**Identificación de Patrones Estructurales**

**Facade:** Se podría implementar este patrón para una mejor gestión en cuanto a funcionalidades, como lo es ComprarProducto y VenderProducto. Gracias a esto, podríamos relacionar sistemas de Producto y Vendedor, para una mejor interacción entre ellos de manera coherente y más sencilla.

**Proxy:** Podríamos utilizar el patrón Proxy como intermediario entre la pestaña de iniciar sesión y hacia donde nos dirige, es como un filtro que verifique que la información ingresada por el usuario sea correcta y haga lo correcto. Esto serviría mucho en cuanto a seguridad y verificación de cada usuario que allí ingresa.

**Composite:** Podríamos hacer el uso de este patrón para la clase Producto, dado que no solamente se podrían vender productos individuales, sino que también paquetes de productos o varios productos.

**Flyweight:** Podríamos hacer uso de él, dado que en marketPlace pueden haber distintos Productos pero que comparten muchos atributos o tienen cosas en común, por lo tanto, si queremos optimizar memoria, esto sería la mejor opción, y más si hablamos de una página de mucha diversidad de productos y tendrán similitudes con otros.