**ARQUITECTURA DEL SOFTWARE**

**INTRODUCCION:**

La arquitectura de software de nuestra app de venta de electrodomésticos en línea está diseñada para garantizar la trazabilidad, mantenimiento y eficiencia en el desarrollo a continuación describiremos los patrones de diseño aplicados, la vista de componentes y vista de despliegue

PATRONES DE DISEÑOS

Para la app se tomó como patrón de creación el Singleton, que asegura que una clase tenga una única instancia y proporciona un punto de acceso global a ella.

Como patrón estructural tomamos Adapter, que permite que dos interfaces incompatibles trabajen juntas, adaptando una interfaz a otra.

Y como patrón de arquitectura Client-Server que toma un modelo en el que los clientes solicitan servicios de los servidores, y los servidores procesan las solicitudes.

VISTA DE COMPONENTES:

**Modulo de autenticación:** maneja el acceso de clientes y administradores de la app

**Módulo de gestión de pagos:** implementa métodos de pagos, validaciones y generación de comprobantes

**Modulo de reportes:** genera las facturas, envíos, reclamos, garantías y stock de marcancia

**Modulo administrativo:** permite gestionar precios, productos y clientes registrados

**IMPLEMENTACION DE LOS PATRONES EN LOS COMPONENTES**

SINGLETON: lo usamos para la conexión a la base de datos para asegurar q solo este conectado por una instancia y no generemos sobre carga

ADAPTER: lo usamos en el momento q desee integrar un sistema de pagos para procesar pagos de los clientes, debe interactuar con diferentes proveedores de pago (como PayPal, PSE y el sistema de pagos de un banco débito o crédito). Cada uno de estos proveedores tiene una API diferente.

CLIENTE-SERVIDOR: lo usamos para separar el Frontend de todas las interacciones de cada cliente q queda alojada en el servidor

**VISTA DE DESPLIEGUE**

**Servidor de aplicación:** se ejecutará la lógica de la app y los servicios REST

**Servidor de Base de Datos:** se aloja en un servicio de nube para la seguridad y respaldo de la información

**Cliente web y móviles:** interfaz de usuario accesible por los dispositivos móviles

**HERRAMIENTAS UTILIZADAS**

**FLUTTER** para desarrollar la interfaz

**NODE.JS + EXPRESS** para el Backend

**MYSQL con MYSQLWORKBRENCH** para gestionar la base de datos

**CONCLUSION:**

Utilizando una buena arquitectura de software en este caso a través de Singleton y SERVIDOR-CLIENTE garantizamos buenas practicas al momento del despliegue, documentación y desarrollo de nuestra APP asegurando q en un futuro sea fácil implementar actualizaciones al sistema