

# Recuperatorio

## Programación

---

### Examen de Consumo de APIs en Python

**Importante:** La entrega del examen debe realizarse exclusivamente a través de un repositorio de GitHub.

#### Ejercicio 1

Implementa un script en Python que consuma la API provista por JSONPlaceholder para obtener una lista de usuarios en formato JSON.

El script debe comenzar solicitando al usuario que ingrese la cantidad de usuarios (`n`) que desea obtener, con `n > 0` y `n <= 10`. En caso de ingresar valores incorrectos, como valores no numéricos o fuera del rango deseado, se deberá mostrar un mensaje de error y solicitar el ingreso de `n` nuevamente.

Una vez obtenido el valor de `n`, el script debe realizar la solicitud a la API y almacenar los datos de cada usuario en una lista de diccionarios. A continuación:

1. Divide los usuarios en dos grupos: aquellos cuyo nombre comienza con una vocal (a, e, i, o, u) y aquellos cuyo nombre comienza con una consonante.
2. Crea dos archivos JSON en un directorio llamado “UsersData” con los siguientes nombres:
  - `usersVowels.json` para usuarios cuyo nombre comienza con vocal.
  - `usersConsonants.json` para usuarios cuyo nombre comienza con consonante.

#### Requisitos:

- Usa el módulo `requests` para hacer la solicitud a la API.
- Usa el módulo `json` para escribir los datos en formato JSON en los archivos.
- Asegúrate de manejar errores de entrada del usuario y errores de conexión.

## Ejercicio 2

Extiende el script del ejercicio anterior para permitir que el usuario ingrese una ubicación (ciudad o estado) y filtre la lista de usuarios en base a la ubicación ingresada.

1. Crea una función que reciba la lista de usuarios y filtre aquellos cuyo campo "address" contenga la ubicación indicada, ya sea en el campo `city` o `street`.
2. Muestra en consola la cantidad de usuarios que coinciden con la ubicación ingresada.
3. Genera un archivo llamado `filteredUsers.json` en el que almacenes los datos de los usuarios filtrados.

**Ejemplo de uso:** Si el usuario ingresa "`South Christy`" como ubicación, el script debe buscar usuarios con esta ubicación y mostrar el conteo de coincidencias en consola.

### Requisitos:

- Usa listas y diccionarios para almacenar y filtrar los datos.
- Muestra un mensaje de error en caso de que no haya coincidencias.
- Guarda los resultados en el archivo `filteredUsers.json` en el directorio `UserData`.

---

## Ejercicio 3

Implementa una función en el mismo script que permita realizar una búsqueda avanzada en los posts de JSONPlaceholder en base a un usuario y una palabra clave.

El flujo del ejercicio es el siguiente:

1. Solicita al usuario el ID de usuario (un número entre 1 y 10) y verifica que sea válido.
2. Solicita una palabra clave para buscar en el contenido de los posts.
3. Filtra los posts del usuario indicado y cuenta la cantidad de veces que la palabra clave aparece en el título y el cuerpo de cada post.
4. Muestra en la consola:
  - El `ID` de cada post que contiene la palabra clave.
  - La cantidad de veces que la palabra clave aparece en cada post.
  - El total de apariciones de la palabra clave en todos los posts del usuario.

**Ejemplo de salida:** Si el usuario ingresa el ID 2 y la palabra clave "vida", la salida debería mostrar algo como:

Post ID      Ocurrencias de "vida"

1                2

3                1

Total de ocurrencias de "vida": 3 veces

#### Requisitos:

- Usa `requests` para obtener los datos de los posts.
- Usa diccionarios para organizar las ocurrencias de la palabra clave en cada post.
- Muestra un mensaje de error si no hay coincidencias con la palabra clave o si el ID del usuario no es válido.