

MÓDULO 1: PROGRAMACIÓN ORIENTADA A IA

OBJETIVO GENERAL: BRINDAR HERRAMIENTAS DE PROGRAMACIÓN FUNDAMENTALES CON PYTHON PARA ABORDAR PROBLEMAS EN INTELIGENCIA ARTIFICIAL.

- CLASES 1-2: FUNDAMENTOS DE PYTHON. TIPOS DE DATOS. ESTRUCTURAS DE CONTROL. ESTRUCTURAS DE DATOS.
- CLASE 3: PROGRAMACIÓN MODULAR. PROGRAMACIÓN ORIENTADA A OBJETOS.
- CLASE 4: MANEJO DE ARCHIVOS Y LECTURA DE DATOS. LIBRERÍAS CIENTÍFICAS.



MÓDULO 1: PROGRAMACIÓN ORIENTADA A IA

OBJETIVO GENERAL: BRINDAR HERRAMIENTAS DE PROGRAMACIÓN FUNDAMENTALES CON PYTHON PARA ABORDAR PROBLEMAS EN INTELIGENCIA ARTIFICIAL.

- CLASES 1-2: FUNDAMENTOS DE PYTHON. TIPOS DE DATOS. ESTRUCTURAS DE CONTROL. ESTRUCTURAS DE DATOS.
- CLASE 3: PROGRAMACIÓN MODULAR. PROGRAMACIÓN ORIENTADA A OBJETOS.
- CLASE 4: MANEJO DE ARCHIVOS Y LECTURA DE DATOS. LIBRERÍAS CIENTÍFICAS.



MÓDULO 1: PROGRAMACIÓN ORIENTADA A IA

- CLASE 4: MANEJO DE ARCHIVOS Y LECTURA DE DATOS. LIBRERÍAS CIENTÍFICAS.

OBJETIVO: APRENDER A IMPORTAR, EXPLORAR Y MANIPULAR ARCHIVOS DE DATOS ESTRUCTURADOS EN PYTHON, UTILIZANDO LIBRERÍAS ESTÁNDAR Y HERRAMIENTAS DE ANÁLISIS DE DATOS.



MÓDULO 1: PROGRAMACIÓN ORIENTADA A IA

- CLASE 4: MANEJO DE ARCHIVOS Y LECTURA DE DATOS. LIBRERÍAS CIENTÍFICAS.
 - CONCEPTOS BÁSICOS SOBRE ARCHIVOS
 - FUENTES DE DATOS
 - LECTURA Y ESCRITURA EN PYTHON
 - LIBRERÍA PANDAS
 - LIBRERÍAS ESENCIALES PARA IA



MÓDULO 1: PROGRAMACIÓN ORIENTADA A IA

- CLASE 4: MANEJO DE ARCHIVOS Y LECTURA DE DATOS. LIBRERÍAS CIENTÍFICAS.
 - CONCEPTOS BÁSICOS SOBRE ARCHIVOS
 - FUENTES DE DATOS
 - LECTURA Y ESCRITURA EN PYTHON
 - LIBRERÍA PANDAS
 - LIBRERÍAS ESENCIALES PARA IA



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

¿QUÉ ES UN DATO?

- ✓ REPRESENTACIÓN SIMBÓLICA (NUMÉRICA, ALFABÉTICA, ETC) DE UNA VARIABLE CUANTITATIVA O CUALITATIVA.
- ✓ POR SI MISMO NO CONSTITUYE INFORMACIÓN.
- ✓ EL PROCESAMIENTO DE LOS DATOS ES LO QUE NOS PROPORCIONA INFORMACIÓN.



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

¿QUÉ ES UN ARCHIVO?

✓ UN ARCHIVO ES UN CONJUNTO DE DATOS ALMACENADOS EN UN DISPOSITIVO.

✓ PUEDE CONTENER TEXTO, NÚMEROS, IMÁGENES, AUDIO, ETC.



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

DATOS ESTRUCTURADOS VS NO ESTRUCTURADOS

✓ *DATOS ESTRUCTURADOS:*

- SON INFORMACIÓN ORGANIZADA EN UN FORMATO ESPECÍFICO QUE LA HACE FÁCILMENTE ACCESIBLE Y COMPRENSIBLE TANTO PARA HUMANOS COMO PARA MÁQUINAS.
- INCLUYEN LA INFORMACIÓN ALMACENADA EN HOJAS DE CÁLCULO O BASES DE DATOS QUE SE AJUSTAN A UNA ESTRUCTURA BIEN DEFINIDA.



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

DATOS ESTRUCTURADOS VS NO ESTRUCTURADOS

✓ *DATOS NO ESTRUCTURADOS:*

- SON INFORMACIÓN QUE NO ESTÁ ORGANIZADA EN UNA ESTRUCTURA PREDEFINIDA. ESTO SIGNIFICA QUE SON DIFÍCILES DE BUSCAR, ANALIZAR Y GESTIONAR.
- LA MAYORÍA DE LOS DATOS NO ESTÁN ESTRUCTURADOS.



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

DATOS ESTRUCTURADOS VS NO ESTRUCTURADOS

✓ *DATOS NO ESTRUCTURADOS:*

- INCLUYEN UNA AMPLIA GAMA DE FORMATOS, COMO CORREOS ELECTRÓNICOS, ARCHIVOS DE TEXTO, PUBLICACIONES EN REDES SOCIALES, VÍDEOS, IMÁGENES, AUDIO, DATOS DE SENSORES, ETC.
- SUELEN SER UN RECURSO SIN EXPLOTAR Y PUEDEN PROPORCIONAR INFORMACIÓN VALIOSA QUE NO OFRECEN LOS DATOS ESTRUCTURADOS POR SÍ SOLOS.



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

TIPOS DE ARCHIVOS:

- ✓ *TXT*
- ✓ *CSV*
- ✓ *JSON*
- ✓ *XLSX*



- CADA UNO TIENE VENTAJAS SEGÚN EL CASO
- EXISTEN SITIOS WEB GRATUITOS QUE CONVIERTEN DE UN TIPO DE ARCHIVO A OTRO, POR EJEMPLO DE JSON A CSV O VICEVERSA



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

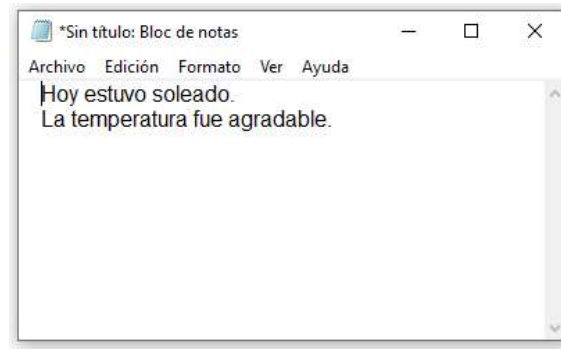
TIPOS DE ARCHIVOS:

✓ *TXT*

- FORMATO DE TEXTO PLANO. NO CONTIENE FORMATO ESPECIAL
- GENERALMENTE SE USAN PARA DATOS NO ESTRUCTURADOS (TEXTO LIBRE)

EJEMPLO:

Hoy estuvo soleado.
La temperatura fue agradable.



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

TIPOS DE ARCHIVOS:

✓ *TXT*

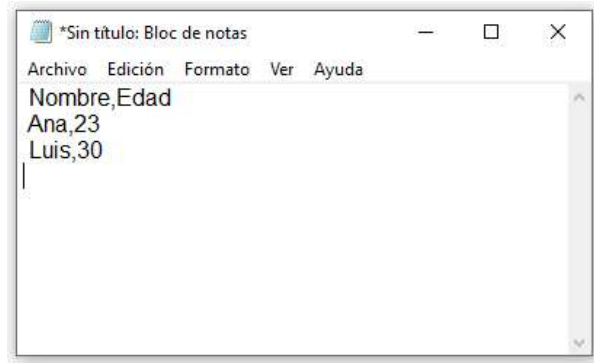
○ TAMBIÉN PUEDEN CONTENER DATOS ESTRUCTURADOS SI SE USAN DELIMITADORES

EJEMPLO:

Nombre,Edad

Ana,23

Luis,30



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

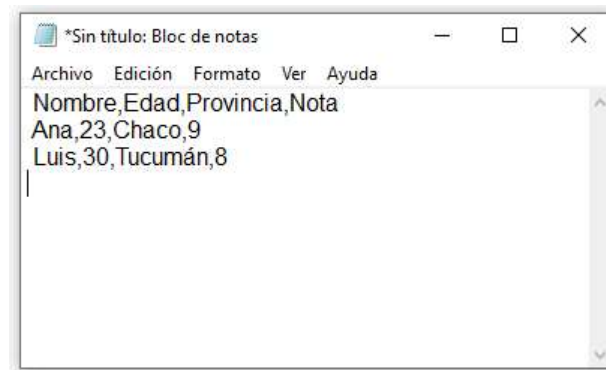
TIPOS DE ARCHIVOS:

✓ CSV

- DATOS TABULARES
- VALORES SEPARADOS POR COMA (*COMMA SEPARATED VALUES*)
- CADA FILA = UN REGISTRO
- CADA COLUMNA = UNA VARIABLE

EJEMPLO:

```
Nombre,Edad,Provincia,Nota  
Ana,23,Chaco,9  
Luis,30,Tucumán,8
```



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

TIPOS DE ARCHIVOS:

✓ *JSON*

- NOTACIÓN DE OBJETOS JAVASCRIPT (*JAVASCRIPT OBJECT NOTATION*)
- REPRESENTA DATOS ESTRUCTURADOS SEGÚN LA SINTAXIS DE OBJETOS DE JAVASCRIPT, AUNQUE ES COMPATIBLE CON LA MAYORÍA DE LOS LENGUAJES
- MUY USADO PARA TRANSMITIR DATOS EN APLICACIONES WEB



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

TIPOS DE ARCHIVOS:

✓ *JSON*

- PERMITE REPRESENTAR INFORMACIÓN ESTRUCTURADA DE MANERA JERÁRQUICA.
- SE BASA EN LAS ESTRUCTURAS CLAVE-VALOR

EJEMPLO:

```
{  
  'Nombre': 'Ana',  
  'Edad': 23  
}
```

```
1 {  
2   "array": [  
3     1,  
4     2,  
5     3  
6   ],  
7   "boolean": true,  
8   "color": "gold",  
9   "null": null,  
10  "number": 123,  
11  "object": {  
12    "a": "b",  
13    "c": "d"  
14  },  
15  "string": "Hello World"  
16 }
```



CONCEPTOS BÁSICOS SOBRE ARCHIVOS

TIPOS DE ARCHIVOS:

✓ XLSX

- FORMATO DE HOJAS DE CÁLCULO DE EXCEL
- PERMITE VARIAS HOJAS, FÓRMULAS Y TABLAS
- MUY USADO EN EMPRESAS Y PARA USO PERSONAL



MÓDULO 1: PROGRAMACIÓN ORIENTADA A IA

- CLASE 4: MANEJO DE ARCHIVOS Y LECTURA DE DATOS. LIBRERÍAS CIENTÍFICAS.
 - CONCEPTOS BÁSICOS SOBRE ARCHIVOS
 - FUENTES DE DATOS
 - LECTURA Y ESCRITURA EN PYTHON
 - LIBRERÍA PANDAS
 - LIBRERÍAS ESENCIALES PARA IA



FUENTES DE DATOS

¿DE DÓNDE PODEMOS OBTENER LOS DATOS?

- ✓ DISCO LOCAL
- ✓ ALMACENAMIENTO EN LA NUBE (GOOGLE DRIVE, ONEDRIVE, DROPBOX)
- ✓ PÁGINAS WEB
- ✓ REPOSITORIOS ONLINE (GITHUB, KAGGLE)
- ✓ APIs



FUENTES DE DATOS

PÁGINAS WEB (EJEMPLOS)

- ✓ INDEC (INSTITUTO NACIONAL DE ESTADÍSTICA Y CENSOS)
(<https://www.indec.gob.ar/indec/web/institucional-indec-basesdedatos>)
- ✓ INTA (INSTITUTO NACIONAL DE TECNOLOGÍA AGROPECUARIA) - SISTEMA DE INFORMACIÓN Y GESTIÓN AGROMETEOROLÓGICA (SIGA)
(<https://sig.inta.gob.ar/#/>)
- ✓ DATOS ARGENTINA (<https://datos.gob.ar/>)
- ✓ DATOS PROVINCIA BUENOS AIRES (<https://catalogo.datos.gba.gob.ar/>)



FUENTES DE DATOS

¿QUÉ ES UN REPOSITORIO?

- ✓ SON SITIOS PÚBLICOS (O PRIVADOS, SI SE CONFIGURAN ASÍ) DONDE SE ALMACENAN DATASETS O PROYECTOS COMPLETOS, NORMALMENTE PENSADOS PARA COMPARTIR, VERSIONAR Y COLABORAR.
- ✓ EJEMPLOS: GITHUB, KAGGLE (Debemos tener usuario y contraseña).
- ✓ LOS SERVICIOS DE ALMACENAMIENTO EN LA NUBE (GOOGLE DRIVE, ONEDRIVE, DROPBOX, ETC.) NO SON REPOSITORIOS DE DATOS EN SÍ, SINO ESPACIOS PERSONALES DE ALMACENAMIENTO.



FUENTES DE DATOS

¿QUÉ ES GITHUB?

- ✓ ES UNA PLATAFORMA WEB PARA GUARDAR PROYECTOS DE SOFTWARE Y DATASETS (<https://github.com/>)
- ✓ UTILIZA UN CONTROLADOR DE VERSIONES PARA ALMACENAR EL CÓDIGO Y SU HISTORIAL DE CAMBIOS
- ✓ PERMITE A DESARROLLADORES COLABORAR, COMPARTIR Y MANTENER CÓDIGO.



FUENTES DE DATOS



¿QUÉ ES KAGGLE?

- ✓ ES UNA PLATAFORMA WEB Y UNA COMUNIDAD ONLINE PARA CIENTÍFICOS DE DATOS Y PROFESIONALES DEL APRENDIZAJE AUTOMÁTICO (<https://www.kaggle.com/>)
- ✓ OFRECE COMPETICIONES, CONJUNTOS DE DATOS Y CUADERNOS DE CÓDIGO PARA APRENDER Y DESARROLLAR HABILIDADES EN IA.
- ✓ ALMACENA UNA GRAN CANTIDAD DE DATOS PÚBLICOS, PERMITE A LOS USUARIOS CREAR Y COMPARTIR MODELOS, Y FACILITA LA COLABORACIÓN PARA RESOLVER PROBLEMAS COMPLEJOS DE MACHINE LEARNING.



FUENTES DE DATOS

¿QUÉ ES UNA API?

- ✓ API = INTERFAZ DE PROGRAMACIÓN DE APLICACIONES.
- ✓ ES UN CONJUNTO DE REGLAS, PROTOCOLOS Y HERRAMIENTAS QUE PERMITE QUE DIFERENTES APLICACIONES DE SOFTWARE SE COMUNIQUEN ENTRE SÍ PARA INTERCAMBIAR DATOS Y FUNCIONALIDADES.
- ✓ PERMITE ACCEDER A DATOS DIRECTAMENTE DESDE UNA WEB.
- ✓ EJEMPLOS DE APIs:
 - API DEL CLIMA: Permite consultar parámetros y pronósticos en tiempo real
 - APIS DE MAPAS: Permiten integrar un mapa en una aplicación de transporte o en un sitio web de viajes, como es el caso de Google Maps.



MÓDULO 1: PROGRAMACIÓN ORIENTADA A IA

- CLASE 4: MANEJO DE ARCHIVOS Y LECTURA DE DATOS. LIBRERÍAS CIENTÍFICAS.
 - CONCEPTOS BÁSICOS SOBRE ARCHIVOS
 - FUENTES DE DATOS
 - LECTURA Y ESCRITURA EN PYTHON
 - LIBRERÍA PANDAS
 - LIBRERÍAS ESENCIALES PARA IA



LECTURA Y ESCRITURA EN PYTHON

FUNCIONES DE APERTURA DE ARCHIVOS: `open()`, `with open()` as f

MODOS DE APERTURA:

- ✓ 'r' LEER (no es posible realizar modificaciones sobre el archivo, solamente leer su contenido)
- ✓ 'w' ESCRIBIR (el archivo es vaciado si existe, y se lo crea si no existe)
- ✓ 'a' AGREGAR (si no existe, se crea el archivo. Pero en caso de que exista se posiciona al final, manteniendo el contenido original)



LECTURA Y ESCRITURA EN PYTHON

LECTURA DE UN ARCHIVO

```
with open('archivo.txt','r') as f:  
    contenido = f.read()
```

ESCRITURA DE UN ARCHIVO

```
with open('salida.txt','w') as f:  
    f.write('Hola IA!')
```

▼ Lectura de un archivo

```
✓ 0s [6] 1 with open('archivo.txt','r') as f:  
        2     contenido = f.read()
```

▼ Escritura de un archivo

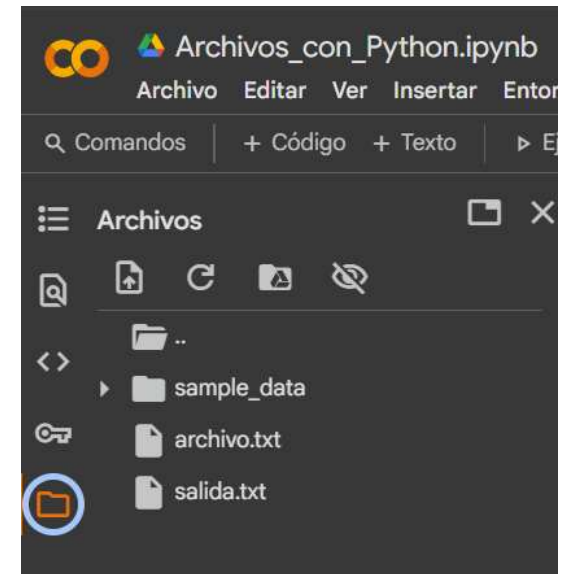
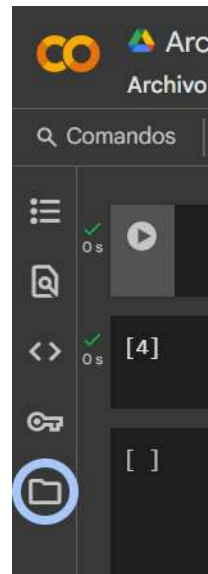
```
✓ 0s [6] 1 with open('salida.txt','w') as f:  
        2     f.write("Hola IA!")
```



LECTURA Y ESCRITURA EN PYTHON

Si usamos la ruta: 'archivo.txt' o 'salida.txt' y trabajamos con el Google Colaboratory, los archivos a leer o los archivos que escribimos deben estar en nuestro Google Drive. Sino, debemos pasarle la ruta completa.

Estos datos se pierden cada vez que finalizó la sesión.



MÓDULO 1: PROGRAMACIÓN ORIENTADA A IA

- CLASE 4: MANEJO DE ARCHIVOS Y LECTURA DE DATOS. LIBRERÍAS CIENTÍFICAS.
 - CONCEPTOS BÁSICOS SOBRE ARCHIVOS
 - FUENTES DE DATOS
 - LECTURA Y ESCRITURA EN PYTHON
 - LIBRERÍA PANDAS
 - LIBRERÍAS ESENCIALES PARA IA



LIBRERÍA PANDAS

¿QUÉ ES PANDAS?

- ✓ LIBRERÍA DE PYTHON PARA ANÁLISIS DE DATOS
- ✓ FACILITA IMPORTAR, MANIPULAR Y EXPORTAR DATASETS
- ✓ PROPORCIONA ESTRUCTURAS DE DATOS COMO LAS SERIES Y LOS DATAFRAMES Y HERRAMIENTAS PARA MANIPULAR Y ANALIZAR DATOS TABULARES, DE FORMA SIMILAR A UNA HOJA DE CÁLCULO DE EXCEL.



LIBRERÍA PANDAS

¿QUÉ SON LAS SERIES Y LOS DATAFRAME?

✓ UNA SERIE EN PYTHON, USANDO LA BIBLIOTECA PANDAS, ES UNA ESTRUCTURA DE DATOS UNIDIMENSIONAL (COMO UNA COLUMNA O LISTA) QUE CONTIENE UN CONJUNTO DE VALORES Y UN ÍNDICE ASOCIADO A ELLOS.

✓ UN DATAFRAME, EN CAMBIO, ES UNA ESTRUCTURA DE DATOS BIDIMENSIONAL QUE SE ASEMEJA A UNA TABLA DE EXCEL O UNA HOJA DE CÁLCULO, COMPUESTA POR FILAS Y COLUMNAS, DONDE CADA COLUMNA ES UNA SERIE.

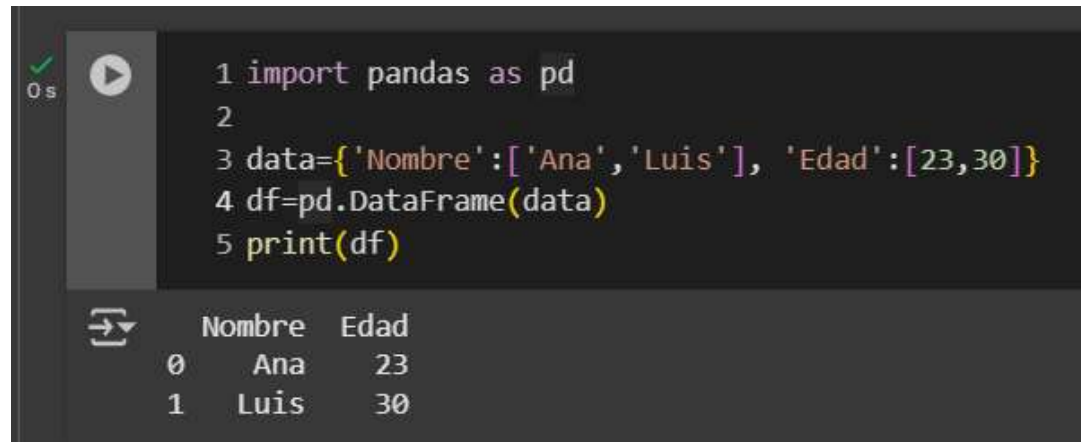


LIBRERÍA PANDAS

EJEMPLO PARA CREAR UN DATAFRAME

```
import pandas as pd
```

```
data={'Nombre':['Ana', 'Luis'], 'Edad':[23,30]}  
df=pd.DataFrame(data)  
print(df)
```



The screenshot shows a Jupyter Notebook interface. On the left, there is a green checkmark and a play button icon, with '0 s' indicating execution time. The main area displays the following code:

```
1 import pandas as pd  
2  
3 data={'Nombre':['Ana', 'Luis'], 'Edad':[23,30]}  
4 df=pd.DataFrame(data)  
5 print(df)
```

Below the code, the output of the print statement is shown as a table:

	Nombre	Edad
0	Ana	23
1	Luis	30

LIBRERÍA PANDAS

EJEMPLO PARA LEER UN ARCHIVO CON PANDAS

```
import pandas as pd
```

```
df1=pd.read_csv('archivo.csv')  
df2=pd.read_excel('archivo.xlsx')  
df3=pd.read_json('archivo.json')
```

```
print(df1)
```

```
1 import pandas as pd  
2  
3 df1=pd.read_csv('archivo.csv')  
4 #df2=pd.read_excel('archivo.xlsx')  
5 #df3=pd.read_json('archivo.json')  
6  
7 print(df1)  
8
```

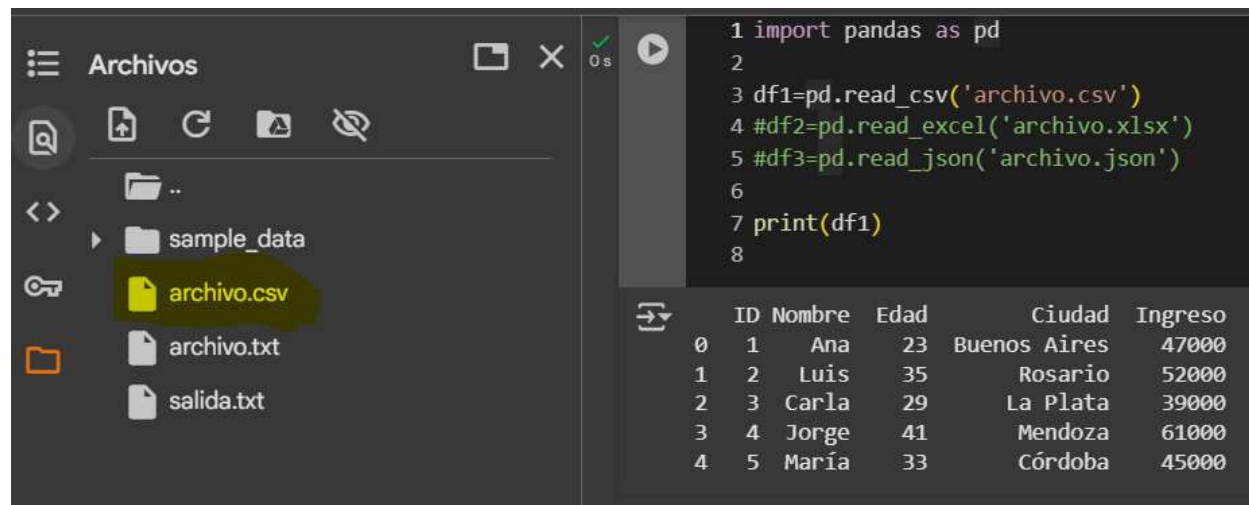
	ID	Nombre	Edad	Ciudad	Ingreso
0	1	Ana	23	Buenos Aires	47000
1	2	Luis	35	Rosario	52000
2	3	Carla	29	La Plata	39000
3	4	Jorge	41	Mendoza	61000
4	5	María	33	Córdoba	45000



LIBRERÍA PANDAS

EJEMPLO PARA LEER UN ARCHIVO CON PANDAS

Nuevamente, la ruta: 'archivo.csv' implica que el archivo debe estar en nuestro Google Drive. Sino, debemos pasarle la ruta completa. Estos datos se pierden cada vez que finalizó la sesión.



The screenshot shows a Jupyter Notebook interface. On the left, the 'Archivos' (Files) sidebar displays a directory structure with a folder named 'sample_data' and three files: 'archivo.csv' (highlighted in yellow), 'archivo.txt', and 'salida.txt'. The main area contains a code cell with the following Python code:

```
1 import pandas as pd
2
3 df1=pd.read_csv('archivo.csv')
4 #df2=pd.read_excel('archivo.xlsx')
5 #df3=pd.read_json('archivo.json')
6
7 print(df1)
8
```

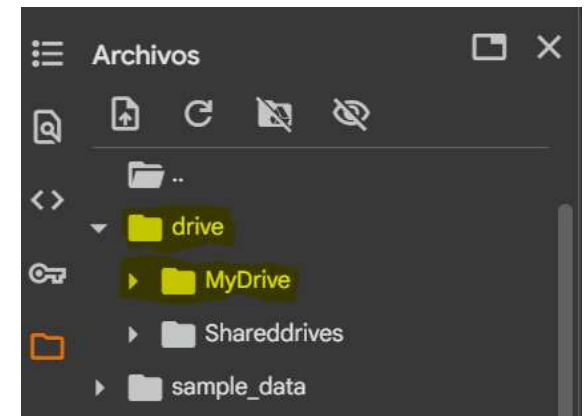
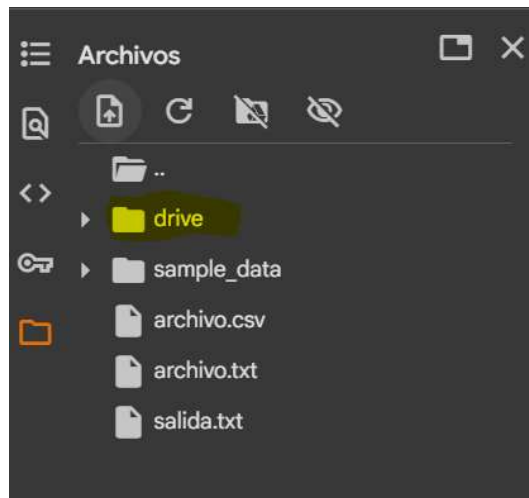
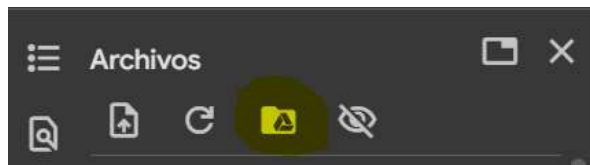
Below the code cell, the output is displayed as a table with 6 columns: ID, Nombre, Edad, Ciudad, and Ingreso. The table contains 5 rows of data:

	ID	Nombre	Edad	Ciudad	Ingreso
0	1	Ana	23	Buenos Aires	47000
1	2	Luis	35	Rosario	52000
2	3	Carla	29	La Plata	39000
3	4	Jorge	41	Mendoza	61000
4	5	María	33	Córdoba	45000

LIBRERÍA PANDAS

EJEMPLO PARA LEER UN ARCHIVO CON PANDAS

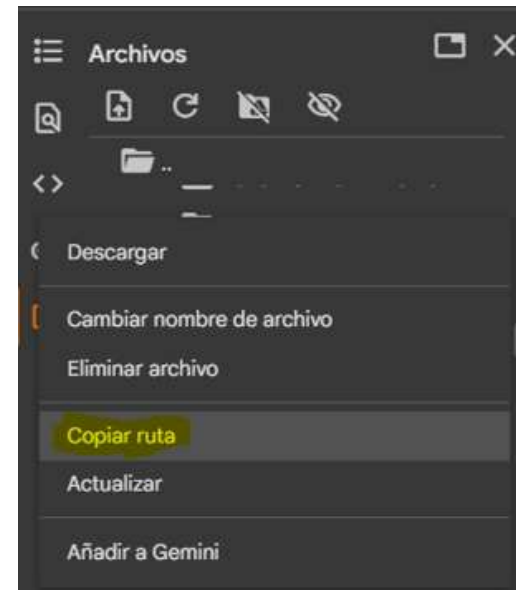
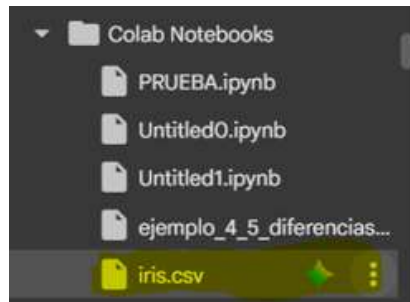
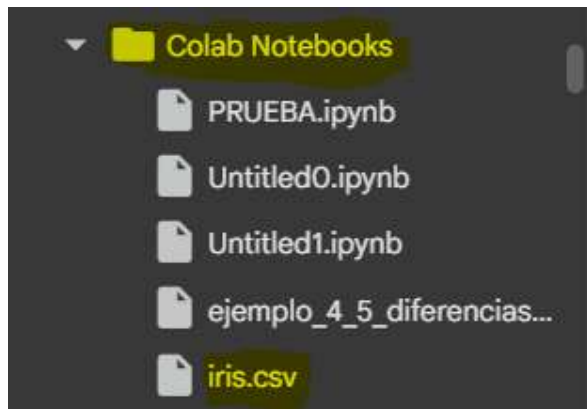
Si el archivo a leer está en nuestro Drive pero en otra carpeta debemos copiar la ruta completa.



LIBRERÍA PANDAS

EJEMPLO PARA LEER UN ARCHIVO CON PANDAS

Si el archivo a leer está en nuestro Drive pero en otra carpeta debemos copiar la ruta completa.



LIBRERÍA PANDAS

EJEMPLO PARA LEER UN ARCHIVO CON PANDAS

Si el archivo a leer está en nuestro Drive pero en otra carpeta debemos copiar la ruta completa.

✓ Leer un archivo con Pandas ubicado en Drive.

Le especificamos la ruta:

```
[12] 1 import pandas as pd
      2
      3 # Leer un archivo CSV
      4 df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/iris.csv")
      5
      6 print(df.head()) # Muestra las primeras filas
```

	sepal_length	sepal_width	petal_length	petal_width	name
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Si la ruta es muy larga, se puede guardar en una variable:

```
1 import pandas as pd
2
3 # Leer un archivo CSV
4 ruta = "/content/drive/MyDrive/Colab Notebooks/iris.csv"
5 df = pd.read_csv(ruta)
6
7 print(df.head()) # Muestra las primeras filas
8
```

	sepal_length	sepal_width	petal_length	petal_width	name
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

LIBRERÍA PANDAS

EJEMPLO PARA LEER UN ARCHIVO CON PANDAS DESDE URL

(URL: dirección que identifica la ubicación de un recurso único en Internet, como una página web, una imagen o un documento)

Por ejemplo, si el archivo está en un repositorio público de GitHub, se lo puede leer directamente con Pandas usando la URL cruda (raw) del archivo.

```
import pandas as pd

# URL cruda (raw) del archivo en GitHub
url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/iris.csv"

# Leer CSV directamente desde la URL
df = pd.read_csv(url)

print(df.head())
```



LIBRERÍA PANDAS

EJEMPLO PARA LEER UN ARCHIVO CON PANDAS DESDE URL

```
sepal_length,sepal_width,petal_length,petal_width,species
5.1,3.5,1.4,0.2,setosa
4.9,3.0,1.4,0.2,setosa
4.7,3.2,1.3,0.2,setosa
4.6,3.1,1.5,0.2,setosa
5.0,3.6,1.4,0.2,setosa
5.4,3.9,1.7,0.4,setosa
4.6,3.4,1.4,0.3,setosa
5.0,3.4,1.5,0.2,setosa
4.4,2.9,1.4,0.2,setosa
4.9,3.1,1.5,0.1,setosa
5.4,3.7,1.5,0.2,setosa
4.8,3.4,1.6,0.2,setosa
4.8,3.0,1.4,0.1,setosa
4.3,3.0,1.1,0.1,setosa
5.8,4.0,1.2,0.2,setosa
5.7,4.4,1.5,0.4,setosa
5.4,3.9,1.3,0.4,setosa
5.1,3.5,1.4,0.3,setosa
5.7,3.8,1.7,0.3,setosa
5.1,3.8,1.5,0.3,setosa
5.4,3.4,1.7,0.2,setosa
5.1,3.7,1.5,0.4,setosa
4.6,3.6,1.0,0.2,setosa
5.1,3.3,1.7,0.5,setosa
```

Leer un archivo con pandas desde una URL

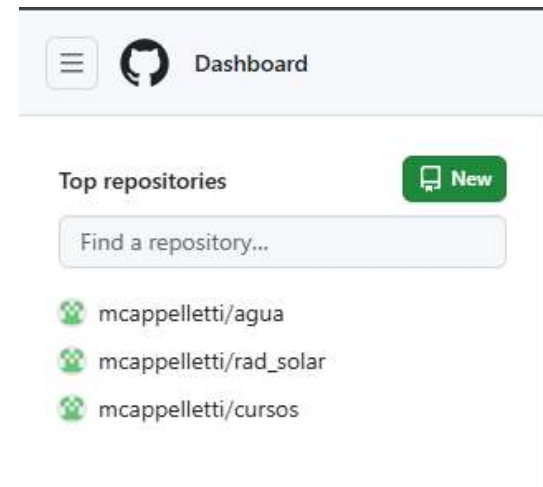
```
1 import pandas as pd
2
3 # URL cruda (raw) del archivo en GitHub
4 url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/iris.csv"
5
6 # Leer CSV directamente desde la URL
7 df = pd.read_csv(url)
8
9 print(df.head())
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

LIBRERÍA PANDAS



EJEMPLO PARA LEER UN ARCHIVO CON PANDAS DESDE NUESTRO GITHUB



- 1- Iniciar sesión
- 2- Buscar el archivo o agregarlo en algún repositorio.
- 3- Hacer clic en el botón “Raw”.
- 4- Copiar la URL.
- 5- Usarla en `pd.read_csv(url)`.





LIBRERÍA PANDAS






EJEMPLO PARA LEER UN ARCHIVO CON PANDAS DESDE NUESTRO GITHUB

 main `cursos / 02_Buenos_Aires_San_Pedro.csv` 


Go to file  

We can make this file [beautiful and searchable](#) if this error is corrected: No commas found in this CSV file in line 0.

 **mcappelletti** Add files via upload 77680b1 · 29 minutes ago  History

Preview Code Blame 507 lines (507 loc) · 25.9 KB  **Raw**    

```
1 Fecha;Temperatura_Abrigo_150cm;Humedad_Media;Presion_Media;Radiacion_Global
2 2023-05-01 00:00:00.0;14.05764;63;1016.089;10.39776
3 2023-05-02 00:00:00.0;13.39375;54;1017.172;10.38144
4 2023-05-03 00:00:00.0;11.17778;74;1015.728;10.23486
5 2023-05-04 00:00:00.0;14.56736;76;1013.973;9.822361
6 2023-05-05 00:00:00.0;16.99792;92;1013.477;1.57902
7 2023-05-06 00:00:00.0;17.6618;99;1007.203;1.37706
8 2023-05-07 00:00:00.0;16.50278;83;1009.491;8.02122
9 2023-05-08 00:00:00.0;16.30694;76;1013.322;9.237181
10 2023-05-09 00:00:00.0;13.45944;83;1015.764;6.94512
```



LIBRERÍA PANDAS

EJEMPLO PARA ESCRIBIR UN ARCHIVO CON PANDAS

`df.to_csv` → guarda el contenido del DataFrame `df` en un archivo CSV.

`df.to_excel` → guarda el contenido del DataFrame `df` en un archivo XLSX.

```
import pandas as pd
```

```
df.to_csv('salida.csv', index=False) # guardar sin índice
```

```
df.to_excel('salida.xlsx')
```



MÓDULO 1: PROGRAMACIÓN ORIENTADA A IA

- CLASE 4: MANEJO DE ARCHIVOS Y LECTURA DE DATOS. LIBRERÍAS CIENTÍFICAS.
 - CONCEPTOS BÁSICOS SOBRE ARCHIVOS
 - FUENTES DE DATOS
 - LECTURA Y ESCRITURA EN PYTHON
 - LIBRERÍA PANDAS
 - LIBRERÍAS ESENCIALES PARA IA



LIBRERÍAS ESENCIALES PARA IA

- ❑ LAS LIBRERÍAS SON CONJUNTOS DE MÓDULOS, QUE A SU VEZ SON ARCHIVOS QUE CONTIENEN UNA COLECCIÓN DE FUNCIONES, CLASES Y VARIABLES RELACIONADAS QUE PERMITEN REALIZAR CIERTAS TAREAS SIN TENER QUE ESCRIBIR EL CÓDIGO DESDE CERO.
- ❑ LAS LIBRERÍAS ESTÁN DISEÑADAS PARA SER REUTILIZABLES, LO QUE SIGNIFICA QUE UNA VEZ QUE SE HA ESCRITO UNA FUNCIÓN O CLASE EN UNA LIBRERÍA, SE PUEDE USAR EN MÚLTIPLES PROGRAMAS Y PROYECTOS.



LIBRERÍAS ESENCIALES PARA IA

- ❑ PYTHON CUENTA CON UN ECOSISTEMA MUY AMPLIO DE LIBRERÍAS QUE PERMITEN REALIZAR ANÁLISIS DE DATOS, VISUALIZACIÓN, MODELADO ESTADÍSTICO Y APRENDIZAJE AUTOMÁTICO.



LIBRERÍAS ESENCIALES PARA IA

PRINCIPALES UTILIDADES DE LAS LIBRERÍAS

- ✓ **FACILITAR LA PROGRAMACIÓN:** proporcionan herramientas y funciones ya desarrolladas y evita escribir código desde cero para tareas comunes.
- ✓ **ESPECIFICIDAD:** existen librerías con herramientas especializadas para áreas específicas lo que proporciona un camino más simple y eficiente.
- ✓ **AHORRO DE TIEMPO:** utilizar funciones y métodos predefinidos de las librerías permite ahorrar una cantidad significativa de tiempo en el desarrollo de software.
- ✓ **OPTIMIZACIÓN:** muchas librerías están optimizadas para ofrecer un rendimiento superior, lo que significa que el código que utilizan puede ser más rápido o eficiente que una implementación manual.



LIBRERÍAS ESENCIALES PARA IA

PRINCIPALES UTILIDADES DE LAS LIBRERÍAS

- ✓ **COMUNIDAD Y SOPORTE:** Las librerías populares suelen tener una comunidad activa detrás de ellas. Esto significa que es más fácil encontrar ayuda, tutoriales, documentación y soluciones a problemas comunes.
- ✓ **INTEGRACIÓN:** Las librerías permiten la integración con otras herramientas, sistemas y tecnologías. Por ejemplo, una librería de análisis de datos puede ofrecer herramientas para importar datos de diversas fuentes o para visualizar resultados en gráficos interactivos.
- ✓ **ESTÁNDARES Y BUENAS PRÁCTICAS:** El uso de librerías consolidadas permite aprovechar de las mejores prácticas y estándares de la industria.
- ✓ **PORTABILIDAD:** Suelen ser compatibles con diferentes sistemas y plataformas, lo que facilita la portabilidad del código entre diferentes entornos.



LIBRERÍAS ESENCIALES PARA IA

❑ LIBRERÍAS DE ANÁLISIS DE DATOS:

Pandas – NumPy – SciPy

PROPÓSITO: Manipulación, análisis y visualización de datos.



LIBRERÍAS ESENCIALES PARA IA

❑ Pandas



- Es una librería que ofrece estructuras de datos y operaciones para manipular y analizar tablas numéricas y series temporales.
- Es especialmente útil para la manipulación de datos, limpieza, filtrado y agregación.
- Con Pandas, es fácil cargar, procesar y analizar datos desde diferentes fuentes, facilitando así la preparación de datos para análisis posteriores.



LIBRERÍAS ESENCIALES PARA IA

❑ NumPy



- Proviene de “Numerical Python”.
- Es una de las librerías más importantes para operaciones numéricas en Python.
- Permite trabajar con arreglos, matrices y objetos de dimensiones superiores de una manera eficiente.
- Gracias a su capacidad para realizar operaciones matemáticas rápidas en arreglos de datos sin requerir bucles explícitos, es fundamental para cualquier trabajo científico con Python.



LIBRERÍAS ESENCIALES PARA IA

❑ SciPy



- Es una extensión de NumPy.
- Proporciona una gran cantidad de funciones que son útiles en matemáticas, ciencias e ingeniería.
- Incluye módulos para optimización, álgebra lineal, integración, interpolación, funciones especiales, Transformada Rápida de Fourier (FFT), procesamiento de señales y de imagen, resolución de Ecuaciones Diferenciales Ordinarias (ODEs) y más.



LIBRERÍAS ESENCIALES PARA IA

❑ LIBRERÍAS DE APRENDIZAJE AUTOMÁTICO Y APRENDIZAJE PROFUNDO:

TensorFlow – Keras – Scikit-learn

PROPÓSITO: Creación y entrenamiento de modelos de aprendizaje automático y aprendizaje profundo.



LIBRERÍAS ESENCIALES PARA IA

❑ TensorFlow



- Es una librería de código abierto desarrollada por Google que se utiliza para implementar redes neuronales y otras técnicas de aprendizaje automático.
- Es conocida por su flexibilidad y capacidad para trabajar con grandes conjuntos de datos.



LIBRERÍAS ESENCIALES PARA IA

❑ Keras



- Es una librería que está orientada al desarrollo y entrenamiento de redes neuronales.
- Ofrece una API sencilla y de alto nivel para crear modelos de Deep Learning.
- Está principalmente integrada en TensorFlow. (Se puede pensar a Keras como la capa amigable de TensorFlow: facilita mucho la construcción de modelos sin tener que escribir tanto código complejo).



LIBRERÍAS ESENCIALES PARA IA

❑ Scikit-learn



- Es una herramienta simple y eficiente para el análisis de datos y la minería de datos.
- Proporciona algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad, así como herramientas de selección y evaluación de modelos.



LIBRERÍAS ESENCIALES PARA IA

❑ LIBRERÍAS DE VISUALIZACIÓN:

Matplotlib – Seaborn

PROPÓSITO: Generar gráficos para el análisis visual a partir de datos.



LIBRERÍAS ESENCIALES PARA IA

❑ Matplotlib



- Es una librería de visualización en Python que produce figuras de calidad en una variedad de formatos impresos y entornos interactivos.
- Es ideal para generar gráficos simples con datos rápidamente y sin mucho esfuerzo.
- También tiene soporte para visualizaciones más avanzadas y personalizadas.



LIBRERÍAS ESENCIALES PARA IA

❑ Seaborn



- Es una librería de visualización en Python basada en Matplotlib
- Crea gráficos atractivos e informativos de manera sencilla.
- Facilita principalmente la creación de gráficos estadísticos, como diagramas de dispersión, diagramas de caja, gráficos de barras, mapas de calor, y gráficos de distribución (histogramas y estimaciones de densidad del núcleo).

