.xticks(range(len(df_creditos_x_año)), arr_años)
plt.plot(arr_frecuencias, '.r-', linewidth = 1)
plt.show()
```



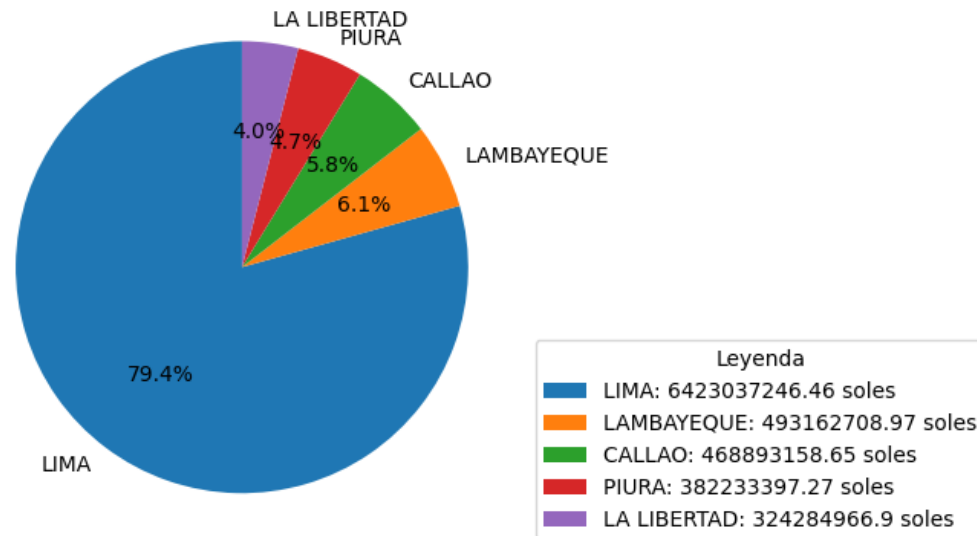
## 4. Presentación de resultados:

```
arr_deptos = np.array(df_TotalCredito_x_Depto["DEPARTAMENTO"])
arr_totales = np.array(df_TotalCredito_x_Depto["TOTAL_CREDITO"])

grafico, ejes = plt.subplots()
sectores, textos, autotextos = ejes.pie(arr_totales[0:5], labels=arr_deptos[0:5], autopct='%1.1f%%', startangle=90)

ejes.legend(sectores, [f'{label}: {size} soles' for label, size in zip(arr_deptos[0:5], arr_totales[0:5])], \
             title="Leyenda", loc="lower left", bbox_to_anchor=(1, 0, 0.5, 1))

plt.show()
```





## Conclusiones del Análisis

- Lima es, con significativa diferencia, la ciudad que más créditos ha gestionado, seguida por Lambayeque, Callao, Piura y La Libertad.
- La mayor cantidad de créditos han sido otorgados a través de Cajas de Ahorro y Crédito.
- Los años de mayor crecimiento fueron 2021 y 2022, notándose una tendencia decreciente significativa en los dos años siguientes.

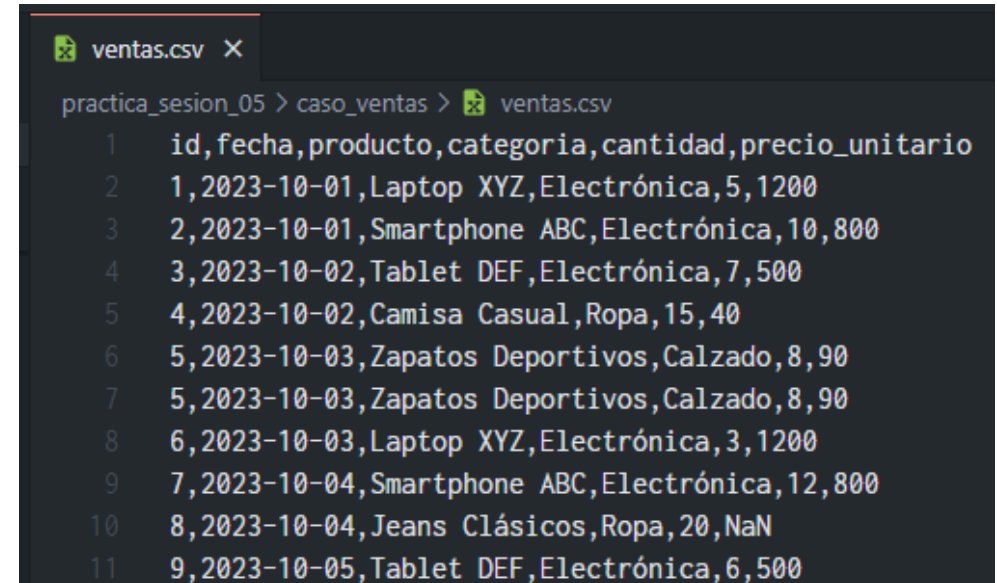
# PRÁCTICA – Sesión 5

## Caso Práctico: Análisis de ventas de una tienda en línea.

Una tienda en línea desea analizar sus ventas para identificar tendencias, productos más vendidos y áreas de mejora. Los datos están almacenados en un archivo CSV llamado *ventas.csv*, que contiene información sobre las transacciones realizadas.

### Objetivos del Análisis:

1. Cargar los datos desde el archivo CSV.
2. Limpiar los datos (valores nulos, duplicados, etc.).
3. Realizar un análisis exploratorio y responder:
  - a) ¿Cuál es el producto más vendido?
  - b) ¿Cuál es el total recaudado en ventas por categoría?
  - c) ¿Cuál es el día de la semana con más ventas?
4. Visualizar resultados y presentar un reporte final con los hallazgos.



```
ventas.csv X
practica_sesion_05 > caso_ventas > ventas.csv
1 id,fecha,producto,categoria,cantidad,precio_unitario
2 1,2023-10-01,Laptop XYZ,Electrónica,5,1200
3 2,2023-10-01,Smartphone ABC,Electrónica,10,800
4 3,2023-10-02,Tablet DEF,Electrónica,7,500
5 4,2023-10-02,Camisa Casual,Ropa,15,40
6 5,2023-10-03,Zapatos Deportivos,Calzado,8,90
7 5,2023-10-03,Zapatos Deportivos,Calzado,8,90
8 6,2023-10-03,Laptop XYZ,Electrónica,3,1200
9 7,2023-10-04,Smartphone ABC,Electrónica,12,800
10 8,2023-10-04,Jeans Clásicos,Ropa,20,NaN
11 9,2023-10-05,Tablet DEF,Electrónica,6,500
```

# PRÁCTICA – Sesión 5

## 1. Carga de datos desde el archivo .csv

```
import numpy as np
import pandas as pd

# 1. Carga de datos desde el archivo .csv
df = pd.read_csv("<ruta>\\ventas.csv")

print(df.head()) # Mostramos los primeros registros

print("Nro de registros: ", df.shape[0]) # Usando Pandas
print("Nro de registros: ", np.shape(df)[0]) # Usando NumPy
```

	id	fecha	producto	categoria	cantidad	precio_unitario
0	1	2023-10-01	Laptop XYZ	Electrónica	5	1200.0
1	2	2023-10-01	Smartphone ABC	Electrónica	10	800.0
2	3	2023-10-02	Tablet DEF	Electrónica	7	500.0
3	4	2023-10-02	Camisa Casual	Ropa	15	40.0
4	5	2023-10-03	Zapatos Deportivos	Calzado	8	90.0
Nro de registros:		23				
Nro de registros:		23				

# PRÁCTICA – Sesión 5

## 2. Limpieza de los datos:

```
# Verificamos valores nulos  
print(df.isnull().sum())
```



id	0
fecha	0
producto	1
categoria	0
cantidad	0
precio_unitario	2

```
# Eliminamos filas con valores nulos en columnas críticas.  
df = df.dropna(subset = ['producto', \  
                        'cantidad', 'precio_unitario'])  
  
print("Nro de registros: ", df.shape[0])
```

Nro de registros: 21

# PRÁCTICA – Sesión 5

## 2. Limpieza de los datos:

```
# Verificar duplicados  
print("Filas duplicadas: ", df.duplicated().sum())
```

Filas duplicadas: 1

```
# Eliminación de filas duplicadas  
df = df.drop_duplicates()
```

```
print("Nro de registros: ", df.shape[0])
```

Nro de registros: 20

```
# Convertimos la columna fecha a tipo datetime  
df['fecha'] = pd.to_datetime(df["fecha"])
```

```
# Mostramos información del DataFrame  
df.info()
```

```
Data columns (total 6 columns):  
#      Column      Non-Null Count  Dtype  
---  -  
0     id           20 non-null      int64  
1     fecha        20 non-null      datetime64[ns]  
2     producto     20 non-null      object  
3     categoria    20 non-null      object  
4     cantidad     20 non-null      int64  
5     precio_unitario 20 non-null      float64
```

# PRÁCTICA – Sesión 5

## 3. Análisis de datos. Pregunta 1: ¿Cuál es el producto más vendido?

```
productos_vendidos = df.groupby('producto')['cantidad'].sum().sort_values(ascending = False)
print(productos_vendidos)
```

```
producto
Smartphone ABC      41
Camisa Casual       37
Jeans Clásicos      35
Tablet DEF          22
Laptop XYZ          18
Zapatos Deportivos  15
Zapatos Formales     9
Name: cantidad, dtype: int64
```

```
print(productos_vendidos.idxmax(), 'con', productos_vendidos.max(), 'unidades vendidas')
```

```
Smartphone ABC con 41 unidades vendidas
```



# PRÁCTICA – Sesión 5

## 3. Análisis de datos.

Pregunta 2: ¿Cuál es el total recaudado en ventas por categoría?

```
df['total'] = df["cantidad"] * df["precio_unitario"]
df_ventas_x_categoria = df.groupby("categoria")['total'].sum() \
                           .sort_values(ascending = False)
print(df_ventas_x_categoria)
```

```
categoria
Electrónica    65400.0
Ropa           3580.0
Calzado        2430.0
Name: total, dtype: float64
```

# PRÁCTICA – Sesión 5

## 3. Análisis de datos.

Pregunta 3: ¿Cuál es el día de la semana con más ventas?

```
df['dia_semana'] = df['fecha'].dt.day_name()
df_ventas_por_dia = df.groupby('dia_semana')['total'].sum().sort_values(ascending = False)
```

```
dia_semana
Monday      20100.0
Wednesday   15300.0
Sunday       14630.0
Friday       11200.0
Tuesday       5280.0
Thursday     3600.0
Saturday     1300.0
Name: total, dtype: float64
```

```
print(df_ventas_por_dia.idxmax(), 'con S/.', \
      df_ventas_por_dia.max(), 'en ventas')
```

```
Monday con S/. 20100.0 en ventas
```

# PRÁCTICA – Sesión 5

## 4. Visualizar resultados y presentar un reporte final con los hallazgos.

Número de registros en ventas.csv: 23

Filas con valores nulos: 2

Filas duplicadas: 2

Registros válidos tras depuración: 20

Producto más vendido:  
Smartphone ABC, con 41 unidades vendidas

Recaudación por categoría:

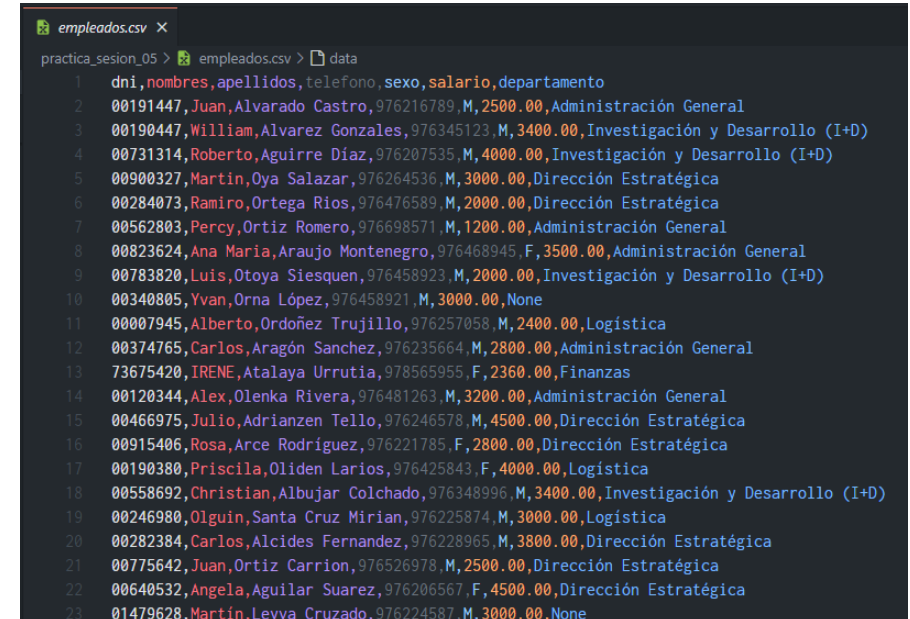
Electrónica	65400.0
Ropa	3580.0
Calzado	2430.0

Día de la semana con más ventas:  
Lunes, con S/.20100.0 en ventas

# ESPACIO PRÁCTICO (tarea) – Sesión 5

Utilice el archivo [empleados.csv](#) proporcionado para realizar el proceso de carga, limpieza y análisis de datos, y responder a las siguientes preguntas:

- Obtener el salario máximo, mínimo y promedio.
- Mostrar la cantidad de empleados que trabajan en cada departamento.
- Mostrar la cantidad de empleados por rango de salario (rangos de 1000 soles)



```
empleados.csv
practica_sesion_05 > empleados.csv > data
1  dni,nombres,apellidos,telefono,sexo,salario,departamento
2  00191447,Juan,Alvarado Castro,976216789,M,2500.00,Administración General
3  00190447,William,Alvarez Gonzales,976345123,M,3400.00,Investigación y Desarrollo (I+D)
4  00731314,Roberto,Aguirre Díaz,976207535,M,4000.00,Investigación y Desarrollo (I+D)
5  00900327,Martin,Oya Salazar,976264536,M,3000.00,Dirección Estratégica
6  00284073,Ramiro,Ortega Rios,976476589,M,2000.00,Dirección Estratégica
7  00562803,Percy,Ortiz Romero,976698571,M,1200.00,Administración General
8  00823624,Ana Maria,Araujo Montenegro,976468945,F,3500.00,Administración General
9  00783820,Luis,Otoya Siesquen,976458923,M,2000.00,Investigación y Desarrollo (I+D)
10 00340805,Yvan,Orna López,976458921,M,3000.00,None
11 00007945,Alberto,Ordoñez Trujillo,976257058,M,2400.00,Logística
12 00374765,Carlos,Aragón Sanchez,976235664,M,2800.00,Administración General
13 73675420,IRENE,Atalaya Urrutia,978565955,F,2360.00,Finanzas
14 00120344,Alex,Olenka Rivera,976481263,M,3200.00,Administración General
15 00466975,Julio,Adrianzen Tello,976246578,M,4500.00,Dirección Estratégica
16 00915406,Rosa,Arce Rodriguez,976221785,F,2800.00,Dirección Estratégica
17 00190380,Priscila,Oliden Larios,976425843,F,4000.00,Logística
18 00558692,Christian,Albujar Colchado,976348996,M,3400.00,Investigación y Desarrollo (I+D)
19 00246980,Olguin,Santa Cruz Mirian,976225874,M,3000.00,Logística
20 00282384,Carlos,Alcides Fernandez,976228965,M,3800.00,Dirección Estratégica
21 00775642,Juan,Ortiz Carrion,976526978,M,2500.00,Dirección Estratégica
22 00640532,Angela,Aguilar Suarez,976206567,F,4500.00,Dirección Estratégica
23 01479628,Martin,Leyva Cruzado,976224587,M,3000.00,None
```

# CIERRE - Sesión 05

Describe cómo  
podemos leer  
un archivo con  
Pandas

¿En qué  
consiste el  
proceso de  
limpieza de  
datos en un  
DataFrame?

Menciona cinco  
operaciones  
que podemos  
realizar sobre  
un DataFrame



# Gracias por su atención