Programación Orientado a Objetos (Herencia y Relaciones)

PAREJAS Alejandro Oquendo Bedoya
Yeison Orozco Vasco
Docente: Alejandro Rodas Vásquez
Universidad Tecnológica de Pereira

1 de noviembre de 2025

Introducción

Una de las claves para realizar este proyecto es aplicar el concepto de modularidad en la construcción de la Arquitectura de Software que soporta la aplicación.

1. Requerimientos Funcionales

Usted ha sido contratado para realizar un sistema de facturación para una tienda agrícola. Donde cada factura (o Pedido) está compuesto de los productos que serán comprados.

Esta tienda solamente maneja Productos de Control (Fertilizantes y Controles de plagas) y medicina para animales de granja, precisamente antibióticos.

Los Productos de Control tendrán como características un registro ICA, el nombre del producto y la frecuencia de aplicación (es decir, cada cuanto periodo se aplica el producto. Cada 15 días, cada 30 días, etc) así como también el valor del producto. Tenga en cuenta que el Control de Plagas y el Control de Fertilizantes son *un tipo de* Productos de Control, donde el primero tiene como característica un periodo de carencia (es el tiempo legalmente establecido, expresado usualmente en número de días que debe transcurrir entre la última aplicación de un fitosanitario y la cosecha) y el segundo la fecha de la última aplicación de este Producto.

Por otro lado, en la tienda se venden antibióticos para bovinos y porcinos donde las características de este producto son: nombre del producto, dosis (entre 400Kg y 600Kg), tipo de animal al que se le puede aplicar (Bovinos, caprinos o porcinos) y precio.

Tenga en cuenta que al ser una tienda agrícola los Clientes (con atributos nombre y cédula) son habituales por lo tanto el mismo cliente puede tener dentro de su historial de

compras, muchas Pedidos (o Facturas) asociadas. Una Factura como tal debe tener fecha en que se realizó la factura y el valor total de la compra.

> Tenga en cuenta que todos los atributos de las clases son *obligatorios*. Con esta información puede diseñar los *casos de prueba*.

2. Requerimientos de la Arquitectura de Software

Esta aplicación debe de ser construida bajo los siguientes parámetros arquitectónicos:

1. Los componentes para separar responsabilidades (Modelo, Test).
2. Utilicen el concepto de *Módulos y NameSpace*.
3. Cada *Clase* debe de estar en un archivo separado dentro del *Componente* de *Modelo*.

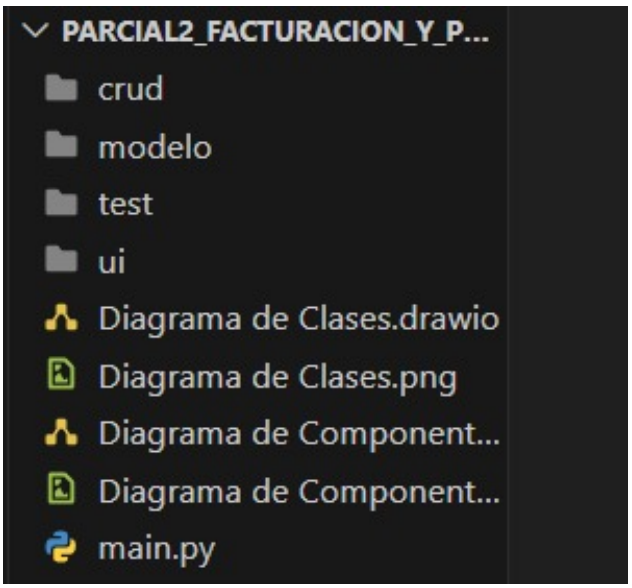

fig. 1: Arquitectura del proyecto

3. ¿Cómo realizo la entrega?

1. Usted debe de entregar el código fuente en su repositorio de *github*.
2. Pantallazos donde se corrobore que las pruebas unitarias han pasado.
3. Pantallazos donde se corrobore el uso del **debug**. En estas imagenes de debe de observar la composición del objeto. Es decir, se envidencia que el objeto x tiene *asociado* n *insntacias* del objeto y. Lo mismo con la herencia.
4. Realizar el Diagrama de Clases y Diagrama de Componentes.

4. Evidencias

Repositorio de Git:
https://github.com/Yeyei121/Parcial2_Facturacion_y_pruebas.git

4.1. Diagrama de Clases

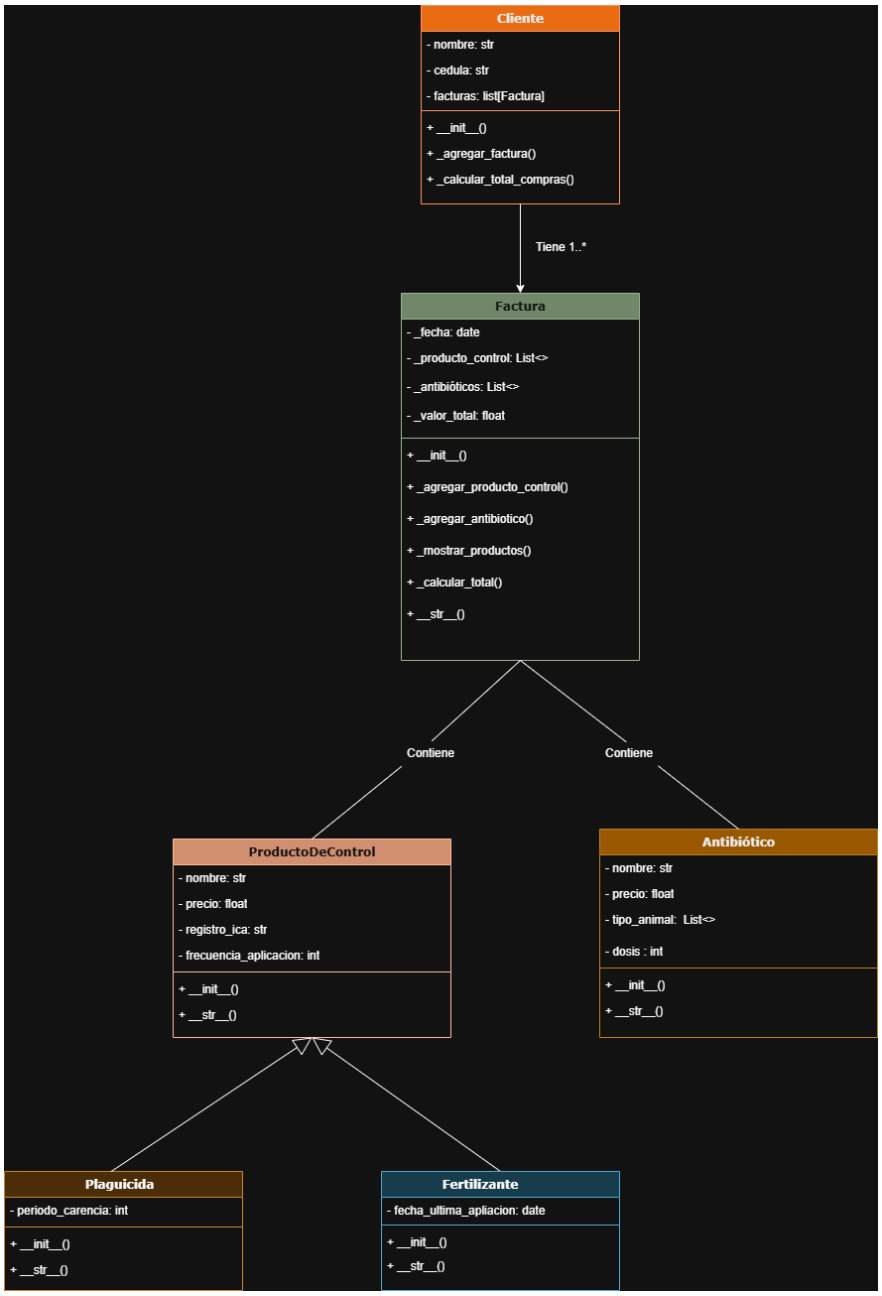

fig. 2: Diagrama de Clases

4.2. Diagrama de Componentes

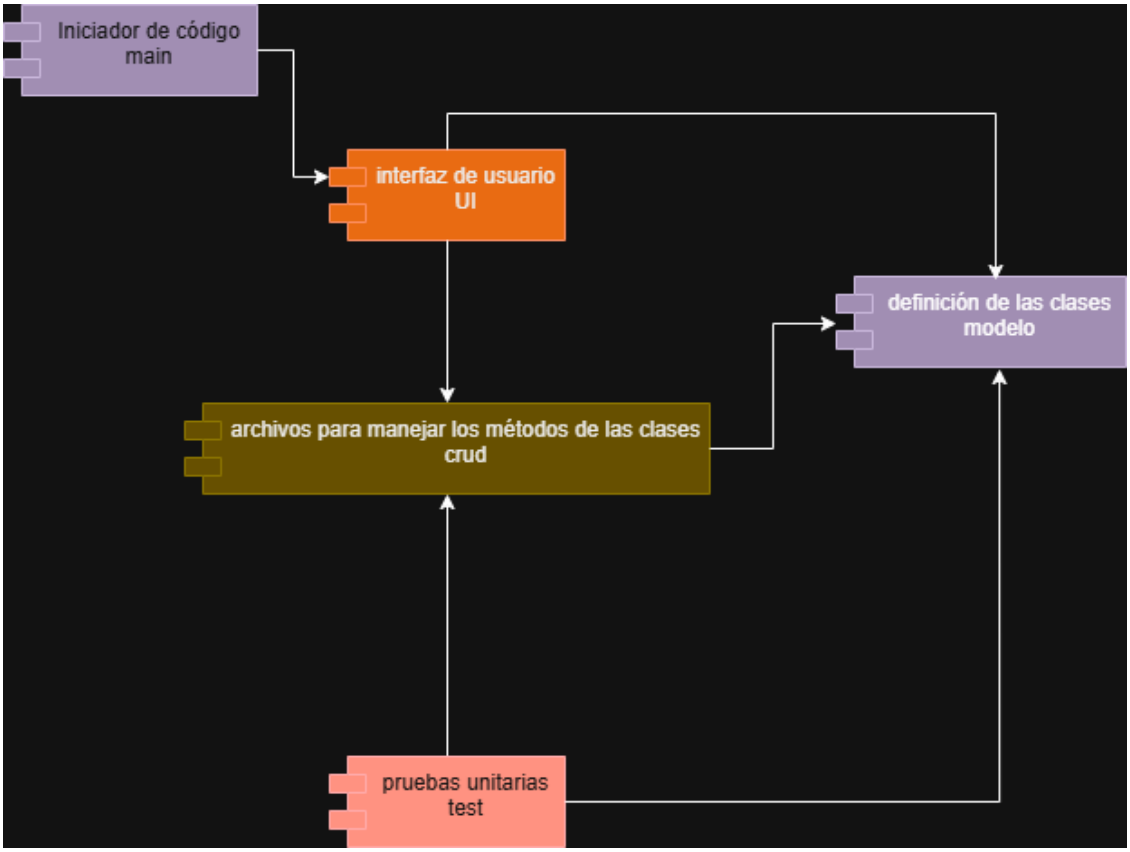

fig. 3: Diagrama de Componentes

4.3. Pantallazos Pruebas Unitarias

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\yeiso\OneDrive\Documentos\Parcial2_Facturacion_y_Pruebas> python -m unittest -v test.test_cliente

----------------------------------------------------------------------
Ran 0 tests in 0.000s

NO TESTS RAN
test_agregar_factura_a_cliente (test.test_cliente.TestFacturacion.test_agregar_factura_a_cliente) ... ok
test_agregar_varias_facturas (test.test_cliente.TestFacturacion.test_agregar_varias_facturas) ... ok
test_creacion_cliente (test.test_cliente.TestFacturacion.test_creacion_cliente) ... ok
test_obtener_facturas_vacias (test.test_cliente.TestFacturacion.test_obtener_facturas_vacias) ... ok

----------------------------------------------------------------------
Ran 4 tests in 0.000s

OK
PS C:\Users\yeiso\OneDrive\Documentos\Parcial2_Facturacion_y_Pruebas>
```

fig. 4: Pruebas unitarias para clientes

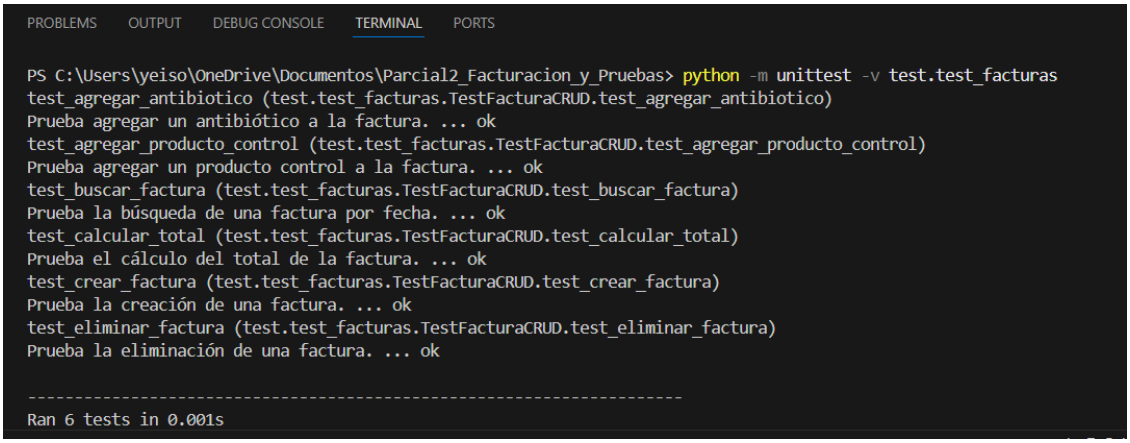

fig. 5: Pruebas unitarias para facturas

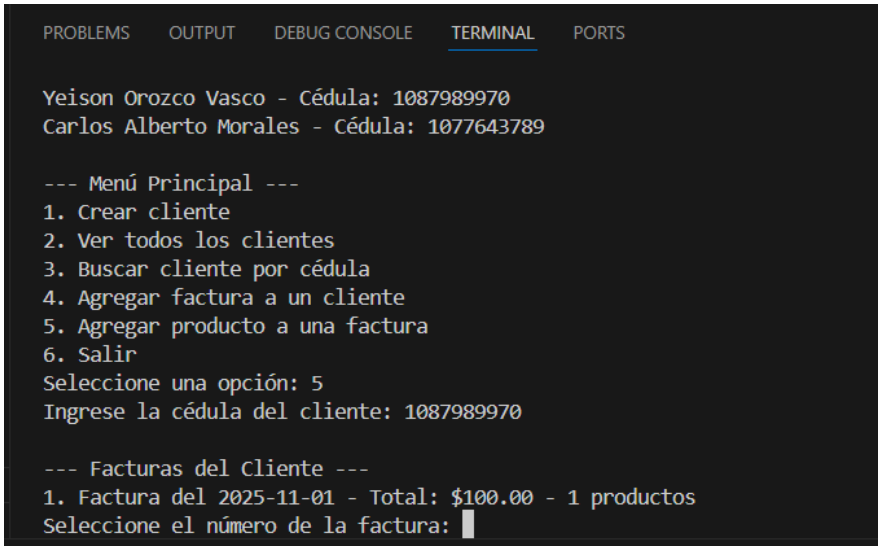

fig. 6: Interfaz del programa

4.4. Pantallazos Debug

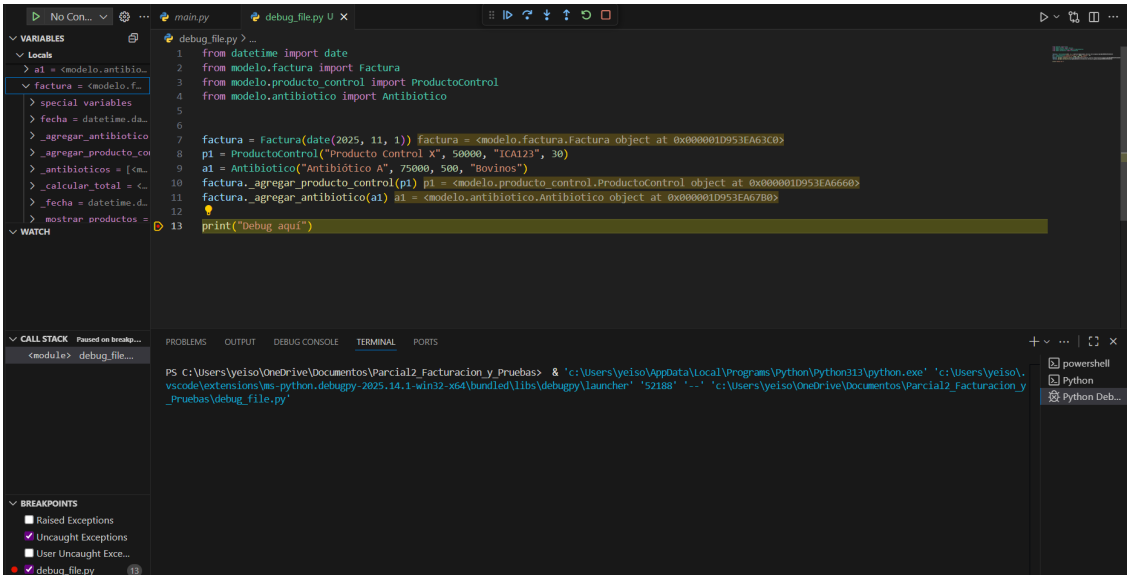

fig. 7: Debug1

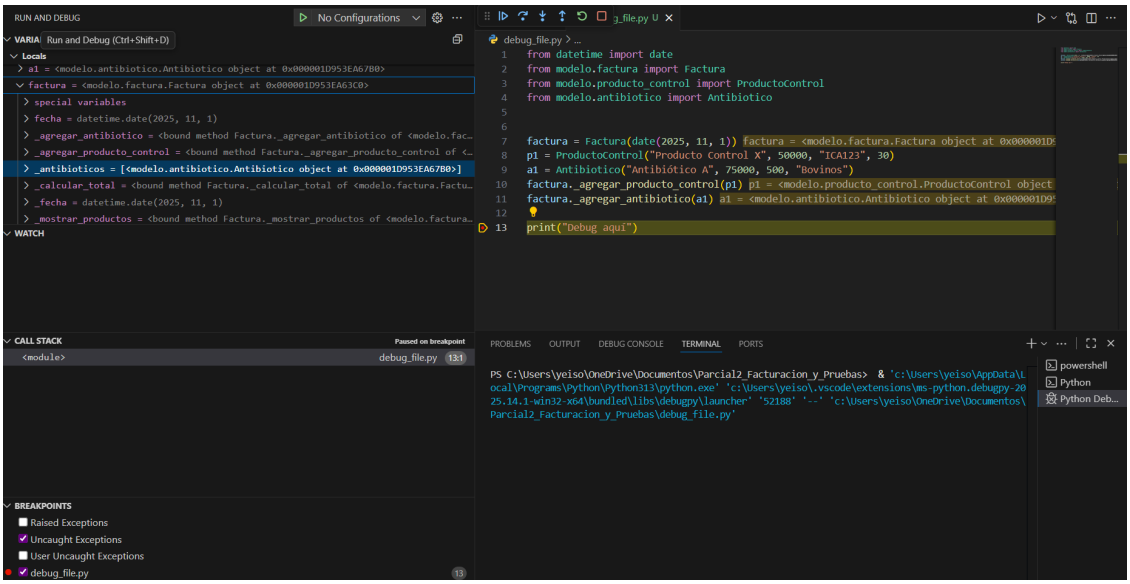

fig. 8: Debug2

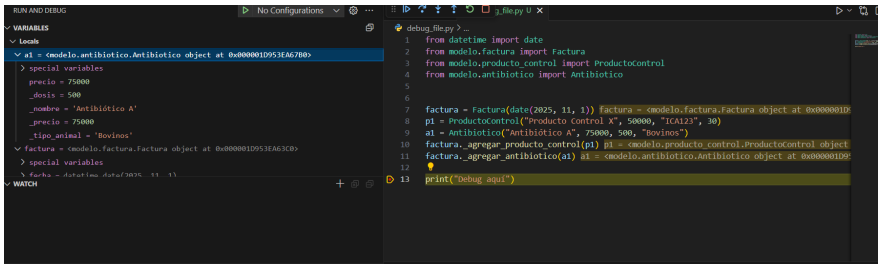

fig. 9: Debug3 (Antibióticos)

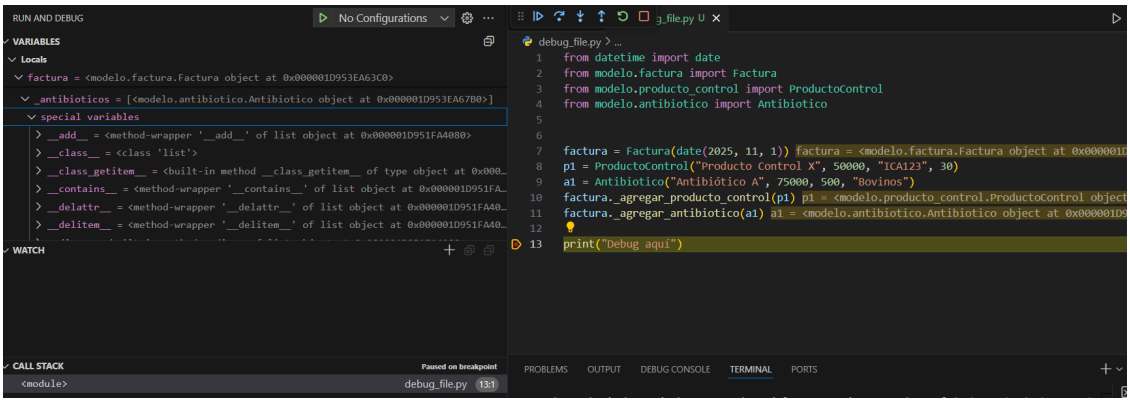

fig. 10: Debug4 Instancia de factura

fig. 11: Codigo Debug