

INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES DE MONTERREY

Maestría en Inteligencia Artificial Aplicada

TC4017.10 Pruebas de software y aseguramiento de la calidad (Gpo 10)

4.2 Ejercicio de programación 1



Alberto Nieves Cisneros

A01793829

Introducción

En el ámbito del desarrollo de software, la adhesión a estándares de codificación reconocidos y la implementación de buenas prácticas son fundamentales para asegurar la calidad, mantenibilidad y escalabilidad de los sistemas. Esta tarea se centra en la importancia del estilo de codificación, reconociendo atributos de estándares de codificación útiles para identificar errores y explorando estándares de codificación reconocidos en la industria junto con sus implicaciones. A través de la implementación de tres ejercicios de programación en Python, siguiendo el estándar PEP-8 y utilizando herramientas como PyLint para la verificación de código, esta tarea busca no solo reforzar las habilidades de programación sino también profundizar en la comprensión y aplicación de prácticas de codificación que son esenciales en el desarrollo de software profesional.

Compute statistics

El script `computeStatistics.py` está diseñado para calcular estadísticas básicas de un conjunto de números extraídos de un archivo de texto. Este proceso incluye la lectura de números del archivo, seguido del cálculo de la media, mediana, moda, varianza y desviación estándar de los números leídos. El script maneja datos no válidos de forma elegante, ignorando cualquier entrada que no pueda convertirse en un número flotante.

Converter

El programa `convertNumbers.py` está diseñado para leer números de un archivo de entrada, convertirlos a formatos binario y hexadecimal, y luego escribir los resultados en un archivo llamado `ConversionResults.txt`. Los números que no se pueden convertir debido a datos no válidos se registran en un archivo `InvalidData.log`. El programa también mide y muestra el tiempo que tarda en completar la operación.

Count Words

El programa `wordCount.py` analiza un archivo de texto, cuenta la frecuencia de cada palabra y escribe los resultados en un archivo llamado `WordCountResults.txt`. Además, calcula y muestra el total de palabras contadas. Este script maneja errores comunes como archivos no encontrados y otros errores de lectura de archivo, asegurando que el usuario reciba retroalimentación adecuada en caso de problemas.

Conclusión

Después de esta práctica, hemos adquirido un entendimiento profundo sobre la relevancia de los estándares de codificación en el desarrollo de software. La aplicación meticulosa de PEP-8 y el análisis riguroso mediante PyLint han reforzado la importancia de un código limpio y bien estructurado, no solo para la mejora de la legibilidad sino también para la eficiencia y la mantenibilidad del software. Esta experiencia subraya que la excelencia en la programación se logra mediante la adhesión disciplinada a las mejores prácticas y normativas establecidas en el sector.