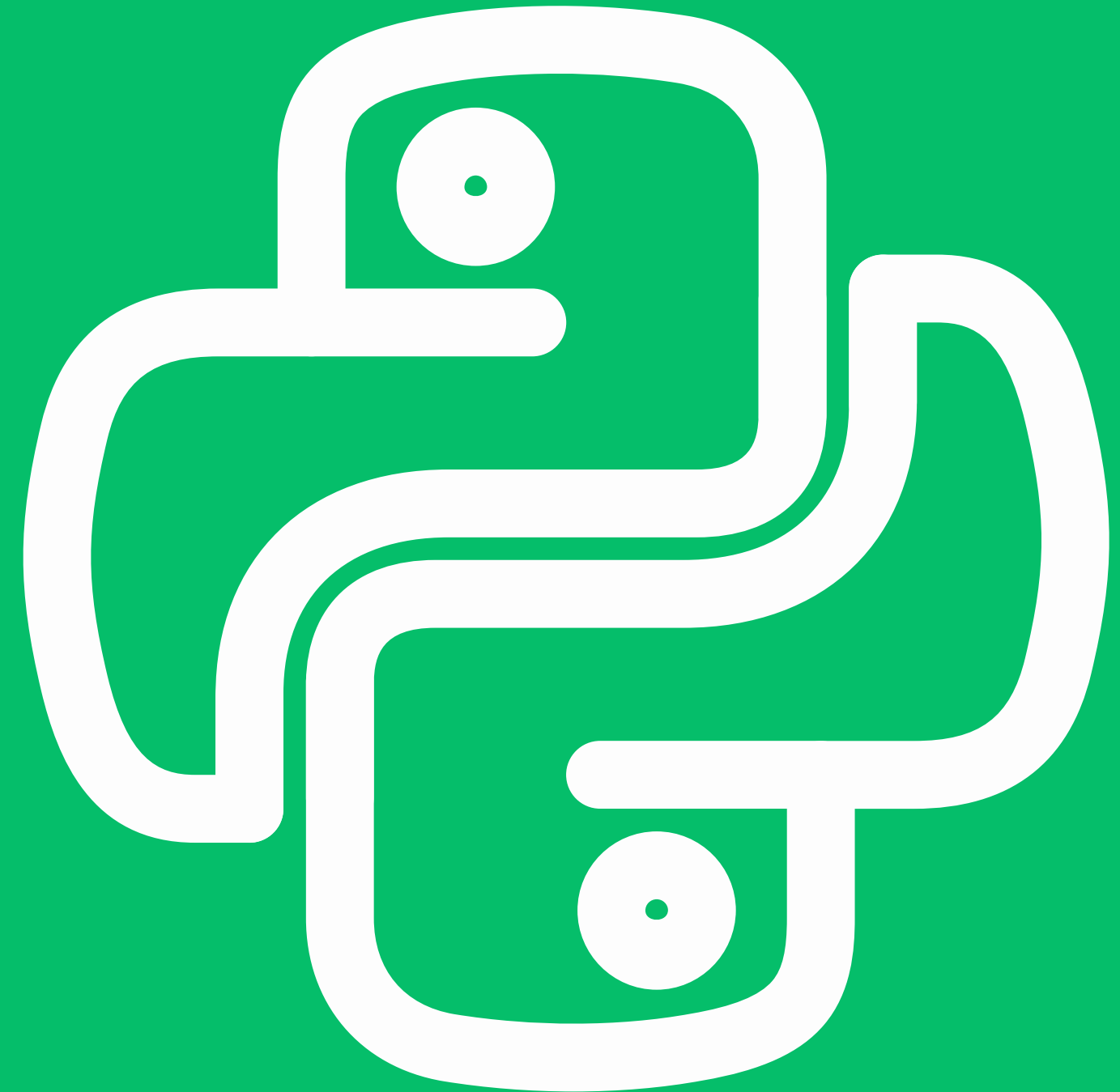


# Archivos



INSTITUTO DE INNOVACIÓN Y  
TECNOLOGÍA APLICADA

I I T A

# Contenido

**01** Introducción

**02** Tipos

**03** ¿Para que sirven?

**04** Abrir y Cerrar

**05** Leer un Archivo

**06** Escribir en un archivo



P Y T H O N

Pregunta Inicial:

¿Qué es un  
archivo?

# ¿Qué es un archivo?



- Un archivo es un lugar donde podemos **guardar información** para que no se pierda, incluso después de **apagar** la computadora.
- Hasta ahora, los datos que usamos en nuestros programas (variables y números) se borraban cuando el programa terminaba.

# ¿Por qué usar archivos?



- Con archivos, podemos guardar esa información en un **medio de almacenamiento** (como un disco duro o una memoria USB).
- Así, más tarde podemos:
  - **Leer** la información.
  - **Escribir** nueva información.
  - **Usar** esa información en otros programas.



P Y T H O N

Otra pregunta:

**¿Diferencia  
entre un archivo  
y un proceso?**

# La diferencia es...

- Un **archivo** permanece guardado incluso si la computadora se apaga. Se usa para almacenar datos como texto, imágenes, música, o programas.
- Mientras que un **proceso** es un programa que **está en ejecución**, es decir, que el sistema operativo está ejecutando activamente. Existe solo mientras se está ejecutando; **desaparece cuando termina**.

# Tipos de Archivos

Podemos dividir los archivos en dos grandes tipos según cómo se almacena su información:

## Archivos binarios

- Contienen información que **no se puede leer directamente** como texto.
- Son útiles para guardar imágenes, videos, programas, y más.

.exe

.jpg

.mp4

.mp3

## Archivos de Texto (ASCII)

- Toda su información **se puede leer como texto** si usamos un esquema de codificación, como ASCII.
- Son ideales para guardar datos simples y comprensibles.

.py

.json

.txt

.xml



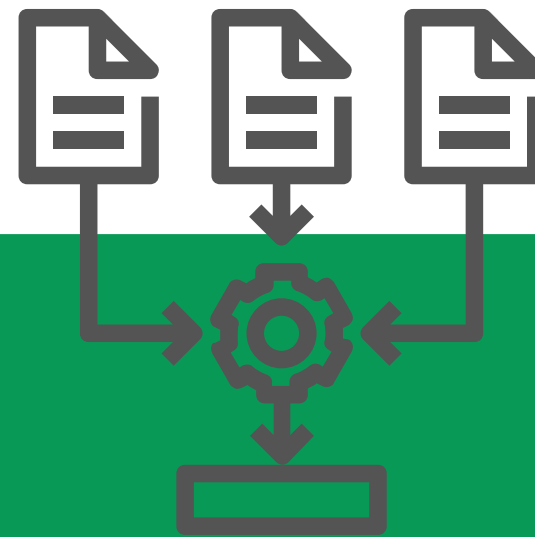
# ¿Para qué sirven?

Además de guardar información, los archivos tienen diversos usos que resultan esenciales en la programación y en el manejo de sistemas. Algunos de ellos son:



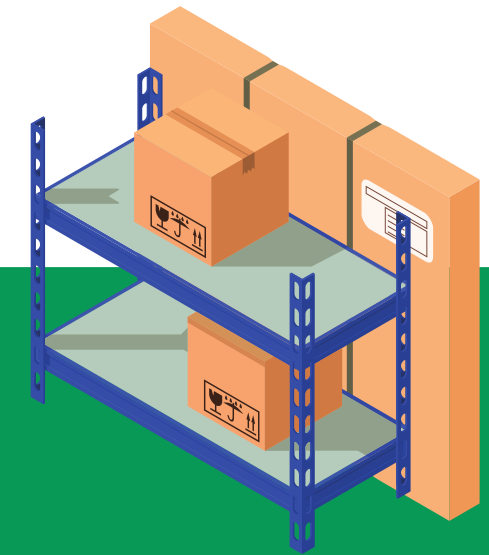
## Comunicación entre Procesos (Pipeline)

Los archivos pueden ser utilizados como una herramienta para que dos procesos distintos se comuniquen.



## Modelado de Información

Ayudan a representar estructuras complejas, como procesos, recursos, conexiones o funciones.



## Persistencia de Datos

Garantizan que la información generada por un programa permanezca disponible incluso después de que el programa termina.

# Abrir y cerrar archivos en python

Para trabajar con archivos en Python usamos la función **open()**. Esta función devuelve un **objeto archivo**, que luego podemos usar para **leer, escribir o modificar su contenido**.

```
file = open(nombre_archivo, modo)
```

referencia al archivo por abrir

nombre completo del archivo.formato

modos de apertura

"r"

"r+"

"w"

"a"



P Y T H O N

Otra pregunta:

¿Cuál es el  
nombre  
completo de un  
archivo?

# Modos de apertura

El segundo parámetro de `open()` indica cómo queremos trabajar con el archivo. Aquí están los principales modos:

- **"r": Read** (sólo lectura)
  - Abre el archivo para leer su contenido.
- **"r+": Read/Write** (lectura y escritura)
  - Permite leer y escribir en el archivo.
- **"a": Append** (agregar información)
  - Abre el archivo para agregar contenido al final sin borrar lo que ya tiene.
- **"w": Write** (escritura)
  - Escribe en el archivo.

P Y T H O N

# Abrir y cerrar archivos en python

Cuando termines de trabajar con un archivo, **debes cerrarlo** para liberar los recursos del sistema asociados al archivo:

```
file.close()
```

# Lectura de archivos en python

Cuando trabajamos con archivos en Python, el objeto file nos ofrece varias formas de leer contenido dependiendo de nuestras necesidades. Aquí están las opciones principales:

**file.read(cantidad)**

Lee una cantidad específica de caracteres (o bytes) del archivo. Si **no se especifica** el argumento **cantidad**, o es negativo, se leerá todo el contenido del archivo.

**file.readline()**

Lee una sola línea del archivo. Si se lo llama varias veces, avanzará línea por línea.

**file.readlines()**

Lee todas las líneas del archivo y las devuelve como una lista de cadenas.

# Escritura de archivos en python

Con Python, podemos escribir en archivos utilizando el objeto file y sus métodos. A continuación, se explican las operaciones más comunes:

**file.write("contenido")**

- Escribe el contenido proporcionado en el archivo.
- Devuelve la cantidad de caracteres escritos.
- El archivo debe abrirse en un modo que permita escritura como 'w', 'r+' o 'a'.

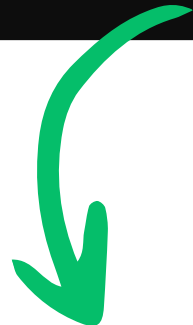
**file.tell()**

- Devuelve un número entero que indica la posición actual del puntero en el archivo.

# Buenas practicas

Trabajar con archivos requiere cuidado para evitar errores y problemas de rendimiento. Aquí te dejo las principales buenas prácticas:

```
with open("example.txt", "w") as file:  
    file.write("Esta es una buena práctica.")  
# Aquí el archivo ya está cerrado automáticamente.
```



Garantiza que los recursos se liberen incluso si ocurre un error dentro del bloque.



**Automáticamente cierra el archivo al salir del bloque.**



01

Escribir un programa que abra un archivo, lea todas sus líneas y cuente cuantas líneas existen en el mismo

02

Utilizar Python para escribir un archivo de texto que tenga 11 líneas, en cada una escribir lo que deseen y cerrar el archivo. Luego mostrar el contenido del archivo.

03

[Escribir una función] que cuente cuantos caracteres existen dentro del archivo creado en el punto anterior

## Ayuda Memoria

```
# abrir archivo
file = open("archivo.txt", "modo")
# leer archivo (todo el contenido)
file.read()
# leer archivo (una línea)
file.readline()
# leer archivo (todas las líneas => list[str])
file.readlines()
# escribir archivo
file.write("contenido")
# cerrar archivo
file.close()
```

04

Escriba un programa que pida al usuario su nombre. Cuando este lo ingrese, muestre un mensaje de bienvenida en la pantalla, y agregue una línea donde registre la visita del usuario a un archivo llamado `libro_invitados.txt`. Asegúrese de que cada registro figure en una nueva línea, y de que cada nueva entrada sea grabada en el mismo archivo, incluso entre múltiples ejecuciones del programa (No hay problema con los repetidos)

05

Escribir un programa que copie los contenidos de un archivo A y los vuelque en un archivo B

## Ayuda Memoria

```
# abrir archivo
file = open("archivo.txt", "modo")
# leer archivo (todo el contenido)
file.read()
# leer archivo (una linea)
file.readline()
# leer archivo (todas las lineas => list[str])
file.readlines()
# escribir archivo
file.write("contenido")
# cerrar archivo
file.close()
```

I I T A

# ¡Muchas Gracias!

Por su atención

