

## Detección de Viajes Fantasmas (Uber/Bolt + Webfleet)

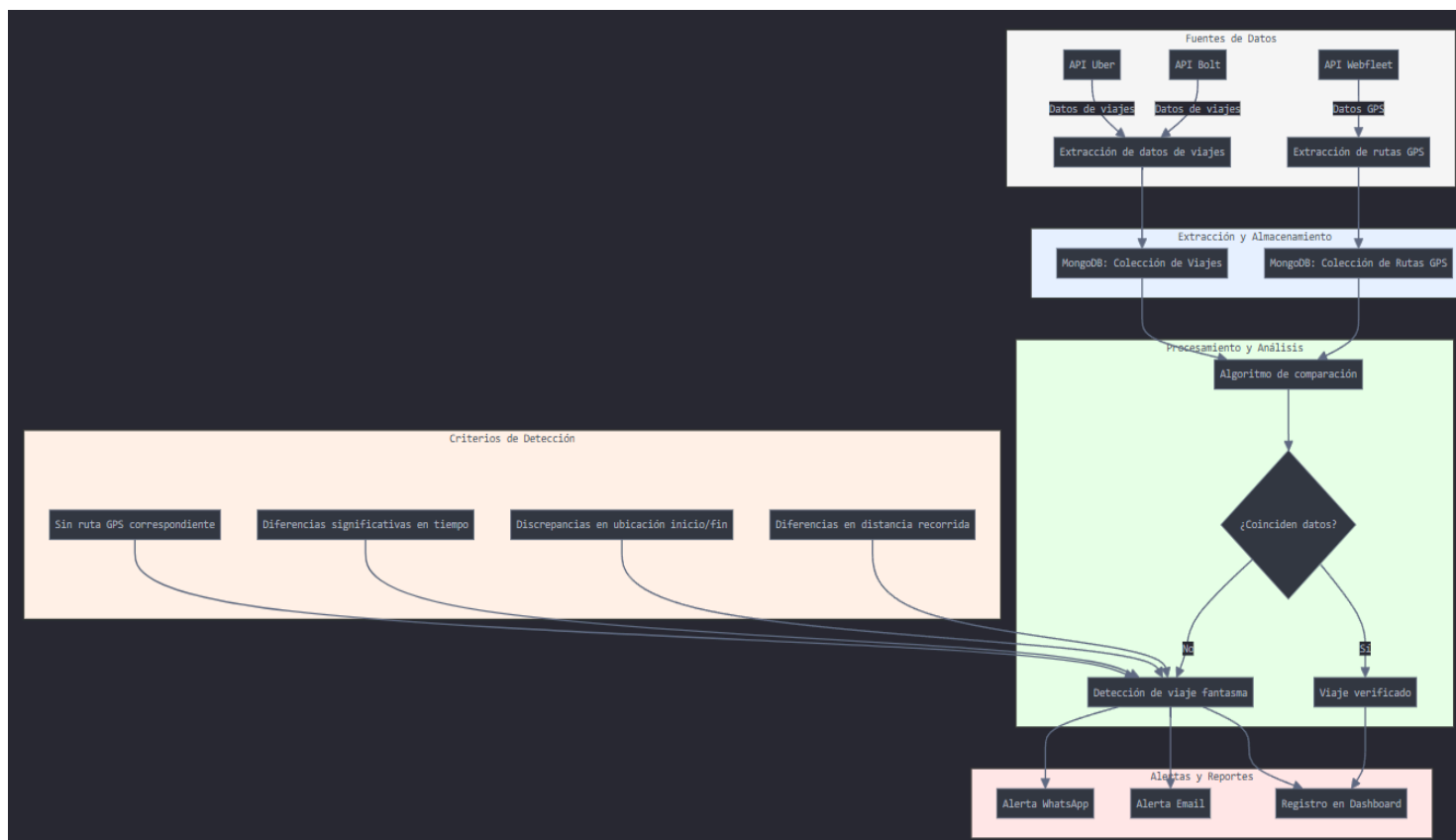
Se desarrollará un sistema para detectar "viajes fantasmas" comparando datos de viajes de Uber/Bolt con rutas GPS de Webfleet.

### Proceso:

1. **Obtención de Datos:** Se extraerán datos de viajes (ID, tiempo, ubicaciones, conductor, vehículo, tarifa, pago) de las APIs de Uber y Bolt, y datos de geolocalización (timestamp, latitud, longitud) de la API de Webfleet.
2. **Almacenamiento:** Los datos se almacenarán estructuradamente en MongoDB.
3. **Comparación:** Un algoritmo comparará los viajes de Uber/Bolt con las rutas GPS basándose en la coincidencia temporal, la proximidad de las ubicaciones de inicio/fin y la duración del viaje.
4. **Detección de Viajes Fantasmas:** Se identificarán viajes fantasmas si no hay una ruta GPS correspondiente o si hay discrepancias significativas en duración o distancia.
5. **Alertas y Reportes:** Se generarán alertas automáticas (WhatsApp/email) y reportes visuales al detectar posibles viajes fantasmas.

**Tecnologías:** Se utilizarán Python (requests, geopy), y MongoDB para la interacción con APIs, cálculos de distancia y almacenamiento de datos.

### DIAGRAMA DE FLUJO



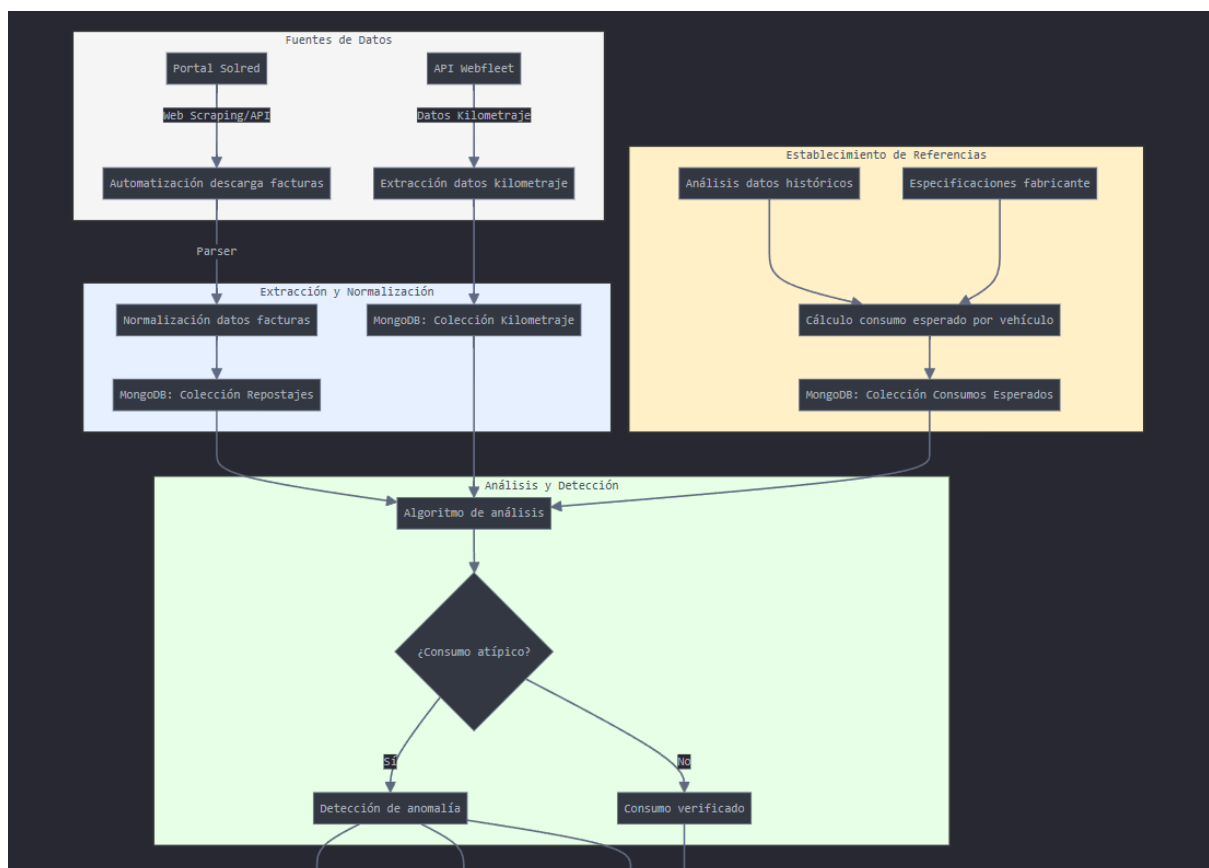
## Detección de Inconsistencias en Consumo de Combustible (Solred + Webfleet)

Se implementará un sistema para detectar inconsistencias en el consumo de combustible mediante el análisis integrado de datos de Solred y Webfleet.

### Proceso:

1. **Obtención de Datos Solred:** Se extraerán automáticamente las facturas de combustible de Solred (web scraping o API), obteniendo fecha/hora, vehículo, estación, litros, importe y tipo de combustible.
2. **Procesamiento de Datos Solred:** Los datos de Solred se normalizarán, validarán y almacenarán en MongoDB.
3. **Obtención de Datos Webfleet:** Se obtendrán datos de kilometraje precisos (km totales, trayectos, conducción) de la API de Webfleet.
4. **Cálculo de Consumo Esperado:** Se crearán perfiles de consumo esperado para cada vehículo basados en datos históricos, especificaciones técnicas, tipo de ruta y factores estacionales.
5. **Detección de Anomalías:** Se calculará el consumo real y se comparará con el esperado, detectando anomalías mediante umbrales, algoritmos estadísticos y considerando factores contextuales.
6. **Alertas y Reportes:** Se generarán alertas inmediatas (WhatsApp/email), se categorizarán por gravedad y se mostrarán en un dashboard interactivo, además de informes periódicos.

**Tecnologías:** Se utilizarán Python (BeautifulSoup/Scrapy, Pandas), MongoDB y algoritmos de machine learning para la extracción, análisis, almacenamiento y detección de patrones anómalos.



## Automatización de la Contabilidad Semanal (Uber/Bolt + Solred)

Se automatizará la contabilidad semanal integrando datos de Uber/Bolt (ingresos, viajes), Solred (combustible) y otra información relevante.

### Proceso:

1. **Integración y Normalización de Datos:** Se integrarán y normalizarán datos de diversas fuentes en un formato uniforme mediante algoritmos específicos.
2. **Uso de la API de OpenAI:** Se utilizará la API de OpenAI para:
  - Realizar cálculos contables mediante prompts diseñados.
  - Generar reportes contables semanales para conductores.
  - Detectar anomalías contables.
3. **Generación de Reportes y Dashboards:** Se crearán reportes semanales individuales con ingresos, gastos y ganancias para cada conductor, y dashboards con métricas clave para la flota.
4. **Alertas:** Se configurarán alertas automáticas vía WhatsApp para conductores y email para el propietario con resúmenes contables o alertas importantes.

**Tecnologías:** Se utilizarán Pandas para manipulación de datos, la librería de Python para la API de OpenAI, Flask para la API del sistema y MongoDB para almacenar datos contables históricos.

