



## Diseño de Arquitectura para Sistema de Comercio Electrónico

### Introducción

El presente documento detalla el diseño de arquitectura para un sistema de comercio electrónico que será desarrollado utilizando HTML, CSS y JavaScript en el frontend, PHP en el backend, y se ejecutará en un servidor local XAMPP con una base de datos MySQL. Este sistema permitirá a los usuarios navegar por productos, realizar compras, gestionar carritos de compra, realizar pagos, entre otras funcionalidades típicas de un sistema de comercio electrónico.

La arquitectura propuesta se basa en una división clara entre el frontend y el backend, con una interfaz de usuario intuitiva y una lógica de negocio robusta. Se prestará especial atención a la seguridad de los datos y a la optimización del rendimiento del sistema.

### 1. Frontend

El frontend del sistema de comercio electrónico se encargará de la presentación de la interfaz de usuario y la interacción con los usuarios. Se utilizarán las siguientes tecnologías y componentes:

- **HTML:** Para la estructura y organización de las páginas web.
- **CSS:** Para el diseño y la apariencia visual de las páginas.
- **JavaScript:** Para la interactividad y dinamismo de la aplicación web.

### Componentes y Funcionalidades

El frontend incluirá las siguientes páginas y funcionalidades:

- Página de inicio
- Página de productos
- Página de detalles de producto
- Carrito de compras
- Página de pago
- Gestión de cuentas de usuario (registro, inicio de sesión, perfil)
- Página de edición de cuentas
- Página de Reportes
- Página de información
- chats

## 2. Backend

El backend del sistema de comercio electrónico será responsable de la lógica de negocio y la gestión de datos. Se utilizará PHP como lenguaje de servidor para implementar esta lógica.

### Componentes y Funcionalidades

El backend incluirá las siguientes funcionalidades:

1. **Gestión de usuarios:** Registro, inicio de sesión, gestión de perfiles.
2. **Gestión de productos:** Añadir, editar, eliminar productos.
3. **Gestión de pedidos:** Crear, procesar, cancelar pedidos.
4. **Autenticación y autorización de usuarios.**
5. **Seguridad:** Validación de datos de entrada, prevención de ataques comunes.
6. **Gestión de chats**
7. **Gestión de reportes**

## 3. Base de Datos

La base de datos será utilizada para almacenar información crítica del sistema, como detalles de productos, información de clientes, historial de pedidos, etc. Se utilizará MySQL como sistema de gestión de base de datos (DBMS).

### Estructura de la Base de Datos

El esquema de la base de datos incluirá las siguientes tablas y relaciones:

- Tabla de usuarios
- Tabla de publicaciones
- Tabla de carritos
- Tabla de ventas
- Tabla de Detalle de ventas
- Tabla de Reportes
- Tabla de Valoraciones
- Tabla de chats

Otras tablas según sea necesario para mantener la integridad de los datos.

## 4. Servidor Local (XAMPP)

El sistema de comercio electrónico se ejecutará en un entorno local utilizando XAMPP, que proporcionará un servidor web Apache y un servidor de base de datos MySQL.

### Configuración del Servidor

Se configurará XAMPP para ejecutar los servicios necesarios y asegurar la conectividad entre el frontend y el backend.

### **Pruebas y Depuración**

Se realizarán pruebas exhaustivas del sistema en el entorno local para identificar y solucionar problemas de rendimiento y errores.

## **5. Seguridad**

Se implementarán medidas de seguridad en todas las capas del sistema para protegerlo contra amenazas comunes.

- Medidas de Seguridad
- Cifrado de datos sensibles.
- Validación de datos de entrada.
- Prevención de inyecciones SQL y otros ataques.
- Autenticación y autorización seguras de usuarios.

### **Conclusiones**

El diseño de arquitectura propuesto proporciona una base sólida para el desarrollo de un sistema de comercio electrónico seguro, escalable y eficiente. Es importante seguir las mejores prácticas de desarrollo de software y seguridad durante todas las etapas del proyecto para garantizar un sistema robusto y confiable.

Además, la elección de tecnologías estándar y ampliamente utilizadas como HTML, CSS, JavaScript, PHP y MySQL asegura la compatibilidad y la interoperabilidad del sistema con otras plataformas y servicios. La ejecución del sistema en un entorno local con XAMPP proporciona un entorno de desarrollo y pruebas controlado, lo que facilita la detección y corrección de errores antes de su implementación en un entorno de producción.