**Universidad Autónoma de Tamaulipas**

**Facultad de Ingeniería Tampico**

A black background with a black square

Description automatically generated with medium confidenceA red and grey logo

Description automatically generated

**ASIGNATURA**

**PROGRAMACION DE INTERFACES Y PUERTOS**

**6to. Semestre – Grupo “i”**

**2025 -1**

**TRABAJO**

**Desarrollo de Prácticas y Proyectos**

**UNIDAD**

**2 -DESARROLLO E IMPLEMENTACIÓN DE CONTROLADORES MEDIANTE INTERFACES Y PUERTOS**

**Docente:** Dr. García Ruiz Alejandro H.

|  |  |
| --- | --- |
| **Integrante del Equipo** | **Nivel de Participación** |
| Izaguirre Cortes Emanuel | 33.33% |
| Turrubiates Mejia Gilberto | 33.33% |
| García Salas Yahir Misael | 33.33% |
| Total: | 100% |

# Índice

[Índice 1](#_Toc191750393)

[Repositorio(s) de Prácticas 2](#_Toc191750394)

[PP1. 2](#_Toc191750395)

[Descripción de la practica 2](#_Toc191750396)

[Introducción 2](#_Toc191750397)

[Componentes para el desarrollo de la practica 2](#_Toc191750398)

[Desarrollo 3](#_Toc191750399)

[Pruebas Realizadas 3](#_Toc191750400)

[Conclusión PP1 5](#_Toc191750401)

[PP2. 6](#_Toc191750402)

[Descripción de la practica 6](#_Toc191750403)

[Introducción 6](#_Toc191750404)

[Componentes para el desarrollo de la practica 6](#_Toc191750405)

[Desarrollo 6](#_Toc191750406)

[Pruebas Realizadas 7](#_Toc191750407)

[Conclusión PP2 8](#_Toc191750408)

# Repositorio(s) de Prácticas

|  |  |
| --- | --- |
| **Practica** | **Repositorio** |
| Practica 1 a la 2 |  |

# PP1.

Esta práctica consiste en el desarrollo de una aplicación de escritorio en Python utilizando PyQt5. La aplicación permite mostrar diferentes imágenes según la selección de un usuario mediante un QSlider y verificar si el texto ingresado por el usuario coincide con la imagen mostrada.

## Descripción de la practica

Esta práctica consiste en el desarrollo de una aplicación de escritorio en Python utilizando PyQt5. La aplicación permite mostrar diferentes imágenes según la selección de un usuario mediante un QSlider y verificar si el texto ingresado por el usuario coincide con la imagen mostrada.

## Introducción

Las interfaces gráficas de usuario (GUI) facilitan la interacción con aplicaciones mediante elementos visuales. PyQt5 es un framework popular para la creación de interfaces en Python. En esta práctica, se desarrolló una aplicación donde el usuario debe identificar correctamente el nombre de una imagen seleccionada aleatoriamente.

## Componentes para el desarrollo de la practica

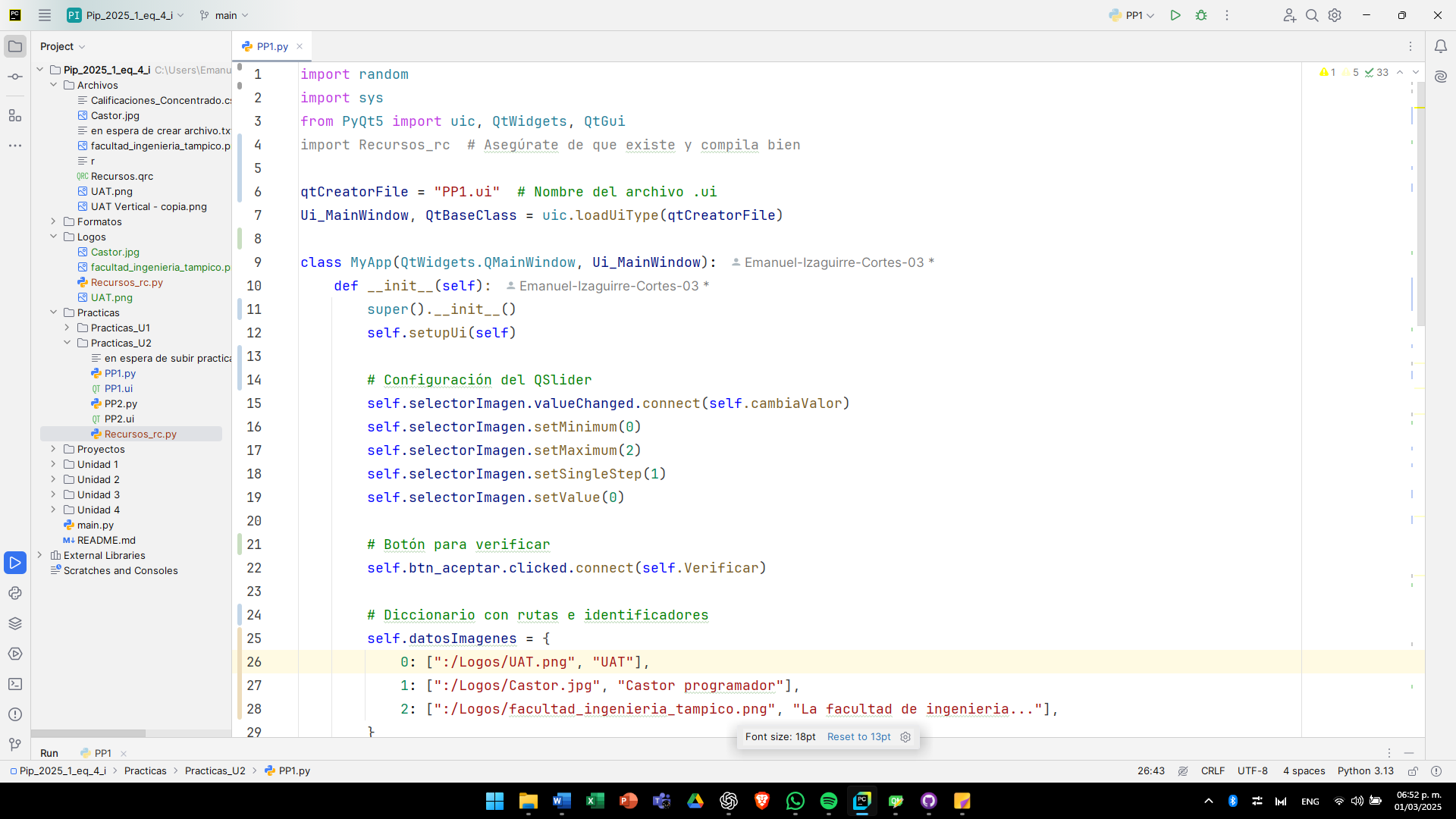
* **Lenguaje de programación:** Python 3
* **Bibliotecas:** PyQt5, random, sys
* **Widgets de PyQt5:** QLabel, QSlider, QPushButton, QMessageBox
* **Archivos:** Archivo .ui para la interfaz diseñada en Qt Designer y un script .py para la lógica de la aplicación

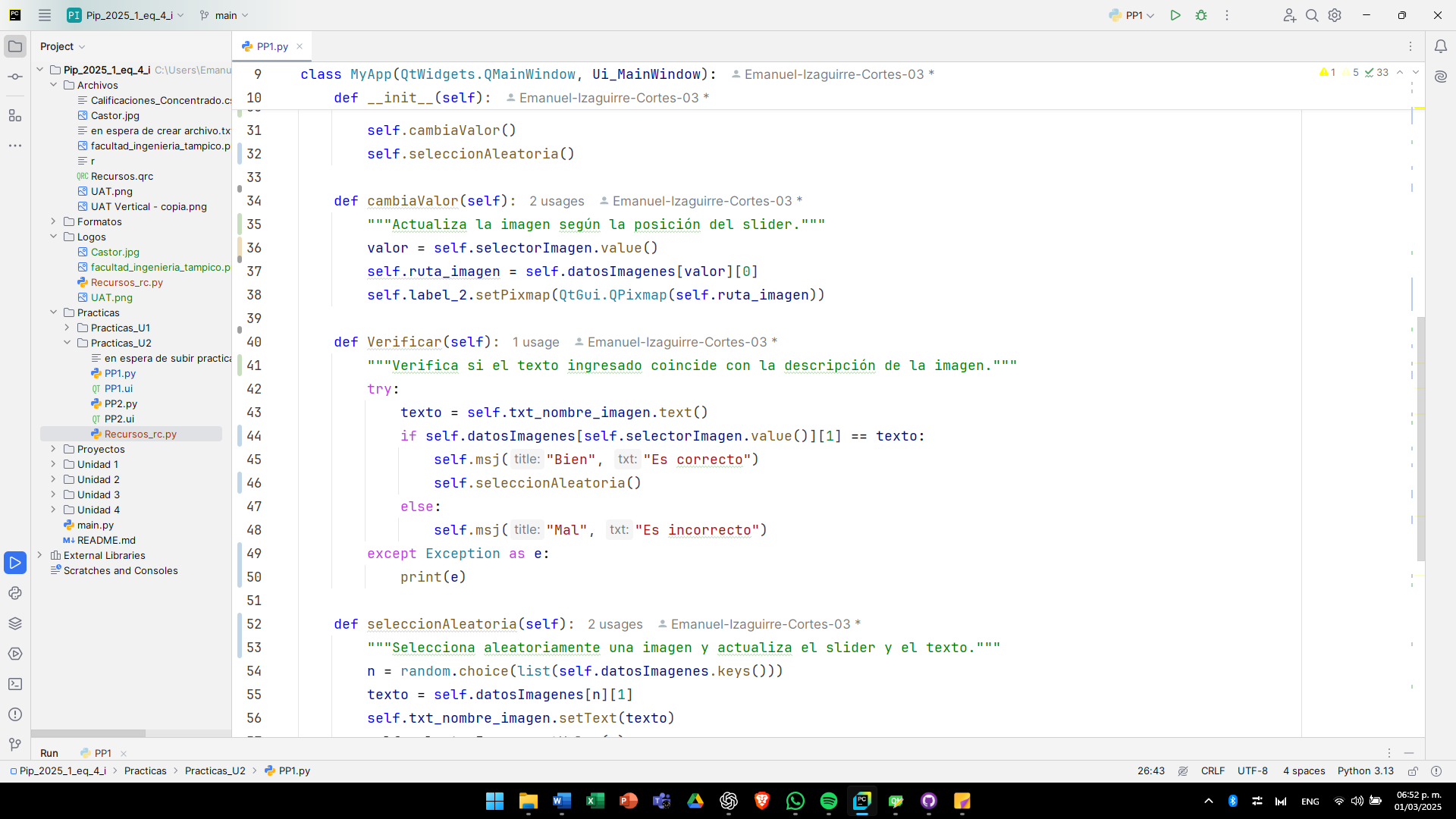
## Desarrollo

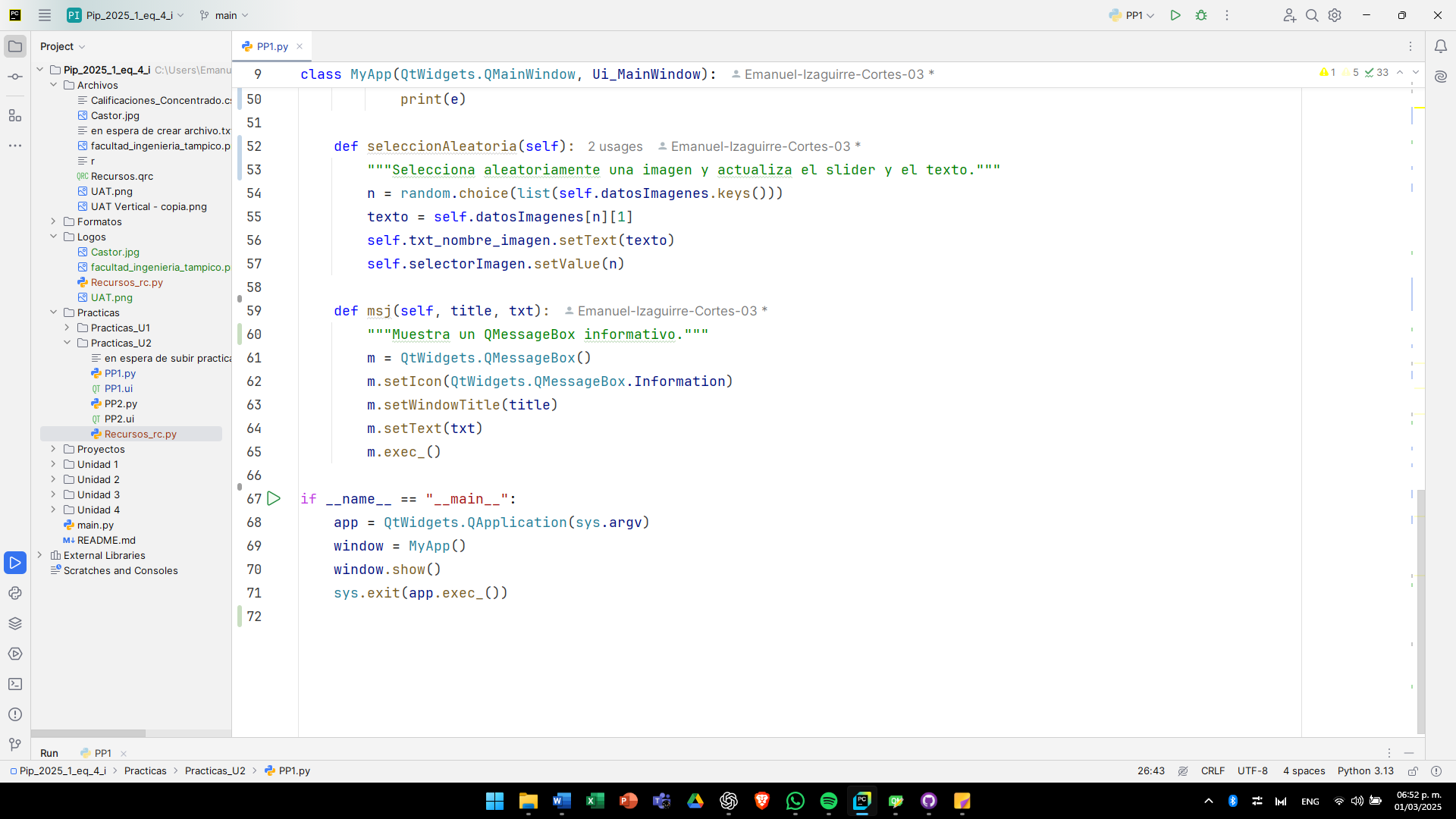
1. Se diseñó la interfaz en Qt Designer y se guardó en un archivo .ui.
2. Se cargó la interfaz en Python usando uic.loadUiType().
3. Se implementó un diccionario con las rutas de las imágenes y sus nombres.
4. Se configuró un QSlider para cambiar las imágenes según su valor.
5. Se implementó un método para seleccionar aleatoriamente una imagen y mostrar su nombre en un QLineEdit.
6. Se programó la verificación de la coincidencia entre el nombre ingresado y el correcto.

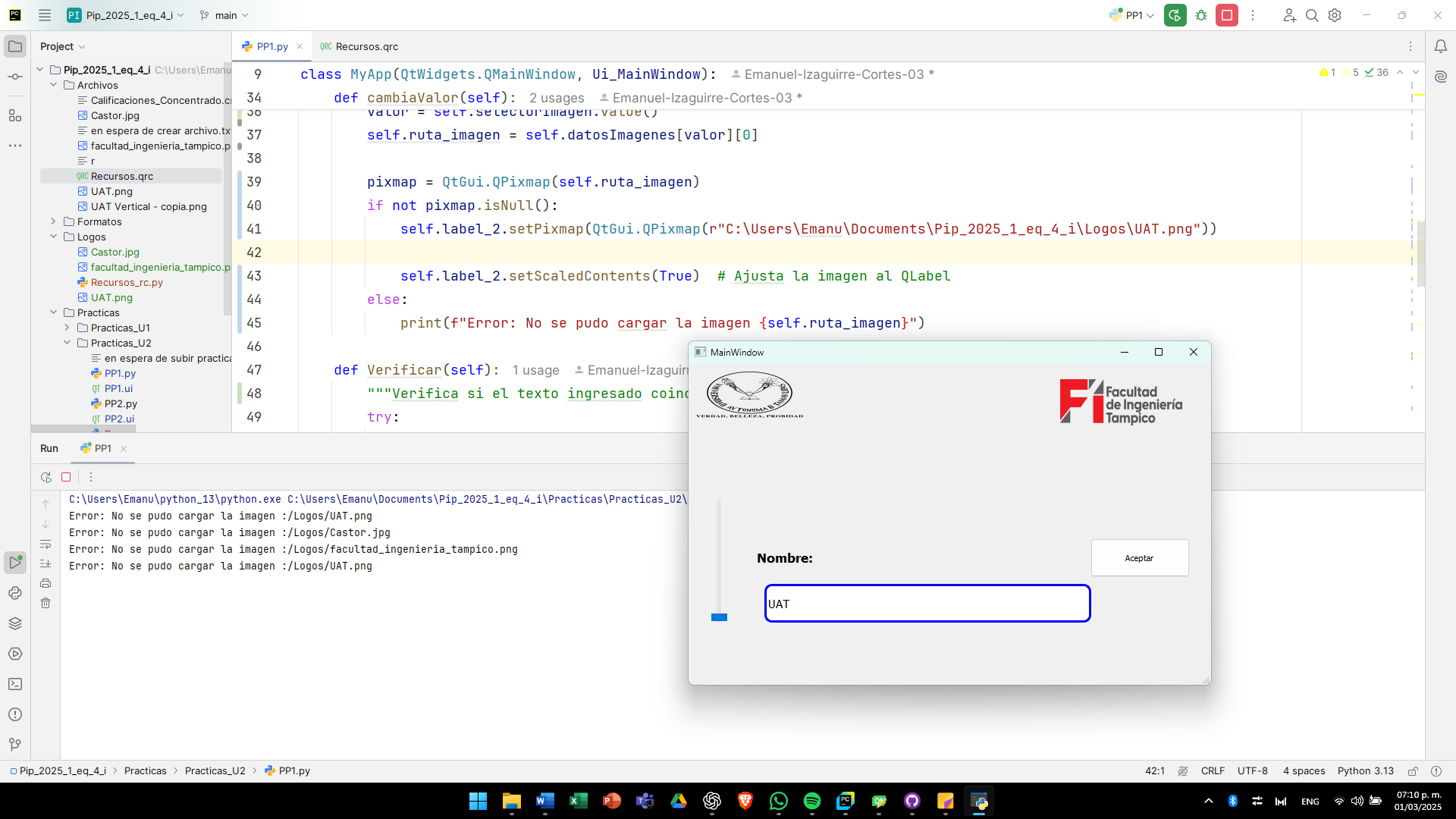
## Pruebas Realizadas

* Se probó la carga de imágenes según la selección del slider.
* Se verificó la correcta selección aleatoria de imágenes.
* Se comprobó la validación de nombres de imágenes con mensajes informativos.
* Se corrigieron errores en la comparación del nombre ingresado.









## Conclusión PP1

En esta práctica, se implementó una interfaz gráfica utilizando **PyQt5**, permitiendo la manipulación de imágenes a través de un **QSlider** y verificando la correspondencia entre el nombre ingresado y la imagen mostrada.

Al concluir el desarrollo, se identificaron varios aspectos clave:

1. **Manejo de Archivos UI:** La correcta carga del archivo .ui con uic.loadUiType() facilita la integración de diseños creados en Qt Designer, reduciendo el tiempo de desarrollo.
2. **Gestión de Rutas de Imágenes:** Se evidenció la importancia de definir rutas de imágenes correctamente, evitando errores relacionados con secuencias de escape (\U) en Windows.
3. **Interactividad con el Usuario:** La validación de texto y el uso de QMessageBox para mostrar mensajes de acierto o error mejoraron la experiencia del usuario, permitiendo una retroalimentación inmediata.
4. **Aleatorización de Datos:** Se utilizó la función random.choice() para seleccionar imágenes de manera aleatoria, lo que diversificó la interacción con el programa y reforzó conceptos de estructuras de datos como diccionarios en Python.
5. **Importancia del Debugging:** Se resolvieron errores de sintaxis y se optimizó la gestión de eventos, resaltando la necesidad de pruebas constantes para garantizar un funcionamiento correcto.

En general, la práctica permitió reforzar conocimientos en **interfaces gráficas, manipulación de imágenes, eventos en GUI y validación de datos**, habilidades esenciales para el desarrollo de software interactivo.

# PP2.

## Descripción de la practica

La aplicación permite iniciar un temporizador que se actualiza cada segundo y muestra el tiempo transcurrido en una etiqueta. Se ha desarrollado utilizando PyQt5 y un diseño de interfaz definido en un archivo de código Python.

## Introducción

Este proyecto es una aplicación en Python con una interfaz gráfica (GUI) desarrollada con PyQt5. La aplicación implementa un temporizador que muestra el tiempo transcurrido en segundos.

## Componentes para el desarrollo de la practica

Lenguaje de programación\*\*: Python

- \*\*Bibliotecas\*\*:

- PyQt5 (para la interfaz gráfica)

- QtCore (para el temporizador)

- QtWidgets (para los elementos de la interfaz)

- \*\*Archivos\*\*:

- `PP2\_ui.py`: Define la interfaz gráfica.

- `PP2.py`: Contiene la lógica del programa y la interacción con la interfaz.

## Desarrollo

1. Se creó la interfaz gráfica en `PP2\_ui.py`, definiendo un botón (`btn\_temporizador`) y una etiqueta (`label\_tiempo`).

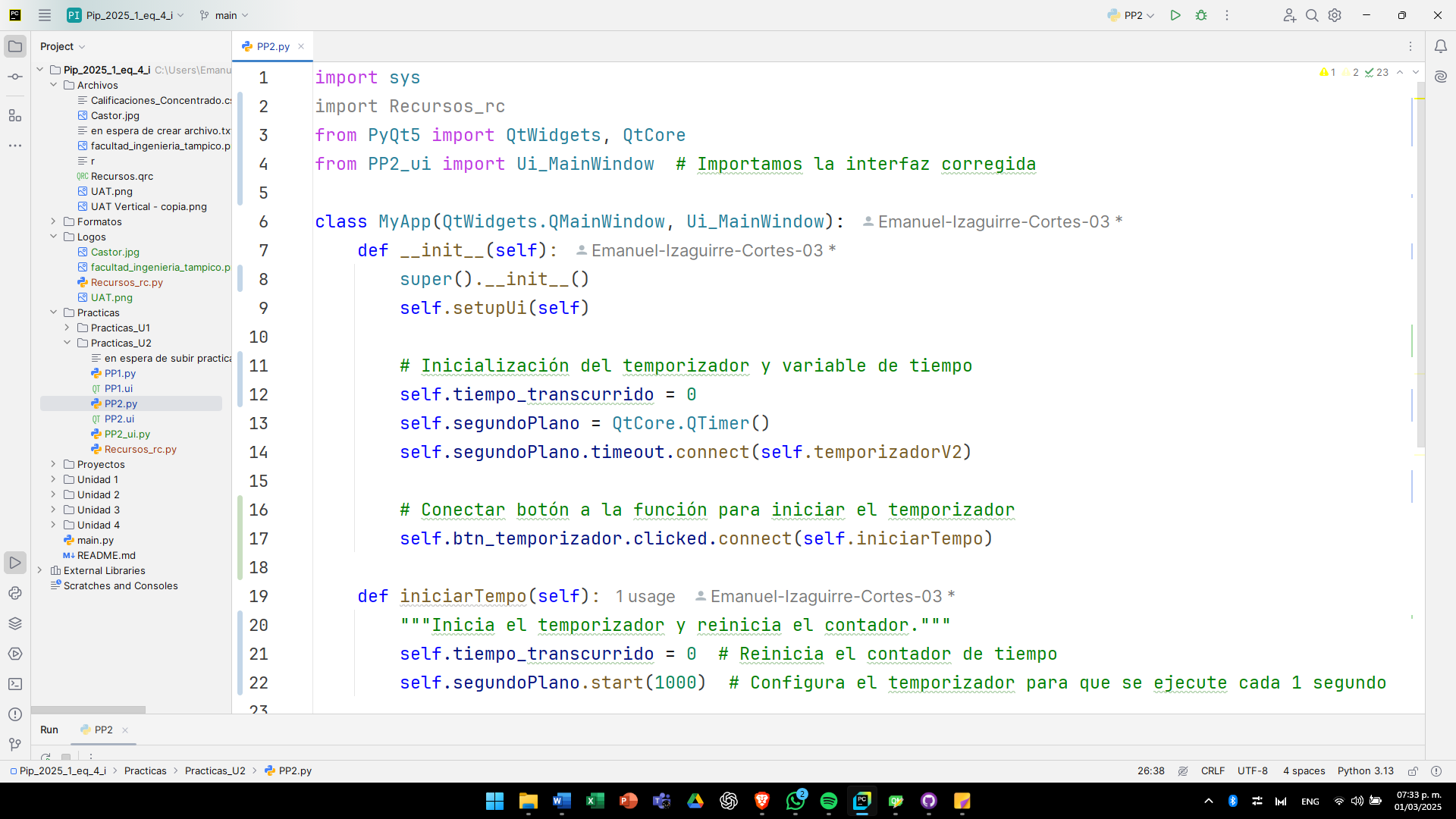
2. Se implementó la lógica en `PP2.py`, donde:

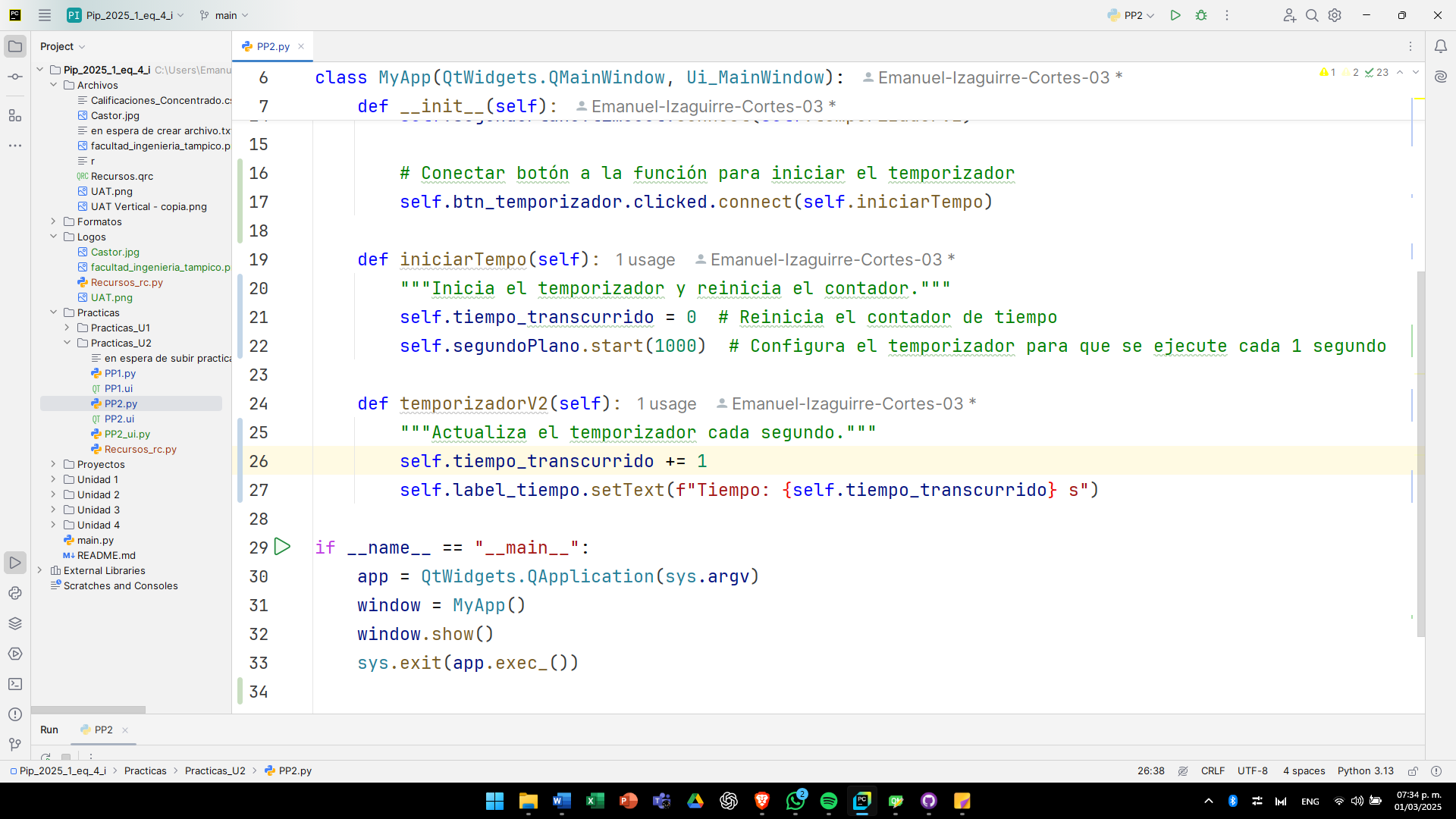
- Se inicializa la interfaz y los componentes.

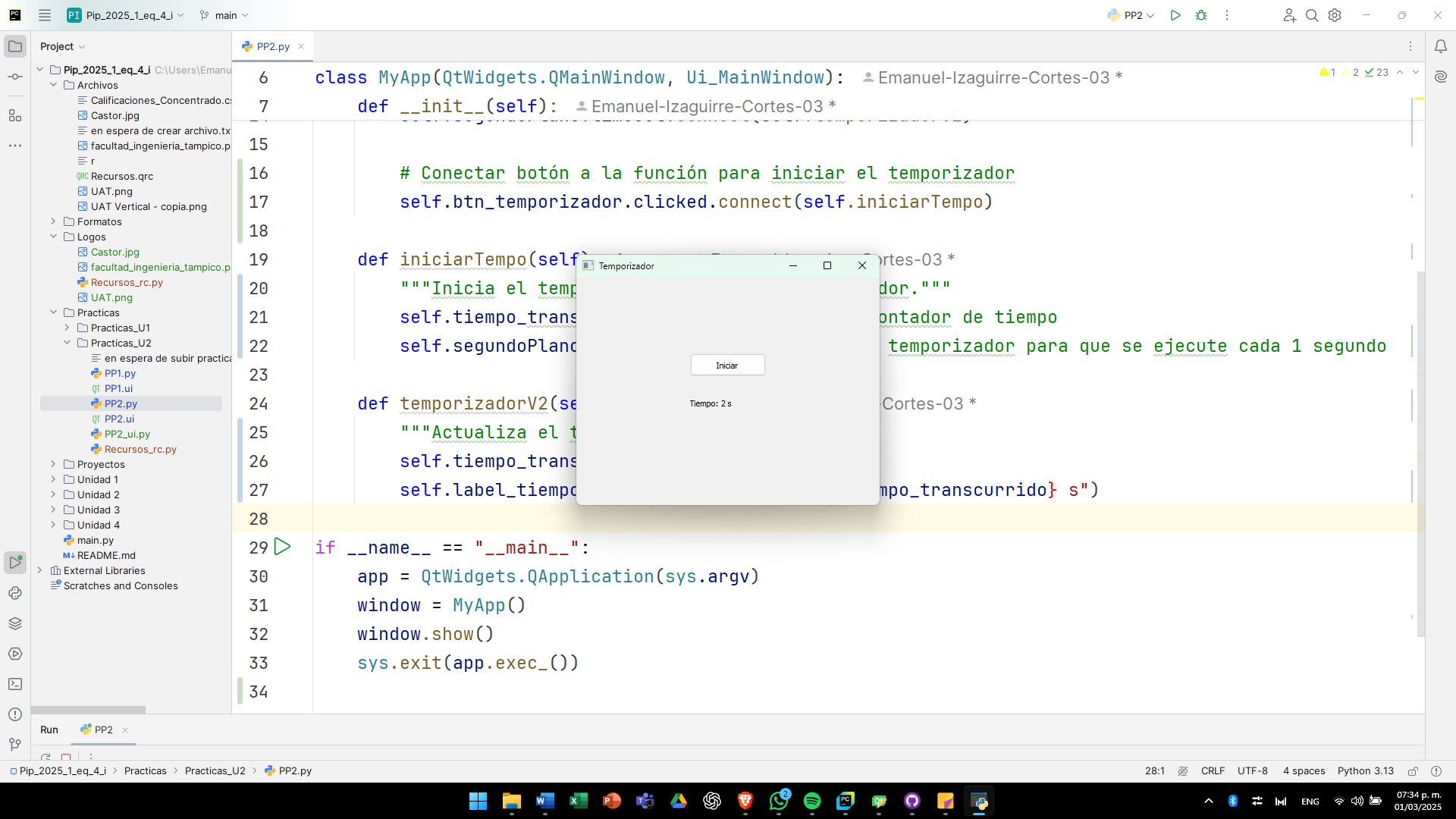
- Se configura un temporizador que se actualiza cada segundo.

- Se maneja el evento del botón para iniciar el temporizador.

## Pruebas Realizadas







## Conclusión PP2

El proyecto cumple con su objetivo de implementar un temporizador simple con una interfaz gráfica en PyQt5. Se logró una integración funcional entre la lógica del programa y la interfaz, permitiendo futuras mejoras o expansión.