Pruebas de software y aseguramiento de la calidad

Erick de Jesus Hernandez Cerecedo

A01066428

México, 11 de febrero de 2024

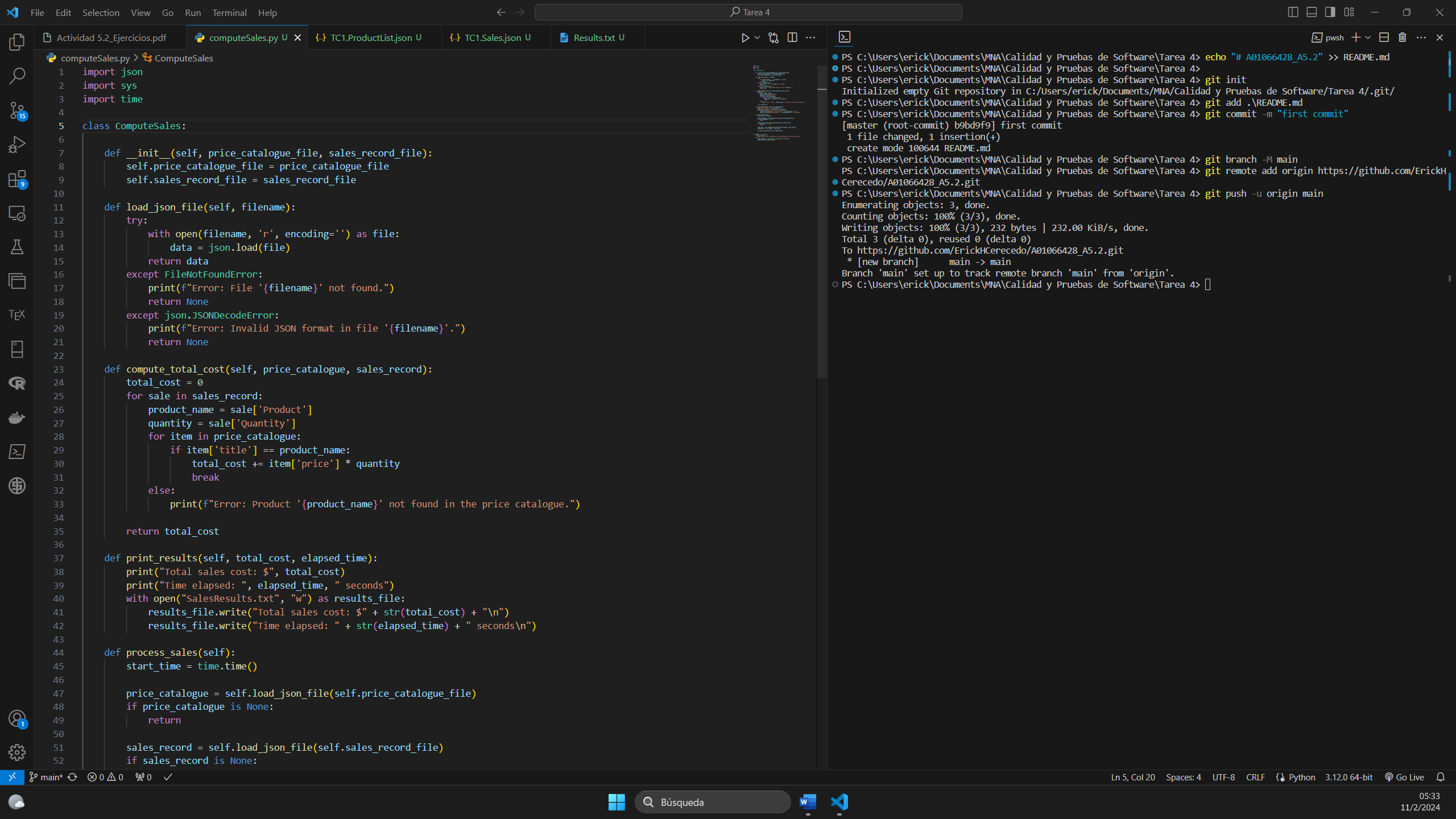
Actividad 5.2 | Ejercicio de programación 2

Enlace al repositorio de GitHub: <https://github.com/ErickHCerecedo/A01066428_A5.2>

El siguiente archivo contiene la solución implementada en Python al ejercicio de programación 2, en el que se reporta el proceso de creación del ejercicio desde la ejecución de revisión de gramática y estilo usando PyLint, análisis de errores empleando Flake y por último la ejecución de pruebas unitarias.

|  |  |
| --- | --- |
| **Problem 1: Compute Sales** | |
| Description | Requirement 1. The program shall be invoked from a command line. The program shall receive two files as parameters. The first file will contain information in a JSON format about a catalogue of prices of products. The second file will contain a record for all sales in a company.  Requirement 2. The program shall compute the total cost for all sales included in the second JSON archive. The results shall be print on a screen and on a file named SalesResults.txt. The total cost should include all items in the sale considering the cost for every item in the first file.  The output must be human readable, so make it easy to read for the user.  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 computeSales.py  Requirement 5. The minimum format to invoke the program shall be as follows:  python computeSales.py priceCatalogue.json salesRecord.json  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 inicial del código:



1. Análisis de errores de PyLint (PEP8):

Captura de pantalla de computadora

Descripción generada automáticamente

Corrección de errores:

Texto

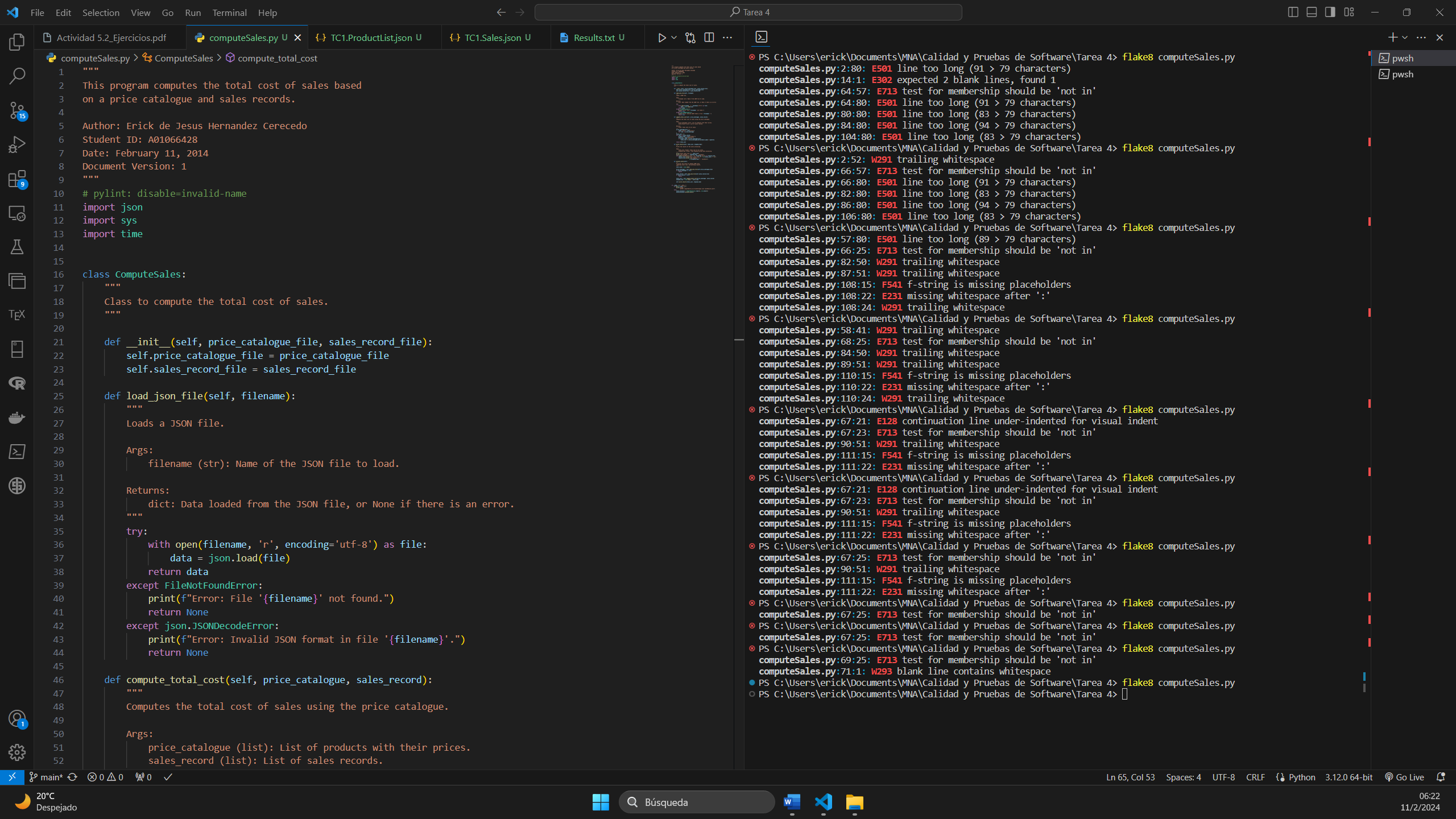
Descripción generada automáticamente

1. Análisis de Errores Flake:

Texto

Descripción generada automáticamente

Corrección de errores con Flake8



1. Pruebas unitarias pasadas correctamente:

Texto

Descripción generada automáticamente