

# **Documento de requerimientos de software**

***APLICACIÓN WEB CON INTELIGENCIA ARTIFICIAL PARA LA  
GESTIÓN DE TRANSPORTE PESADO EN VOLECE.CA***

***Fecha: 10/05/2025***

## Tabla de contenido

Historial de Versiones .....	3
Información del Proyecto .....	3
Aprobaciones .....	3
1. Propósito .....	4
2. Alcance del producto / Software .....	4
3. Referencias .....	5
4. Funcionalidades del producto .....	5
5. Clases y características de usuarios .....	5
6. Entorno operativo .....	5
7. Requerimientos funcionales .....	6
8. Reglas de negocio .....	10
9. Requerimientos de interfaces externas .....	11
10. Requerimientos no funcionales .....	13
11. Otros requerimientos .....	17
12. Glosario .....	17

## Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
15/04/2025	1	Alex Vallejo	ESPOCH	1ERA VERSIÓN DE REQUISITOS
10/05/2025	2	Alex Vallejo	ESPOCH	2DA VERSIÓN DE REQUISITOS (MAS COMPLETOS)

## Información del Proyecto

Empresa / Organización	Compañía de Volquetas El Cedral VOLECE C.A.
Proyecto	VOLECE C.A.
Fecha de preparación	10/05/2025
Cliente	VOLECE C.A.
Patrocinador principal	Alex Vallejo
Gerente / Líder de Proyecto	Alex Vallejo
Gerente / Líder de Análisis de negocio y requerimientos	Alex Vallejo

## Aprobaciones

Nombre y Apellido	Cargo	Departamento u Organización	Fecha	Firma

## 1. Propósito

Este proyecto tiene como propósito desarrollar una aplicación web que ayude con la optimización de procesos en la gestión administrativa y logísticos de la compañía Volquetas el Cedral VOLECE C.A. La aplicación se desarrollará con varios módulos orientados a las áreas mas importantes que necesitan optimizar procesos como la solicitud de servicios, la asignación de turnos equitativamente, gestión de los vehículos, y generación de reportes. Esta aplicación abarca la totalidad del sistema, con la integración de un módulo inteligente que automatizara tareas clave mediante el uso de inteligencia artificial, buscando resolver los principales problemas operativos que enfrenta la compañía: falta de digitalización, asignación manual e injusta de turnos. La solución completa será diseñada desde cero, con enfoque modular y escalable, aplicando metodologías ágiles para asegurar entregas funcionales continuas durante el proceso de implementación.

## 2. Alcance del producto / Software

El alcance de este proyecto es el desarrollo de una aplicación Web para la compañía VOLECE C.A. El objetivo general es automatizar procesos principales como la solicitud de servicios, la asignación de turnos para los transportistas de forma equitativa, gestionar la flota vehicular y generar reportes, lo que ayudara a tener mejor control y transparencia dentro de la organización y mejorando la eficiencia operativa y la experiencia de usuario. Esta aplicación traerá grandes beneficios para que la administración pueda tomar mejores decisiones y defina estrategias que la vuelvan más competitiva en el mercado. El proyecto se alinea con objetivos de crecimiento y modernización al incluir tecnologías eficientes que ayuden a mejorar la productividad. Se desarrollará aplicando una metodología ágil Scrum para poner dar entregas parciales del aplicativo y poner realizar pruebas continuas. El desarrollo estará limitado por el tiempo que está programado para la entrega de producto.

### 3. Referencias

Políticas internas de la compañía  
Estatutos de la compañía  
Documentos de versiones del proyecto

### 4. Funcionalidades del producto

- **Gestión de usuarios:** Registro, inicio de sesión y administración de roles (Administrador, Transportista, Cliente).
- **Solicitud de servicios:** Permite a los clientes ingresar solicitudes de transporte.
- **Asignación de turnos con IA:** Asignación automática y equitativa de turnos de viaje a los transportistas.
- **Gestión de vehículos:** Registro de estado, programación de mantenimiento y notificaciones de uso.
- **Generación de reportes:** Visualización y exportación de reportes de turnos, viajes y estados financieros.

### 5. Clases y características de usuarios

- **Administrador:** Tiene acceso completo al sistema. Puede gestionar usuarios, aprobar solicitudes, revisar reportes del sistema.
- **Transportistas (Socio):** Visualiza turnos asignados, estados financieros y reportes personales.
- **Cliente:** Solicita servicios de transporte, realiza seguimiento de sus pedidos, y accede a su historial de servicios.

### 6. Entorno operativo

**Hardware:** Computadoras personales o laptops estándar, acceso desde navegadores web modernos.

**Sistema operativo:** Compatible con Windows, macOS y Linux.

**Software base:**

- Backend: Python con Django.
- Frontend: React JS.
- Base de datos: PostgreSQL.
- Navegadores: Chrome, Firefox, Edge.

## 7. Requerimientos funcionales

### 7.1 Autenticación de usuarios

#### Descripción

Permitir a los clientes, administradores y transportistas poder registrarse e ingresar a la aplicación de forma segura con credenciales personales (usuario y contraseña), garantizando que solo los usuarios autenticados puedan acceder.

#### Requisitos Funcionales

ID	Requisito Funcional
RF01	El sistema debe permitir el registro público únicamente a usuarios con rol CLIENTE.
RF02	El sistema debe permitir el inicio de sesión a cualquier usuario registrado.
RF03	El sistema debe permitir al Administrador crear nuevos usuarios con rol ADMIN o TRANSPORTISTA.
RF04	El sistema debe redirigir al dashboard adecuado según el rol del usuario autenticado.
RF05	El sistema debe validar que todos los campos obligatorios sean completados en el registro.
RF06	El sistema debe proteger las rutas privadas mediante autenticación JWT.

## 7.2 Gestión de usuarios

La aplicación debe permitir a el administrador poder gestionar a los administradores a los usuarios que están registrados en la aplicación, además poder ingresar a los nuevos transportistas.

ID	Requisito Funcional
RF07	El sistema debe permitir al administrador visualizar a todos los usuarios registrados en una tabla.
RF08	El sistema debe permitir al administrador registrar nuevos usuarios (CLIENTE, ADMIN, TRANSPORTISTA).
RF09	El sistema debe permitir al administrador editar la información de los usuarios existentes.
RF10	El sistema debe permitir al administrador eliminar usuarios del sistema de forma segura.
RF11	La tabla debe permitir ordenar, buscar y paginar los usuarios listados.
RF12	Las operaciones CRUD deben realizarse mediante API protegida por autenticación JWT.

## 7.3 Solicitud de servicios

El sistema permite a los clientes realizar peticiones de servicios de transporte pesado añadiendo campos como origen y destino, y además podrá visualizar si el administrador ya acepto su solicitud.

ID	Requisito Funcional
RF13	El sistema debe permitir a los clientes registrar una solicitud de transporte desde un formulario web.
RF14	El formulario debe incluir campos como: origen, destino, tipo de carga, fecha y observaciones.

RF15	La solicitud registrada debe quedar asociada al cliente autenticado.
RF16	El sistema debe permitir al cliente visualizar el historial de sus solicitudes en una tabla.
RF17	El sistema debe permitir al administrador ver todas las solicitudes de todos los clientes.
RF18	El administrador debe poder asignar o rechazar solicitudes desde una interfaz administrativa.
RF19	El sistema debe mostrar el estado actual de cada solicitud: pendiente, en proceso, completada, rechazada.

## 7.4 Gestión de vehículos

El sistema tiene la funcionalidad en la cual se podrá gestionar los vehículos que tiene cada transportista y por el lado del transportista se podrá visualizar los datos del vehículo y registrar los mantenimientos que realiza a su vehículo.

ID	Requisito Funcional
RF29	El sistema debe permitir al administrador registrar un nuevo vehículo desde un formulario web.
RF30	El sistema debe permitir seleccionar un transportista existente para asociarlo al vehículo.
RF31	El formulario debe incluir los campos: tipo, marca, modelo, placa, año, color, tonelaje, observaciones.
RF32	El backend debe validar que el año del vehículo esté entre 1990 y el año actual.
RF33	El backend debe validar que el tonelaje esté entre 3.5 y 50 toneladas.
RF34	El sistema debe guardar correctamente el vehículo y devolver un mensaje de confirmación.



RF35	El sistema debe ofrecer un endpoint API que permita registrar, consultar y modificar vehículos registrados.
RF36	El sistema debe permitir al transportista visualizar los datos técnicos y estado de su propio vehículo.
RF37	El sistema debe permitir al transportista cambiar el estado del vehículo (Activo, Pasivo, Mantenimiento).
RF38	Si el estado del vehículo se cambia a "Mantenimiento", este debe quedar no disponible para turnos.
RF39	El sistema debe permitir registrar un nuevo mantenimiento preventivo con los datos correspondientes.
RF40	El sistema debe mantener un historial completo de todos los mantenimientos registrados.
RF41	El transportista debe poder filtrar el historial por rango de fechas.
RF42	El sistema debe generar alertas automáticas cuando un mantenimiento esté próximo (menos de 500 km).
RF43	Las alertas deben ser visibles desde el dashboard del transportista.
RF44	Solo el transportista dueño del vehículo debe poder modificar o registrar información sobre su vehículo.

## 7.5 Asignación de turnos con IA

El sistema mostrara al administrador cual es el transportista sugerido para realizar el servicio, este modulo integra IA para seleccionar a que transportista le toca cada turno teniendo en cuenta datos como disponibilidad y estado del vehículo.

ID	Requisito Funcional
RF45	El sistema debe permitir generar un dataset sintético con al menos 5000 registros representativos de turnos.

RF46	El sistema debe entrenar un modelo de IA supervisado para predecir asignaciones de turnos.
RF47	El modelo entrenado debe ser almacenado como archivo .pkl y cargado correctamente por el backend.
RF48	El backend debe exponer un endpoint REST (/asignar-turno) que permita realizar predicciones con el modelo.
RF49	El administrador debe poder visualizar la predicción de turno sugerido por la IA desde el frontend.
RF50	El sistema debe permitir al administrador aceptar o modificar la asignación sugerida.
RF51	La decisión final del turno asignado debe guardarse en la base de datos.
RF52	El sistema debe calcular una métrica de equidad de las asignaciones (por ejemplo, desviación estándar).
RF53	El sistema debe mostrar una visualización gráfica de las asignaciones realizadas por IA.
RF54	El informe de equidad debe poder exportarse en formato PDF.

## 8. Reglas de negocio

El sistema deberá operar bajo un conjunto de reglas de negocio que garanticen el correcto funcionamiento de los procesos administrativos y logísticos de la compañía VOLECE C.A.

Primero, cada usuario registrado deberá ser clasificado obligatoriamente como Administrador, Transportista o Cliente, y su acceso estará restringido según su rol. Solo los administradores podrán gestionar usuarios, validar solicitudes y generar reportes globales. Los clientes únicamente tendrán acceso a funcionalidades relacionadas con la solicitud de servicios y el seguimiento de sus estados. Por otro lado, los turnos de viaje serán asignados automáticamente mediante un algoritmo

de inteligencia artificial que tomará en cuenta criterios preestablecidos, como la disponibilidad del vehículo, frecuencia de asignación y registros anteriores, garantizando así una distribución equitativa entre los socios transportistas.

## 9. Requerimientos de interfaces externas

### 9.1 Interfaces de usuario

La aplicación contará con una interfaz web moderna y responsiva, desarrollada en React. Cada usuario verá pantallas personalizadas según su rol (Administrador, Transportista o Cliente). Se utilizarán formularios simples, menús intuitivos, botones destacados y paneles de control organizados por módulos.

La navegación será clara y guiada, con mensajes visuales que confirmen acciones o alerten sobre errores. Se aplicarán principios de usabilidad para asegurar una experiencia amigable y accesible, incluso para usuarios sin experiencia técnica.

Requisitos para la interfaz principal.

ID	Requisito Funcional
RF20	El sistema debe permitir visualizar los tipos de carga ofrecidos por VOLECE C.A. con imágenes y descripciones.
RF21	La sección de tipos de carga debe ser accesible desde el menú de navegación.
RF22	El visitante debe poder consultar los servicios especializados disponibles con representación visual.
RF23	El sistema debe mostrar la ubicación geográfica de la empresa mediante un mapa integrado.
RF24	El visitante debe poder ver la sección “¿Por qué elegir VOLECE C.A.?” con ventajas competitivas destacadas.

RF25	La página debe mostrar estadísticas clave sobre la empresa (años, clientes, servicios, etc.).
RF26	El visitante debe poder consultar información institucional desde el menú “Nosotros” mediante modales.
RF27	El visitante debe poder navegar desde la página principal e informarse fácilmente sobre todos los servicios.
RF28	La interfaz debe permitir interacción directa mediante botones como WhatsApp o contacto modal.

## 9.2 Interfaces de hardware

El sistema podrá ser utilizado desde computadoras personales, laptops, o teléfonos.

## 9.3 Interfaces de software

El frontend (React) se comunicará con el backend (Django) mediante APIs REST en formato JSON. Estas interfaces permiten enviar y recibir datos de forma rápida y estructurada.

La aplicación también integrará librerías de inteligencia artificial en Python, como *scikit-learn*, que procesarán la información histórica para asignar turnos de manera automática y justa.

## 9.4 Interfaces de comunicación

La plataforma utilizará conexiones seguras HTTPS para proteger los datos transmitidos.

Se implementará un sistema de notificaciones internas (en pantalla) para avisar sobre turnos, mantenimientos, pagos o eventos importantes.

## 10. Requerimientos no funcionales

ID	Requisito No Funcional
RNF01	El tiempo de respuesta del login debe ser menor a 1 segundo bajo condiciones normales.
RNF02	El sistema debe impedir accesos no autorizados, retornando códigos HTTP 401 o 403 según corresponda.
RNF03	La interfaz de login y registro debe ser responsive y usable en dispositivos móviles y escritorio.
RNF04	La autenticación debe utilizar el estándar JWT (JSON Web Token) para mantener sesiones seguras.
RNF05	El formulario de registro debe mostrar mensajes de error claros y contextualizados.
RNF06	Los datos deben estar validados tanto en frontend como backend.
RNF07	Solo los administradores autenticados deben tener acceso al módulo de gestión de usuarios.
RNF08	La tabla de usuarios debe ser responsive y adaptarse a distintos tamaños de pantalla.
RNF09	Los formularios deben mostrar mensajes claros y amigables ante errores de validación o envío.
RNF10	La eliminación de usuarios debe estar protegida por una confirmación previa en el frontend.
RNF11	El rendimiento de carga de la tabla debe ser óptimo, soportando mínimo 100 usuarios sin afectar la fluidez.
RNF12	Las llamadas a la API deben tener un tiempo de respuesta menor a 800ms en condiciones normales.
RNF13	El formulario de solicitud debe tener validación de campos obligatorios en frontend y backend.

RNF14	Las operaciones CRUD deben ser seguras y autenticadas mediante JWT.
RNF15	La interfaz del cliente debe ser responsive y clara en dispositivos móviles y de escritorio.
RNF16	El estado de la solicitud debe actualizarse en tiempo real o con una recarga mínima de datos.
RNF17	El tiempo de respuesta al enviar o actualizar una solicitud debe ser menor a 1 segundo bajo carga normal.
RNF18	Si no hay solicitudes registradas, el sistema debe mostrar un mensaje claro e informativo.
RNF19	Toda la página debe ser responsiva y funcionar correctamente en distintos tamaños de pantalla y navegadores.
RNF20	La interfaz debe ser visualmente clara y mantener una estructura consistente entre secciones.
RNF21	El tiempo de carga inicial de la página debe ser inferior a 2 segundos en condiciones normales.
RNF22	Los modales deben tener comportamiento accesible y fluido (abrir, cerrar, foco, etc.).
RNF23	Los enlaces del menú deben estar operativos y correctamente dirigidos a las secciones respectivas.
RNF24	Las secciones visuales deben incluir íconos, imágenes y descripciones legibles, sin solapamientos.
RNF25	Todos los textos deben tener contraste suficiente y ser legibles en pantallas de baja y alta resolución.
RNF26	La interfaz de registro debe ser responsive y usable en computadoras y dispositivos móviles.
RNF27	El sistema debe mostrar mensajes claros en caso de error de validación o éxito de registro.

RNF28	El tiempo de respuesta al registrar un vehículo debe ser menor a 1 segundo bajo condiciones normales.
RNF29	Las validaciones deben estar implementadas tanto en el frontend como en el backend.
RNF30	El formulario debe prevenir el envío si hay campos obligatorios vacíos o inválidos.
RNF31	El diseño del formulario debe ser intuitivo y estéticamente coherente con el resto de la plataforma VOLECE.
RNF32	Toda la funcionalidad del módulo debe estar protegida por autenticación JWT, asegurando acceso por rol.
RNF33	La interfaz debe ser responsive y accesible en dispositivos móviles y de escritorio.
RNF34	El cambio de estado y registro de mantenimiento deben reflejarse inmediatamente en la interfaz.
RNF35	El tiempo de respuesta de cada operación no debe superar 1 segundo en condiciones normales.
RNF36	Los formularios deben incluir validación en frontend y backend de los campos requeridos.
RNF37	Las alertas generadas deben almacenarse de forma persistente y estar visibles en cada sesión del usuario.
RNF38	Las búsquedas y filtros deben realizarse con mínimo de recarga y con tiempos de respuesta óptimos.
RNF39	El script de generación de datos sintéticos debe ejecutarse en menos de 30 segundos para 5000 registros.
RNF40	El modelo entrenado debe tener precisión $\geq 85\%$ en validación para considerarse aceptable.
RNF41	El endpoint /asignar-turno debe responder en menos de 1 segundo bajo condiciones normales.

RNF42	Toda la lógica de IA debe estar integrada de forma segura en el backend Django, sin exponer rutas críticas.
RNF43	La pantalla de predicción debe ser intuitiva y responsiva para que el administrador tome decisiones fácilmente.
RNF44	La visualización de la equidad debe ser clara, comprensible y generarse sin errores.
RNF45	El sistema debe prevenir sesgos en los datos y garantizar distribución balanceada durante el entrenamiento.



## 11. Otros requerimientos

Correcto diseño de la base de datos.

## 12. Glosario

Por definir