Documento de Arquitectura del Software: Mi Primera Borrachera

Versión	Fecha	Descripción
1.0	04/10/2024	versión inicial del documento de arquitectura

1. Introducción

Este documento describe la arquitectura del software desarrollado para "Mi Primera Borrachera", un bar con cuatro sedes ubicadas en Restrepo, 1ra de Mayo, Galerías y Chía. El objetivo de esta solución tecnológica es mejorar la gestión integral de operaciones mediante un sistema unificado que centraliza la administración del inventario, pedidos y pagos. Cada rol (cajero, mesero, administrador) cuenta con funcionalidades específicas para realizar sus tareas con mayor eficacia y control, asegurando la optimización del servicio en cada sede.

La implementación de este software ofrece una plataforma robusta que responde a las necesidades específicas del bar, tales como la gestión individualizada de inventarios por sede y el control eficiente de las mesas y pedidos. Esta solución asegura un flujo de trabajo eficiente, proporcionando acceso en tiempo real a información crítica para la toma de decisiones.

Buenas Prácticas de Desarrollo:

Selección de Tecnologías:

- Elegir tecnologías adecuadas a los requisitos del sistema, asegurando que sean escalables y de fácil mantenimiento. Por ejemplo, utilizar React o Angular para el frontend y Node.js o Python para el backend, según la compatibilidad con los módulos de inventario, pedidos y pagos.

Control de Versiones de Código:

- Implementar un sistema de control de versiones con Git, permitiendo un seguimiento detallado de los cambios en el código y facilitando la colaboración entre los desarrolladores. Establecer ramas claras para el desarrollo de nuevas funcionalidades, corrección de errores y pruebas.

Desarrollo de Componentes Reutilizables:

- Crear componentes reutilizables que puedan ser implementados en diferentes partes del sistema. Por ejemplo, un módulo de gestión de inventarios puede ser reutilizado en distintas sedes sin necesidad de reescribir el código. Esto mejora la eficiencia y reduce la duplicación de código.

Calidad del Código:

- Mantener altos estándares de calidad mediante el uso de revisiones de código y análisis automatizados con herramientas como ESLint. Asegurar que el código sea legible y mantenible siguiendo principios de diseño limpio (Clean Code).

Pruebas Unitarias y de Integración:

- Implementar pruebas unitarias para validar el correcto funcionamiento de los módulos individuales del sistema, como la gestión de pedidos o el cierre de cuentas. Utilizar herramientas como Jest o Mocha para JavaScript o el entorno correspondiente a las tecnologías seleccionadas.

- Realizar pruebas de integración para asegurar que los distintos módulos del sistema, como inventario y pagos, interactúan correctamente entre sí.

Manejo Eficiente de Errores:

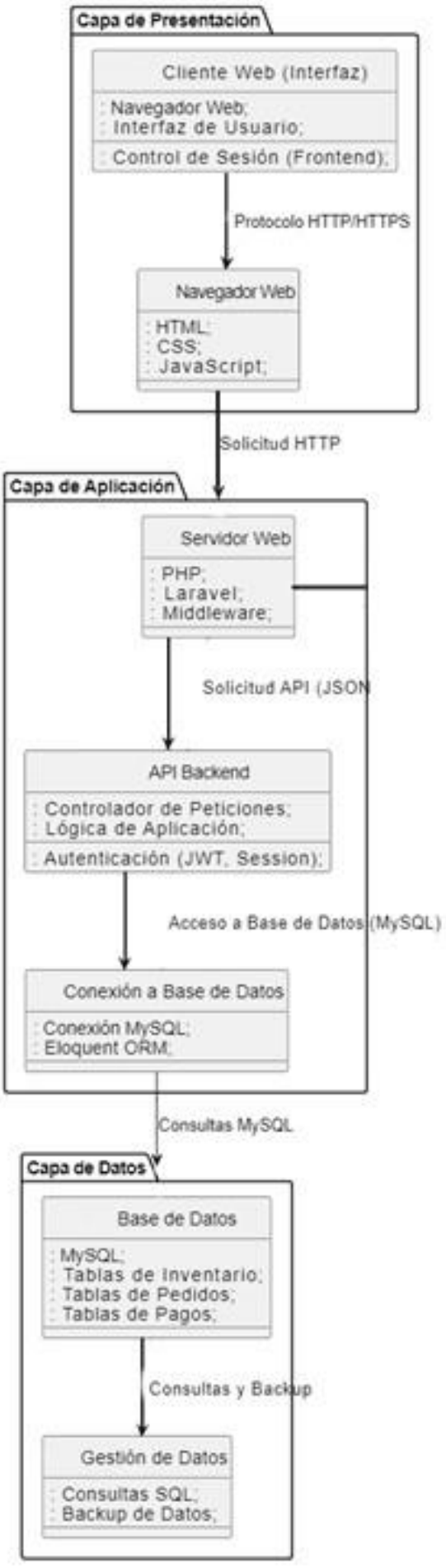
- Implementar un manejo adecuado de errores que permita registrar problemas técnicos sin interrumpir el servicio. Los mensajes de error deben ser claros para los desarrolladores y amigables para el usuario final, evitando exponer detalles técnicos innecesarios.

Características destacadas del software:

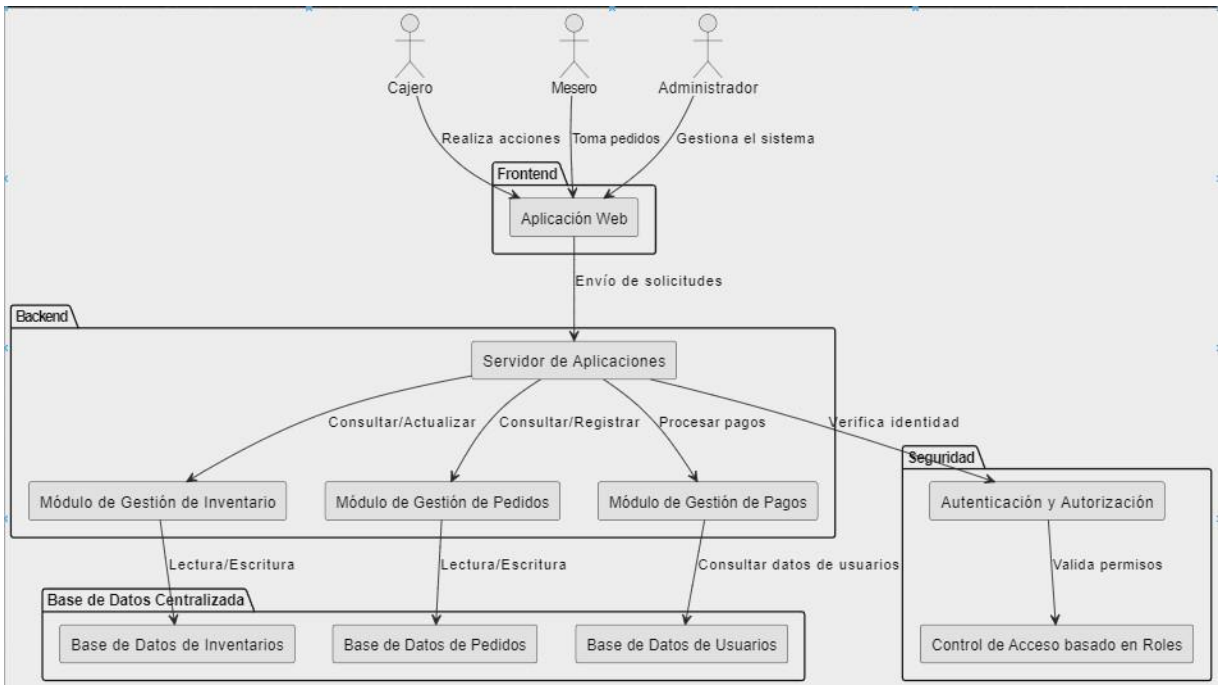
- **Manejo de Inventario por Sede:** Cada cajero es responsable de gestionar el inventario de su sede, sin posibilidad de consultar o modificar el inventario de otras sedes, asegurando una administración precisa y autónoma.

- **Gestión de Pedidos y Mesas:** Los meseros podrán asignar mesas, tomar pedidos y consultar el inventario disponible en su sede para cumplir con los pedidos, evitando la duplicación de tareas al marcar mesas como ocupadas.
- **Pagos Simulados:** Los cajeros podrán cerrar pedidos mediante una simulación de pagos, sin necesidad de utilizar sistemas de pago externos o integraciones con APIs de terceros.
- **Reportes Generados:** Los administradores tendrán acceso a reportes descargables en formatos CSV y XLSX, lo que facilita el análisis y supervisión de las operaciones diarias y de inventario.
- **Sistema de Roles:** El sistema asigna roles específicos a cajeros, meseros y administradores, garantizando que cada uno tenga acceso únicamente a las funciones necesarias para sus tareas operativas. El administrador, además de las funciones del cajero y mesero, podrá parametrizar el sistema y gestionar los usuarios.

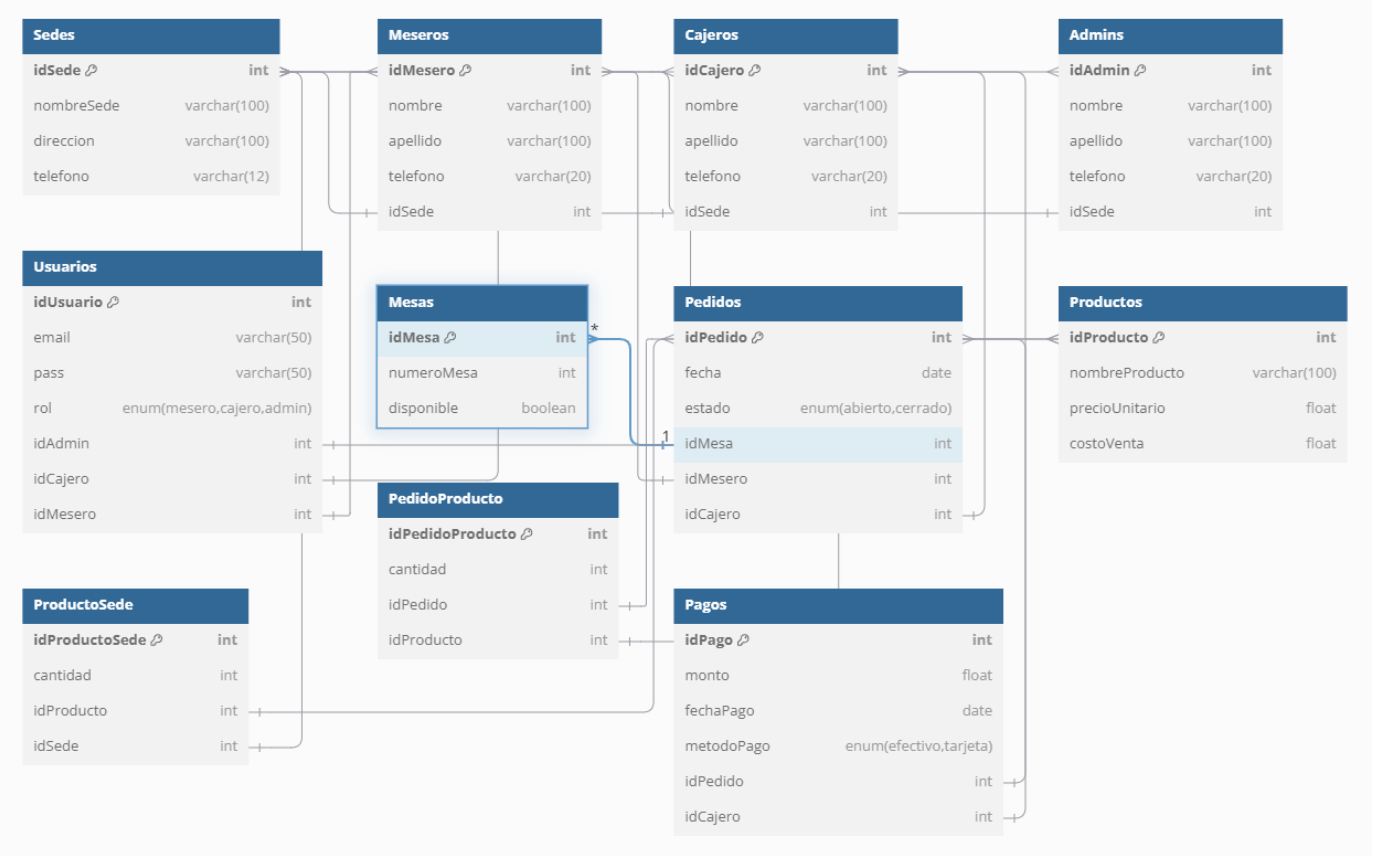
2. Diagrama de Arquitectura



3. Diagrama de Despliegue



4. Modelo Entidad-Relación (ERD)



Relaciones:

Sedes → Meseros:
Cada mesero está asociado a una sede específica.

Sedes → Cajeros:
Cada cajero está asociado a una sede específica.

Sedes → Admins:
Cada administrador está asociado a una sede específica.

Admins → Usuarios:
Un usuario puede estar asociado a un administrador específico.

Cajeros → Usuarios:
Un usuario puede estar asociado a un cajero específico.

Meseros → Usuarios:
Un usuario puede estar asociado a un mesero específico.

Mesas → Pedidos:
Un pedido está asociado a una mesa específica.

Meseros → Pedidos:
Un mesero está asociado a un pedido específico.

Cajeros → Pedidos:
Un cajero está asociado a un pedido específico (al momento de pago).

Productos → ProductoSede:
Un producto puede estar disponible en varias sedes, con una cantidad asociada en cada sede.

Sedes → ProductoSede:
Una sede tiene una cantidad específica de productos.

Pedidos → PedidoProducto:
Un pedido puede contener múltiples productos.

Productos → PedidoProducto:
Un producto puede estar en varios pedidos.

Pedidos → Pagos:
Un pago está asociado a un pedido específico.

Cajeros → Pagos:
Un cajero está asociado a un pago específico (al realizar el pago del pedido).