Pruebas de software y aseguramiento de la calidad
Erick de Jesus Hernandez Cerecedo
A01066428

México, 28 de enero de 2024

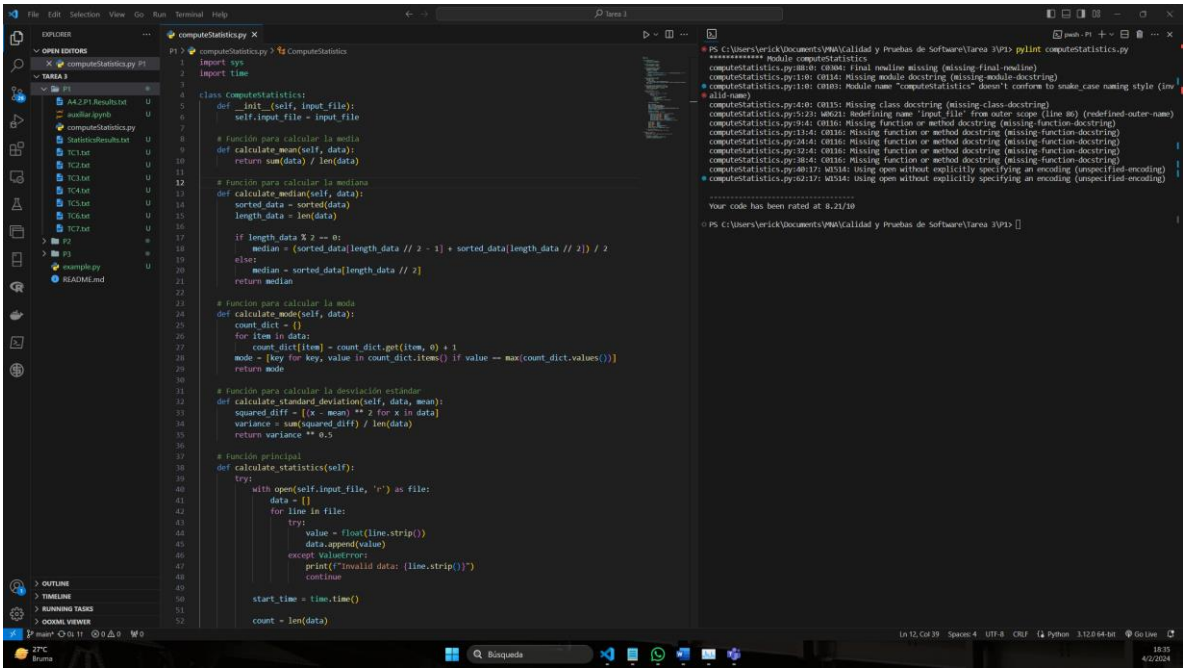
Actividad 4.2 | Ejercicio de programación 1

GitHub: https://github.com/ErickHCerecedo/A01066428_A4.2

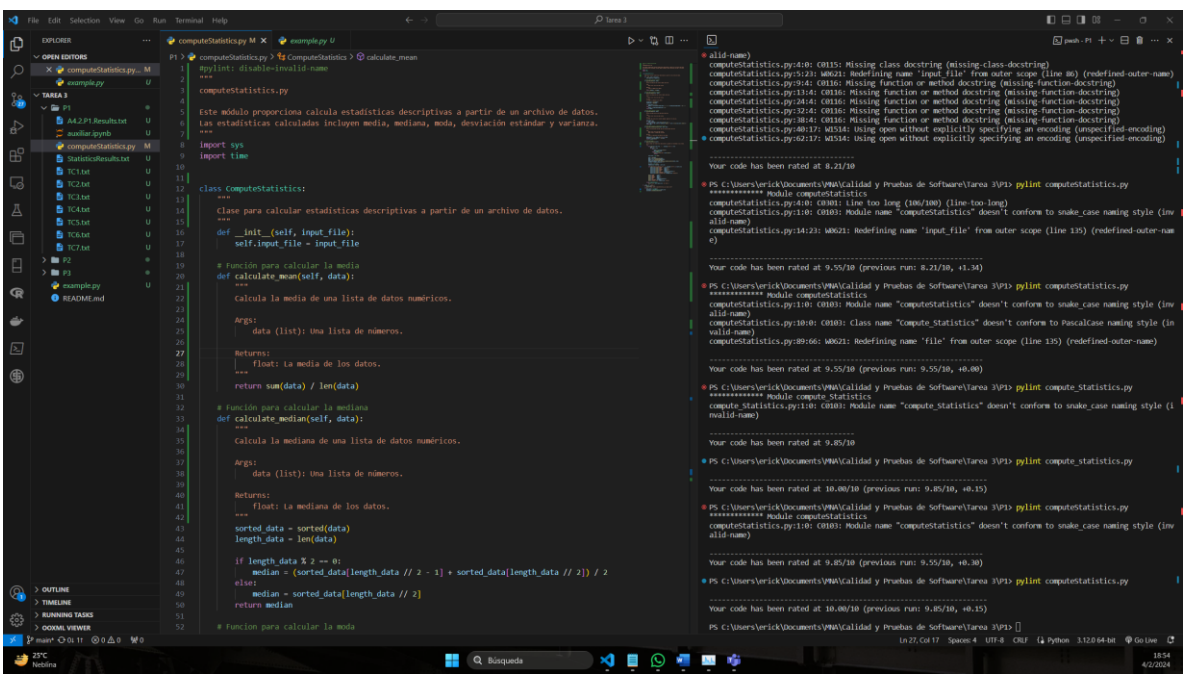
En la siguiente actividad empleando Python se da solución a 3 problemas de programación siguiendo las pautas del estándar de programación PEP-8, una de las herramientas empleadas para el análisis de redacción y estilo es PyLint para asegurar que el programa no genere errores o problemas, en este reporte se encuentra documentado el proceso de creación y corrección de cada uno de los problemas.

Problem 1: Compute Statistics	
Description	Requirement 1. The program shall be invoked from a command line. The program shall receive a file as parameter. The file will contain a list of items (presumable numbers). Requirement 2. The program shall compute all descriptive statistics from a file containing numbers. The results shall be print on a screen and on a file named StatisticsResults.txt. **All computation MUST be calculated using the basic algorithms, not functions or libraries.** The descriptive statistics are mean, median, mode, standard deviation, and variance. Requirement 3. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue. Requirement 4. The name of the program shall be computeStatistics.py Requirement 5. The minimum format to invoke the program shall be as follows: python computeStatistics.py fileWithData.txt Requirement 6. The program shall manage files having from hundreds of items to thousands of items. Requirement 7. The program should include at the end of the execution the time elapsed for the execution and calculus of the data. This number shall be included in the results file and on the screen. Requirement 8. Be compliant with PEP8.

1. Implementación del problema 1 y primera revisión usando PyLint:

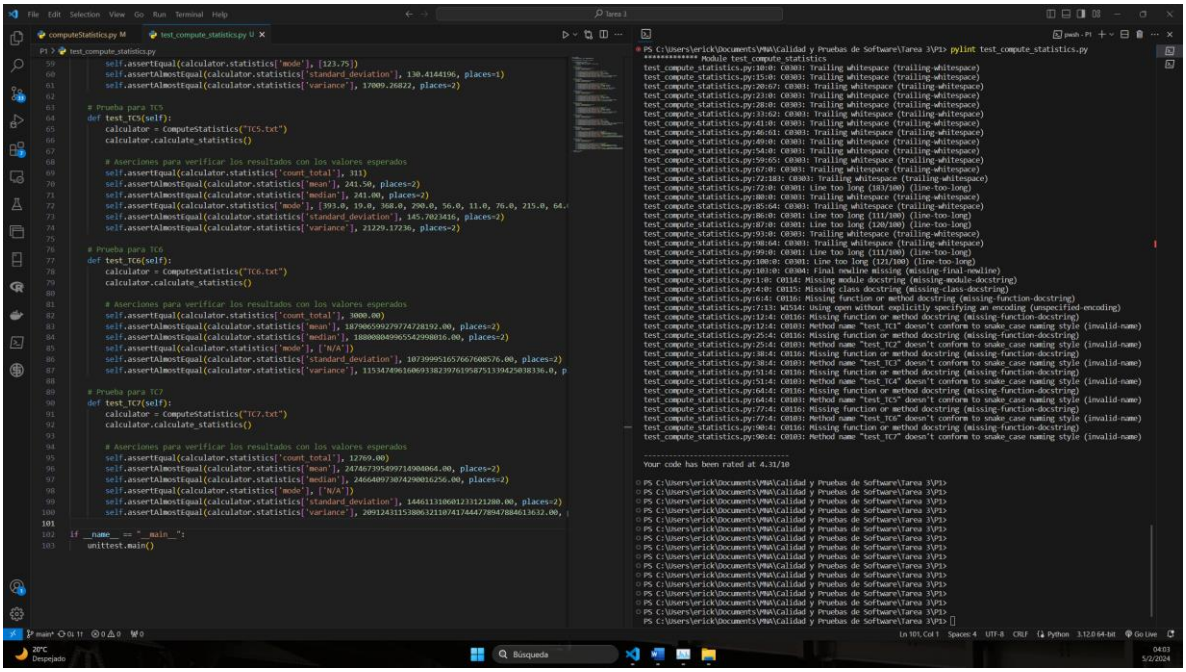


2. Corrección del código usando PyLint:

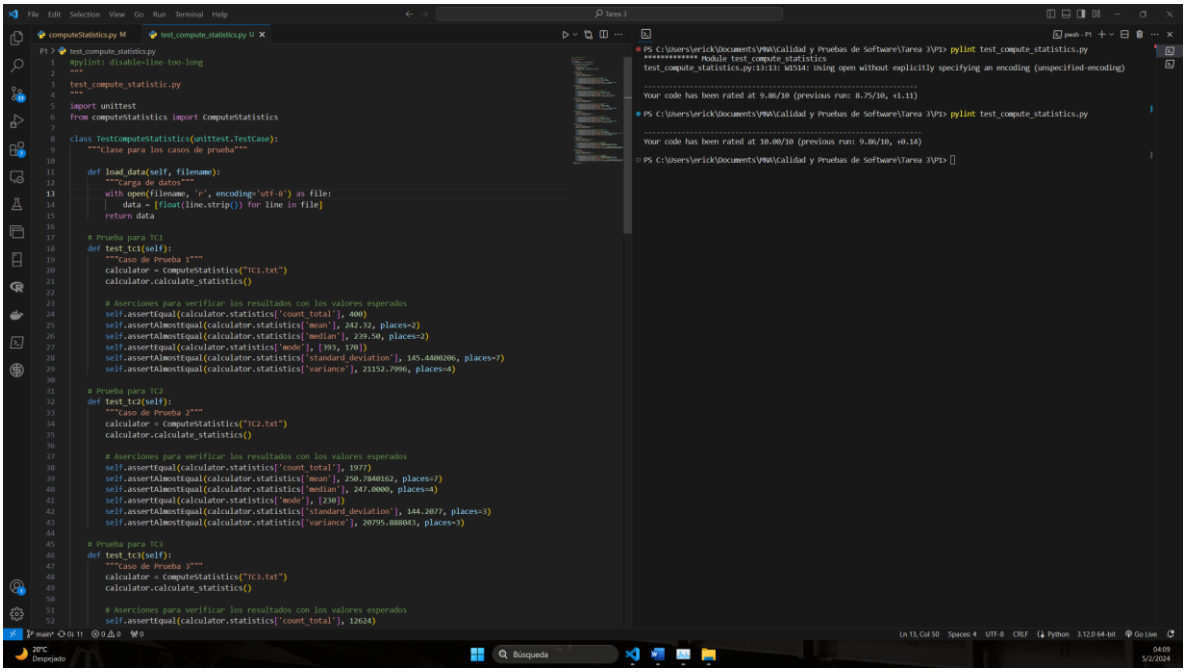


Nota * El nombre del archivo computeStatistics.py no cumple con el estándar snake_case para nombrar archivos, pero debido a ser un requerimiento se usó el comando “disable” de PyLint para ignorar esta advertencia.

3. Creación un archivo de pruebas unitarias:

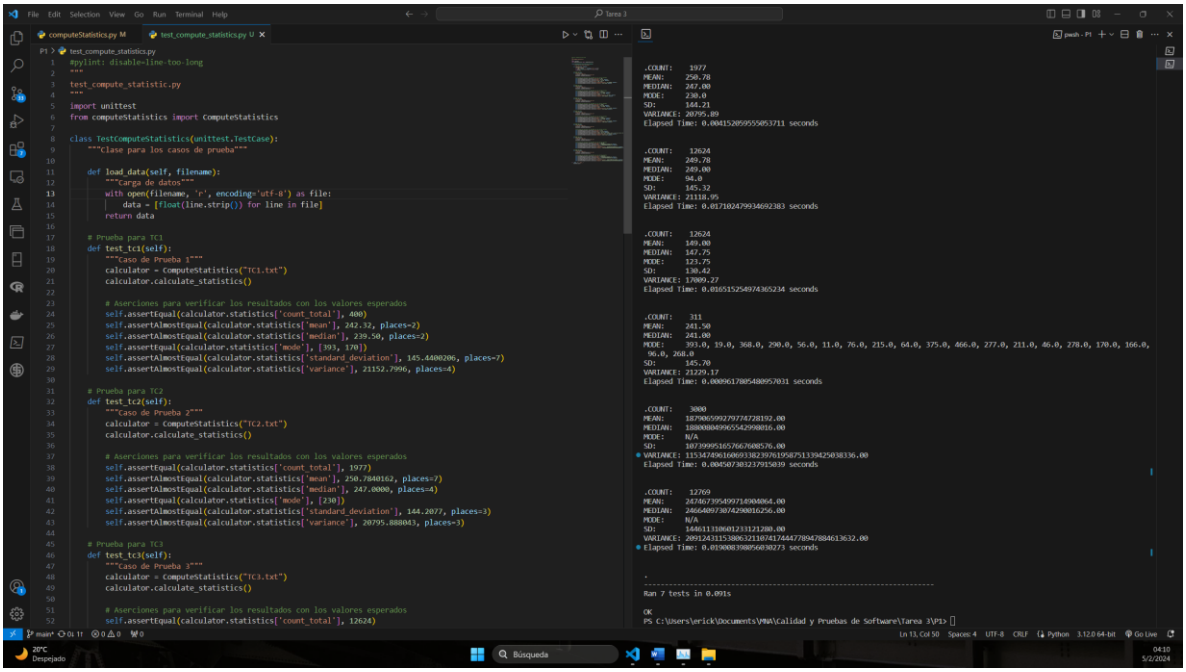


4. Corrección de código usando PyLint:



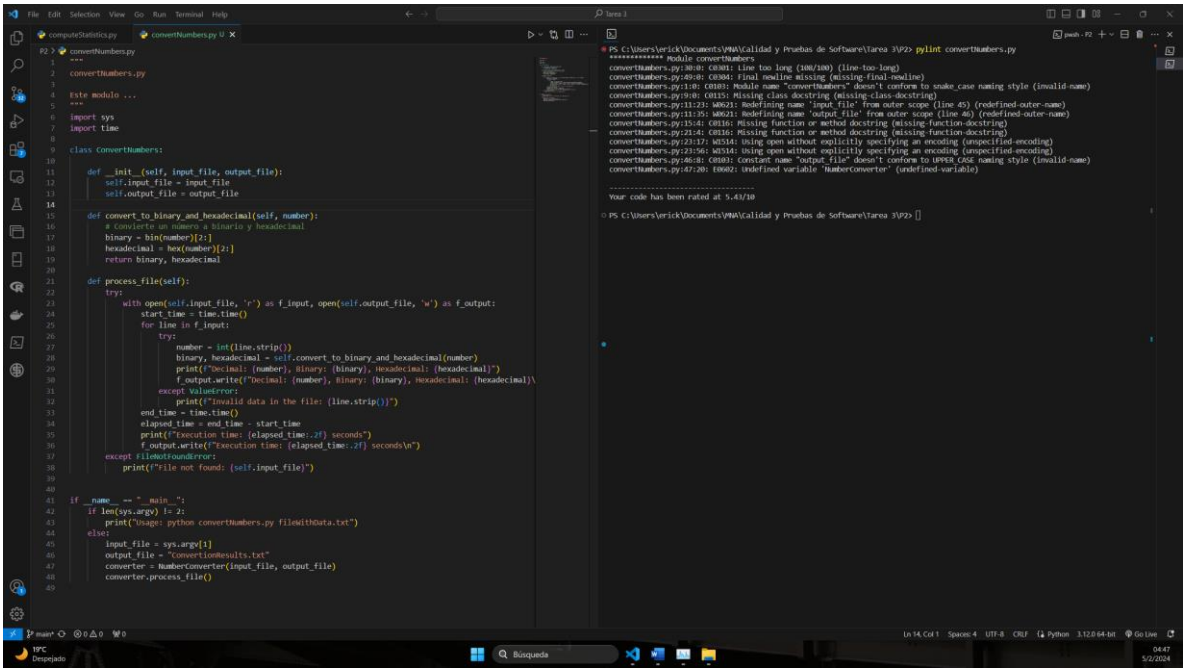
Nota* algunos de los valores proporcionados en el archivo de resultados contenían errores, ya que algunos conjuntos de datos eran multimodales (Contienen más de un valor en moda), los resultados contenidos en casos de pruebas fueron comprobados en otra herramienta de análisis de datos.

5. Casos de prueba aprobados:

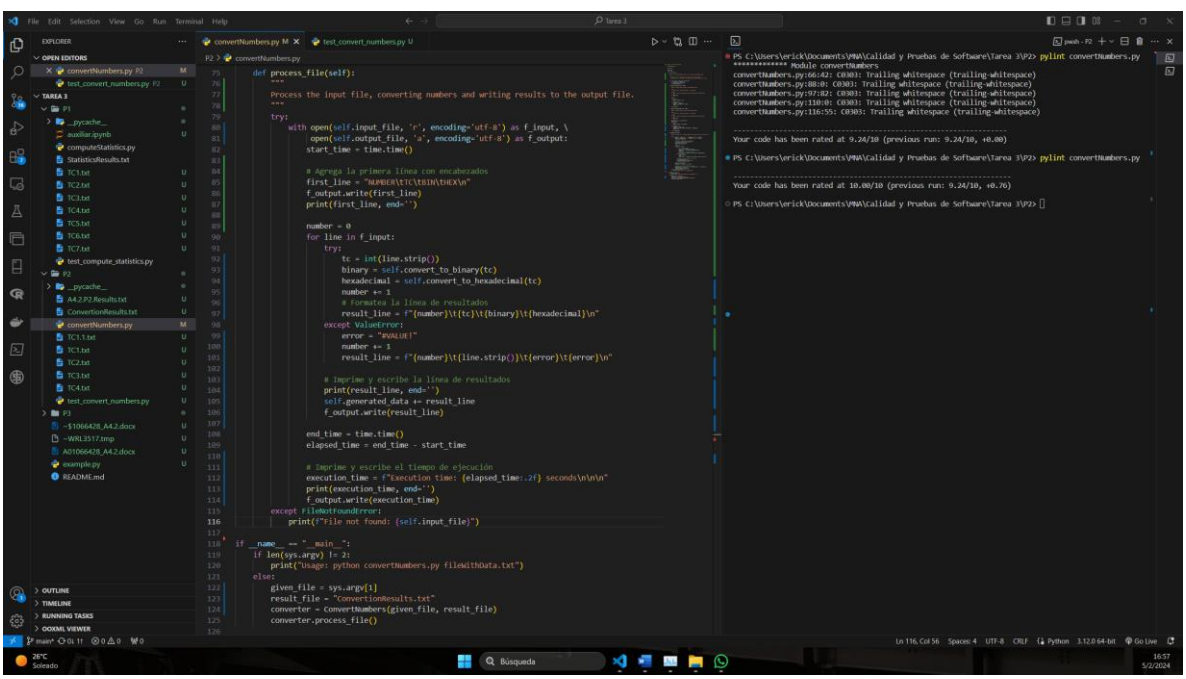


Problem 2: Converter	
Description	Requirement 1. The program shall be invoked from a command line. The program shall receive a file as parameter. The file will contain a list of items (presumable numbers). Requirement 2. The program shall convert the numbers to binary and hexadecimal base. The results shall be print on a screen and on a file named ConvertionResults.txt. **All computation MUST be calculated using the basic algorithms, not functions or libraries.** Requirement 3. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue. Requirement 4. The name of the program shall be convertNumbers.py Requirement 5. The minimum format to invoke the program shall be as follows: python computeStatistics.py fileWithData.txt Requirement 6. The program shall manage files having from hundreds of items to thousands of items. Requirement 7. The program should include at the end of the execution the time elapsed for the execution and calculus of the data. This number shall be included in the results file and on the screen. Requirement 8. Be compliant with PEP8.

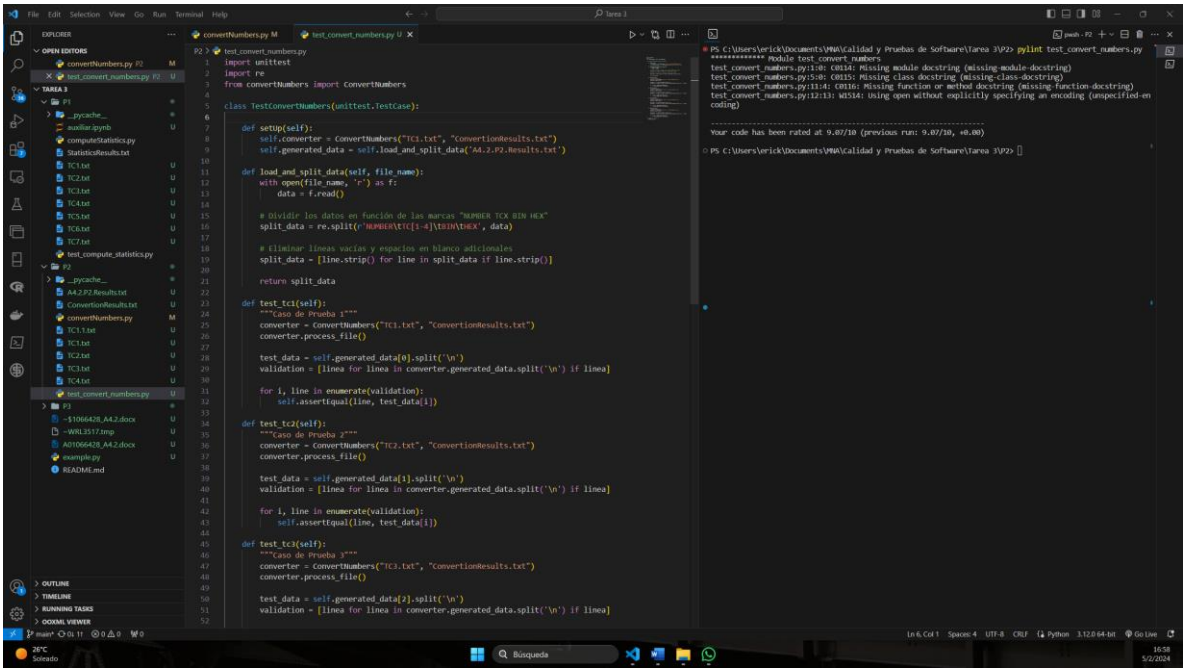
1. Implementación del problema 2 y primera revisión usando PyLint:



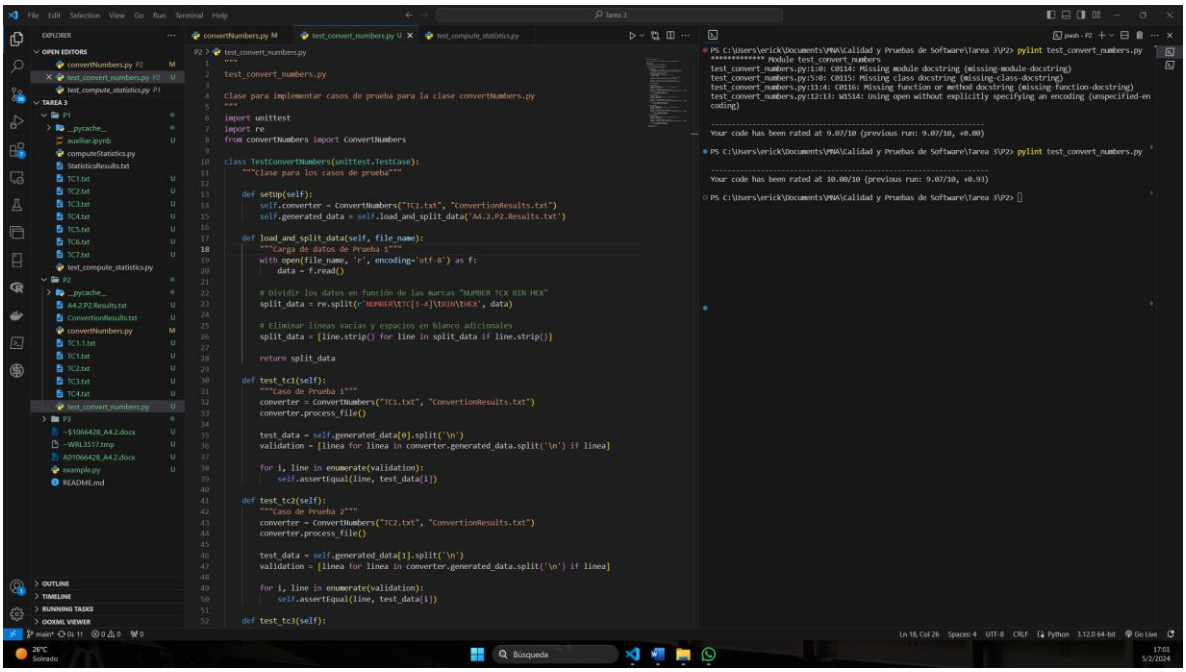
2. Corrección del código usando PyLint:



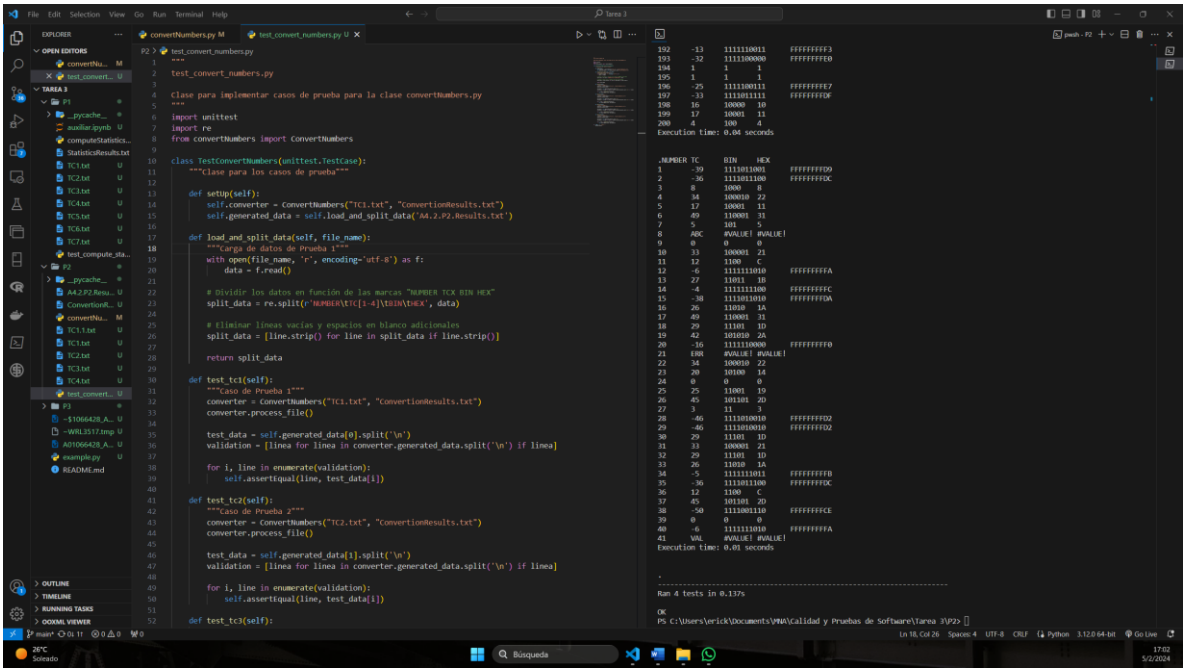
3. Creación un archivo de pruebas unitarias:



4. Corrección de código usando PyLint:



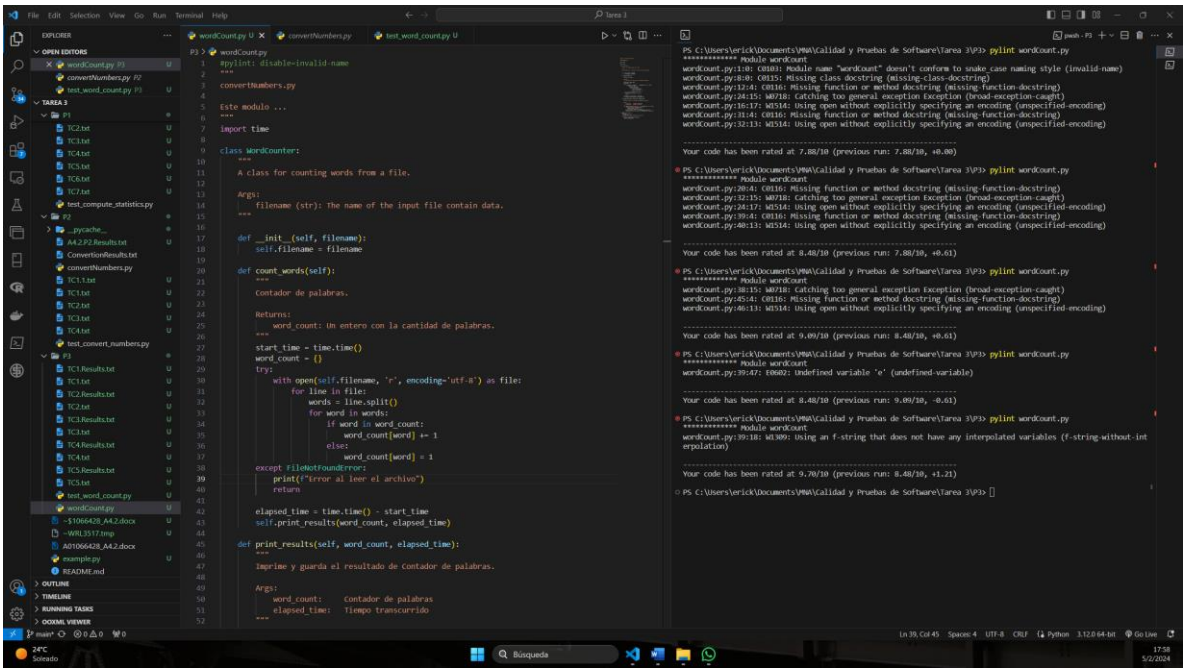
5. Casos de prueba aprobados:



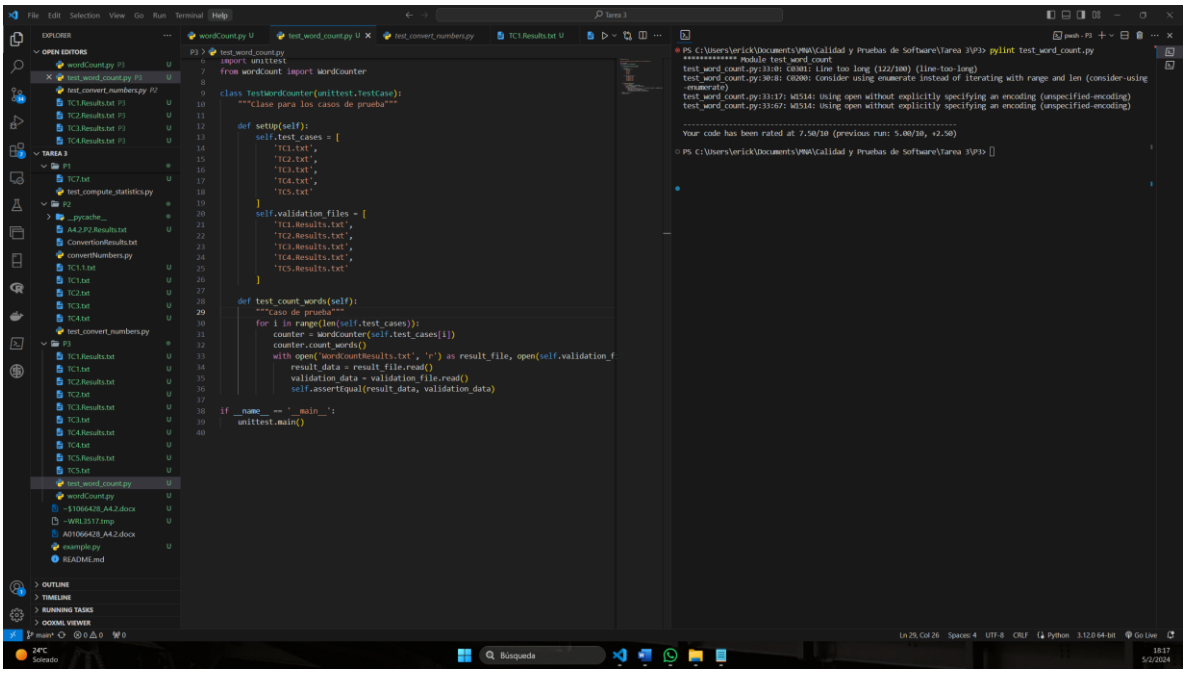
Nota* El archivo de resultados presentaba algunos errores, el archivo de datos TC1.txt no coincide con los datos en el archivo de resultados, los datos manejados como negativos están convertidos con una mascara a 2 bits, 10bits y 32bits, fue más complicado y tardado encontrar estos patrones que completar la actividad.

Problem 3: Count Words	
Description	Requirement 1. The program shall be invoked from a command line. The program shall receive a file as parameter. The file will contain a word (presumable between spaces). Requirement 2. The program shall identify all distinct words and the frequency of them (how many times the word "X" appears in the file). The results shall be print on a screen and on a file named WordCountResults.txt. **All computation MUST be calculated using the basic algorithms, not functions or libraries.** Requirement 3. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue. Requirement 4. The name of the program shall be wordCount.py Requirement 5. The minimum format to invoke the program shall be as follows: python wordCount.py fileWithData.txt Requirement 6. The program shall manage files having from hundreds of items to thousands of items. Requirement 7. The program should include at the end of the execution the time elapsed for the execution and calculus of the data. This number shall be included in the results file and on the screen. Requirement 8. Be compliant with PEP8.

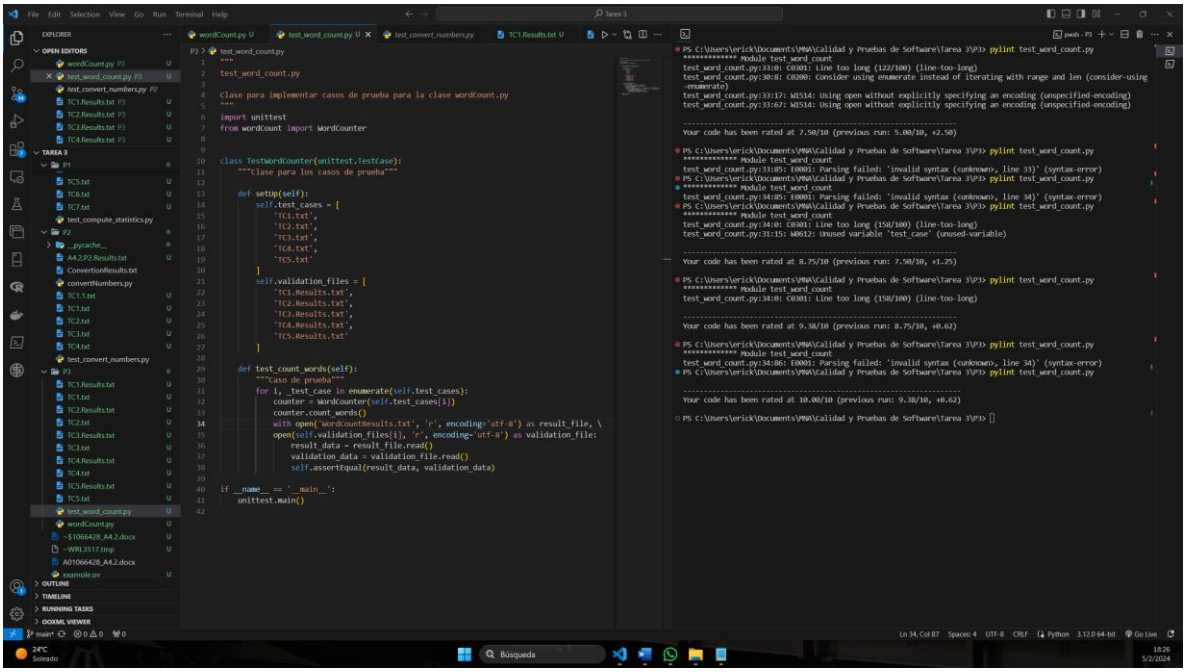
1. Implementación y corrección del problema 3 y primera revisión usando PyLint:



2. Creación un archivo de pruebas unitarias:



3. Corrección de código usando PyLint:



4. Casos de prueba aprobados: