TEST CICLO 1 – GRUPO C4

Título del ejercicio	**Test 2 del ciclo 1 – Grupo C4**
Eje temático	Algoritmia Avanzado
Módulos	Manejo de excepciones y errores en Python. Prácticas de manejo de errores. Lectura y escritura de archivos en Python. Trabajo con diferentes formatos (texto, CSV, JSON).

Tiempo estimado	3 horas
Nivel de dificultad	Básico
Objetivos	Evaluar a través de un caso de prueba las habilidades, capacidades y competencias de los contenidos del ciclo 1 (estructuras de control, modularidad y estructura de datos).

ESCENARIO/PLANTEAMIENTO DEL PROBLEMA

Situación problema
CALCULO DE SPAM EN EL SERVIDOR DE CORREO

La empresa ACME desea saber cual es su indice promedio de SPAM en su servidor de correo. Para calcular se le pide construir un programa que lea el archivo y calcule el promedio de SPAM de todo el servidor. El indicador de SPAM por cada correo se encuentra en el parámetro X-DSPAM-Confidence. Este programa será ejecutado por el administrador de infraestructura por lo tanto el programa debe pedir el nombre del archivo que se va a leer.

Ejemplo:

```
Enter the file name: mbox.txt
Average spam confidence: 0.894128046745

Enter the file name: mbox-short.txt
Average spam confidence: 0.750718518519
```

TIPS DE SOLUCION

- Analice el problema: Lea detenidamente y anote los puntos clave
- Diseñe la solución:
- Desarrolle la solución:
- Pruebe su solución

RUBRICA DE CALIFICACIÓN

Criterio técnico de evaluación
Valor: 20 / 100
Diseño y usabilidad de la aplicación • Diseño visual atractivo y coherente. • Estructura de navegación clara y fácil de usar.
Valor: 30 / 100
Estructura de datos • Estructura que gestione los datos • Mantenibilidad de la estructura (mantenerla durante la vida del programa)
Valor: 40 / 100
Funcionalidad y características técnicas • Desarrollo de la solución de manera modular (funciones) • Evitar el uso de variables globales. Paso de estructura por parámetros a funciones • Organización del código. Código limpio, organizado y legible, estructurado. Uso de una convención en el nombramiento de las variables, funciones y estructura de datos. • Eficiencia y performance de algoritmos • Validación de entrada del usuario y gestión de errores mediante *try / except*