

# EJERCICIOS DE FILES EN PYTHON

link de GITHUB: [https://github.com/villacis-cristian/ejercicios\\_en\\_clase\\_python.git](https://github.com/villacis-cristian/ejercicios_en_clase_python.git)

CRISTIAN ANDRES VILLACIS MENDOZA

## DESARROLLO DE SOFTWARE

1. Escribe una función que reciba una ruta de un fichero de texto y una cadena de caracteres a buscar y determine si la cadena aparece en el fichero.
2. Escribe una función que reciba una lista, una ruta destino y un número  $n$ . La función debe crear un fichero en la ruta especificada. El contenido del fichero serán los primeros  $n$  elementos de la lista. La función debe controlar de manera apropiada los posibles valores de  $n$  que estén fuera de rango.
3. Escribe una función que reciba una ruta de un fichero de texto devuelva un diccionario con la frecuencia de aparición de cada palabra. Ejemplo: un fichero que contenga la frase 'es mejor que venga que que no venga' devolverá el siguiente diccionario: {'es' : 1, 'mejor' : 1, 'que' : 3, 'venga' : 2, 'no' : 1}. Para dividir un string en palabras puedes hacer uso del método *split*.

## EJERCICIO 1

```
In [2]: def buscar_cadena_en_archivo(ruta_archivo, cadena_buscar):
    try:
        with open(ruta_archivo, 'r', encoding='utf-8') as archivo:
            contenido = archivo.read()
            return cadena_buscar in contenido # Devuelve True si la cadena está, False si no
    except FileNotFoundError:
        print("Archivo no encontrado.")
        return False
    except Exception as e:
        print(f"Error inesperado: {e}")
        return False
```

## EJERCICIO 2

```
In [ ]: def guardar_primeros_n(ruta_destino, lista, n):
    try:
        if n < 0:
            raise ValueError("n no puede ser negativo")
        elementos = lista[:n] # Corta la Lista hasta n, aunque n esté fuera del rango
        with open(ruta_destino, 'w', encoding='utf-8') as archivo:
            for item in elementos:
                archivo.write(f"{item}\n") # Escribe cada elemento en una línea
        print("Archivo creado con éxito.")
    except Exception as e:
        print(f"Error al guardar el archivo: {e}")
```

## EJERCICIO 3

```
In [ ]: def contar_palabras_en_archivo(ruta_archivo):
    try:
        with open(ruta_archivo, 'r', encoding='utf-8') as archivo:
            texto = archivo.read()
            palabras = texto.split() # Divide el texto en palabras usando espacios
            frecuencia = {}
            for palabra in palabras:
                palabra = palabra.strip(".,:;!¿?i()[]\'"') # Limpia signos de puntuación
                palabra = palabra.lower()
                frecuencia[palabra] = frecuencia.get(palabra, 0) + 1
            return frecuencia
    except FileNotFoundError:
        print("Archivo no encontrado.")
        return {}
    except Exception as e:
        print(f"Error inesperado: {e}")
        return {}
```