

# Anexo de documentación técnica

## 1. INTRODUCCIÓN

FleetIQ nace con la misión de transformar la logística entre almacenes en diferentes ciudades, respondiendo a una necesidad crítica en el mercado: la optimización de los procesos de envío y gestión de pedidos. A través de inteligencia avanzada y análisis en tiempo real, FleetIQ organiza y optimiza el flujo de envíos, previniendo errores comunes y reduciendo significativamente los costos operativos. Esta plataforma ha sido desarrollada utilizando JavaFX para la interfaz de usuario y Hibernate como herramienta de persistencia.

---

## 2. ANÁLISIS

### 2.1. Definición del sistema

#### 2.1.1. Determinación del alcance del sistema: descripción funcional

FleetIQ permite:

- Registrar y autenticar usuarios (trabajadores y administradores).
- Administrar paquetes, rutas, envíos y vehículos.
- Asignar vehículos a rutas.
- Consultar y modificar perfiles de usuario.
- Cambiar contraseñas.
- Visualizar toda la información de envíos y optimizar rutas entre almacenes.
- Hacer logout desde cualquier pantalla mediante un menú común.

#### 2.1.2. Identificación de usuarios

- **Administrador:** Responsable de supervisar y gestionar el sistema, ver el historial de sesiones, y administrar todos los registros.
- **Trabajador:** Responsable de preparar envíos y gestionar información operativa diaria.
- **Usuario no autenticado:** Tiene acceso limitado únicamente a las pantallas de Splash, SignIn y SignUp.

---

## **2.2. Especificación de casos de uso (jerarquizada)**

### **2.2.1. Casos de uso gestión de usuarios**

- UC1: Iniciar sesión
- UC2: Registrarse
- UC3: Cerrar sesión
- UC4: Cambiar contraseña
- UC5: Consultar y editar perfil

### **2.2.2. Casos de uso gestión de paquetes**

- UC6: Crear paquete
- UC7: Consultar paquete
- UC8: Editar paquete
- UC9: Eliminar paquete

### **2.2.3. Casos de uso gestión de envíos**

- UC10: Crear envío
- UC11: Consultar envío
- UC12: Actualizar estado de envío

### **2.2.4. Casos de uso gestión de rutas**

- UC13: Crear ruta
- UC14: Asignar vehículos a ruta
- UC15: Consultar rutas

### **2.2.5. Casos de uso gestión de vehículos**

- UC16: Registrar vehículo

- UC17: Consultar vehículo
- UC18: Editar o desactivar vehículo

---

### 2.3. Modelo conceptual de datos (incluyendo cardinalidades)

El modelo E-R define relaciones entre:

- **User** (Admin o Trabajador) y **Envío**: N:M (una preparación puede involucrar varios usuarios).
- **Envío** y **Paquete**: 1:N (un envío contiene muchos paquetes).
- **Envío** y **Ruta**: N:1 (cada envío va por una ruta).
- **Ruta** y **Vehículo**: N:M mediante la entidad asociativa **RutaVehículo** con `fecha_asignacion`.

(Referencia: segundo y tercer diagrama compartido)

---

### 2.4. Arquitectura de software de la solución

- **Frontend**: JavaFX (pantallas Splash, SignIn, SignUp, Menú con navegación dinámica, etc.)
- **Persistencia**: Hibernate (mapeo objeto-relacional a las entidades definidas)
- **Backend**: DAO + lógica de servicios (gestión de rutas, paquetes, usuarios, etc.)
- **Base de datos**: PostgreSQL (estructura conforme a modelo E-R)
- **Gestión de sesiones y roles**: Implementado en capa de servicios