



Documentación Técnica - TaxiDay



1. Introducción

TaxiDay es una solución multiplataforma que he desarrollado para facilitar la gestión diaria de los taxistas. El proyecto nace desde una necesidad personal y muy cercana: mi padre es taxista, y desde el primer momento quise diseñar una herramienta que le ayudara en su día a día de forma rápida, práctica y sin complicaciones innecesarias. He creado esta app pensando en lo que realmente necesita un conductor durante su jornada: registrar sus turnos, carreras, pausas e ingresos de manera ágil, efectiva y sin distracciones. No quería formularios largos ni pasos innecesarios, sino una experiencia sencilla que le permitiera centrarse en lo importante: conducir.

Este enfoque desde la experiencia real ha sido clave para tomar decisiones de diseño, tanto a nivel técnico como funcional.

Factores que Contribuyen a la Efectividad de TaxiDay



2. Objetivos del Proyecto

- Digitalizar el control diario de actividad de los taxistas.
- Facilitar el registro de jornadas, turnos y carreras.
- Proveer acceso multiplataforma (móvil y web).
- Asegurar la privacidad de los datos mediante mecanismos de autenticación robustos.
- Ofrecer consultas históricas y herramientas de análisis de actividad.



3. Tecnologías Utilizadas

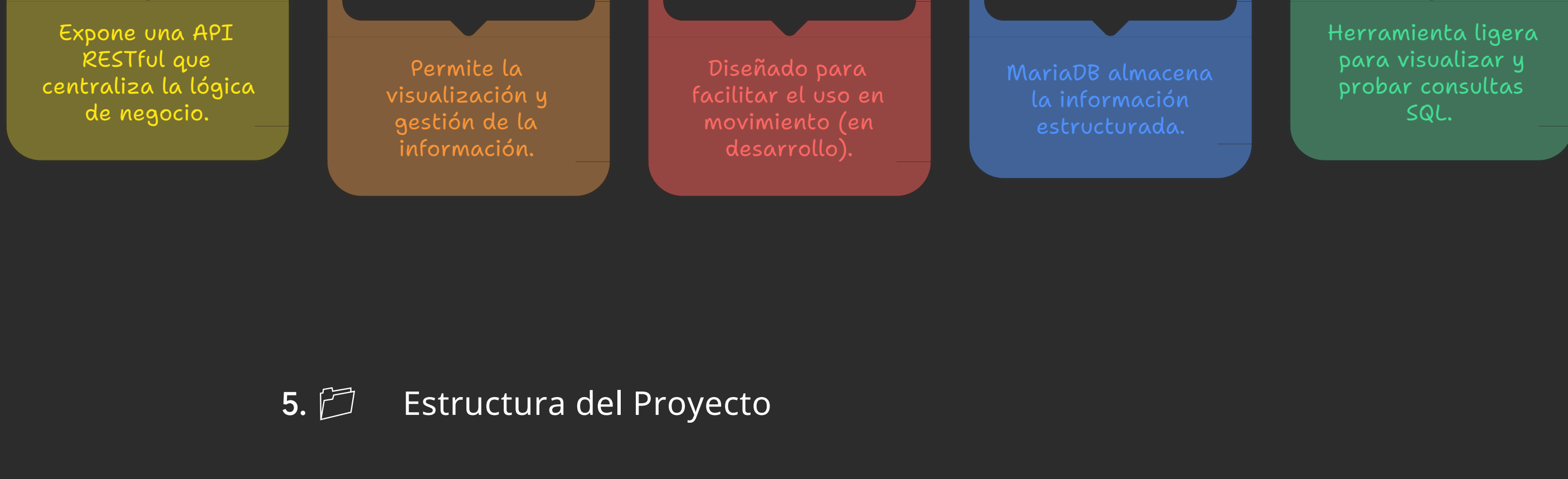
BackendJava 17 + Spring Boot 3.2.3
Frontend WebReact 18 [CRA]
Frontend MóvilReact Native + Expo (en progreso)
Base de Datos MariaDB 10.7
Contenedores Docker + Docker Compose
Visualización BBDDAdminer
Control de Versiones Git

4. Arquitectura General

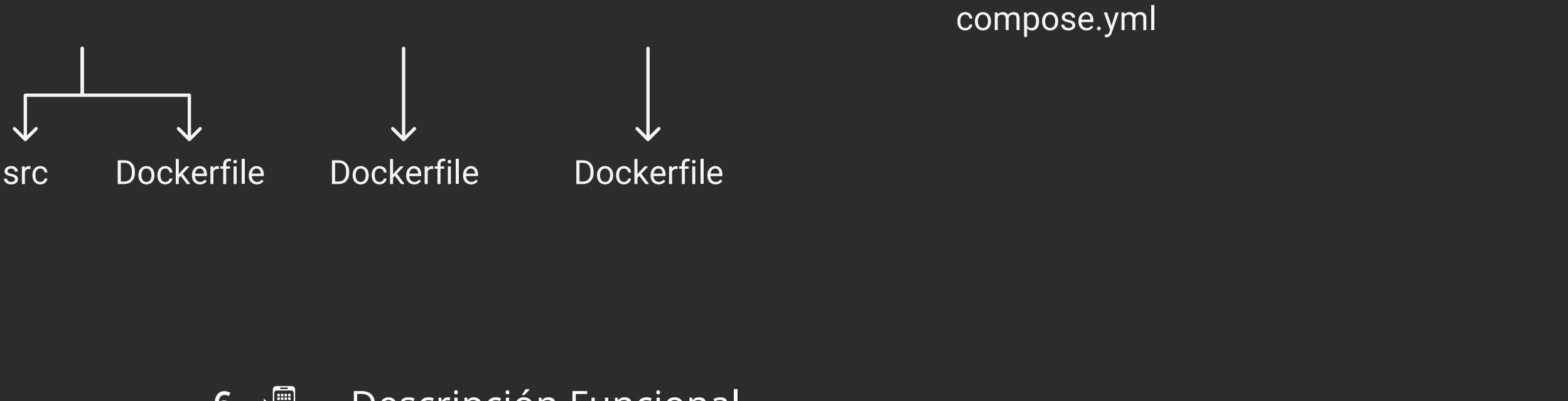
La arquitectura sigue un modelo de microservicios desacoplados mediante Docker Compose. Cada componente es independiente pero comunicable mediante red interna. Componentes:

- Backend: expone una API RESTful que centraliza toda la lógica de negocio.
- Frontend Web: permite la visualización y gestión de la información.
- Frontend Móvil: diseñado para facilitar el uso en movimiento (en desarrollo).
- Base de Datos: MariaDB almacena la información estructurada.
- Adminer: herramienta ligera para visualización y prueba de queries SQL.

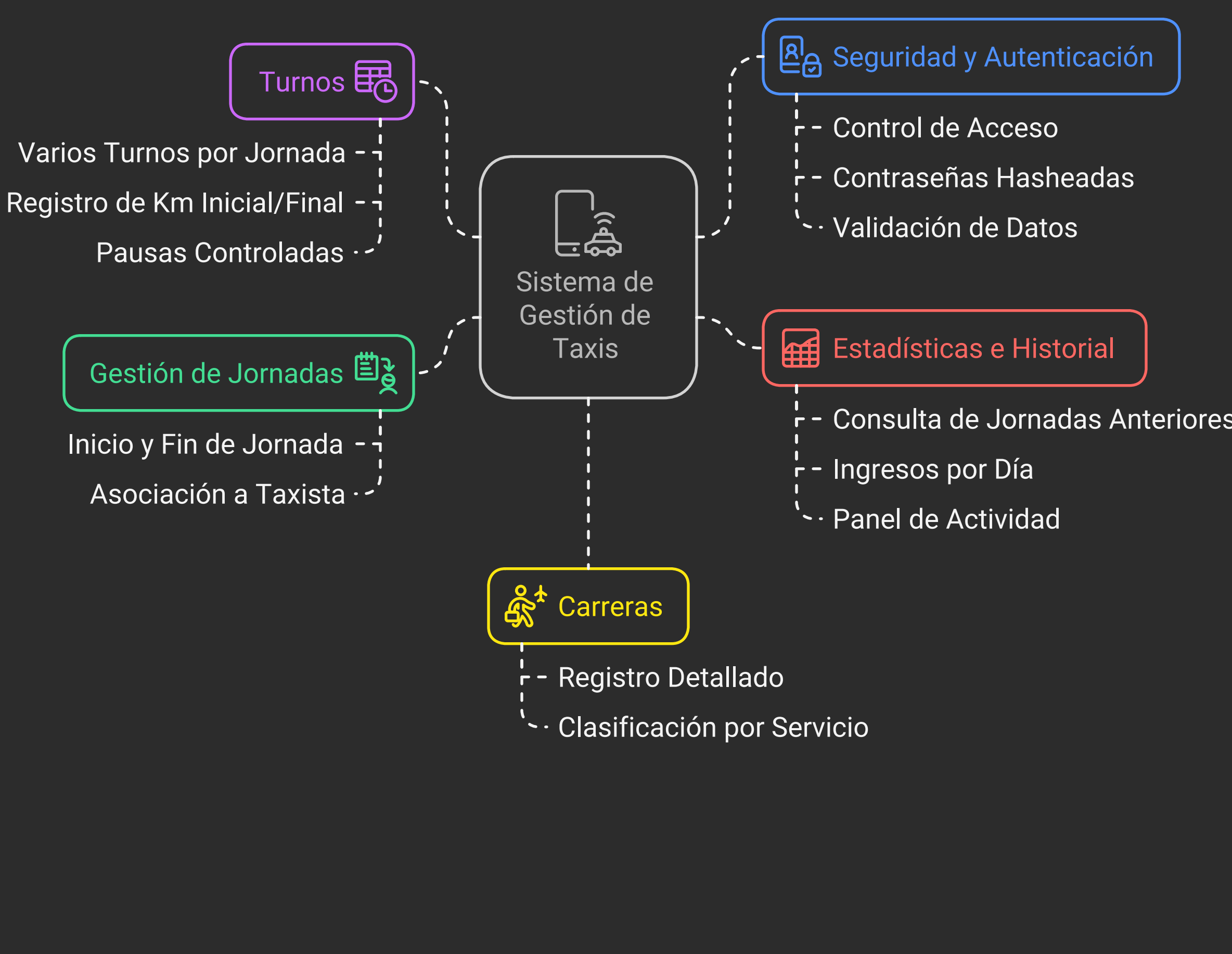
Componentes del Sistema



5. Estructura del Proyecto



6. Descripción Funcional



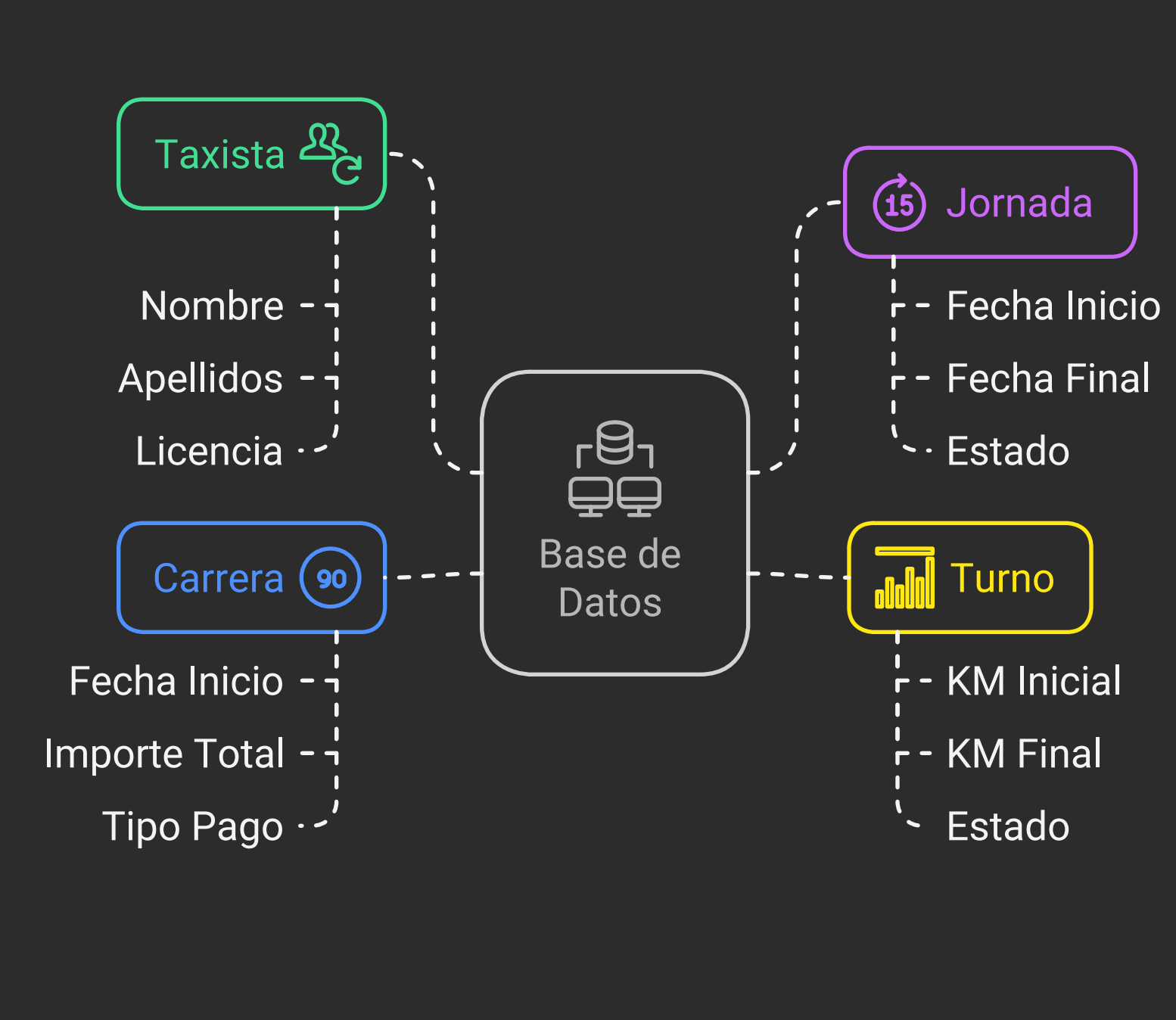
7. Base de Datos

7.1. Modelo Relacional

Tablas principales: • taxista: usuarios del sistema. • jornada: días de trabajo. • turno: subdivisiones de la jornada. • carrera: registro individual de cada trayecto.

7.2. Esquema SQL

Modelo Relacional de Base de Datos para Sistema de Taxis



8. Despliegue con Docker Compose

Comandos esenciales

- Levantar servicios:

```
docker compose up -d --build
```

- Parar servicios:

```
docker compose down
```

- Ver logs:

```
docker compose logs -f
```

- Acceder a MariaDB:

```
docker exec -it taxiway_db bash  
mariadb -uroot -proot
```

9. Acceso a la Aplicación

- Frontend Web: <http://localhost>
- Backend API: <http://localhost:8085>
- Adminer [BBDD]: <http://localhost:8082>
- App móvil: se ejecutará mediante Expo CLI (próximamente)



10. Buenas Prácticas, Seguridad y Experiencia de Desarrollo

10.1. Control de Versiones con GitHub

El proyecto TaxiDay se gestiona completamente mediante Git y GitHub, lo que ha permitido mantener un flujo de trabajo organizado y claro. Se han definido ramas específicas para facilitar el desarrollo y la evolución modular del sistema:

- proyecto-entrega: rama principal para la versión estable y funcional que se presenta como entrega principal del proyecto.
- proyecto-movil: rama dedicada a la implementación y pruebas del frontend móvil con React Native y Expo, manteniéndolo separado del core funcional hasta estar completado.

Este enfoque ha facilitado:

- Aislar cambios experimentales sin afectar la versión estable.
- Revisar y testear funcionalidades por separado.
- Favorecer la colaboración futura si se incorpora a un equipo.

10.2. Uso Extensivo de Docker y Beneficios

Uno de los mayores aprendizajes e implementaciones ha sido el uso de Docker como entorno de desarrollo y despliegue.

Ventajas clave que me ha aportado Docker en TaxiDay:

- Entornos reproducibles con solo ejecutar `docker compose up`.
- Aislamiento de servicios (backend, frontend, base de datos, Adminer) en contenedores separados.
- Control completo de puertos, imágenes y dependencias sin necesidad de instalaciones locales.
- Facilidad de escalado y despliegue en producción futura.

El archivo `docker-compose.yml` orquesta todos los servicios necesarios, simplificando radicalmente el arranque del proyecto.

Este aprendizaje ha sido uno de los pilares más importantes para profesionalizar el desarrollo del proyecto, permitiéndome ver cómo trabajan proyectos reales y escalables.

10.3. Descubrimiento de Adminer

Durante el desarrollo, descubrí Adminer, una herramienta web ligera y potente para interactuar visualmente con bases de datos SQL. A diferencia de interfaces más complejas, Adminer ofrece:

- Interfaz clara y sencilla.
- Edición directa de datos y estructura de tablas.
- Consulta y ejecución de queries SQL en tiempo real.

Incluso como contenedor dentro de Docker (adminer:latest) fue una mejora inesperada pero muy útil, que optimizó mi productividad y permitiendo testear fácilmente la integridad de las relaciones de la base de datos.

10.4. Aprendizajes con el Frontend

Tanto con React Web como al iniciar el trabajo en React Native, este proyecto me ha permitido afianzar y expandir mis conocimientos en frontend:

- Componentes reutilizables y separación clara por vistas.
- Uso de Docker para encapsular también el entorno de desarrollo frontend, asegurando consistencia en versiones de Node y dependencias.

He aprendido a dar prioridad a la usabilidad, simplificando la interacción para los taxistas, priorizando accesibilidad y claridad de los formularios.

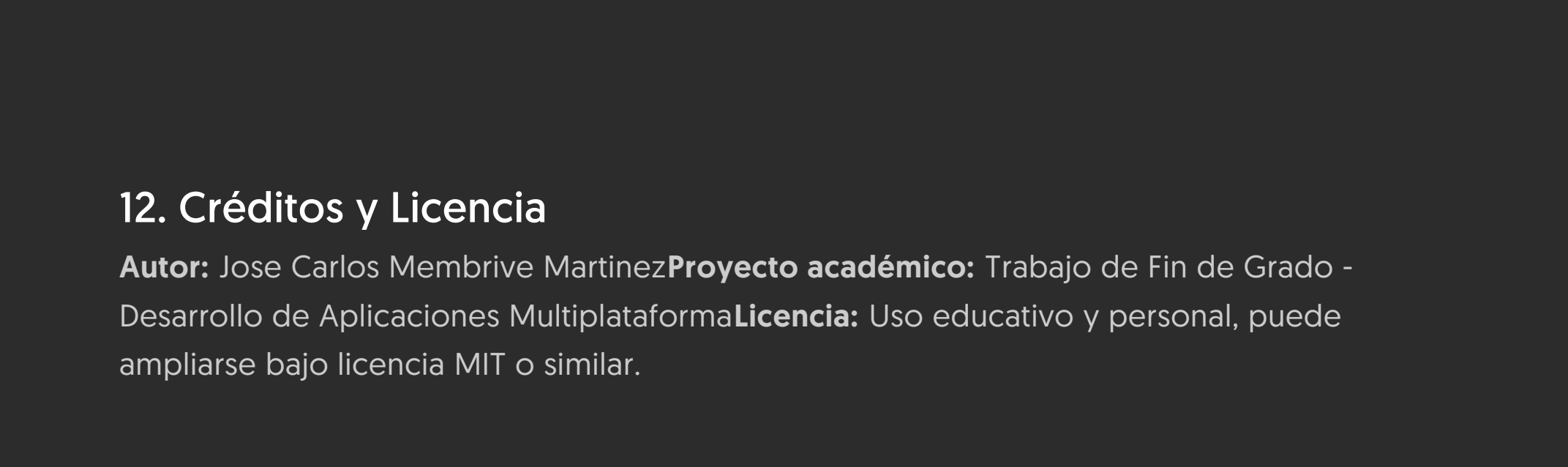
Flujo de Trabajo de Desarrollo del Proyecto TaxiDay



11. Mejoras Futuras

- Finalizar e integrar el frontend móvil.
- Incorporar GPS y localización en tiempo real.
- Crear un panel administrativo para estadísticas y exportación.
- Soporte para facturación automática.
- Notificaciones push y alertas de mantenimiento.

Características de la aplicación móvil



12. Créditos y Licencia

Autor: Jose Carlos Membrive Martinez Proyecto académico: Trabajo de Fin de Grado - Desarrollo de Aplicaciones Multiplataforma Licencia: Uso educativo y personal, puede ampliarse bajo licencia MIT o similar.