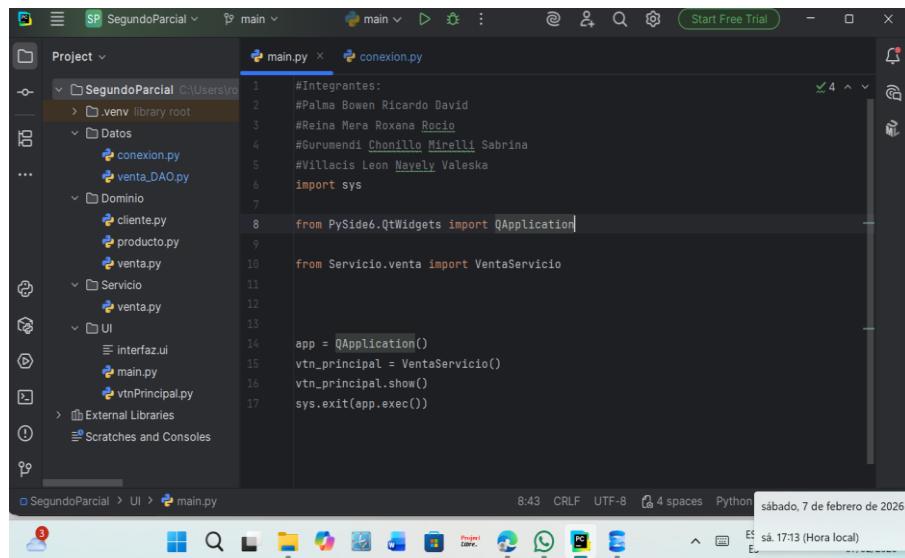


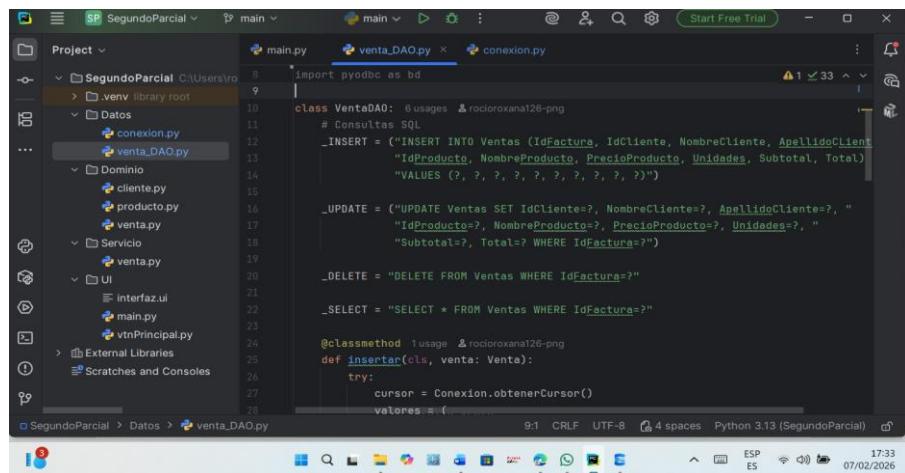
Capturas Proyecto: Venta Online



Project: SegundoParcial C:\Users\ro... / main.py / conexión.py

```
1 #Integrantes:  
2 #Palma Bowen Ricardo David  
3 #Reina Mera Roxana Rocio  
4 #Gurumendi Chonillo Mirella Sabrina  
5 #Villacis Leon Nayely Valeska  
6 import sys  
7  
8 from PySide6.QtWidgets import QApplication  
9  
10 from Servicio.venta import VentaServicio  
11  
12  
13 app = QApplication()  
14 vtn_principal = VentaServicio()  
15 vtn_principal.show()  
16 sys.exit(app.exec())
```

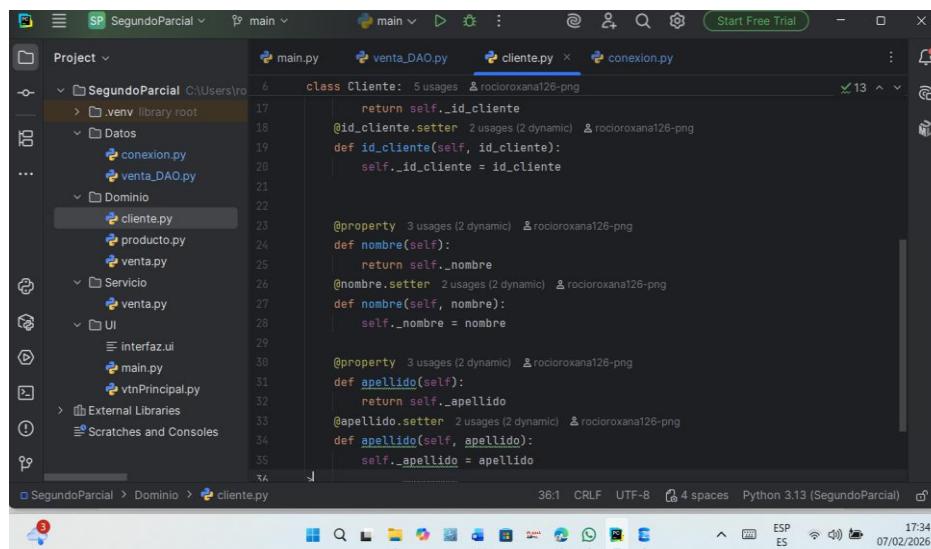
8:43 CRLF UTF-8 4 spaces Python sábado, 7 de febrero de 2026



Project: SegundoParcial C:\Users\ro... / main.py / venta.DAO.py / conexión.py

```
8 import pyodbc as bd  
9  
10 class VentaDAO: 6 usages & rocoroxana126.png  
11     # Consultas SQL  
12     _INSERT = ("INSERT INTO Ventas (IdFactura, IdCliente, NombreCliente, ApellidoCliente,  
13         "IdProducto, NombreProducto, PrecioProducto, Unidades, Subtotal, Total)  
14         "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)")  
15  
16     _UPDATE = ("UPDATE Ventas SET IdCliente=?, NombreCliente=?, ApellidoCliente=?, "  
17         "IdProducto=?, NombreProducto=?, PrecioProducto=?, Unidades=?, "  
18         "Subtotal=?, Total=? WHERE IdFactura=?")  
19  
20     _DELETE = "DELETE FROM Ventas WHERE IdFactura=?"  
21  
22     _SELECT = "SELECT * FROM Ventas WHERE IdFactura=?"  
23  
24     @classmethod 1 usage & rocoroxana126.png  
25     def insertar(cls, venta: Venta):  
26         try:  
27             cursor = Conexion.obtenerCursor()  
28             valores = [
```

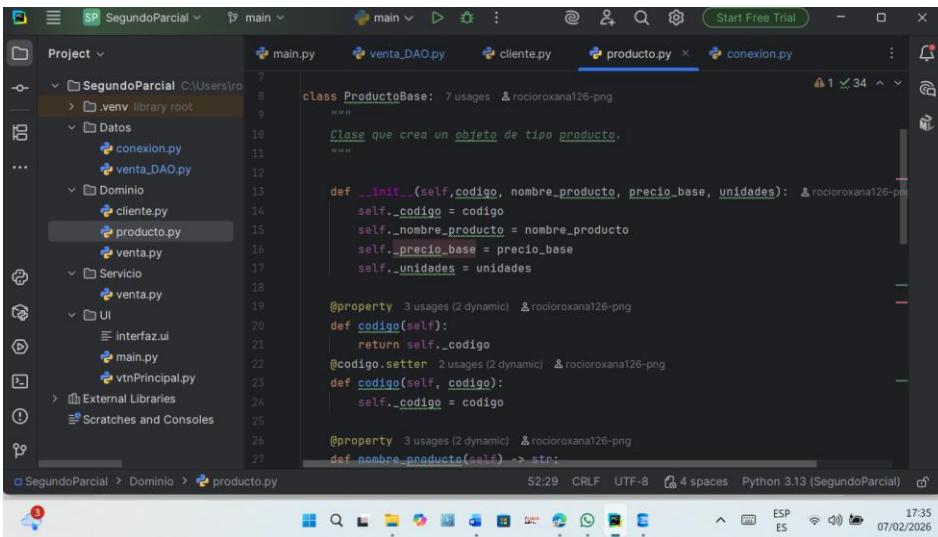
9:1 CRLF UTF-8 4 spaces Python 3.13 (SegundoParcial) 17:33



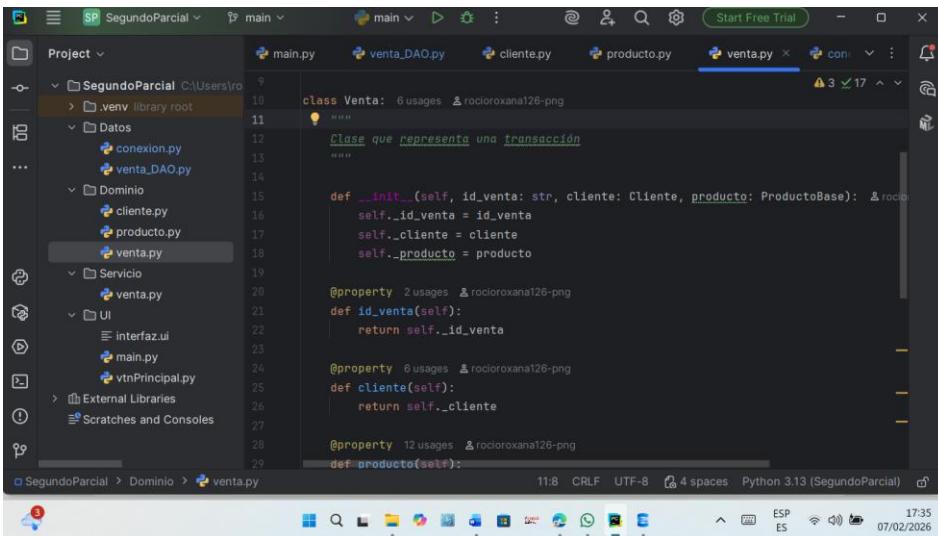
Project: SegundoParcial C:\Users\ro... / main.py / venta.DAO.py / cliente.py / conexión.py

```
6 class Cliente: 5 usages & rocoroxana126.png  
7     return self._id_cliente  
8     @id_cliente.setter 2 usages (2 dynamic) & rocoroxana126.png  
9     def id_cliente(self, id_cliente):  
10         self._id_cliente = id_cliente  
11  
12         @property 3 usages (2 dynamic) & rocoroxana126.png  
13         def nombre(self):  
14             return self._nombre  
15         @nombre.setter 2 usages (2 dynamic) & rocoroxana126.png  
16         def nombre(self, nombre):  
17             self._nombre = nombre  
18  
19         @property 3 usages (2 dynamic) & rocoroxana126.png  
20         def apellido(self):  
21             return self._apellido  
22         @apellido.setter 2 usages (2 dynamic) & rocoroxana126.png  
23         def apellido(self, apellido):  
24             self._apellido = apellido
```

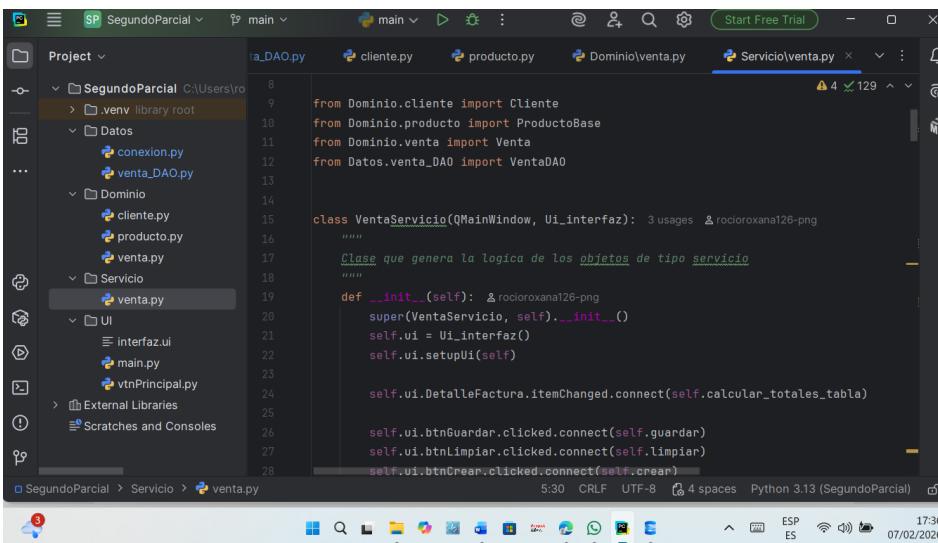
36:1 CRLF UTF-8 4 spaces Python 3.13 (SegundoParcial) 17:34



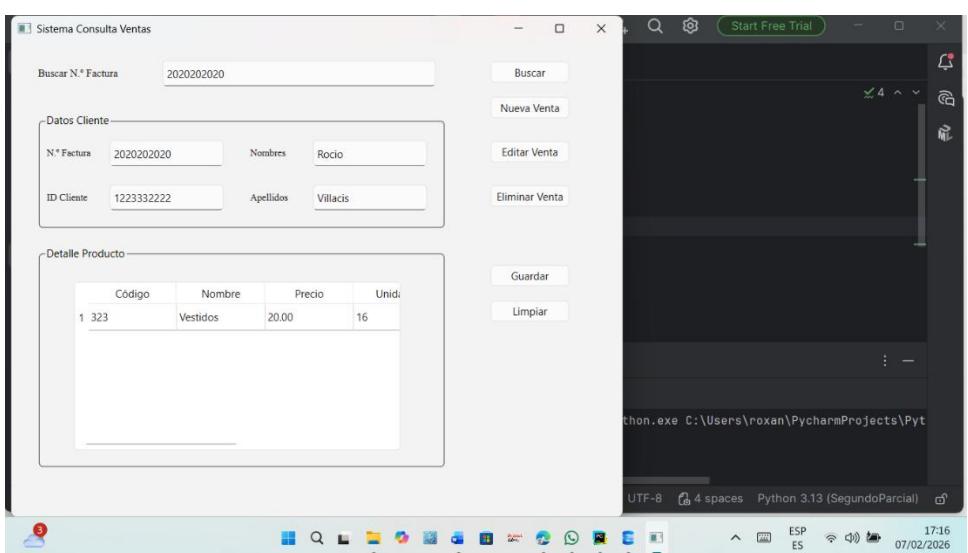
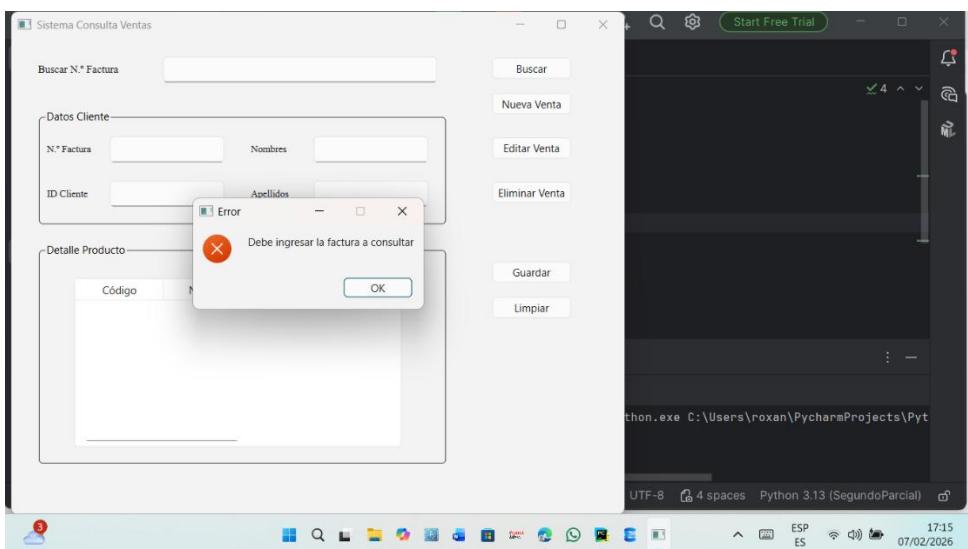
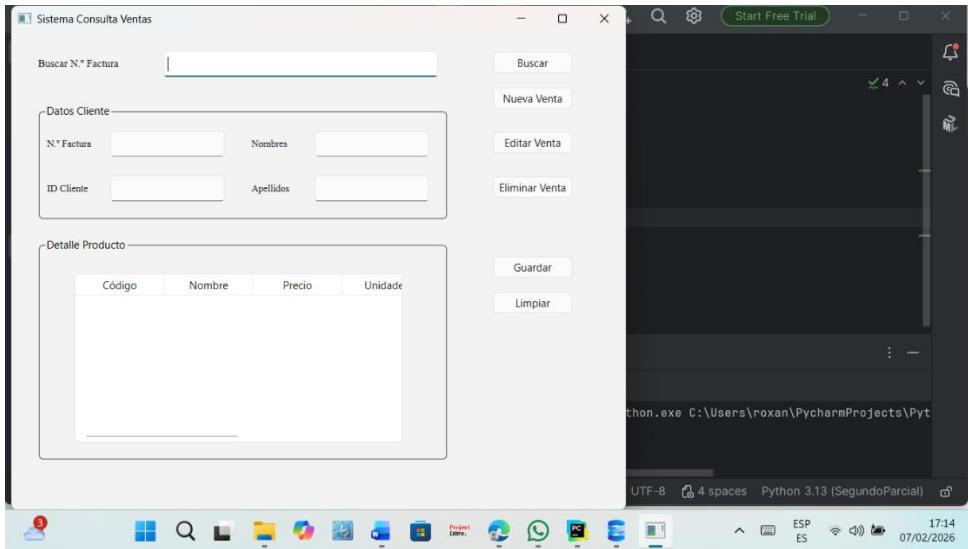
```
7
8     class ProductoBase: 7 usages & rocoroxana126.png
9         """
10            Clase que crea un objeto de tipo producto.
11        """
12
13        def __init__(self, codigo, nombre_producto, precio_base, unidades):
14            self._codigo = codigo
15            self._nombre_producto = nombre_producto
16            self._precio_base = precio_base
17            self._unidades = unidades
18
19        @property 3 usages (2 dynamic) & rocoroxana126.png
20        def codigo(self):
21            return self._codigo
22
23        @codigo.setter 2 usages (2 dynamic) & rocoroxana126.png
24        def codigo(self, codigo):
25            self._codigo = codigo
26
27        @property 3 usages (2 dynamic) & rocoroxana126.png
28        def nombre_producto(self) -> str:
29
```

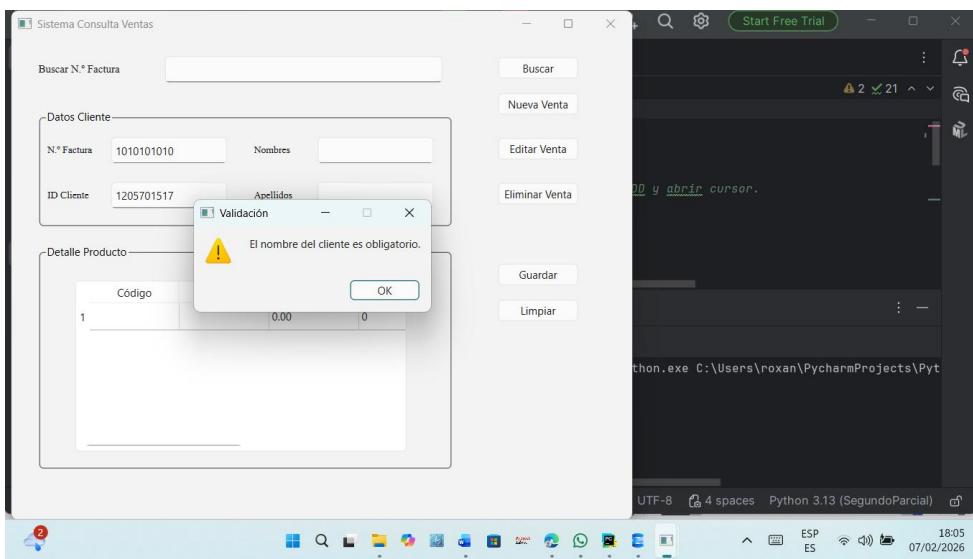
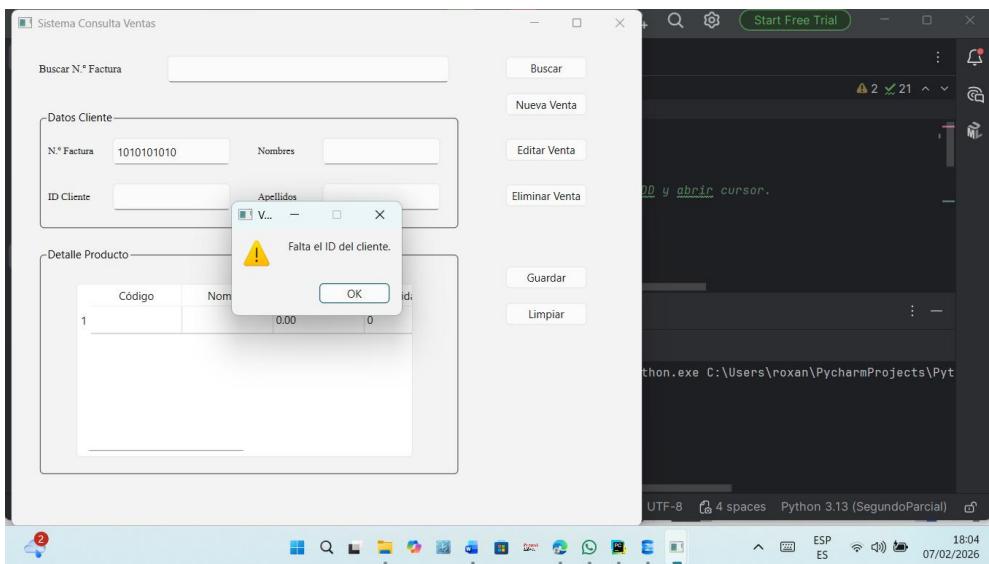
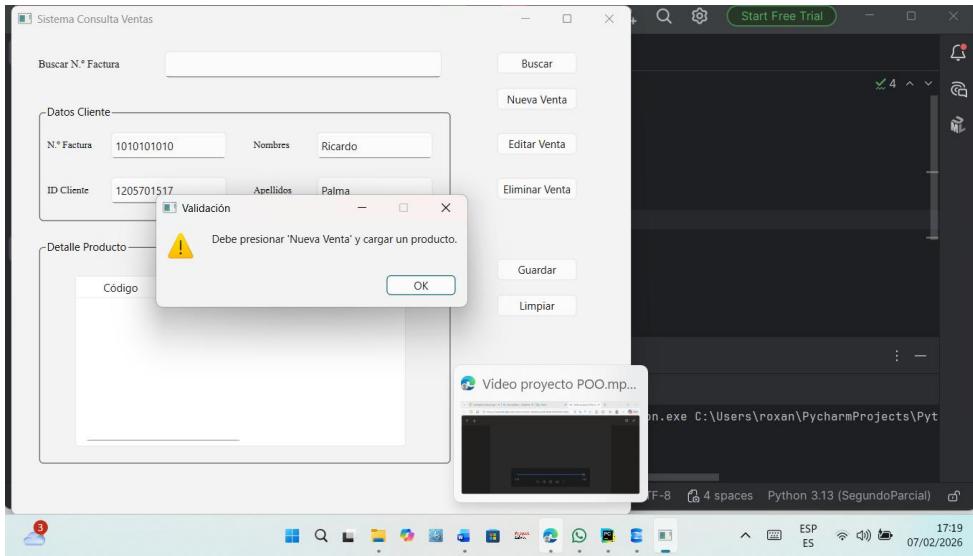


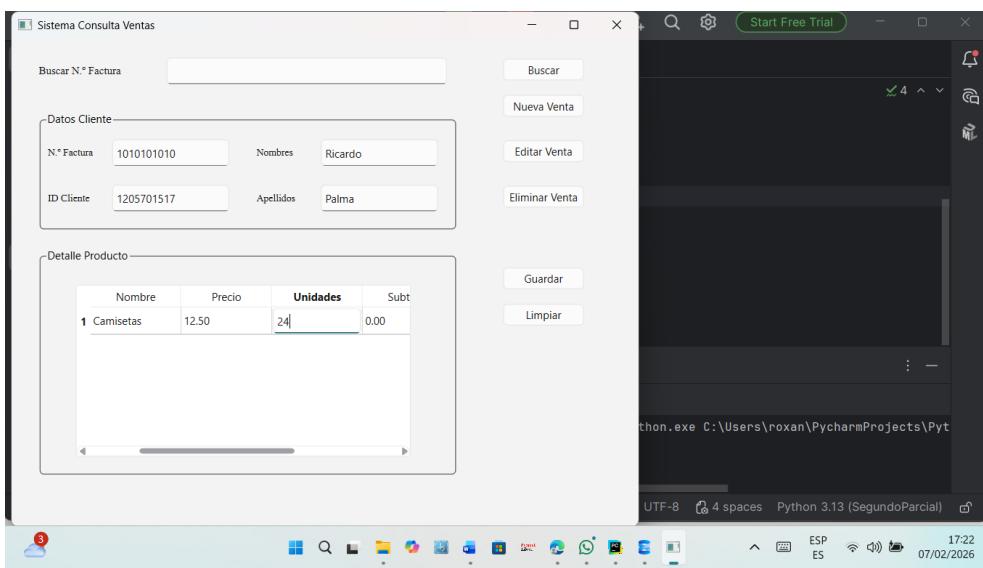
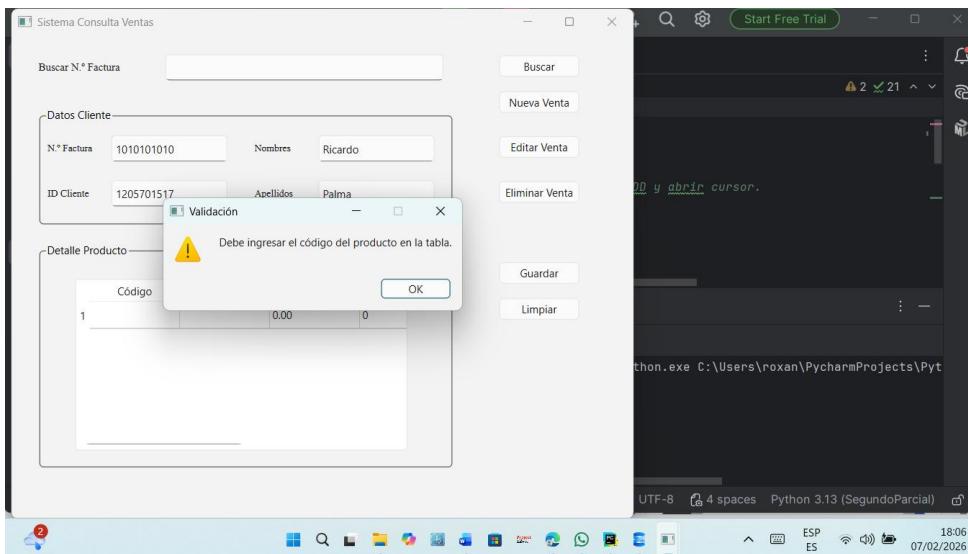
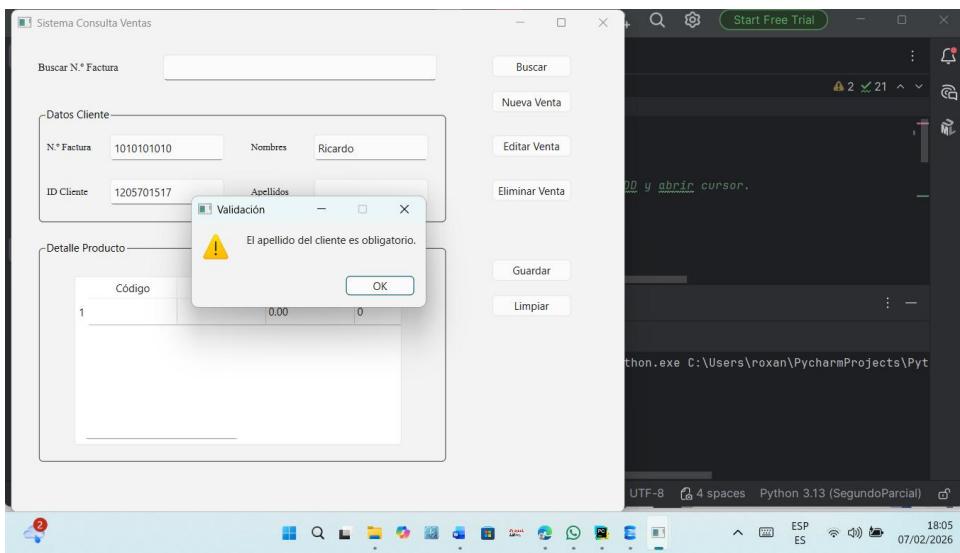
```
9
10    class Venta: 6 usages & rocoroxana126.png
11        """
12            Clase que representa una transacción.
13        """
14
15        def __init__(self, id_venta: str, cliente: Cliente, producto: ProductoBase):
16            self._id_venta = id_venta
17            self._cliente = cliente
18            self._producto = producto
19
20        @property 2 usages & rocoroxana126.png
21        def id_venta(self):
22            return self._id_venta
23
24        @property 6 usages & rocoroxana126.png
25        def cliente(self):
26            return self._cliente
27
28        @property 12 usages & rocoroxana126.png
29        def producto(self):
30
```

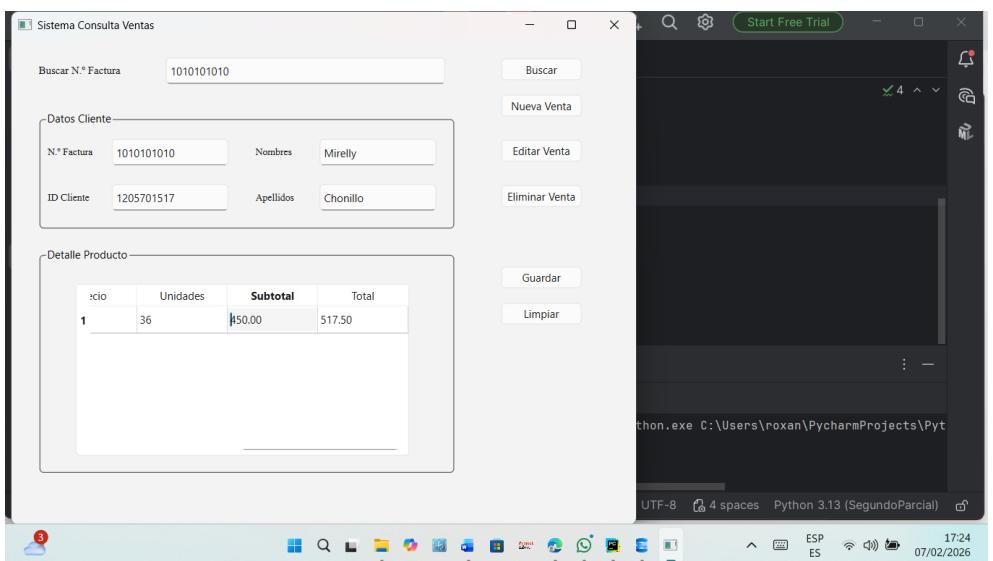
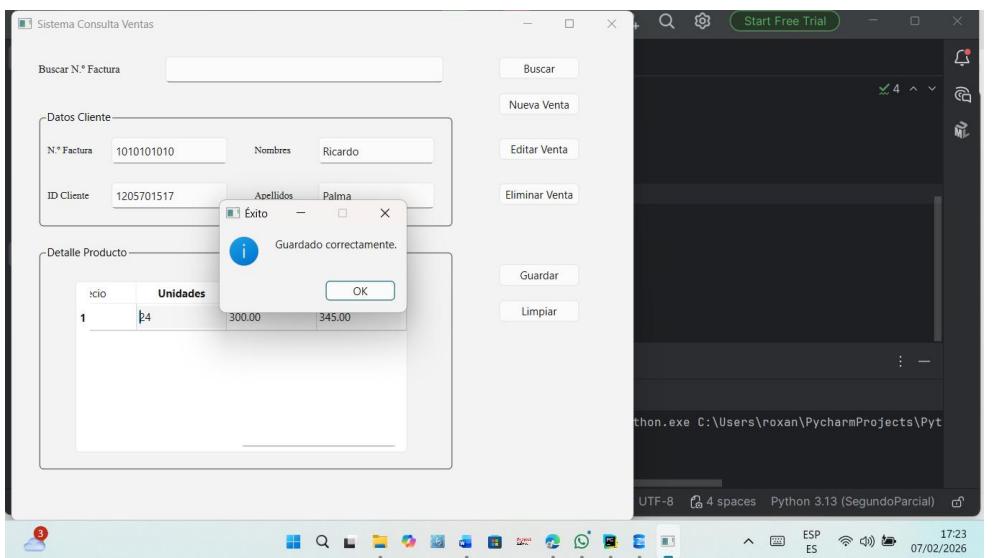
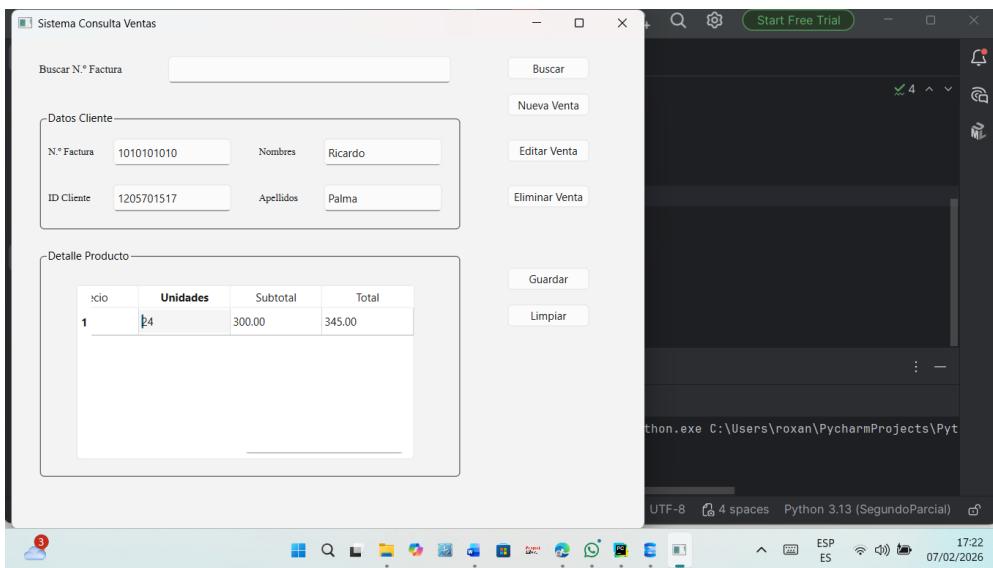


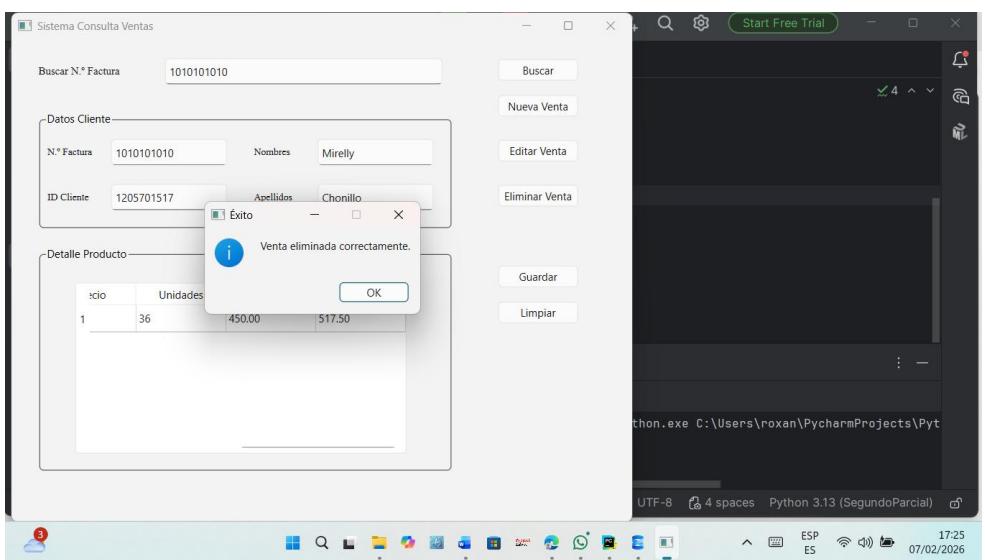
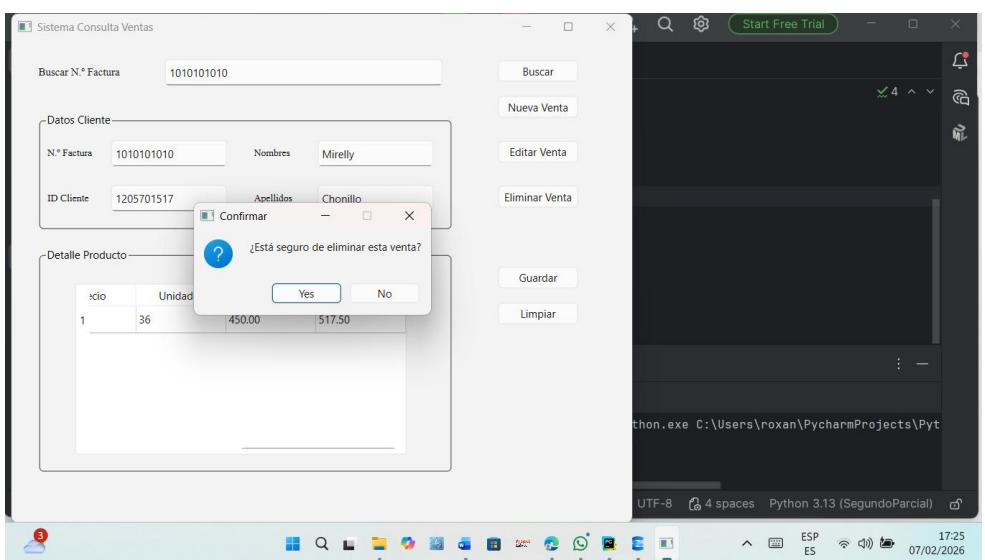
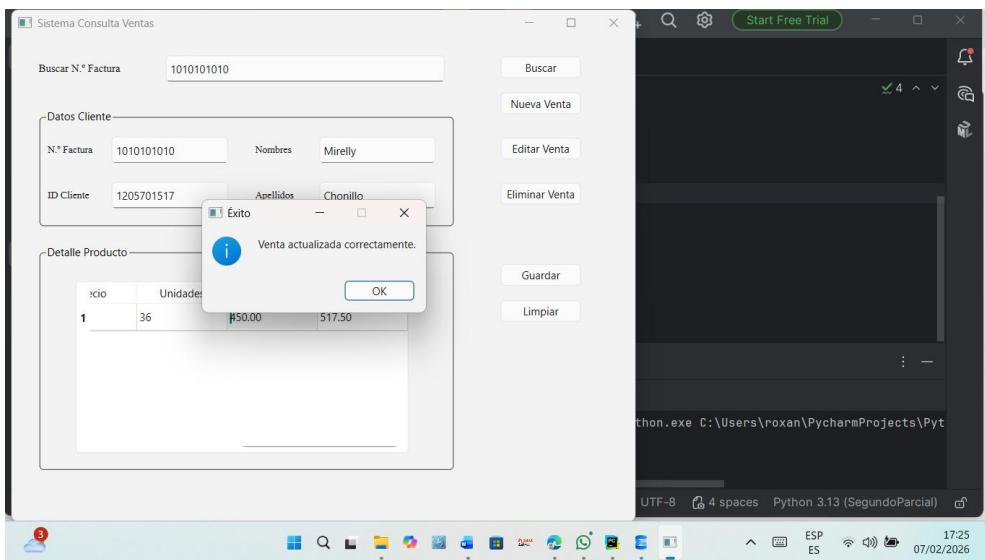
```
8
9     from Dominio.cliente import Cliente
10    from Dominio.producto import ProductoBase
11    from Dominio.venta import Venta
12    from Datos.venta.DAO import VentaDAO
13
14
15    class VentaServicio(QMainWindow, Ui_interfaz): 3 usages & rocoroxana126.png
16        """
17            Clase que genera la logica de los objetos de tipo servicio.
18        """
19
20        def __init__(self): & rocoroxana126.png
21            super(VentaServicio, self).__init__()
22            self.ui = Ui_interfaz()
23            self.ui.setupUi(self)
24
25            self.ui.DetalleFactura.itemChanged.connect(self.calcular_totales_tabla)
26
27            self.ui.btnGuardar.clicked.connect(self.guardar)
28            self.ui.btnLimpiar.clicked.connect(self.limpiar)
29            self.ui.btnCrear.clicked.connect(self.crear)
```











Link del video con la explicación de las clases:

<https://1drv.ms/v/c/420e760fd39caeac/IQDmYnk9Ef38SKo8wT1HKw8CAZ3XpyAh3z72q1FhAOm2DbY?e=4c8iem>

Nota:

En algunas capturas con coincide la hora porque me estaba olvidando de esas capturas y tuve que volver a ejecutar para sacar las capturas que me hacían falta.

Debido a que el video solo podía tener 2 minutos tuve que sacar mas capturas para completar con el requerimiento