

INFORME TECNICO COFFEE MANAGER APP

Alumno: Jhonatan Ocubere Saucedo

Docente: ING. Heber Tito Zúñiga

Materia: Diseño de Pruebas

Carrera: Ingeniería de sistemas

1. Introducción y Objetivos

CoffeeManagerApp es una aplicación desarrollada en Streamlit para la gestión de pedidos en una cafetería. Su propósito es ofrecer una interfaz accesible, visualmente atractiva y funcional, orientada tanto a la experiencia del usuario como a la presentación académica. El sistema permite registrar clientes y productos, realizar pedidos, calcular facturas con IVA, y visualizar el historial de operaciones.

Objetivos específicos:

- Diseñar una interfaz horizontal intuitiva y estética.
- Implementar funcionalidades completas de registro, listado y facturación.
- Asegurar accesibilidad visual mediante contrastes, legibilidad y estructura modular.
- Validar el sistema mediante pruebas funcionales y visuales.

2. Diseño del Sistema

La aplicación se estructura en módulos independientes:

cliente.py: define la clase Cliente.

producto.py: define la clase Producto.

pedido.py: gestiona la lógica de pedidos.

facturacion.py: calcula subtotal, IVA y total.

streamlit_app.py: interfaz principal en Streamlit.

Características visuales:

- Fondo personalizado con imagen fija.

- Paleta de colores cálidos y contrastantes.
- Tarjetas negras con bordes dorados para resúmenes.
- Sidebar con navegación clara y botones icónicos.

Flujo de navegación:

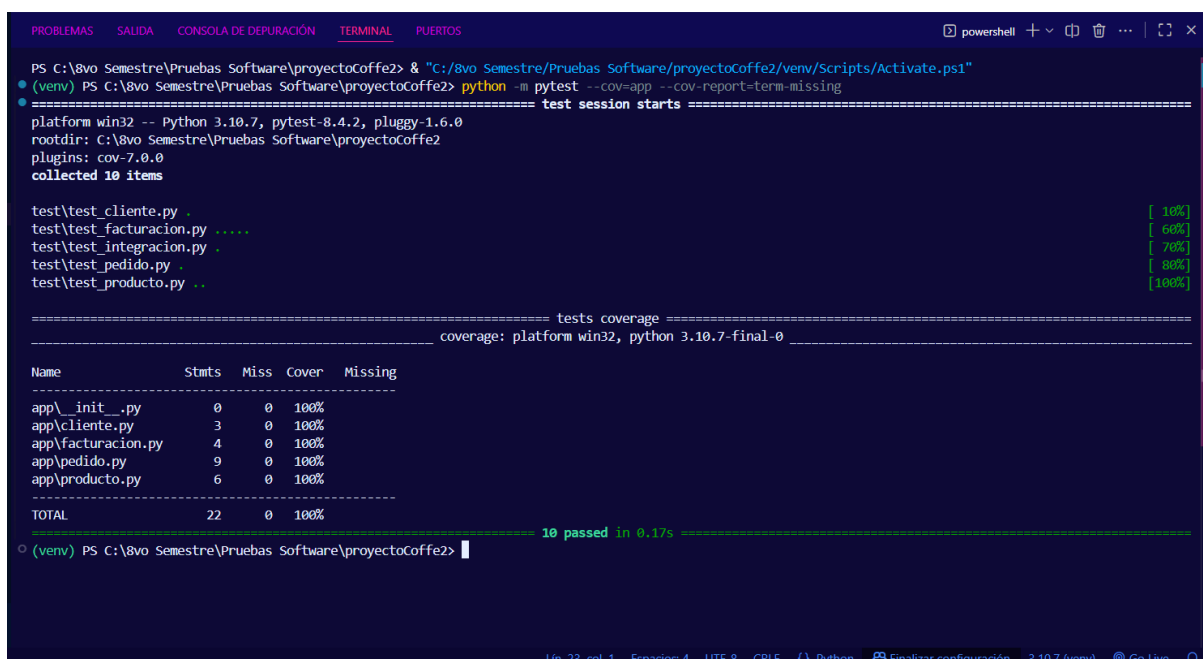
1. Realizar Pedido
2. Listar Clientes
3. Listar Productos
4. Registrar Cliente
5. Registrar Producto
6. Listar Pedidos
7. Cada vista se adapta dinámicamente según la interacción del usuario.

3. Resultados de Pruebas

Se realizaron pruebas manuales y visuales para validar:

- Registro correcto de clientes y productos.
- Cálculo preciso de facturas con IVA del 13%.
- Visualización clara del resumen de pedido.
- Persistencia de datos en `st.session_state`.
- Reinicio automático del resumen al cambiar de vista.

Captura de consola



```

PS C:\8vo Semestre\Pruebas Software\proyectoCoffe2> & "C:/8vo Semestre/Pruebas Software/proyectoCoffe2/venv/Scripts/Activate.ps1"
(venv) PS C:\8vo Semestre\Pruebas Software\proyectoCoffe2> python -m pytest --cov=app --cov-report=term-missing
===== test session starts =====
platform win32 -- Python 3.10.7, pytest-8.4.2, pluggy-1.6.0
rootdir: C:\8vo Semestre\Pruebas Software\proyectoCoffe2
plugins: cov-7.0.0
collected 10 items

test\test_cliente.py . [ 10%]
test\test_facturacion.py ..... [ 60%]
test\test_integracion.py . [ 70%]
test\test_pedido.py . [ 80%]
test\test_producto.py .. [100%]

===== tests coverage =====
coverage: platform win32, python 3.10.7-final-0

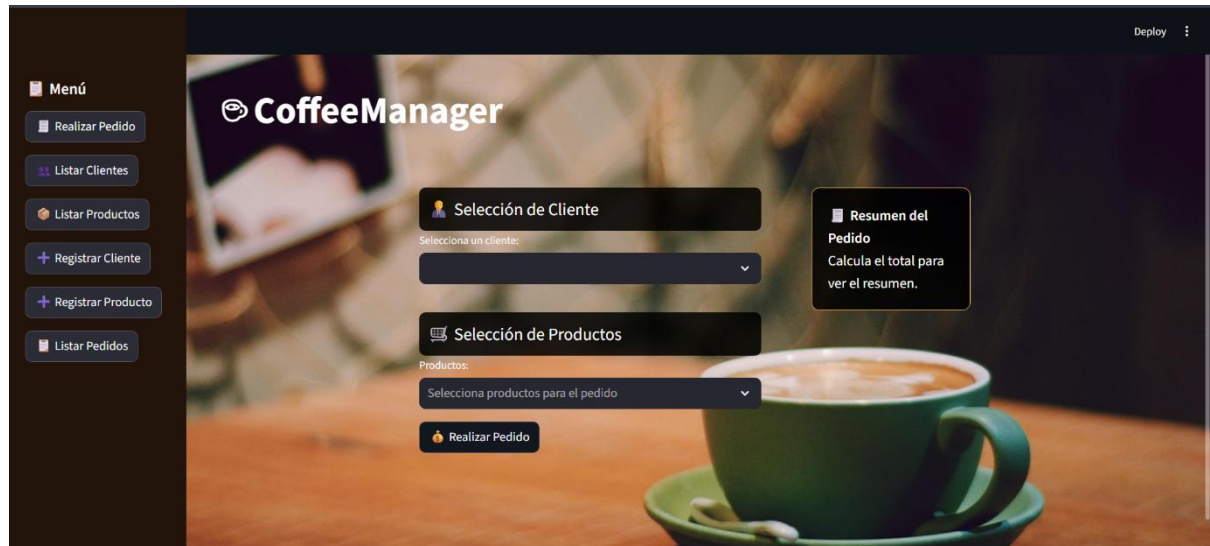
Name                Stmts  Miss  Cover   Missing
-----
app\__init__.py       0      0  100%
app\cliente.py        3      0  100%
app\facturacion.py     4      0  100%
app\pedido.py         9      0  100%
app\producto.py       6      0  100%
-----
TOTAL                 22      0  100%

===== 10 passed in 0.17s =====
(venv) PS C:\8vo Semestre\Pruebas Software\proyectoCoffe2>

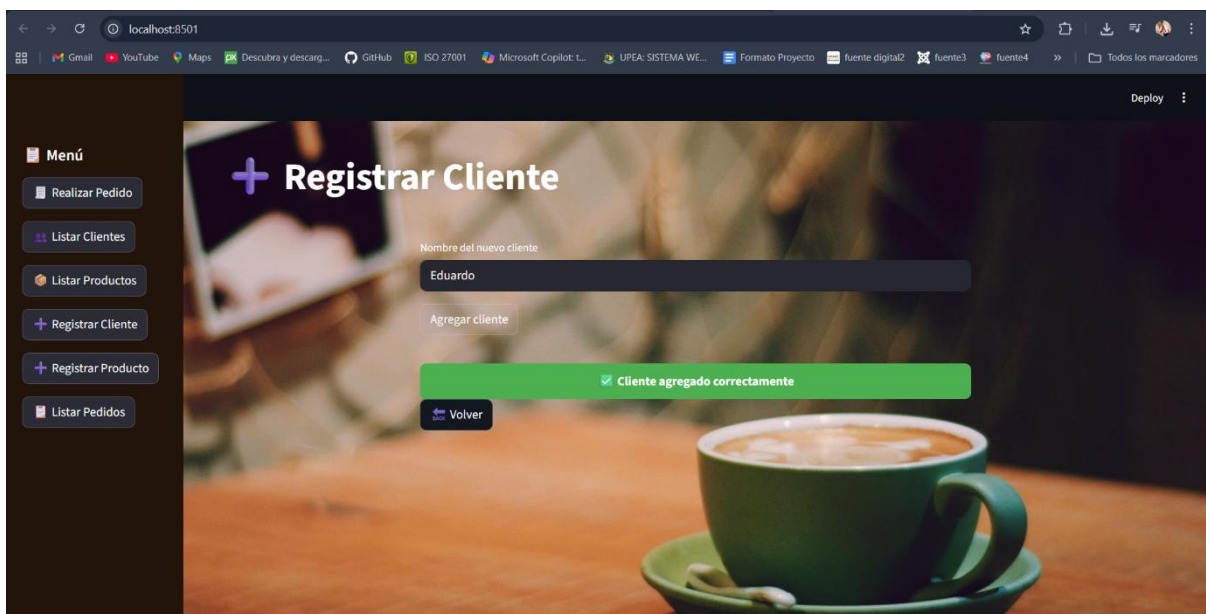
```

Capturas de Streamlit

Desplegado en Streamlit



Registro de Cliente



Registro de Producto

The screenshot shows the 'Registrar Producto' form in the CoffeeManager application. The form is displayed on a web browser at localhost:8501. The background image shows a cup of coffee on a wooden table. The form has a sidebar menu on the left with options: Realizar Pedido, Listar Clientes, Listar Productos, Registrar Cliente, Registrar Producto, and Listar Pedidos. The main form area has a title '+ Registrar Producto' and a background image of a cup of coffee. The form fields are: 'Nombre del producto' with the value 'Empanada de queso', 'Precio del producto' with the value '4.50', and a 'Agregar producto' button. A green success message 'Producto agregado correctamente' is displayed at the bottom of the form, with a 'Volver' button next to it.

localhost:8501

Menú

- Realizar Pedido
- Listar Clientes
- Listar Productos
- Registrar Cliente
- Registrar Producto
- Listar Pedidos

+ Registrar Producto

Nombre del producto

Empanada de queso

Precio del producto

4.50

Agregar producto

Producto agregado correctamente

Volver

Registro de Pedido

The screenshot shows the 'Registro de Pedido' form in the CoffeeManager application. The form is displayed on a web browser at localhost:8501. The background image shows a cup of coffee on a wooden table. The form has a sidebar menu on the left with options: Realizar Pedido, Listar Clientes, Listar Productos, Registrar Cliente, Registrar Producto, and Listar Pedidos. The main form area has a title 'CoffeeManager' and a background image of a cup of coffee. The form fields are: 'Selección de Cliente' with a dropdown menu showing 'Eduardo', 'Selección de Productos' with a dropdown menu showing 'Café' and 'Empanada de queso', and a 'Realizar Pedido' button. A green success message 'Pedido realizado correctamente' is displayed at the bottom of the form. A 'Resumen del Pedido' box on the right shows the order details: Café: 10.00 Bs, Empanada de queso: 4.50 Bs, Subtotal: 14.50 Bs, IVA (13%): 1.89 Bs, Total: 16.39 Bs.

localhost:8501

Menú

- Realizar Pedido
- Listar Clientes
- Listar Productos
- Registrar Cliente
- Registrar Producto
- Listar Pedidos

CoffeeManager

Selección de Cliente

Selecciona un cliente:

Eduardo

Selección de Productos

Productos:

Café x Empanada de queso x

Realizar Pedido

Pedido realizado correctamente

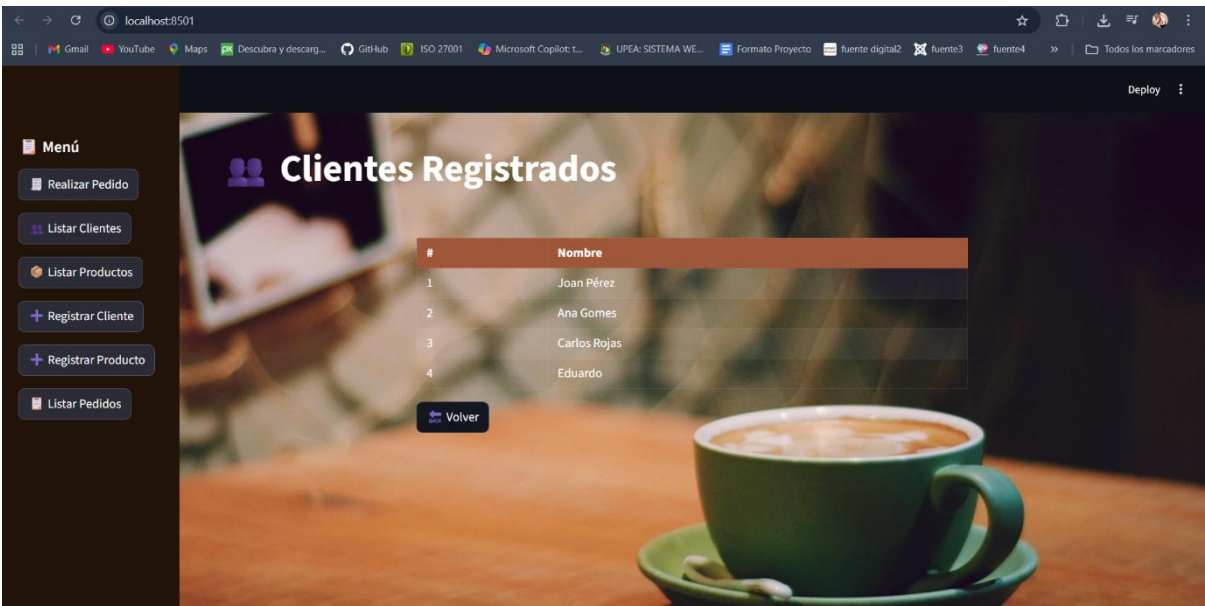
Resumen del Pedido

- Café: 10.00 Bs
- Empanada de queso: 4.50 Bs
- Subtotal: 14.50 Bs
- IVA (13%): 1.89 Bs
- Total: 16.39 Bs

Lista de pedidos realizados



Lista de Clientes



Lista de Productos



3. Enlace Publico de la App

Este es sin base de datos:

<https://coffeemanagerapp2-kqrpxiph2s9fwqpt7ugj4n.streamlit.app/>

este es con base de datos en la nube

<https://coffeemanagerapp-zhrxka8o42erdcaeljgywq.streamlit.app/>

4. Conclusiones

El desarrollo de CoffeeManagerApp permitió integrar diseño visual, lógica funcional y documentación técnica en una solución completa. La experiencia de trabajar con Streamlit facilitó la iteración rápida sobre interfaz y funcionalidad, mientras que el enfoque modular permitió mantener el código limpio y escalable.

