

## **Análisis de Solución.**

La solución propuesta para el sistema de gestión de pedidos es una aplicación web basada en una arquitectura Modelo-Vista-Controlador (MVC) que utiliza observadores y comandos. El sistema consta de dos aplicaciones: una aplicación cliente y una aplicación servidor, además de un balanceador de carga.

En cuanto a la arquitectura, el modelo se encarga de la gestión de los datos y la lógica de negocio, la vista se encarga de la presentación de la información al usuario y el controlador maneja las interacciones del usuario y coordina la comunicación entre el modelo y la vista. Esta arquitectura proporciona una separación clara de las responsabilidades y facilita el mantenimiento y la evolución del sistema.

La incorporación de observadores en el sistema permite que los componentes se suscriban a eventos y sean notificados cuando ocurran cambios relevantes en el estado. Esto facilita la sincronización de datos y la actualización de las vistas en respuesta a cambios en el modelo. Los observadores mejoran la modularidad y la extensibilidad del sistema al permitir una comunicación eficiente entre los componentes.

Además, se utilizan comandos para encapsular acciones específicas y desacoplar las solicitudes de los objetos que las ejecutan. Esto facilita la reutilización y la extensibilidad del sistema, ya que se pueden agregar nuevos comandos sin necesidad de modificar el código existente. Los comandos también contribuyen a la claridad y legibilidad del código al separar las responsabilidades de ejecución de acciones.

El sistema consta de dos aplicaciones: una aplicación cliente y una aplicación servidor. La aplicación cliente se ejecuta en los dispositivos de los usuarios y se encarga de interactuar con ellos a través de interfaces gráficas o interfaces de línea de comandos. La aplicación servidor gestiona la lógica de negocio, procesa las solicitudes de los clientes y proporciona los recursos necesarios para el funcionamiento del sistema. Esta separación de responsabilidades permite escalar y distribuir la carga de trabajo de manera eficiente.

Para garantizar la escalabilidad y la alta disponibilidad del sistema, se utiliza un balanceador de carga. El balanceador de carga distribuye las solicitudes entrantes de los clientes entre varios servidores para evitar la sobrecarga de un solo servidor y garantizar un rendimiento óptimo. Esto asegura que el sistema pueda manejar un alto volumen de pedidos y proporcionar una experiencia fluida a los usuarios.

En cuanto a los atributos de calidad, la solución aborda los siguientes aspectos:

**Escalabilidad:** La utilización de un balanceador de carga permite escalar horizontalmente el sistema mediante la adición de más servidores según sea

necesario. Esto garantiza que el sistema pueda manejar un aumento en la carga de trabajo y adaptarse a nuevas demandas.

Desempeño: El uso de un algoritmo eficiente de enrutamiento, como Round Robin o Hash IP, contribuye a ofrecer un alto rendimiento y tiempos de respuesta rápidos para manejar eficientemente las solicitudes de los clientes.

Modificabilidad: El patrón de diseño Adaptador utilizado en los repositorios facilita la modificabilidad del sistema al separar la lógica de negocio de la implementación específica del acceso a datos. Esto permite realizar modificaciones o mejoras en los repositorios sin afectar otras partes del sistema, lo que facilita su mantenimiento a largo plazo.

En resumen, la solución propuesta para OpenMarket es una aplicación basada en una arquitectura MVC que utiliza observadores y comandos. La utilización de un balanceador de carga, junto con el uso de observadores y comandos, garantiza la escalabilidad, el rendimiento eficiente y la modificabilidad del sistema. Estos atributos de calidad son fundamentales para ofrecer un sistema robusto y flexible en la gestión de pedidos.