

## SISTEMA DE VENTA POR CATÁLOGO

**Alumno:** Estela Pimentel Richar Nikson.

**Pruebas de caja negra:** Imagina que queremos asegurarnos de que el sistema responde correctamente cuando un usuario realiza una búsqueda de productos. Probaríamos diferentes entradas de búsqueda para ver si el sistema siempre devuelve los resultados esperados sin revisar el código fuente.

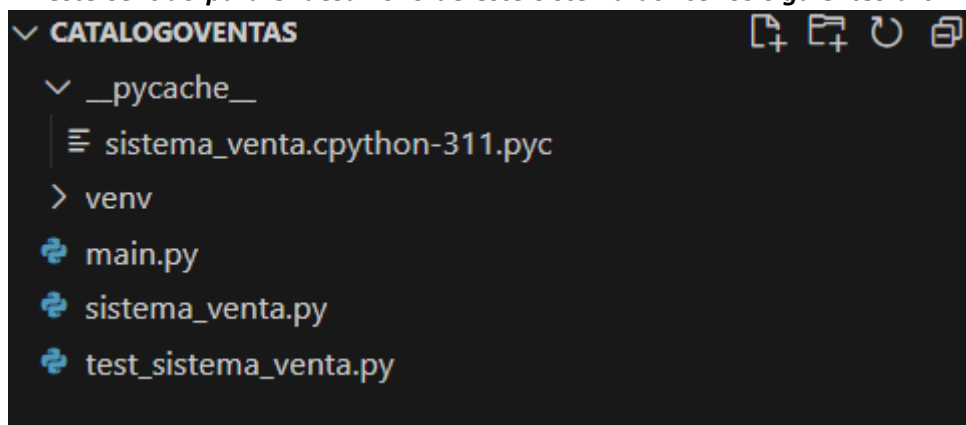
**Partición de equivalencias:** Supongamos que tenemos un campo para ingresar el número de teléfono del cliente. Podemos dividir las entradas en clases de equivalencia: números válidos, números inválidos y campos vacíos. Luego, seleccionamos un representante de cada clase para probar que el sistema maneja cada tipo de entrada adecuadamente.

**Valores límites:** Consideremos que el sistema permite a los usuarios aplicar descuentos en sus compras. Probaríamos valores justo en el límite, como el descuento mínimo y máximo permitido, así como valores fuera de estos límites, para asegurarnos de que el sistema maneja correctamente estas situaciones.

**Transición de estado:** Imaginemos que un pedido pasa por varios estados: pendiente, en proceso, enviado y entregado. Probaríamos la transición de un pedido a través de estos estados para asegurarnos de que el sistema actualiza correctamente el estado del pedido en cada paso del proceso.

**Tablas de decisión:** Supongamos que el sistema tiene diferentes reglas de envío dependiendo del país de destino y del peso del paquete. Usaríamos una tabla de decisión para probar todas las combinaciones posibles de país y peso para asegurarnos de que el sistema calcula los costos de envío correctamente en cada caso.

*En este sentido para el desarrollo de este sistema utilicé los siguientes archivos necesarios:*



Donde se habla que el **sistema\_venta.py** es un código que define un sistema de ventas por catálogo que incluye productos, un catálogo para gestionar los productos, carritos de compras para los clientes y pedidos que pueden ser confirmados. El sistema gestiona el stock de los productos y permite a los clientes agregar y quitar productos de sus carritos, calculando el total del pedido cuando se confirma. Mientras que el **test\_sistema\_venta.py** es el código que proporciona una serie de pruebas automatizadas para un sistema de ventas por catálogo, utilizando el marco de pruebas **unittest** de Python. Las pruebas cubren varios aspectos del sistema: agregar y quitar productos del catálogo y del carrito de compras, manejando casos

válidos e inválidos de acuerdo al stock disponible. Además, se prueban límites específicos de cantidad, transiciones de estado de un pedido y diferentes escenarios basados en reglas de negocio. Estas pruebas aseguran que el sistema funcione correctamente bajo diversas condiciones sin necesidad de revisar el código fuente internamente. Finalmente, el **main.py** es el código que simula el flujo de una compra en un sistema de ventas por catálogo, desde la adición de productos al carrito hasta la confirmación del pedido, y muestra cómo se manipulan y muestran los datos en cada paso.

***Seguidamente se dio a la misma ejecución de los tres archivos votando lo siguiente:***

```
PS C:\Users\Usuario\Desktop\CatalogoVentas> & C:/Users/Usuario/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Usuario/Desktop/CatalogoVentas/sistema_venta.py
PS C:\Users\Usuario\Desktop\CatalogoVentas> & C:/Users/Usuario/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Usuario/Desktop/CatalogoVentas/test_sistema_venta.py
.....
Ran 7 tests in 0.001s

OK
PS C:\Users\Usuario\Desktop\CatalogoVentas> & C:/Users/Usuario/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Usuario/Desktop/CatalogoVentas/main.py
Productos en el carrito:
+-----+-----+-----+-----+
| Producto | Cantidad | Precio Unitario | Total |
+-----+-----+-----+-----+
| Laptop   | 2        | 1500            | 3000  |
+-----+-----+-----+-----+
| Mouse    | 5        | 25              | 125   |
+-----+-----+-----+-----+
Después de quitar 3 unidades del producto Mouse:
+-----+-----+-----+-----+
| Producto | Cantidad | Precio Unitario | Total |
+-----+-----+-----+-----+
| Laptop   | 2        | 1500            | 3000  |
+-----+-----+-----+-----+
| Mouse    | 2        | 25              | 50    |
+-----+-----+-----+-----+
Pedido confirmado para el cliente Juan por un total de $3050
```

Votando así los pedidos totales, en este caso los pedidos del cliente Juan que es un total de \$3050.