

# **EJERCICIOS PYTHON AVANZADO**

## **HOJA 2**

### **TEMA: MANEJO DE ARCHIVOS Y EXCEPCIONES**

#### **SUMA Y VALUEERROR**

Un problema común al solicitar una entrada numérica ocurre cuando las personas ingresan texto en lugar de números. Cuando intentas convertir la entrada a un entero (int), obtendrás un ValueError. Escribe un programa que solicite dos números. Suma los números y muestra el resultado. Captura el ValueError si alguno de los valores de entrada no es un número e imprime un mensaje de error amigable. Prueba tu programa ingresando dos números y luego ingresando texto en lugar de un número. Envuelve tu código del en un bucle while para que el usuario pueda continuar ingresando números incluso si comete un error ingresando texto en lugar de un número.

#### **MANIPULACION DE ARCHIVOS Y FILENOTFOUNDERROR**

Crea dos archivos, cats.txt y dogs.txt. Almacena al menos tres nombres de gatos en el primer archivo y tres nombres de perros en el segundo archivo. Escribe un programa que intente leer estos archivos e imprima el contenido de cada archivo en la pantalla. Envuelve tu código en un bloque try-except para capturar el error de FileNotFoundError, e imprime un mensaje amigable si falta algún archivo. Mueve uno de los archivos a una ubicación diferente en tu sistema y asegúrate de que el código en el bloque except se ejecute correctamente. Modifica tu bloque except para que falle en silencio si falta alguno de los archivos.

#### **PALABRAS COMUNES**

Encuentra o crea algunos textos que te gustaría analizar (puedes visitar Project Gutenberg (<http://gutenberg.org/>) o crear textos usando ChatGPT). Copia el texto sin formato desde tu navegador en un archivo de texto en tu computadora (o descarga los archivos). Averigua cuántas veces aparece una palabra o frase en el texto (puedes usar el método count()).

## BUSCANDO EN PI

Busca si tu fecha de nacimiento esta en los primeros 10000 digitos de pi (y en que posición. Puedes usar `find()`).  
Puedes usar el archivo `pi_10000.txt`

## NUMERO FAVORITO

Escribe un programa que solicite al usuario su número favorito. Utiliza `json.dump()` para almacenar este número en un archivo. Escribe un programa separado que lea este valor e imprima el mensaje: "Sé cuál es tu número favorito... Es \_\_\_\_." Combina ambos programas en un solo archivo (puedes crear tantas funciones como necesites). Si el número ya está almacenado, muestra el número favorito al usuario. Si no lo está, solicita al usuario su número favorito y guárdalo en un archivo. Ejecuta el programa al menos dos veces para asegurarte de que funciona correctamente.

## REFACTORIZACIÓN:

Revisa los ejercicios del modulo "Python para Principiantes". ¿Hay algún ejercicio que pudiese dividirse en funciones? ¿Y alguno que podría optimizarse usando bloques `try-except`? Si es así reescríbelos usando estas estructuras.