**TFC Transportes – Especificación del Proyecto (Flutter + FastAPI)**

Documento listo para pegar en Cursor. Incluye alcance, historias de usuario, arquitectura, contratos de API, modelos de datos, integraciones y plan de entrega.

**1) Objetivo y Contexto**

Construir una plataforma integral para **captación, operación logística, HSEQ, flota, facturación y trazabilidad** de TFC Transportes, con **front-end en Flutter** (móvil Android/iOS, desktop y web) y **back-end en FastAPI (Python)**, diseñada con arquitectura modular, multi-sede y preparada para crecer.

**Metas clave**

* Reducir procesos manuales y errores operativos.
* Aumentar conversión comercial y velocidad de respuesta.
* Dar **trazabilidad** en tiempo real a clientes y cumplir **normativa** (ISO 9001/14001/45001, RUC, PESV).
* Medir KPIs operativos y financieros (costos por placa, tiempos, cierres, cartera).

**2) Alcance por Módulos**

**2.1 CRM & Ventas**

* Registro y calificación de leads (fuente: llamadas, web, Ads, referidos).
* Pipeline de oportunidades, cotizaciones y órdenes de servicio.
* Gestión de documentos comerciales por cliente (formatos, anexos).
* SLA de respuesta y notificaciones.

**2.2 Operaciones & Asignación**

* Creación/planificación de servicios (rutas, origen/destino, materiales: líquidos/sólidos, equipos).
* Asignación de vehículos/aliados y operadores (propios/terceros).
* Seguimiento de estado (programado → en curso → completado → soportes).
* Carga de evidencias: remisiones, fotos, manifiestos, DD/DA, certificados de disposición final.

**2.3 Flota & Mantenimiento**

* Inventario de flota (propia y tercerizada), documentos y vigencias.
* Órdenes de trabajo, consumos (combustible, llantas, filtros), mantenimientos preventivos/correctivos.
* Centros de costo por placa y rentabilidad por proyecto.

**2.4 HSEQ & PESV**

* Gestión de inducciones, EPP, exámenes médicos y restricciones.
* Control de horarios/fatiga, inspecciones preoperativas, incidentes/siniestros.
* Reportería normativa (mensual y anual) y auditorías.

**2.5 Talento Humano**

* Hoja de vida completa, historial laboral, vetos/antecedentes.
* Alertas de vencimientos (licencias, cursos, exámenes), control documental.
* Accesos de autogestión para subir documentos.

**2.6 Finanzas (operativo)**

* Integración con contabilidad (cartera y proveedores).
* Anticipos, hitos de facturación por servicio y validación de soportes.
* Alertas de cobranza y conciliación de pagos.

**2.7 Portal Cliente (Extranet)**

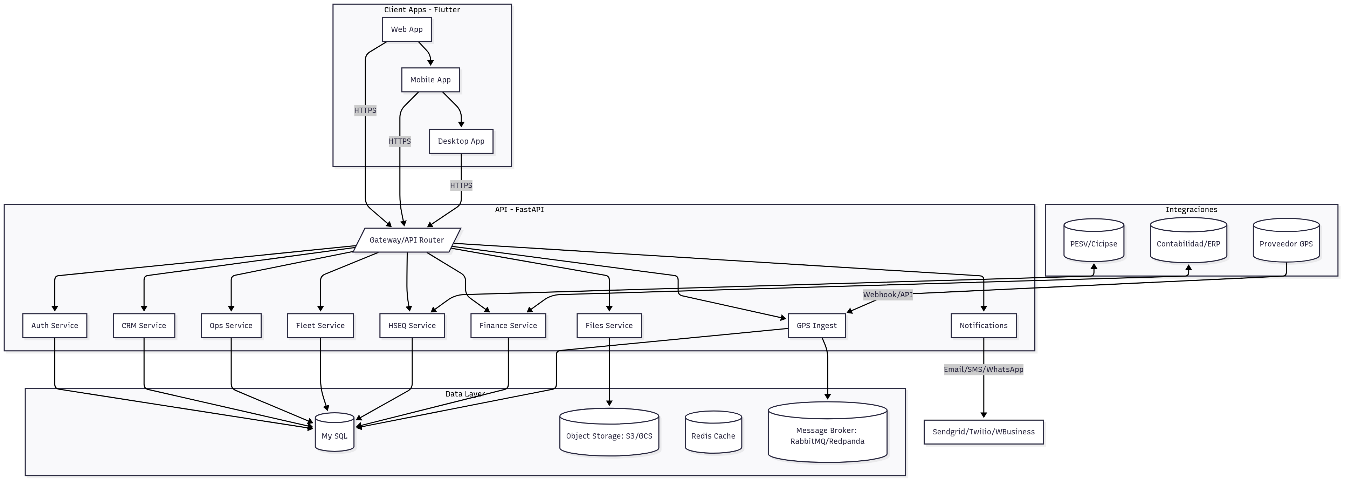
* Estado de servicios en tiempo real (tracking).
* Descarga de certificados y facturas; interacción y PQR.
* KPIs por contrato (cumplimiento, volúmenes, tiempos, costos).

**3) Roles y Permisos**

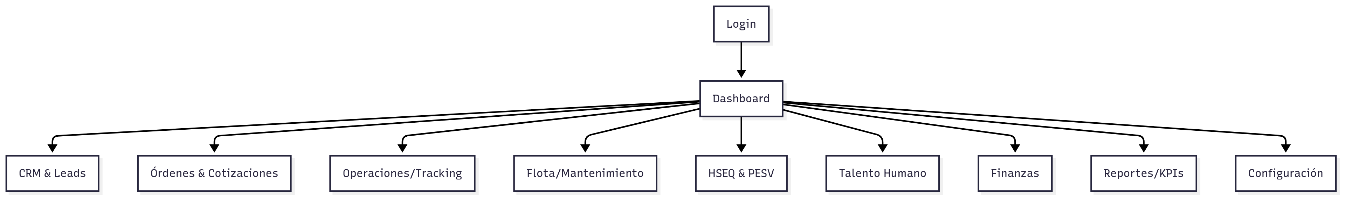
* **Admin**: configuración, seguridad, catálogos y auditoría.
* **Comercial**: leads, cotizaciones, órdenes, adjuntos.
* **Operaciones**: planeación, asignación, seguimiento y soportes.
* **Flota**: mantenimiento, inventario, talleres y repuestos.
* **HSEQ**: inducciones, EPP, PESV, incidentes, reportes.
* **Talento Humano**: perfiles, documentos, vencimientos.
* **Finanzas**: anticipos, facturación, pagos, cartera.
* **Cliente**: consulta de servicios e historial, descargas.

**4) Arquitectura**

**4.1 Panorama General (Mermaid)**

****

**4.2 Front-end (Flutter) – Jerarquía de Pantallas**

****

**Stack sugerido**

* Flutter 3.x, Material 3, **Riverpod** (estado) + **GoRouter** (routing).
* **Dio** (HTTP), **Freezed/JsonSerializable** (modelado), **Drift** (offline cache), **intl**.
* Soporte offline-first para formularios de campo y sincronización diferida.

**4.3 Back-end (FastAPI)**

* FastAPI + Uvicorn.
* Auth **JWT** (Access + Refresh), OAuth2 Password Flow.
* Pydantic v2 para esquemas; SQLAlchemy 2.0 + Alembic (migraciones);
* Task runner (Celery/RQ) para jobs asíncronos (ETL GPS, generación de reportes).
* Observabilidad: Prometheus + Grafana, Sentry.
* Infra: Docker Compose (dev), K8s (prod), CI/CD (GitHub Actions).

**5) Historias de Usuario (extracto operativo)**

**CRM & Ventas**

* **Como Comercial**, quiero registrar un lead con su industria, ubicación, tipos de residuos y cobertura requerida, para priorizar visitas técnicas y acelerar una cotización.
* **Como Comercial**, quiero generar una **cotización** con plantillas por tipo de servicio y enviar al cliente con vencimiento y trazabilidad de lectura.

**Operaciones**

* **Como Operaciones**, quiero convertir una cotización aceptada en **orden de servicio**, asignar **vehículos/aliados** y operadores, y notificar a todos los implicados.
* **Como Operaciones**, necesito capturar **evidencias** (fotos, remisiones, certificados) desde el móvil, incluso offline.

**Flota**

* **Como Flota**, quiero ver **KPIs por placa** (consumo, mantenimiento, costos) y programar **preventivos**.

**HSEQ/PESV**

* **Como HSEQ**, quiero gestionar **inducciones** y EPP, controlar **horas de conducción** y reportar **siniestros** mensual/anual.

**Talento Humano**

* **Como TH**, necesito el **historial laboral** de cada conductor/operador, con alertas de vencimientos.

**Finanzas**

* **Como Finanzas**, quiero gestionar **anticipos** y **hitos de facturación** por servicio, y disparar alertas de **cobranza** automáticas.

**Cliente (Extranet)**

* **Como Cliente**, quiero **rastrear** el estado de mis servicios en tiempo real y descargar **certificados** y **facturas**.

**6) Modelo de Datos (resumen)**

**Catálogos**

* industrias, residuos (líquidos/sólidos/equipos), tipos\_vehículo, EPP, exámenes\_médicos, cursos, centros\_costo, documentos\_requeridos.

**Entidad núcleo**

* **cliente(id, nit, razón\_social, sector, ubicaciones[])**
* **lead(id, cliente\_id?, fuente, industria, residuos[], cobertura, estado, score, notas)**
* **oportunidad(id, lead\_id, valor\_estimado, probabilidad, etapa)**
* **cotización(id, oportunidad\_id, ítems[], vigencia, total, estado, adjuntos[])**
* **orden\_servicio(id, cotización\_id, cliente\_id, ruta(origen,destino), tipo\_material, ventanas\_tiempo, estado, anticipos[], soportes[])**
* **vehículo(id, placa, tipo, propio, aliado\_id?, docs[], costos\_cc, odómetro, gps\_id?)**
* **operador(id, nombre, rol, licencias[], restricciones\_médicas[], vetos?)**
* **asignación(id, orden\_id, vehículo\_id, operador\_id, aliado\_id?, turno, horas\_conducción)**
* **mantenimiento(id, vehículo\_id, tipo, fecha, odómetro, costo, evidencias[])**
* **hseq\_evento(id, tipo, fecha, orden\_id?, vehículo\_id?, operador\_id?, evidencias[])**
* **usuario(id, rol, email, hash\_pw, cliente\_id?)**
* **documento(id, tipo, entidad, url, hash, firmado, vencimiento)**
* **factura(id, orden\_id, total, estado, fecha\_emisión, fecha\_vencimiento, pagos[])**
* **alerta(id, tipo, target, payload, leído)**

**7) Contratos de API (ejemplos)**

**Auth**

* POST /auth/login → {access\_token, refresh\_token}
* POST /auth/refresh → {access\_token}

**CRM**

* POST /crm/leads → crea lead
* GET /crm/leads?estado=...&q=...
* POST /crm/oportunidades → crea
* POST /crm/cotizaciones → crea; payload incluye ítems y vigencia
* POST /crm/cotizaciones/{id}/enviar → dispara notificación al cliente

**Operaciones**

* POST /ops/ordenes (desde cotización)
* POST /ops/ordenes/{id}/asignaciones
* POST /ops/ordenes/{id}/soportes (multipart)
* PATCH /ops/ordenes/{id} estado

**Flota**

* GET /fleet/vehiculos?propio=true
* POST /fleet/mantenimientos
* GET /fleet/kpis?placa=XYZ123&periodo=2025-08

**HSEQ**

* POST /hseq/inducciones
* POST /hseq/inspecciones\_pre
* POST /hseq/eventos
* GET /hseq/reportes?periodo=2025-08

**Finanzas**

* POST /fin/anticipos (vinculado a orden)
* POST /fin/facturas
* POST /fin/cobranza/{factura\_id}/recordatorio
* GET /fin/cartera?cliente\_id=...

**Archivos**

* POST /files/upload (S3 presigned)
* GET /files/{id} (signed URL)

**GPS**

* POST /gps/webhook (ingesta de posiciones)
* GET /gps/vehiculos/{id}/posicion

**8) Integraciones**

* **GPS**: conector por **Webhook** y **pull** periódico; mapeo de credenciales por proveedor.
* **ERP/Contabilidad**: exportación/importación de movimientos (cartera, proveedores, facturas) vía CSV/SFTP o API.
* **CICIPSE/PESV**: endpoints para subir indicadores y descargar plantillas.
* **Correo certificado**: proveedor con API para evidencias de lectura.

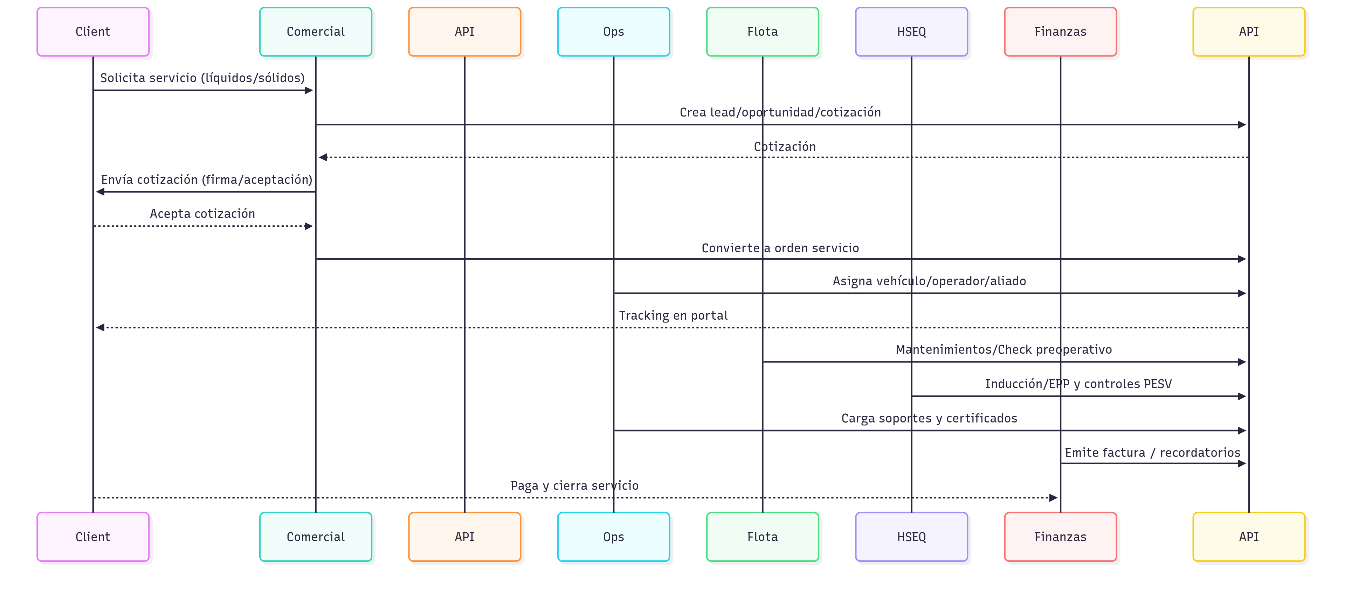
**9) Seguridad y Cumplimiento**

* Roles/Permisos por recurso; **Audit Log** por acción.
* Archivos en object storage con **firma** y **hash**.
* Cifrado en tránsito (TLS) y en reposo.
* Políticas de retención documental y backups.

**10) KPIs Clave (mínimos)**

* **Comercial**: tasa de conversión, tiempo de respuesta, valor vs. meta.
* **Operaciones**: OTIF, % servicios con soportes completos, tiempo ciclo.
* **Flota**: costo/km, cumplimiento preventivos, disponibilidad.
* **HSEQ**: incidentes por millón de km, cumplimiento inducciones, inspecciones al día.
* **Finanzas**: DSO (días de cobro), moras por tramo, % anticipos conciliados.

**11) Flujo E2E (secuencia)**



**12) Entregables & Roadmap**

**Fase 0 – Fundaciones (1-2 sprints)**

* Repos, CI/CD, Docker Compose, scaffolding FastAPI + Flutter.
* Autenticación y RBAC.

**Fase 1 – CRM & Órdenes (2-3 sprints)**

* Leads, cotizaciones, órdenes, portal cliente básico.

**Fase 2 – Operaciones & HSEQ (2-3 sprints)**

* Asignaciones, soportes móviles offline, controles PESV.

**Fase 3 – Flota & Finanzas (2-3 sprints)**

* Mantenimientos, KPIs por placa, anticipos/facturación/cobranza.

**Fase 4 – Integraciones & Reportes (2-3 sprints)**

* GPS, ERP, Cicipse, tableros KPI, auditoría.

**13) Repos y Estructura**

**Monorepo** (sugerido):

/tfc

/apps

/frontend\_flutter

/backend\_fastapi

/packages

/shared\_models

/infra

docker-compose.yml

k8s/

/docs

**Backend**

backend\_fastapi/

app/

main.py

api/

v1/

auth.py, crm.py, ops.py, fleet.py, hseq.py, fin.py, files.py, gps.py

core/

config.py, security.py, deps.py

models/

schemas/

services/

tasks/

db/

base.py, session.py

tests/

**Frontend**

frontend\_flutter/

lib/

app.dart

router.dart

features/

auth/

crm/

orders/

ops/

fleet/

hseq/

hr/

finance/

reports/

settings/

shared/

widgets/

services/

theme/

utils/

**14) Definición de Listas de Tareas para Cursor**

**Tarea 1: Scaffold Backend**

* Crear proyecto FastAPI con módulos, SQLAlchemy, Alembic, JWT y tests.
* Endpoints de auth + healthcheck.

**Tarea 2: Modelo de Datos Inicial**

* Tablas: usuario, cliente, lead, oportunidad, cotización, orden\_servicio, vehículo, operador, asignación, documento, mantenimiento, hseq\_evento, factura, alerta.

**Tarea 3: Scaffold Frontend**

* Proyecto Flutter con Riverpod, GoRouter, layout responsive y tema M3.
* Pantallas: Login, Dashboard, Leads (CRUD), Cotizaciones (lista/detalle), Órdenes (lista/detalle).

**Tarea 4: Portal Cliente**

* Autenticación separada (scopes de cliente) y vista de tracking + descargas.

**Tarea 5: Sincronización Offline**

* Implementar cache local (Drift) y colas de sincronización para evidencias.

**Tarea 6: Observabilidad**

* Prometheus/Grafana y Sentry en backend; logger estructurado.

**Tarea 7: Integración GPS (Mock)**

* Webhook de ingestión + job de consolidación. Endpoint de última posición.

**Tarea 8: Reportes KPI**

* Consultas agregadas por placa/orden/cliente; endpoints y widgets en Flutter.

**15) Normativa & Auditoría (checklist)**

* ISO 9001/14001/45001: control documental, trazabilidad y acciones correctivas.
* RUC: puntajes y requisitos mínimos, mantener evidencias actualizadas.
* PESV: límites de conducción, reportes mensuales, inspecciones.
* Correo certificado: integración para notificaciones con validez jurídica.

**16) Definiciones No Funcionales**

* Disponibilidad 99.5% mensual, RTO ≤ 4h, RPO ≤ 1h.
* Performance: p95 API < 400ms (consultas estándar); subidas de archivo con colas asincrónicas.
* Seguridad: OWASP ASVS nivel 2, hardening en contenedores, rotación de secretos.

**17) Pruebas y Aceptación**

* **Backend**: unitarias (pytest), contract tests (schemathesis), integración (docker).
* **Frontend**: widget tests, golden tests, e2e (Flutter Driver/Patrol).
* **UAT**: scripts por módulo con casos de borde (sin conectividad, documentos faltantes, vetos, etc.).

**Fin del documento.**