Vinoteca MP&D

Elegancia, tradición y pasión por el vino

Universidad Libre

David Felipe Barragan Bustos, Maria Paula Barbosa Quito

Tabla de contenido

[Introducción: 2](#_Toc206500610)

[Objetivo General: 2](#_Toc206500611)

[Objetivos Específicos: 2](#_Toc206500612)

[Idea Principal del Proyecto: 2](#_Toc206500613)

[Incluye: 2](#_Toc206500614)

[No incluye (en esta fase): 2](#_Toc206500615)

[Público Objetivo: 3](#_Toc206500616)

[Modelo Arquitectónico Elegido 3](#_Toc206500617)

[ Vista Lógica: 3](#_Toc206500618)

[ Vista de Procesos: 3](#_Toc206500619)

[ Vista de Desarrollo: 3](#_Toc206500620)

[ Vista Física: 3](#_Toc206500621)

[ Vista de Escenarios: 3](#_Toc206500622)

[Componentes de la Arquitectura: 4](#_Toc206500623)

[Conectores 4](#_Toc206500624)

[Requisitos del Sistema 5](#_Toc206500625)

[Requisitos Funcionales: 5](#_Toc206500626)

[Requisitos No Funcionales: 5](#_Toc206500627)

[Stakeholders y sus Preocupaciones: 5](#_Toc206500628)

[Decisiones Arquitectónicas Clave 5](#_Toc206500629)

[Ciclo de Vida Arquitectónico 6](#_Toc206500630)

[Patrones Arquitectónicos Aplicados 6](#_Toc206500631)

[Principios de Diseño Aplicados 6](#_Toc206500632)

[Conclusión: 6](#_Toc206500633)

[Diseño: 7](#_Toc206500634)

[Portada: 7](#_Toc206500635)

# Introducción:

El comercio electrónico ha transformado la manera en que las personas adquieren productos, incluyendo alimentos y bebidas. En este contexto, el vino se ha consolidado como una bebida con creciente demanda en diversos sectores. Nuestro proyecto propone el desarrollo de una tienda virtual especializada en vinos, brindando a los usuarios una experiencia personalizada, segura y eficiente de compra en línea.

# Objetivo General:

Desarrollar una plataforma web funcional para la comercialización de vinos de distintos tipos, que facilite su compra y promueva el acceso a productos de calidad mediante una experiencia de usuario sencilla e intuitiva.

## Objetivos Específicos:

* Diseñar una interfaz moderna y accesible para el usuario final.
* Implementar una base de datos que clasifique los vinos según tipo, origen, marca y precio.
* Desarrollar un sistema de carrito de compras con métodos de pago electrónicos.
* Incorporar un módulo administrativo para la gestión de productos, usuarios y pedidos.

# Idea Principal del Proyecto:

La idea central del proyecto es crear una página web donde los usuarios puedan navegar, buscar, comparar y comprar vinos de diferentes tipos (blanco, tinto, rosado, espumoso, entre otros). La plataforma permitirá a los clientes registrarse, gestionar sus pedidos, realizar pagos en línea y acceder a recomendaciones según sus preferencias

Alcance del Proyecto:

## Incluye:

* Registro de usuarios y administración de cuentas.
* Catálogo de vinos con filtros por tipo, marca y precio.
* Carrito de compras y sistema de pagos electrónicos.
* Panel administrativo para gestionar productos y usuarios.
* Interfaz adaptable a dispositivos móviles.

## No incluye (en esta fase):

* Envío físico del producto.
* Aplicación móvil nativa.
* Recomendaciones basadas en inteligencia artificial.

# Público Objetivo:

El público objetivo está compuesto por adultos mayores de 18 años interesados en la compra de vinos para consumo personal, regalos, coleccionismo o consumo en eventos. También está dirigido a restaurantes y negocios que deseen adquirir productos de manera sencilla y rápida.

# Modelo Arquitectónico Elegido

Para el desarrollo de la plataforma Vinoteca MP&D se adopta el modelo 4+1 de vistas de Kruchten, que permite representar la arquitectura del sistema desde cinco perspectivas complementarias:

* Vista Lógica:

Representa las funcionalidades clave para el usuario, como el registro, navegación por catálogo, búsqueda de vinos, carrito de compras, pagos en línea y gestión de pedidos.

* Vista de Procesos:

Describe la concurrencia y comunicación entre componentes, como la interacción entre frontend y backend a través de API REST, y las validaciones de pago mediante la pasarela de pagos.

* Vista de Desarrollo:

Expone cómo está organizado el software en el entorno de desarrollo. Se estructura en módulos según el principio de separación de capas: presentación, negocio, acceso a datos y persistencia.

* Vista Física:

Describe el entorno de despliegue, usando servidores web en la nube, balanceadores de carga, bases de datos y almacenamiento escalable, utilizando plataformas como Heroku, AWS o Azure.

* Vista de Escenarios:

Incluye casos de uso representativos como: “usuario se registra y realiza una compra”, “administrador gestiona inventario” o “cliente consulta historial de pedidos”.

# Componentes de la Arquitectura:

La arquitectura del sistema está compuesta por los siguientes elementos:

* Frontend: Interfaz web responsive creada con HTML, CSS, JavaScript y frameworks como React o Vue. Permite la interacción directa con los usuarios y ofrece navegación por el catálogo, carrito y pagos.
* Backend: Servidor de aplicaciones basado en tecnologías como Node.js, Express o Django. Gestiona las reglas del negocio, autenticación, pagos, y operaciones de CRUD para usuarios y productos.
* Base de Datos: Motor relacional (PostgreSQL o MySQL) que almacena información de usuarios, vinos, pedidos, historial de compras y métodos de pago.
* Módulo Administrativo: Permite gestionar productos, usuarios y pedidos. Restringido a personal autorizado.
* Pasarela de pagos: Integración con servicios como Wompi, Stripe o PayU para procesar pagos en línea de manera segura.
* Sistema de autenticación: Controla los roles (cliente, administrador) y protege el acceso a las funcionalidades del sistema.

# Conectores

Los conectores permiten la interacción entre los diferentes componentes del sistema, y se definen así:

* API REST: Facilita la comunicación entre frontend y backend usando HTTP y formato JSON.
* ORM: Herramienta como Sequelize (Node.js) o Django ORM que permite manipular la base de datos como objetos desde el backend.
* HTTPS: Protocolo seguro para transmitir información sensible como credenciales y pagos.
* Webhooks: Permitirán enviar notificaciones automáticas a servicios externos (por ejemplo, confirmaciones de compra).

# Requisitos del Sistema

## Requisitos Funcionales:

* Registro e inicio de sesión de usuarios.
* Catálogo con filtros por tipo de vino, marca, país y precio.
* Carrito de compras con posibilidad de editar y eliminar productos.
* Confirmación y seguimiento de pedidos.
* Módulo administrativo para gestión de inventario y usuarios.

## Requisitos No Funcionales:

* Seguridad: Cifrado de contraseñas, protección de datos, HTTPS.
* Rendimiento: Respuesta rápida ante múltiples usuarios simultáneos.
* Escalabilidad: Capacidad de crecimiento sin afectar el servicio.
* Mantenibilidad: Código desacoplado y documentado.
* Compatibilidad: Multiplataforma y multidispositivo.

# Stakeholders y sus Preocupaciones:

|  |  |  |
| --- | --- | --- |
| Stakeholder | Preocupación Principal | Métrica de Éxito |
| Dueño | Rentabilidad del sistema | ROI, ventas mensuales |
| Usuario Final | Experiencia de compra | Satisfacción, tiempo de respuesta |
| Desarrolladores | Modularidad y extensibilidad | Facilidad de cambios |
| Administradores | Control eficiente del sistema | Tiempo de gestión de pedidos |

# Decisiones Arquitectónicas Clave

* **Arquitectura en capas:** Facilita el mantenimiento y evolución del sistema.
* **Tecnologías abiertas y documentadas:** Node.js, React, PostgreSQL.
* **Seguridad como prioridad:** HTTPS, cifrado, autenticación por roles.
* **Despliegue en la nube:** Flexibilidad para escalar y mejorar disponibilidad.
* **Diseño centrado en el usuario:** Enfocado en accesibilidad, velocidad y facilidad de uso.

# Ciclo de Vida Arquitectónico

* **Conceptualización:** Definición de la visión y necesidades del cliente.
* **Diseño:** Creación de la arquitectura, selección de patrones, tecnologías y modelos.
* **Implementación:** Desarrollo, pruebas, integración y despliegue.
* **Evolución:** Escalamiento del sistema, mejoras continuas, nuevas funcionalidades

# Patrones Arquitectónicos Aplicados

* **Arquitectura en capas:** Para separar responsabilidades y facilitar pruebas/modificaciones.
* **Cliente-servidor:** Navegador del cliente como consumidor y servidor web como procesador central.
* **MVC:** Organización del frontend y backend para separar lógica, datos e interfaz.
* **Microservicios:** Considerados para escalar módulos críticos como pagos o recomendaciones.

# Principios de Diseño Aplicados

* **SRP:** Cada clase o módulo tiene una única responsabilidad.
* **DRY:** Reutilización de componentes y reducción de código duplicado.
* **KISS:** Soluciones simples, claras y fáciles de mantener.
* **Open/Closed Principle:** Abierto a extensiones, pero cerrado a modificaciones directas.
* **Encapsulación:** Protección del acceso a datos mediante métodos controlados.
* **Inversión de dependencias:** Uso de interfaces para permitir flexibilidad de componentes internos.

# Conclusión:

Este proyecto busca ofrecer una solución digital efectiva para la venta de vinos en línea, facilitando la experiencia de compra tanto para usuarios particulares como para negocios, mediante una interfaz amigable y una estructura tecnológica sólida.

# Diseño:

## Portada:

