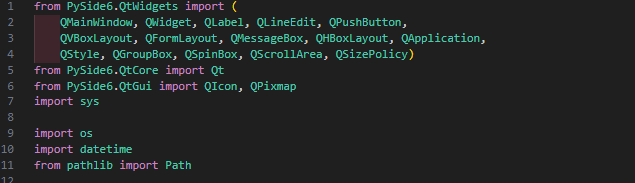
1. **Importaciones**  
   
2. **Ventana principal**

**os.system("cls"): limpia la consola.**

**absPath(file): da la ruta completa a un archivo (como imágenes).**

**Variables globales**

**usuarios = {}: diccionario para guardar usuarios.**

**usuario\_actual = None: usuario que está usando el sistema.**

**Clase MainWindow**  
Representa la ventana principal de la aplicación.

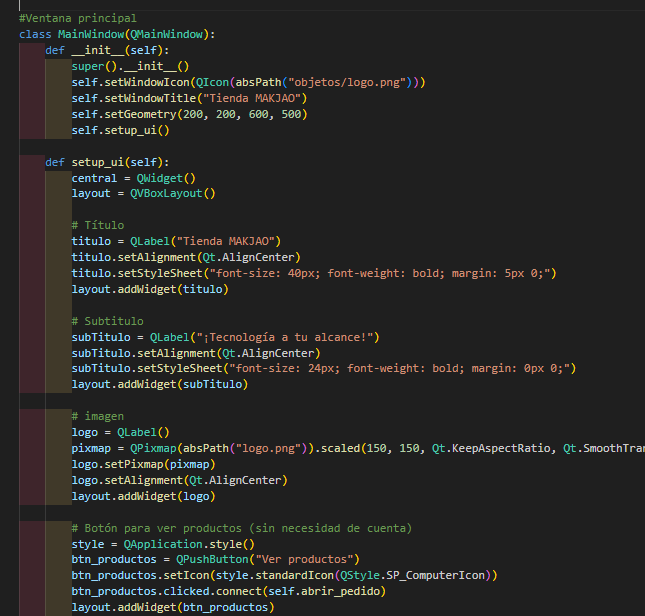
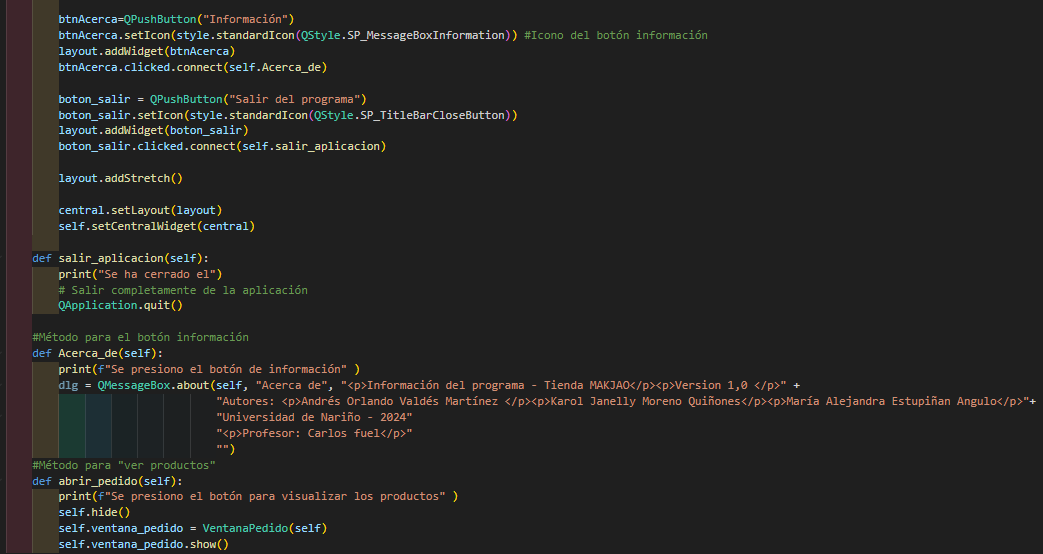
**Constructor \_\_init\_\_**  
Define el ícono, el título, tamaño y llama a setup\_ui() para construir la interfaz.

**Método setup\_ui**  
Crea la parte visual:

* Título, subtítulo y logo.
* Botones: "Ver productos" e "Información".
* Todo alineado verticalmente.

**Método Acerca\_de**  
Muestra un mensaje con los autores, versión y universidad.

**abrir\_pedido**  
Oculta la ventana principal y abre otra (llamada VentanaPedido, no incluida en este fragmento).

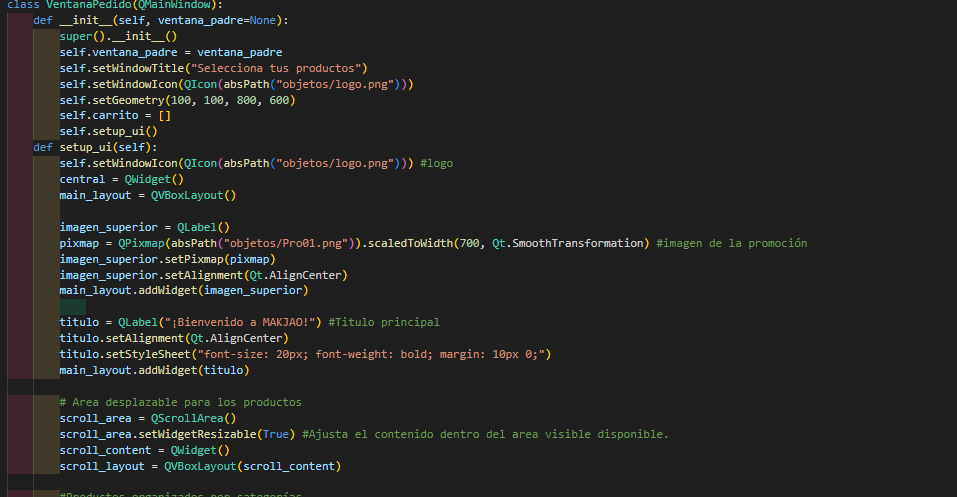
****

1. **Ventana ver productos**

**1. Constructor \_\_init\_\_**:

* Inicializa la ventana principal de la tienda.
* Establece el título, tamaño y estilo.
* Define una lista vacía para el carrito de compras.
* Llama al método setup\_ui() para construir la interfaz.

1. **Método setup\_ui()**:

* Crea el diseño principal vertical.
* Agrega un título con estilo.
* Añade una zona desplazable para mostrar los productos.
* Organiza los productos en categorías con su respectiva sección.

1. **Organización de productos**:

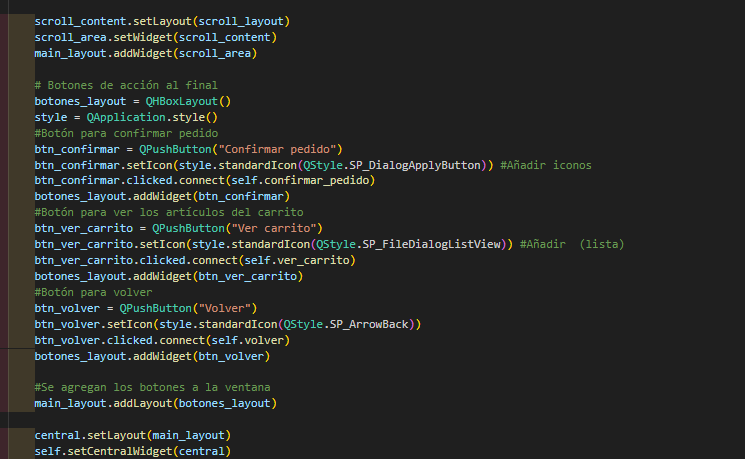
Cada producto tiene:

* + - Nombre del producto.
    - Precio.
    - Campo para seleccionar cantidad.
    - Botón para agregar al carrito.



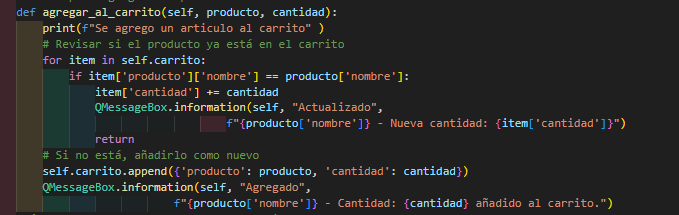
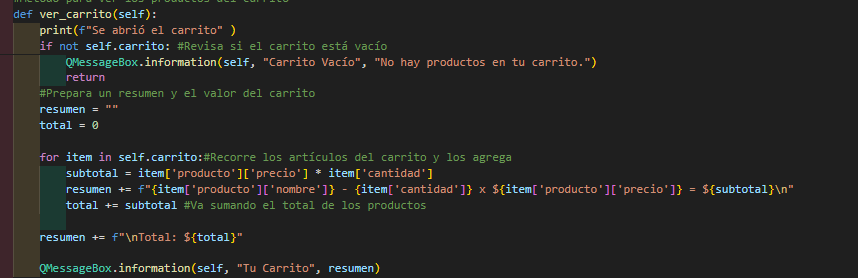
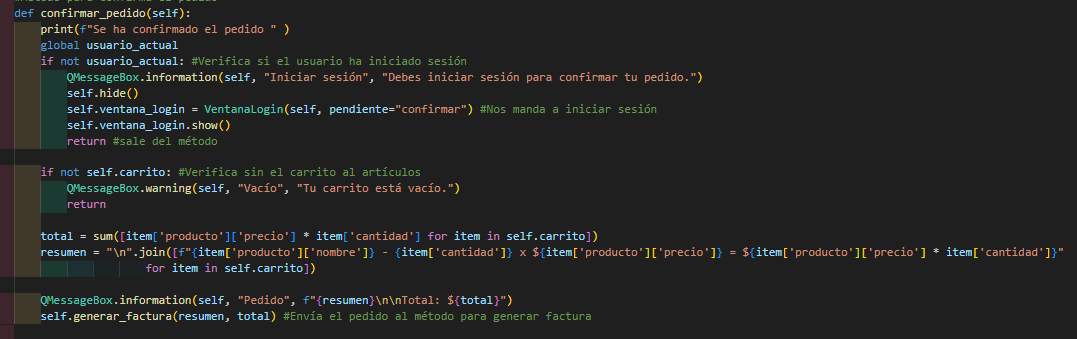
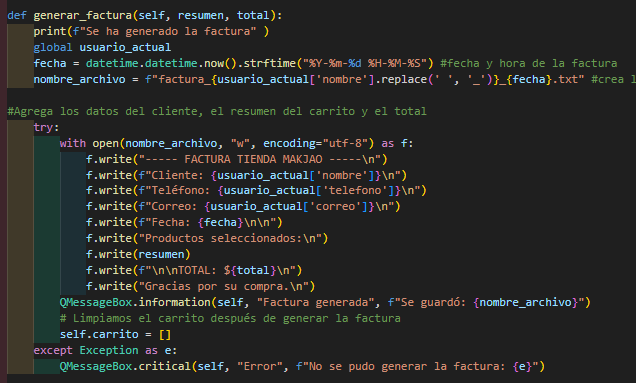
1. **Botones finales en la interfaz**:

* Botón “Confirmar pedido”: muestra el resumen y genera la factura.
* Botón “Ver carrito”: muestra los productos agregados con cantidades.
* Botón “Volver”: vuelve a la ventana anterior.

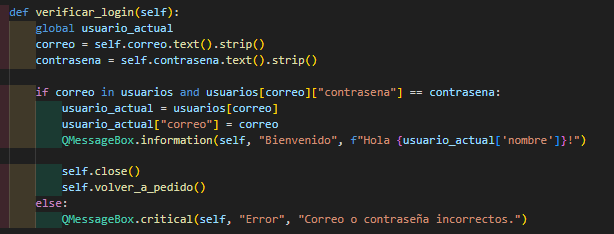


1. **Método verificar\_autenticacion()**:

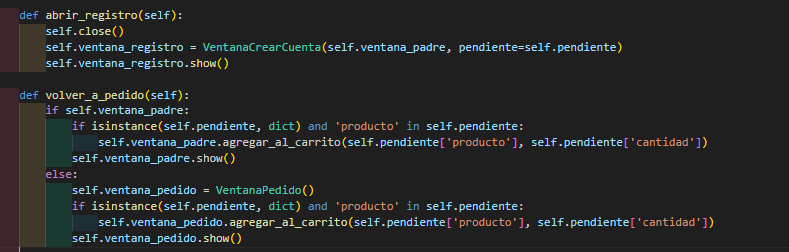
* Verifica si el usuario ha iniciado sesión.
* Si no, abre la ventana de inicio de sesión.
* Si sí, llama al método para agregar el producto al carrito.

1. **Método agregar\_al\_carrito()**:
   * Agrega un producto al carrito con la cantidad seleccionada.
   * Si el producto ya está en el carrito, actualiza su cantidad.
   * Muestra un mensaje confirmando que el producto se agregó.
2. **Método ver\_carrito()**:
   * Muestra en pantalla el contenido actual del carrito.
   * Si está vacío, muestra un mensaje indicando que no hay productos.
   * Si hay productos, muestra cada uno con su cantidad, precio unitario y subtotal.
   * Calcula y muestra el total.
3. **Método confirmar\_pedido()**:
   * Verifica si el usuario ha iniciado sesión y si el carrito tiene productos.
   * Genera un resumen de los productos con cantidades y subtotales.
   * Calcula el total.
   * Muestra el resumen y llama a generar\_factura().
4. **Método generar\_factura()**:
   * Genera un archivo de texto llamado factura.txt.
   * Incluye el nombre del usuario, la fecha y hora actual.
   * Lista todos los productos con sus cantidades, precios y subtotales.
   * Agrega el total de la compra.
5. **Condiciones de funcionamiento**:
   * El sistema requiere que haya un usuario autenticado (usuario\_actual).
   * Las categorías y productos deben estar definidos en listas.
   * El sistema está diseñado para integrarse con otras ventanas, como la de login (VentanaLogin).
6. **Método verificar\_login():**

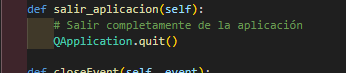
Lee los datos ingresados por el usuario. Comprueba si el correo y la contraseña coinciden con los registros. Si son válidos, guarda la sesión en usuario\_actual, muestra un saludo, cierra esta ventana y vuelve a la ventana de pedido. Si no son válidos, muestra un mensaje de error.



1. **Método abrir\_registro():** Cierra la ventana actual. Abre la ventana de creación de cuenta (VentanaCrearCuenta) pasando la ventana padre y la acción pendiente.



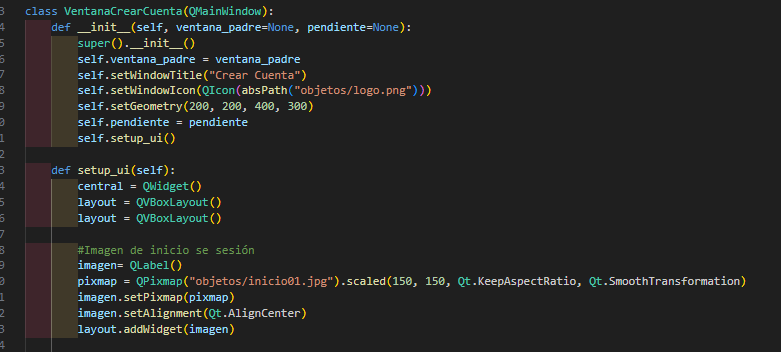
1. **Método volver\_a\_pedido():** Si existe una ventana padre: la muestra. Si hay un producto pendiente, lo agrega al carrito. Si no hay ventana padre: crea una nueva instancia de VentanaPedido. Agrega el producto pendiente si lo hay. Muestra la ventana de pedido.
2. **Método volver():** Si hay ventana padre, la muestra al cerrar esta ventana. Si no, abre la ventana principal (MainWindow).
3. **Método salir\_aplicacion():** Cierra completamente la aplicación.

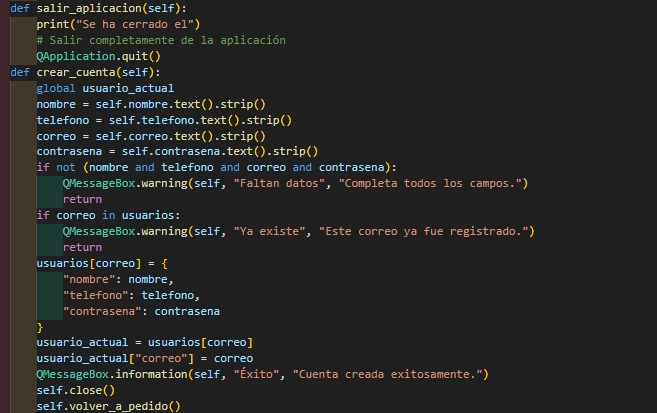


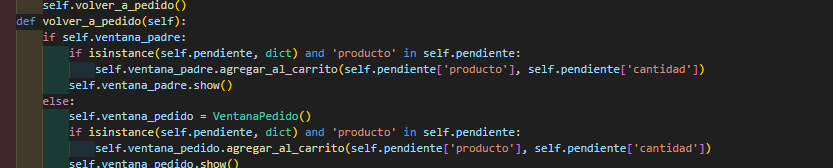
1. **Método closeEvent():** Si hay ventana padre, la vuelve a mostrar automáticamente cuando se cierra esta ventana. Acepta el evento de cierre.

**4. Ventana crear cuenta**

1. Se configura la ventana con un título, icono y tamaño.



1. La función setup\_ui() organiza los elementos visuales: una imagen, campos de texto y botones.
2. Se añade una imagen al centro de la ventana para darle una apariencia amigable.
3. Se crea un formulario con campos de nombre, teléfono, correo y contraseña (con ocultamiento del texto).
4. Se agrega un botón para crear la cuenta, el cual llama al método `crear\_cuenta()`.
5. También se agregan botones de 'Volver' y 'Salir', que permiten regresar o cerrar la aplicación.
6. **crear\_cuenta**() valida que todos los campos estén llenos, verifica que el correo no esté registrado y guarda los datos en el diccionario global `usuarios`.
7. Si todo está correcto, se muestra un mensaje de éxito, se guarda al usuario como usuario\_actual y se cierra la ventana, redirigiendo al pedido.
8. volver\_a\_pedido() muestra la ventana anterior o abre una nueva ventana de pedido, y si había un producto pendiente, lo agrega al carrito.



Estilo del programa.

