PAQUEXPRESS S.A. DE C.V. - Módulo de Entregas y Trazabilidad

Este proyecto implementa una solución Full Stack para la logística de última milla, permitiendo a los agentes de campo registrar entregas de manera segura, capturando evidencia fotográfica y ubicación GPS.

Tecnologías: Flutter (Móvil), FastAPI (API RESTful), MySQL (Base de Datos).

🛠️ 1. Estructura del Repositorio

Directorio / Archivo	Contenido	Responsabilidad
api_fastapi/	Contiene main.py y dependencias de Python.	Lógica de negocio, seguridad (hashing MD5) y conexión a MySQL.
app_flutter/	Contiene main.dart y dependencias de Dart.	Interfaz de usuario, acceso a sensores (GPS, Cámara) y Mapa Interactivo.
database/	Contiene db_u3.sql.	Script de creación de la base de datos MySQL y datos iniciales de prueba.
docs/	Contiene los diagramas UML/DER y el Reporte Digital.	Documentación técnica del proyecto.

🚀 2. Guía de Instalación y Configuración

Sigue estos pasos para levantar el entorno completo (BD, API y App).

2.1. Configuración de la Base de Datos (MySQL)

1. Asegúrate de tener un servidor MySQL instalado y corriendo (ej. XAMPP, Docker, o MySQL Workbench).
2. Accede a tu herramienta de administración (ej. HeidiSQL, Workbench).

3. Ejecuta el script completo ubicado en database/db_u3.sql.
   Resultado Esperado: Se creará la base de datos bd_paquexpress con las tablas agentes, paquetes, y entregas, además de insertar dos agentes de prueba.

2.2. Configuración e Inicio de la API (FastAPI)

1. Requisitos: Python 3.8+ y Pip.
2. Navega al directorio de la API: cd api_fastapi/

Instala las dependencias:
Bash
pip install -r requirements.txt
(Si no tienes requirements.txt, usa: pip install fastapi uvicorn sqlalchemy pymysql python-multipart python-cors)

3.

Verifica la URL de conexión a la base de datos dentro de main.py:
Python
DATABASE_URL = "mysql+pymysql://root:root@localhost:3306/bd_paquexpress"

4. *(Ajusta root:root si tus credenciales de MySQL son diferentes).*

Inicia el servidor (en modo desarrollo):
Bash
uvicorn main:app --reload

5. Verificación: Abre tu navegador y accede a: http://127.0.0.1:8000/docs. Deberías ver la documentación Swagger de la API.

2.3. Configuración y Ejecución de la Aplicación (Flutter)

1. Requisitos: Flutter SDK (instalado y configurado), y un emulador/dispositivo móvil o navegador Chrome.
2. Navega al directorio de la aplicación: cd app_flutter/

Instala las dependencias de Dart:
Bash
flutter pub get

3.
4. Configuración del Mapa (Importante para Web/Desktop):
   - Asegúrate de que la API Key de Google Maps esté en web/index.html (ya está incluida en este repositorio para fines de demostración).

Ejecuta la aplicación:
Bash
flutter run

5.

🔑 3. Credenciales de Prueba y Uso

Utiliza las siguientes credenciales para el inicio de sesión:

Agente	Usuario	Contraseña
fabri	fabricio	123456

Flujo de Prueba de Entrega (Agente: fabri)

1. Login: Inicia sesión con fabri / 123456.
2. Lista de Paquetes: La aplicación cargará los paquetes asignados a fabri (agregalos en docs de la api)(ej. PEX-001, PEX-002, o como el mio pq-prueba2).
3. Detalle de Entrega: Selecciona el paquete PEX-001.
    - Verás la dirección de destino y el mapa interactivo (que intenta geocodificar la dirección).
4. Captura de Evidencia:
    - Presiona "1. CAPTURAR FOTO" (Utiliza el sensor de la cámara del emulador/dispositivo).
    - Presiona "2. OBTENER UBICACIÓN GPS" (Utiliza el sensor de Geolocator).
5. Registro Final: Presiona "3. PAQUETE ENTREGADO" (Esto dispara la petición *Multipart* a la API, que guarda la foto, registra la evidencia en la tabla entregas, y cambia el estado del paquete a ENTREGADO en la BD).
6. Verificación: El paquete PEX-001 desaparecerá de la lista al regresar a la pantalla principal.