Juan Alejandro Guadalupe Rosas

[Dirección de correo electrónico]

Autor:

Juan Guadalupe

ACCEPTANCE TESTING workshop

Ingeniería de Software II

Contenido

[Repository 1](#_Toc155210978)

[Introduction 1](#_Toc155210979)

[Developement 1](#_Toc155210980)

[Conclutions and recomendations 7](#_Toc155210981)

# Repository

<https://github.com/rexman10/Taller6-IS>

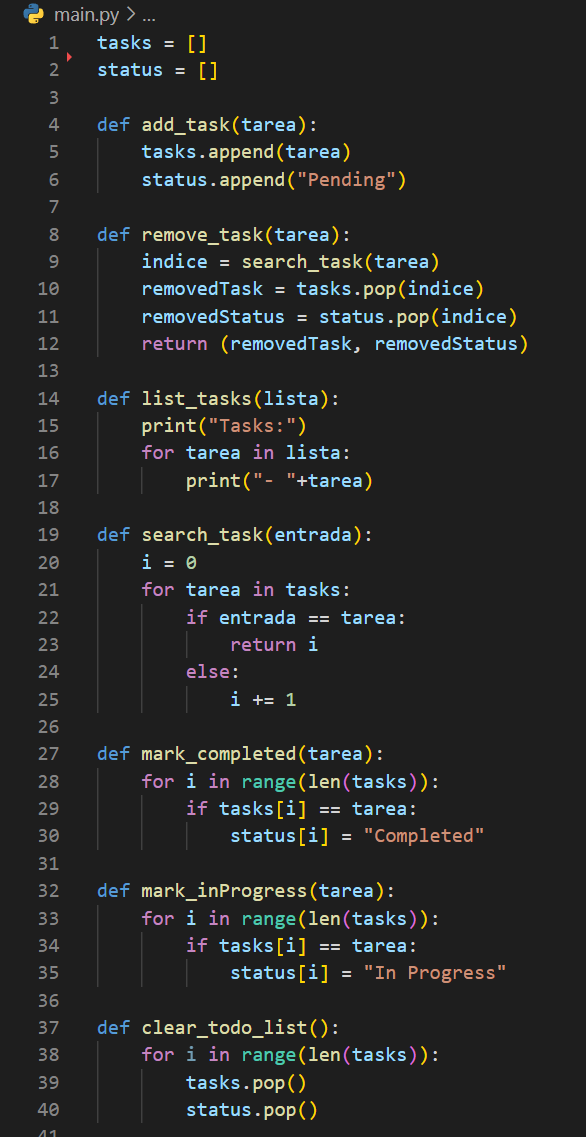
# Introduction

The workshop report is all about this neat Python tool that helps handle to-do lists using the console. We taught people how to do stuff like add, delete, and update tasks with simple commands. Everybody got hands-on with Python scripting, and we made to-do list management really simple. The report tells the story of how people from different backgrounds learned to use the console and Python commands to make their task management super easy. It shares what we all learned together, how we used it in real situations, and the practical skills we gained in a straightforward and easy-to-understand way.

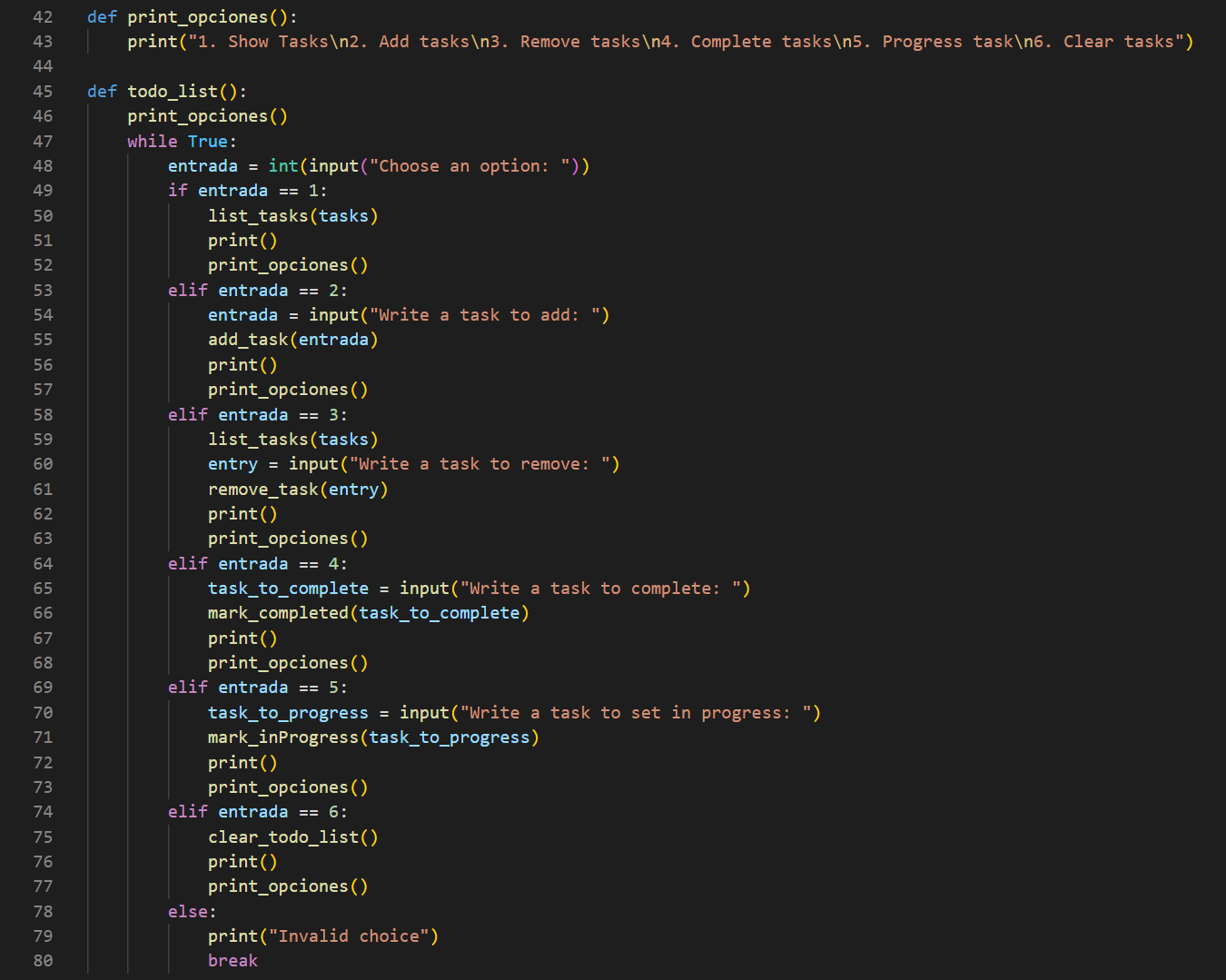
# Developement

The development of this program consists on a series of functions that work together to complete the functionality of the to-do list.

To begin the process it’s needed to write the core functions to be able to manipulate the tasks of the list.

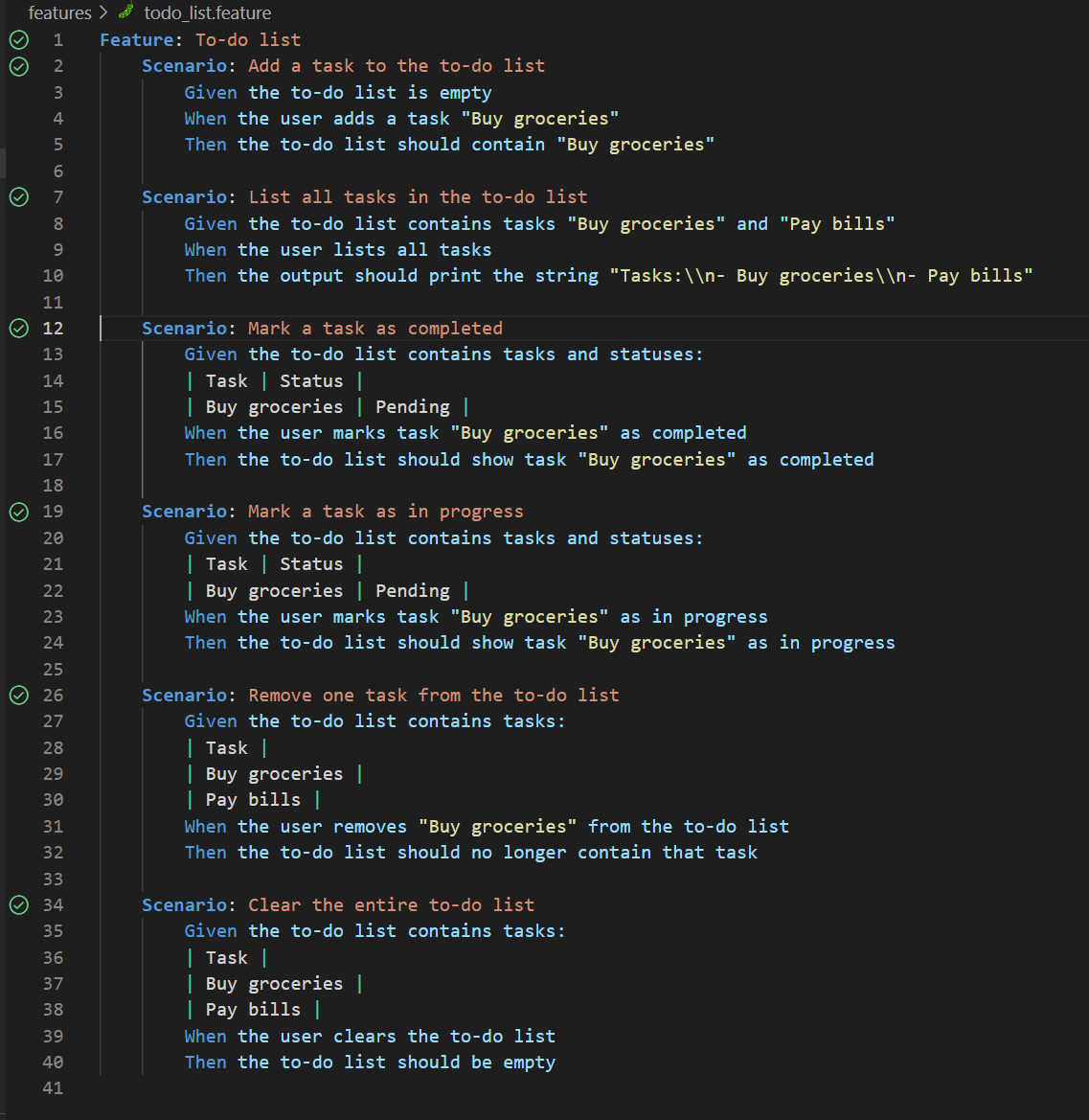


Once the core functions are done we need to manage the loop for the console program to works constantly until a certain input is entered by the user. This loop is achieved by implementing the following method.



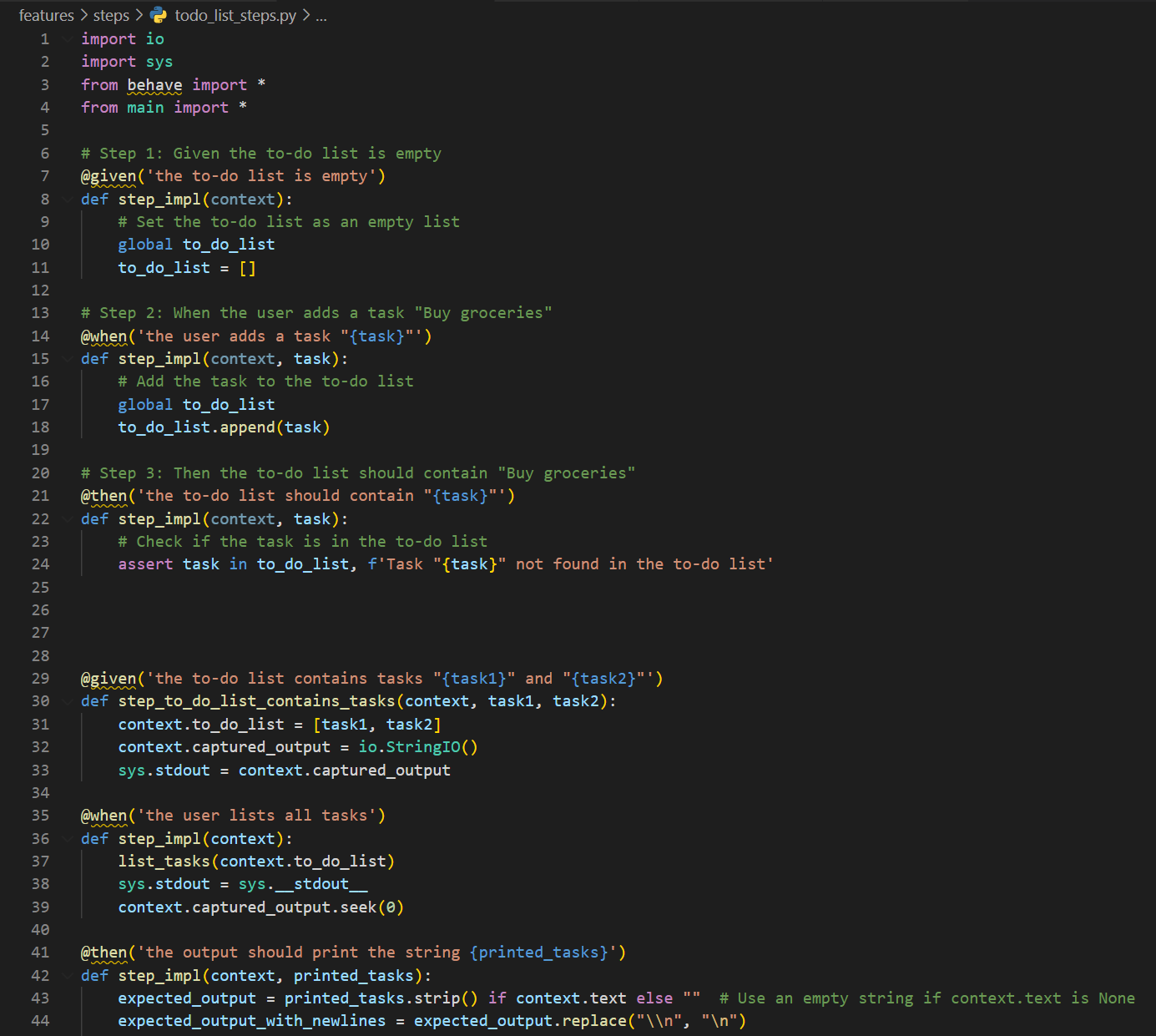
The combination of functions implemented in the main.py file make possible to fulfill the requieremtns.

Now its required to create the acceptance test statements for every functionality in the program. The code to be implemented is required to be written according to a certain syntaxis, which is Gherkin, this includes every function as an acceptance test in the form of a “Feature”.



Each of these scenarios establishes certain circumstances for each test to be validated.

The next step for these tests to be validated is to write the assertion methods related to every scenario previously established, this is achieved by using the behave python library, it uses certain keywords to link the scenario to a Junit based test. The code written with behave is the following.



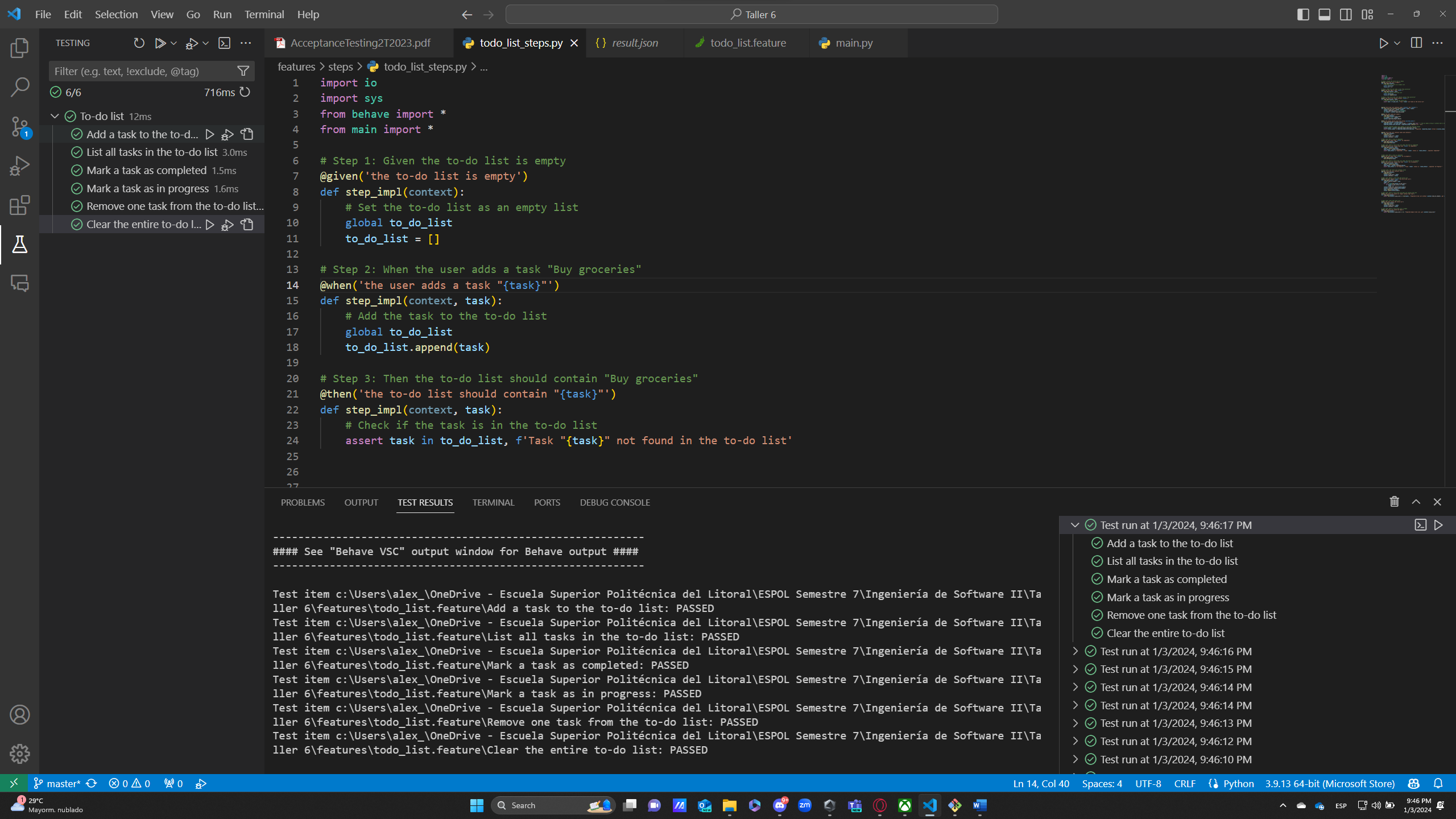
Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Once all the tests have been created the code must be tweaked until every one of them passes.



# Conclutions and recomendations

In summary, employing Gherkin language in tandem with Behave for testing our To-Do List Management program has emerged as a crucial strategy for validating its functionality and dependability. Gherkin's straightforward and human-readable syntax enabled us to articulate test scenarios in a manner easily comprehensible to both technical and non-technical stakeholders. Behave, serving as the connector between Gherkin and Python, facilitated smooth integration with our underlying codebase. Through a well-defined set of scenarios, we conducted thorough testing across various aspects of our program, encompassing tasks such as adding, completing, and handling diverse inputs. The combination of Gherkin and Behave not only delivered a robust suite of tests but also acted as living documentation, enhancing the transparency and manageability of our codebase. Rooted in behavior-driven development principles, this approach empowers our team to iterate confidently on the To-Do List Management program, ensuring both reliability and a user-centric experience.