

- QUE ES UN CRUB?
- **Create:** crear o añadir registros.
- **Read:** consultar registros.
- **Update:** modificar datos existentes.
- **Delete:** eliminar registro

Importar librería y definir archivo de trabajo

En este fragmento se importa la librería estándar json, que permite trabajar con archivos en formato JSON. La constante ARCHIVO define el nombre del archivo donde se guardarán los datos de los estudiantes.

```
import json

ARCHIVO = "estudiantes.json"
```

Inicializar datos

La función inicializar abre el archivo JSON y devuelve su contenido como un diccionario. Si el archivo no existe, retorna un diccionario vacío. Esto asegura que siempre haya una estructura de datos lista para usar.

```
def inicializar():
    try:
        with open(ARCHIVO, "r") as f:
            return json.load(f)
    except FileNotFoundError:
        return {}
```

Guardar datos en el archivo

La función guardar recibe un diccionario y lo escribe en el archivo JSON. El parámetro indent=4 hace que el archivo quede legible, con sangrías apropiadas.

```
def guardar(data):
    with open(ARCHIVO, "w") as f:
        json.dump(data, f, indent=4)
```

Crear un nuevo registro (Create)

La función crear añade un nuevo estudiante. Usa el código como clave en el diccionario y guarda un objeto con nombre y edad como valor. Después, guarda la información en el archivo.

```
def crear(codigo, nombre, edad):
    data = inicializar()
    data[codigo] = {"nombre": nombre, "edad": edad}
    guardar(data)
    print("Estudiante creado.")
```

Leer registros (Read)

La función leer obtiene todos los estudiantes almacenados y los muestra recorriendo el diccionario con un ciclo for.

```
def leer():
    data = inicializar()
    for cod, info in data.items():
        print(cod, info)
```

Actualizar registros (Update)

La función actualizar modifica los datos de un estudiante existente. Si se encuentra el código, actualiza el nombre y/o la edad. Si no existe, muestra un mensaje de error.

```
def actualizar(codigo, nombre=None, edad=None):
    data = inicializar()
    if codigo in data:
        if nombre:
            data[codigo]["nombre"] = nombre
        if edad:
            data[codigo]["edad"] = edad
        guardar(data)
        print("Estudiante actualizado.")
    else:
        print("Estudiante no encontrado.")
```

Eliminar registros (Delete)

La función eliminar borra un estudiante según su código. Si el código no existe en los registros, informa que no se encontró.

```
def eliminar(codigo):
    data = inicializar()
    if codigo in data:
        del data[codigo]
        guardar(data)
        print("Estudiante eliminado.")
    else:
        print("Estudiante no encontrado.")
```

Ejecución de pruebas

```
crear("A001", "Ana", 21)
crear("A002", "Luis", 20)
leer()
actualizar("A001", edad=22)
eliminar("A002")
leer()
```

Salida:

```
Estudiante creado.
Estudiante creado.
A001 {'nombre': 'Ana', 'edad': 21}
A002 {'nombre': 'Luis', 'edad': 20}
Estudiante actualizado.
Estudiante eliminado.
A001 {'nombre': 'Ana', 'edad': 22}
```