

Laboratorio Arquitectura de Software

Juan José Campo López-2453629
Juan Diego Holguin Perez-2453866
Luis Alfonso Mosquera Nene-2453885

Tecnología en Desarrollo de Software

Desarrollo de Software I

Luis Fernando Abanis

Universidad del Valle
Sede Santander de Quilichao

Taller Práctico: Diseño de Software y Arquitectura de Sistemas

Caso de Estudio: sistema de pedidos de transporte especializada para personas con discapacidad visual

Objetivo del Taller

Aplicar conceptos de Diseño de Software y Arquitectura de Sistemas, utilizando Patrones de Diseño, Arquitectura en Capas y el Modelo Vista Controlador (MVC), en el desarrollo de un Sistema de pedidos de transporte tanto público (taxi) como privado (particular), incorporando pagos en línea e integración con aplicaciones de transporte privado

Paso 1: Definición del Caso de Uso

El sistema permitirá a los usuarios finales pedir, modificar y cancelar rutas por medio de comando de voz con distintas aplicaciones, realizar pagos en línea y visualizar historial de viajes con los sistemas de transporte externo.

Actores Principales:

1.-Usuario final → Registra una cuenta, solicita ruta, realiza el pago y consulta su historial de viajes mediante comando de voz.

2.-Conductor → Configura su disponibilidad y tiene que seguir la ruta establecida obligatoriamente por la app.

3.-Administrador → Supervisa el sistema y la integración con las aplicaciones de transporte externo.

4.-Pasarela de Pago → Procesa pagos con tarjeta de débito y billeteras digitales externa o de la app.

5.-Sistema de transporte externo → visualiza historiales de rutas y proporciona datos sobre rutas previas.

Requisitos Funcionales:

- Registro y autenticación de usuarios.
- Reserva, modificación y cancelación de rutas.
- Notificaciones de confirmación por notificación de voz.
- Historial de rutas del usuario final.
- Pagos en línea seguros mediante plataformas como nequi o billeteras virtuales.
- Sincronización de datos con sistema de transporte externo.

Paso 2: Elección de la Arquitectura del Sistema

Arquitectura en Capas:

1.- Capa de Presentación (Front-End):

- Aplicación móvil para acceso rápido.
- Aplicación opcional para reloj inteligente de sincronización

2.- Capa de Aplicación (Back-End):

Microservicios en Node.js para la gestión de usuarios, rutas, pagos e integración con sistemas de transporte externo.

3.- Capa de Datos:

- Base de datos SQL para almacenar rutas de usuarios y datos básicos.

Patrones de Diseño Utilizados:

- ✓ Factory Pattern → Para la creación de objetos como usuarios, rutas y pagos.
- ✓ Singleton Pattern → Para gestionar la conexión con la base de datos.
- ✓ Repository Pattern → Para centralizar las operaciones sobre la base de datos.
- ✓ Adapter Pattern → Para conectar con sistemas de transporte externo.

Modelo Vista Controlador (MVC):

- Modelo: Define la estructura de datos de usuarios, rutas y pagos.
- Vista: Interfaces gráficas para la gestión de rutas y pagos.
- Controlador: Procesa las solicitudes del usuario y actualiza la vista.

Paso 3: Diseño de la Base de Datos

La base de datos incluirá las siguientes tablas:

Usuarios: Contendrá información de usuarios.

Rutas: Registro de todas las rutas seguras y utilizadas anteriormente.

Pagos: Historial de pagos realizados por los usuarios finales.

Historiales de rutas: Datos de las rutas más frecuentes de los usuarios y visualizarlos con sistemas de transporte externos.

Notificaciones: Almacenará alertas y confirmaciones enviadas a los usuarios.

Relación entre Tablas:

- Un usuario puede agendar varias rutas frecuentes.
- Una ruta segura pertenece a un usuario y un conductor la tiene que seguir.
- Un historial de rutas pertenece solo al usuario y se visualiza con sistemas de transporte externo.

Paso 4: Definición de los Módulos del Sistema

1.- Módulo de Autenticación y Gestión de Usuarios

- Permite el registro e inicio de sesión de usuarios.
- Los usuarios deben proporcionar sus datos básicos.

2.- Módulo para solicitar transporte

- Los usuarios pueden solicitar búsqueda de conductores de la aplicación de transporte externo con el que se sientan más cómodo.
- El sistema verifica el precio y la aplicación de transporte externo antes de confirmar la ruta.
- Se envía confirmación por notificación por voz al usuario.

3.- Módulo de Pagos en Línea

- Los usuarios pueden pagar su ruta después de confirmar.
- Se integran pasarelas de pago como nequi o billeteras digitales.
- Se envía un recibo de pago al correo electrónico del usuario final.

4.- Módulo de Integración con Sistemas de Transporte Externo

- Los conductores pueden visualizar la ruta por la cual se van a transportar.
- Se sincroniza una visualización del historial de rutas del usuario.
- La información no se almacena en la base de datos de las aplicaciones de transporte externo para garantizar la seguridad de los datos.

5.- Módulo de Notificaciones

- El sistema envía recordatorios automáticos a los conductores.
- Se notifica cualquier cambio en la disponibilidad por notificación por voz.

Paso 5: Implementación del Flujo de Trabajo

Escenario 1: Pedir transporte

Paso 1: El usuario inicia sesión y accede a la sección "solicitar ruta".

Paso 2: Selecciona una ruta frecuente o en específico.

Paso 3: al haber un conductor externo que haya aceptado la ruta, el sistema redirige al módulo de pagos en línea.

Paso 4: El usuario paga con tarjeta o billetera digital y recibe confirmación por voz.

Paso 5: Se envía una notificación por voz con los detalles de la ruta y el pago.

Escenario 2: Modificación de Ruta con Reembolso Automático en caso de cancelación

Paso 1: El Usuario accede a la aplicación y va hacia la ruta que solicito.

Paso 2: Selecciona la ruta que desea modificar o cancelar por comando de voz.

Paso 3: Si cancela la ruta, el sistema inicia automáticamente el proceso de reembolso.

Paso 4: Se notifica por notificación de voz el estado del reembolso dependiendo el método de pago que haya utilizado.

Paso 6: Beneficios de la Arquitectura Utilizada

✓ **Modularidad:** Se pueden modificar módulos sin afectar el resto del sistema.

✓ **Escalabilidad:** Se pueden agregar más funcionalidades en el futuro.

✓ **Seguridad:** Implementación de autenticación para proteger datos del usuario y pagos.

✓ **Eficiencia:** Optimización de los pagos, solicitud de rutas por comando de voz y visualización de historiales de rutas.