Proyecto Final Estructura de Datos I: Burgerway

1st José Alejandro Gayón Parra   
*Ingenieria de Sistemas*  
*Universidad del Norte*Barranquilla, Colombia  
jgayon@uninorte.edu.co

*Abstract*—This document confronts the creation and development of a software in a Python environment, based on the resolution of a problem (social, economic, healthcare, etc.) using the Data Structures seen during the class “Data Structure I” of the first semester of the year 2023.

Keywords—software, development, Data Structure, Python.

# Introducción

Actualmente uno de los factores determinantes en el negocio de comidas rápidas es la competencia. Esta situación se agudiza con el tiempo al aumentar y diversificar la oferta. Con el fin de mantenerse a flote y superar la competencia es imprescindible implementar estrategias orientadas a captar la atención de los clientes. Un empresario (hipotético) decidió abrir un nuevo negocio de Hamburguesas, su estrategia de venta es ofrecer opciones de ingredientes para que el cliente cree su hamburguesa de acuerdo con sus preferencias. De esta manera elegirá el tipo de pan, carne, acompañantes y salsas. Para facilitar el proceso es necesario crear una aplicación que permita a los clientes crear combinaciones innovadoras a su gusto y de acuerdo con el stock disponible.

# Metodologia

Para solucionar el problema dado, se decidio la creacion de un software basado en el entorno de *Python* y utilizando *Visual Studio Code*. El software permitira al usuario a crear su pedido seleccionando los elementos que el usuario desee que tenga*.* También debe generar el balance total del pedido, la disponibilidad de los recursos (ingredientes) y permitir que el usuario escoja el método de pago.

## Requerimientos Funcionales

* La creación del pedido.
* La creación y selección de los elementos de la Hamburguesas.
* El balance total del pedido.
* La disponibilidad del producto.
* Los datos del Usuario (Dirección, Medio de Pago).

## Requerimientos no funcionales

* Lenguaje de programación: Python.
* IDE: Visual Studio Code.
* Software ejecutable en PC.
* Interfaz amigable para el Usuario.
* Instrucciones de uso.
* Requerimientos legales.
* La aplicación de escritorio debe funcionar en Windows.

# Plan UML

Diagrama

Descripción generada automáticamente

Fig. 1 Plan UML

# Observaciones

Utilizando el IDE Visual Studio Code y el lenguaje de programación Python se crearon dos archivos principales, el “MainWindow.py” y el “BurgerClass.py”. En el primero de estos se establece el código que creara el interfaz grafico con el cual el usuario va a interactuar. En “BurgerClass.py” se crearon las clases y métodos para el funcionamiento de la aplicación y en donde se utilizo aquello aprendido en la clase de Estructuras de Datos I.

Las estructuras de datos utilizadas en este proyecto fueron las siguientes:

* Archivos
* Listas Enlazadas
* Pilas

Las estructuras de Archivos se ven utilizadas en la lectura de los ingredientes, su precio y su stock desde un archivo Excel llamado “Ingredientes.xlsx”. Se utilizo una implementación simple de listas enlazadas para almacenar los ingredientes de una hamburguesa. La clase “**ListaEnlazada**” encontrada en “BurgerClass.py”, representa la lista enlazada y contiene métodos como “**append**”, que agrega un nuevo nodo al final de la lista, y “**obtener\_elementos**”, que devuelve una lista con los elementos de la lista enlazada. La clase “**Pila**” que se encuentra en el mismo modulo, es una implementación básica de una pila [1]. En el código, se utiliza para realizar la función de deshacer una selección en la construcción de la hamburguesa. Cuando el usuario deshace una selección, el último ingrediente seleccionado se elimina de la pila.

En la seccion VI. Resultados del informe se adjuntaron imágenes donde se puede observar el codigo utilizado.

El proyecto y todos sus archivos relacionados se encuentra en el Github con los siguientes links:

<https://github.com/jgayon/OOPProyectFinal.git> en la carpeta Version Datos-1

ó

https://github.com/jgayon/OOPProyectFinal/tree/main/Version%20Datos-1

ó

https://github.com/jgayon/OOPProyectFinal/tree/c32edcf639811bde4f98d0809d5612b630492836/Version%20Datos-1

# Conclusiones

Se cumplieron la mayoría de los requerimientos funcionales y no funcionales que se habían establecido, con la excepción de la disponibilidad del producto. Siendo este una de las posibles mejoras que se pueden realizar. Como la mejora del interfaz grafico otra de estas posibles mejoras que se podría realizar a futuro.

Al momento la aplicación tiene varias limitaciones tanto generales como específicas. Entre estas limitaciones generales podemos encontrar una interfaz de usuario limitada, la falta de persistencia de datos, una gestión de errores limitada, una falta de seguridad ya que no aborda aspectos de seguridad, como la autenticación de usuarios o la protección de datos sensibles. Finalmente, en las especificas se encuentra que al momento el usuario solo puede crear una hamburguesa por pedido y que no hay división clara entre los tipos de ingredientes.

El impacto de esta aplicación se encuentra en la innovación que trae al momento de pedir comida, ya que el usuario va a poder satisfacer sus antojos todo el tiempo y evitar la inclusión de ingredientes que comúnmente terminen siendo sacados del producto final ya en manos del cliente, y su posible perdida.

# Resultados

**Código “MainWindow.py”**

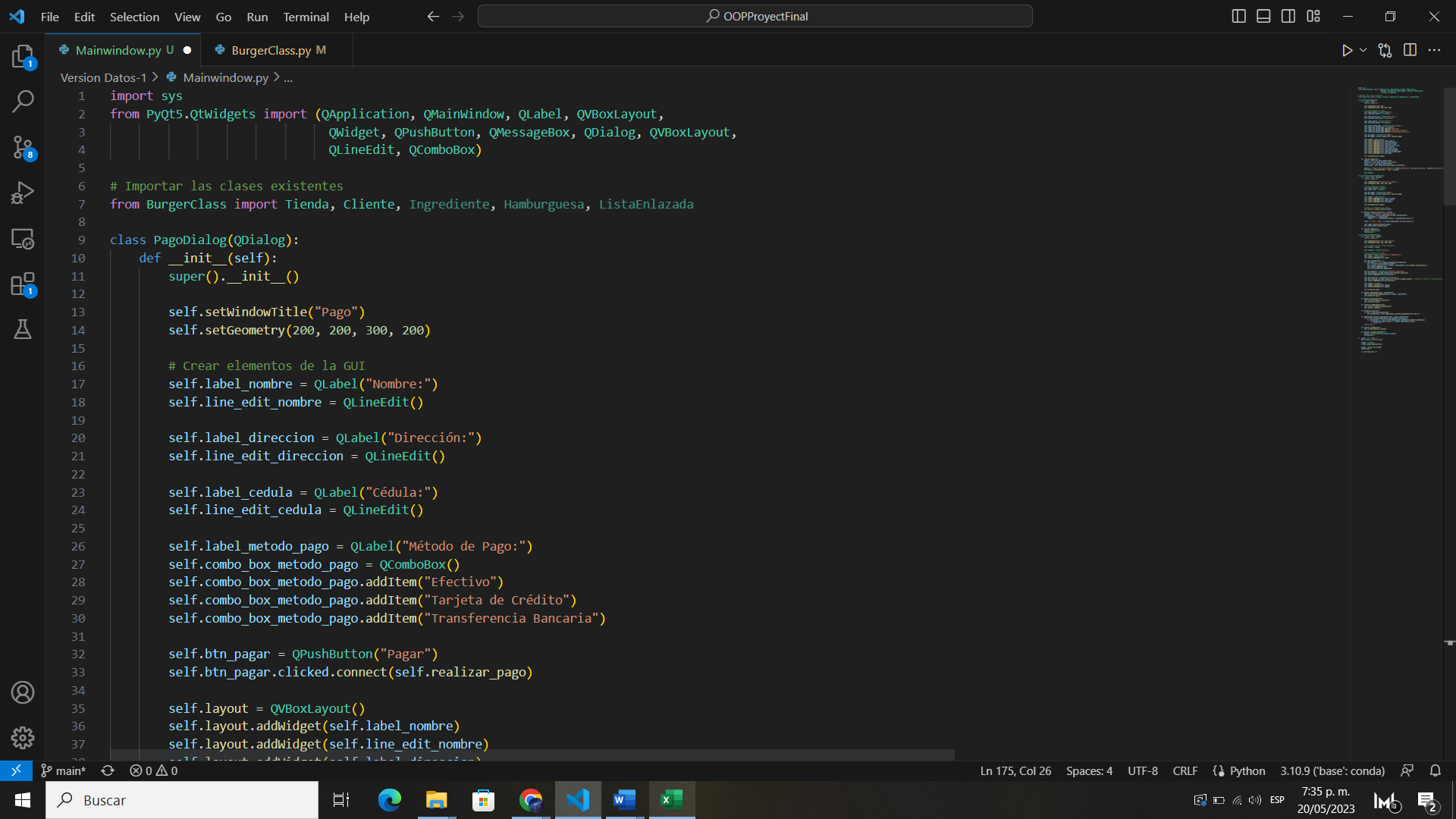


Fig. 2 Código de MainWindow.py

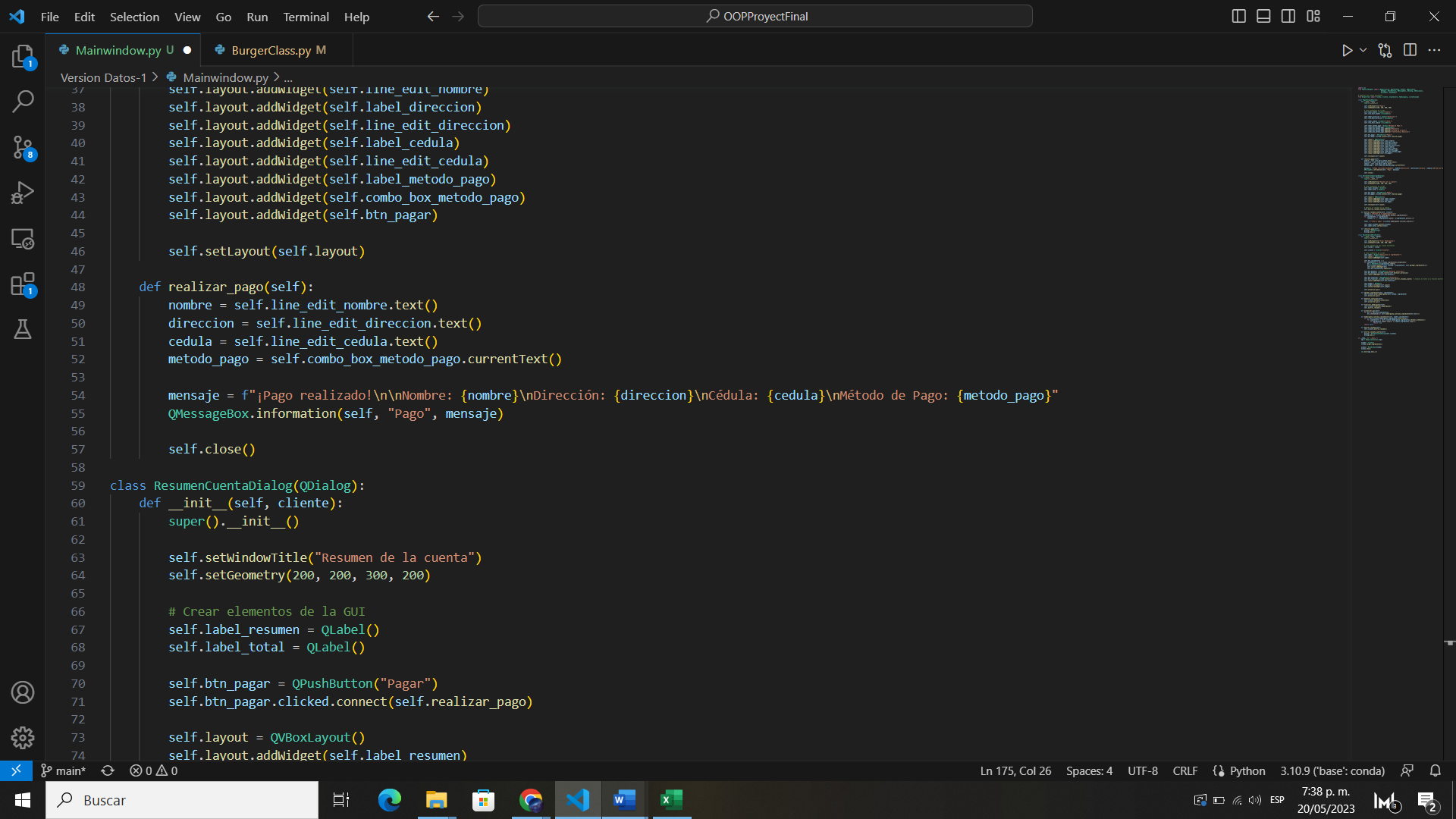


Fig. 3 Código de MainWindow.py

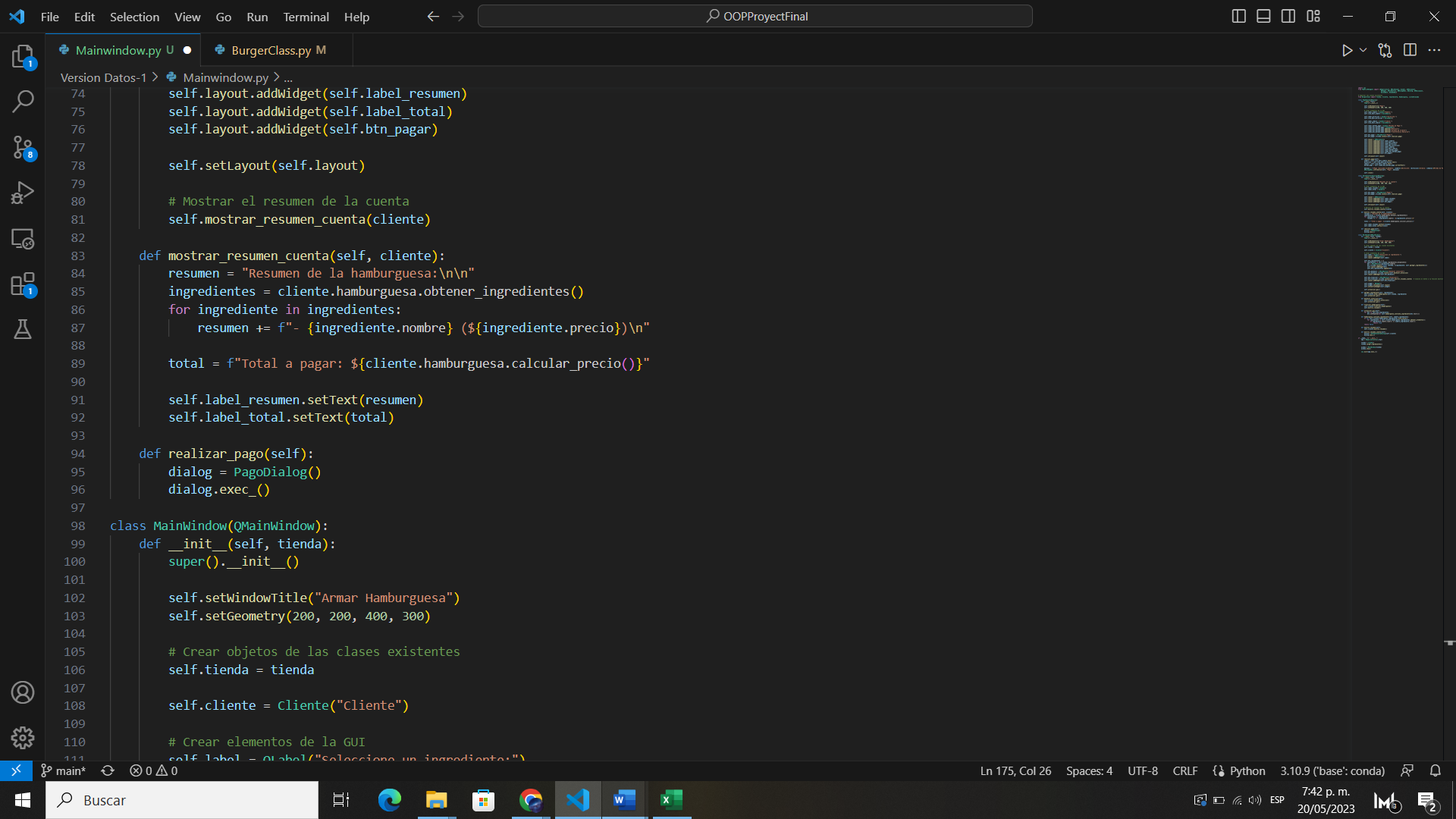


Fig. 4 Código de MainWindow.py

Captura de pantalla de computadora

Descripción generada automáticamente

Fig. 5 Código de MainWindow.py

Captura de pantalla de computadora

Descripción generada automáticamente

Fig. 6 Código de MainWindow.py

**Código “BurgerClass.py”**

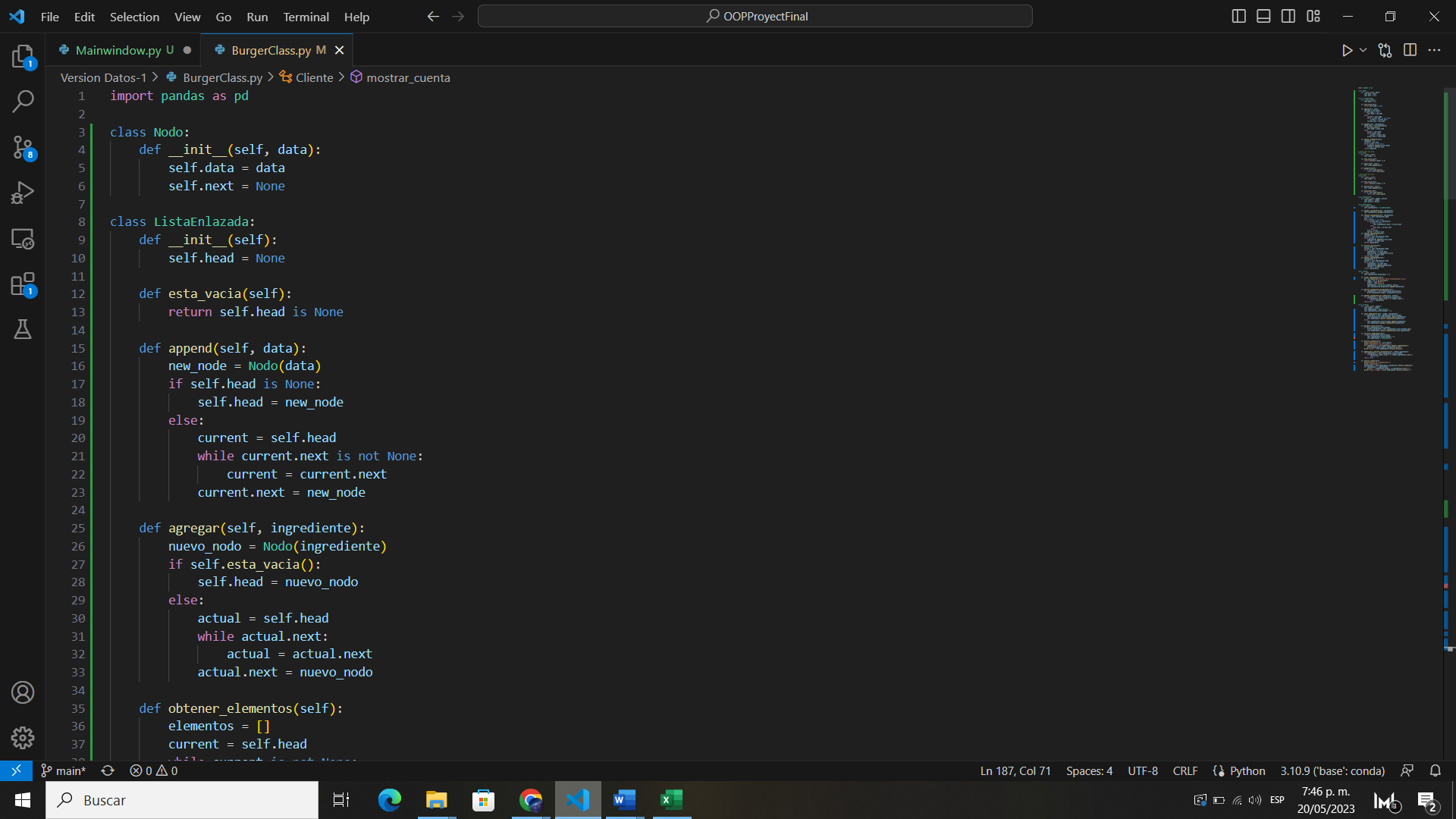


Fig. 7 Código de BurgerClass.py

Captura de pantalla de computadora

Descripción generada automáticamente

Fig. 8 Código de BurgerClass.py

Captura de pantalla de computadora

Descripción generada automáticamente

Fig. 9 Código de BurgerClass.py

Captura de pantalla de computadora

Descripción generada automáticamente

Fig. 10 Código de BurgerClass.py

Captura de pantalla de computadora

Descripción generada automáticamente

Fig. 11 Código de BurgerClass.py

Captura de pantalla de computadora

Descripción generada automáticamente

Fig. 12 Código de BurgerClass.py

# Referencias

[1] jgayon, «OOPProyectFinal». 5 de octubre de 2022. Accedido: 20 de mayo de 2023. [En línea]. Disponible en: https://github.com/jgayon/OOPProyectFinal

[2] «Estructuras de datos básicas (Pilas, Colas y Listas Enlazadas)». https://pier.guillen.com.mx/algorithms/11-otros/11.4-estructuras\_datos.htm (accedido 20 de mayo de 2023).

[3] D. S. A. Gómez, «Estructuras de Datos Clase 4 – Pilas y colas».

[4] «Operaciones en las Listas Enlazadas». http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro9/operaciones\_en\_las\_listas\_enlazadas.html (accedido 20 de mayo de 2023).

[5] «Qt for Python». https://doc.qt.io/qtforpython-6/ (accedido 20 de mayo de 2023).

[6] «User Guide — pandas 2.0.1 documentation». https://pandas.pydata.org/docs/user\_guide/index.html#user-guide (accedido 20 de mayo de 2023).