**Documento de Requerimientos de software**

***Desarrollo de una aplicación web con un asistente inteligente para la gestión de pedidos y entregas de la empresa FERKONSA S.A.***

***Fecha: 17/06/2025***

Contenido

[Historial de Versiones 4](#_Toc196558613)

[Información del Proyecto 4](#_Toc196558614)

[Aprobaciones 4](#_Toc196558615)

[1. Propósito 5](#_Toc196558616)

[2. Alcance del producto / Software 5](#_Toc196558617)

[3. Referencias 6](#_Toc196558618)

[4. Funcionalidades del producto 7](#_Toc196558619)

[5. Clases y características de usuarios 8](#_Toc196558620)

[6. Entorno operativo 9](#_Toc196558621)

[7. Requerimientos funcionales 9](#_Toc196558622)

[7.1 Inicio de sesión y autenticación 9](#_Toc196558623)

[7.2 Validación de nuevos usuarios por el superadministrador 10](#_Toc196558624)

[7.3 Registro de pedidos vía chatbot de WhatsApp 10](#_Toc196558625)

[7.4 Verificación automática de disponibilidad de productos (API) 10](#_Toc196558626)

[7.5 Registro automático del pedido en el sistema 10](#_Toc196558627)

[7.6 Consulta del estado del pedido (cliente vía chatbot) 11](#_Toc196558628)

[7.7 Gestión y visualización de pedidos (administrador) 11](#_Toc196558629)

[7.8 Asignación de camiones y choferes 11](#_Toc196558630)

[7.9 Actualización del estado del envío (chofer) 11](#_Toc196558631)

[7.10 Historial de entregas 11](#_Toc196558632)

[7.11 Reporte financiero por envío 12](#_Toc196558633)

[7.12 Gestión de usuarios y camiones 12](#_Toc196558634)

[8. Reglas de negocio 12](#_Toc196558635)

[9. Requerimientos de interfaces externas 13](#_Toc196558636)

[9.1. Interfaces de usuario 13](#_Toc196558637)

[9.2. Interfaces de hardware 16](#_Toc196558638)

[9.3. Interfaces de software 17](#_Toc196558639)

[9.4. Interfaces de comunicación 22](#_Toc196558640)

[10. Requerimientos no funcionales 25](#_Toc196558641)

[11. Otros Requerimientos 26](#_Toc196558642)

[12. Glosario 27](#_Toc196558643)

# Historial de Versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| 16/04/2025 | 1 | Franklin Satuquinga  Alejandro Yaucen | KERKONSA S.A. |  |
| 27/04/2025 | 2 | Franklin Satuquinga  Alejandro Yaucen | KERKONSA S.A. |  |
| 19/05/2025 | 3 | Franklin Satuquinga  Alejandro Yaucen | KERKONSA S.A. |  |
| 18/06/2025 | 4 | Franklin Satuquinga  Alejandro Yaucen | KERKONSA S.A. |  |

# Información del Proyecto

|  |  |
| --- | --- |
| Empresa / Organización | FERKONSA S.A. |
| Proyecto | Desarrollo de una aplicación web con un asistente inteligente para la gestión de pedidos y entregas de la empresa FERKONSA S.A. |
| Fecha de preparación | 16/04/2025 |
| Cliente | Miguel Santillán |
| Patrocinador principal | ESPOCH |
| Gerente / Líder de Proyecto | Alejandro Yaucen |
| Gerente / Líder de Análisis de negocio y requerimientos | Franklin Satuquinga |

# Aprobaciones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Propósito

Construcción de una aplicación web con un asistente inteligente para la gestión de pedidos y entregas de la empresa FERKONSA S.A.

Este documento tiene como propósito especificar los requisitos de la aplicación web para FERKONSA S.A., una ferretería que busca modernizar y optimizar el proceso de atención de pedidos y distribución de productos.

El documento cubre el alcance completo del sistema, que incluye:

* Módulo del administrador, con gestión de usuarios, camiones, choferes, pedidos, asignaciones y reportes.
* Módulo del chofer, con visualización de entregas asignadas, cambio de estado y visualización de historial.
* Integración con un asistente inteligente para registro automatizado de pedidos vía WhatsApp Business mediante Infobip.

# Alcance del producto / Software

El sistema web está orientado a digitalizar, automatizar y centralizar la gestión de pedidos y entregas de materiales en FERKONSA S.A., integrando una interfaz web moderna para los roles internos y un asistente inteligente para clientes externos.

**Objetivo General:**

Desarrollar una aplicación web automatizada con la integración de un asistente inteligente para mejorar la gestión de pedidos y entregas en FERKONSA S.A

**Beneficios:**

* Optimización del tiempo y procesos logísticos.
* Reducción de errores humanos en la gestión de pedidos.
* Mejora en la trazabilidad y control de entregas.
* Mayor satisfacción del cliente al tener visibilidad del estado de sus pedidos.
* Ahorro de recursos al reemplazar procesos manuales por automatizados.

**Objetivos y Metas:**

* Facilitar la recepción automatizada de pedidos mediante chatbot.
* Agilizar la asignación de camiones y choferes para entregas.
* Permitir el seguimiento del estado del pedido por parte del cliente y la empresa.
* Gestionar de manera eficiente los usuarios del sistema, tanto internos como externos.
* Generar reportes históricos para análisis logístico y operativo.

# Referencias

Título: Integración de Chatbots de IA para la Optimización de Pedidos en Servicios Alimentarios

Autor: Vivas-Naranjo, Martín; Cueva-Costales, Andrés

Fecha: 2021-02-04

Ubicación: Ecuador

[1] M. E. Vivas-Naranjo y A. J. Cueva-Costales, “Integración de Chatbots de IA para la Optimización de Pedidos en Servicios Alimentarios: Caso de Estudio en ‘Cafetería La Estación 04 – Ibarra Ecuador’,” 593 Digital Publisher CEIT, vol. 9, no. 3, pp. 790–802, 2024. [En línea]. Disponible en:

<https://dialnet.unirioja.es/servlet/articulo?codigo=9535934>​

Título: Análisis, Diseño, e Implementación de un Software, para la recepción de pedidos a domicilio en la empresa Chef’s Subs & Grill Delivery, aplicando una metodología ágil

Autor: Eduardo Ramiro Vizuete Albán

Fecha: 2020

Ubicación: Ecuador

[2] E. R. Vizuete Albán, Análisis, Diseño, e Implementación de un Software, para la recepción de pedidos a domicilio en la empresa Chef’s Subs & Grill Delivery, aplicando una metodología ágil, Ecuador, 2020. [Tesis de grado]. Disponible en: <http://204.199.82.243:8080/handle/123456789/1147>

Título: Propuesta de mejora para reducir los retrasos en la entrega de pedidos, mediante la aplicación de Lean Manufacturing en una empresa metalmecánica

Autor: Rodriguez Valdivia, Maria Jessica; Wong Herrera, Josiel Yosiana

Fecha: 2021

Ubicación: Perú

[3] M. J. Rodriguez Valdivia y J. Y. Wong Herrera, “Propuesta de mejora para reducir los retrasos en la entrega de pedidos, mediante la aplicación de Lean Manufacturing en una empresa metalmecánica,” Tesis de Bachiller en Ingeniería Industrial, Universidad Peruana de Ciencias Aplicadas (UPC), Lima, Perú, 2021. [En línea]. Disponible en: <https://repositorioacademico.upc.edu.pe/handle/10757/654994>

[1] M. Langarano Guerrero, F. Montaluisa Yugla y M. Navas Moya, “Implementación de un chatbot con NLP para recibir pedidos en una plataforma de delivery,” *Revista Tecnológica - ESPOL*, vol. 34, no. 3, pp. 157–170, 2022. [En línea]. Disponible: <https://doi.org/10.37815/rte.v34n3.958>

[2] B. A. Pelaes Dávalos, “Desarrollo de un chatbot para automatizar el proceso de reserva de citas del centro de terapias alternativas Dharana,” Tesis de grado, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2022. [En línea]. Disponible: <http://dspace.espoch.edu.ec/handle/123456789/21948>

[3] B. S. Guaylla Ashqui y J. S. Burgos Meléndrez, “Desarrollo de un chatbot basado en inteligencia artificial para atención al cliente de la farmacia San Vicente del cantón Chambo,” Tesis de grado, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2024. [En línea]. Disponible: <http://dspace.espoch.edu.ec/handle/123456789/22006>

[4] J. A. Bejarano Villalobos, “Desarrollo de una aplicación web para la gestión de ventas de la ferretería Improplac mejorando la atención al cliente usando un chatbot,” Tesis de grado, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2024. [En línea]. Disponible: <http://dspace.espoch.edu.ec/handle/123456789/23467>

# Funcionalidades del producto

El software incluirá las siguientes funcionalidades principales:

1. Registro e inicio de sesión de usuarios.
2. Aprobación o rechazo de nuevos registros por parte del superadministrador.
3. Registro de pedidos por parte del cliente mediante un chatbot.
4. Generación de alertas al administrador cuando un nuevo pedido es registrado.
5. Asignación de camiones y choferes a pedidos pendientes.
6. Actualización del estado de la entrega por parte del chofer.
7. Seguimiento del estado del pedido por parte del cliente a través del chatbot.
8. Visualización del historial de entregas tanto por parte del administrador como por parte del chofer.
9. Gestión de usuarios y camiones desde el panel del administrador.
10. Generación de reportes administrativos.

# Clases y características de usuarios

El sistema estará diseñado para ser utilizado por diversos tipos de usuarios, cada uno con diferentes privilegios y responsabilidades. A continuación, se detallan las clases de usuarios:

**1. Cliente**

* **Descripción:** Persona que realiza pedidos de materiales de construcción a través del chatbot de WhatsApp.
* **Nivel de experiencia:** Usuario general sin conocimientos técnicos, interacción simplificada vía chatbot.
* **Frecuencia de uso:** Ocasional, dependiendo de la necesidad de compra.
* **Medio de interacción:** Chatbot (no accede a la aplicación web).
* **Privilegios:** Solo puede registrar pedidos y consultar su estado.
* **Funcionalidades asociadas:**
  + Funcionalidad 3
  + Funcionalidad 7

**2. Administrador**

* **Descripción:** Usuario responsable de gestionar pedidos, camiones, choferes, usuarios y reportes. Tiene acceso completo a las funciones administrativas del sistema.
* **Nivel de experiencia:** Medio-alto. Personal de planta con conocimientos del proceso logístico.
* **Frecuencia de uso:** Alta, uso diario del sistema.
* **Privilegios:** Acceso completo a módulos administrativos.
* **Funcionalidades asociadas:**
  + Funcionalidad 1
  + Funcionalidad 5
  + Funcionalidad 8
  + Funcionalidad 9
  + Funcionalidad 10

**3. Chofer**

* **Descripción:** Usuario encargado de realizar la entrega de los pedidos asignados y actualizar su estado en el sistema.
* **Nivel de experiencia:** Bajo-medio. Solo requiere conocimientos básicos del sistema.
* **Frecuencia de uso:** Media, según la cantidad de entregas asignadas.
* **Privilegios:** Puede visualizar pedidos asignados y actualizar su estado.
* **Funcionalidades asociadas:**
  + Funcionalidad 1
  + Funcionalidad 6
  + Funcionalidad 8

**4. Superadministrador**

* **Descripción:** Usuario con el rol de validar y aprobar nuevos registros de usuarios antes de que accedan al sistema.
* **Nivel de experiencia:** Alto. Encargado del control y autorización de accesos.
* **Frecuencia de uso:** Baja, solo cuando se registran nuevos usuarios.
* **Privilegios:** Exclusivamente validación de accesos.
* **Funcionalidades asociadas:**
  + Funcionalidad 2

# Entorno operativo

Pila de Herramientas y Tecnologías:

**Frontend**:

* Lenguaje principal: JavaScript.
* Framework: React.js.
* Estilizado: Tailwind CSS.
* Librerías útiles:
  + Axios (para consumir APIs).
  + React Router (para navegación).

**Backend:**

* Lenguaje: JavaScript.
* Framework: Node.js con Express.js.
* Base de datos: PostgreSQL.
* Autenticación: JWT (JSON Web Tokens).
* Chatbot + IA:
  + Integración con WhatsApp Business API (Infobip).
  + Modelo NLP básico con Dialogflow.

# Requerimientos funcionales

## Inicio de sesión y autenticación

* RF1.1: El sistema debe permitir el inicio de sesión mediante usuario y contraseña.
* RF1.2: El sistema debe validar las credenciales y permitir el acceso solo a usuarios activos.
* RF1.3: El sistema debe mostrar mensajes de error cuando las credenciales sean incorrectas.
* RF1.4: El sistema debe permitir la recuperación de contraseña mediante correo electrónico.

## 7.2 Validación de nuevos usuarios por el superadministrador

* RF2.1: El sistema debe permitir el registro de nuevos usuarios tipo chofer o administrador.
* RF2.2: El sistema debe enviar una notificación por correo electrónico al superadministrador cuando un usuario se registre.
* RF2.3: El sistema debe permitir al superadministrador aprobar o rechazar el acceso de nuevos usuarios.
* RF2.4: Solo usuarios aprobados podrán iniciar sesión en el sistema.

## 7.3 Registro de pedidos vía chatbot de WhatsApp

* RF3.1: El chatbot debe permitir al cliente iniciar el proceso de pedido mediante mensajes en WhatsApp.
* RF3.2: El chatbot debe solicitar al cliente la información necesaria para generar el pedido (productos, cantidades, dirección, cliente).
* RF3.3: El sistema debe registrar automáticamente el pedido una vez completado por el cliente en el chatbot.
* RF3.4: El chatbot debe confirmar al cliente que el pedido fue registrado con éxito.

## 7.4 Registro automático del pedido en el sistema

* RF5.1: El sistema debe almacenar los datos del pedido una vez que el chatbot los reciba correctamente.
* RF5.2: El sistema debe asignar un estado inicial "pendiente" a todo nuevo pedido.

## 7.6 Consulta del estado del pedido (cliente vía chatbot)

* RF6.1: El chatbot debe permitir al cliente consultar el estado actual de su pedido.
* RF6.2: El chatbot debe mostrar el estado del pedido en texto simple (pendiente, en camino, entregado, no entregado).

## 7.7 Gestión y visualización de pedidos (administrador)

* RF7.1: El administrador debe recibir una notificación cada que se registre un pedido
* RF7.2: El administrador debe poder filtrar pedidos
* RF7.3: El administrador debe poder editar detalles de un pedido antes de ser despachado.

## 7.8 Asignación de camiones y choferes

* RF8.1: El administrador debe poder ver la disponibilidad de camiones y choferes.
* RF8.2: El administrador debe poder asignar manualmente un camión y un chofer a un pedido.
* RF8.3: El sistema debe impedir la asignación si el camión o chofer ya están ocupados.
* RF8.4: El sistema debe actualizar el estado del camión y chofer a “ocupado” al ser asignados.

## 7.9 Actualización del estado del envío (chofer)

* RF9.1: El chofer debe recibir una notificación al ser asignado a un pedido
* RF9.2: El chofer debe poder actualizar el estado del pedido a “en camino”, “entregado” o “no entregado”.
* RF9.4: El sistema debe evitar cambios de estado si no siguen la secuencia lógica definida.

## 7.10 Historial de entregas

* RF10.1: El chofer debe poder visualizar su historial de entregas completadas.
* RF10.2: El administrador debe poder consultar el historial de entregas de todos los choferes.
* RF10.3: El historial debe incluir fechas, pedidos, estados y camiones usados.

## 7.11 Reporte financiero por envío

* RF11.1: El sistema debe calcular el valor del envío en función de la distancia y políticas definidas.
* RF11.2: El sistema debe permitir identificar envíos gratuitos.
* RF11.3: El administrador debe poder generar reportes por fecha, chofer, tipo de envío y valor cobrado.
* RF11.4: El reporte debe estar disponible en formato visual y exportable.

## 7.12 Gestión de usuarios y camiones

* RF12.1: El administrador debe poder editar y eliminar usuarios (choferes y administradores).
* RF12.2: El administrador debe poder registrar, editar y eliminar camiones.
* RF12.3: El sistema debe validar datos únicos como correo electrónico y placa del camión.
* RF12.4: El sistema debe impedir acciones si los datos están incompletos o inválidos.

# Reglas de negocio

**RB-01. Registro de usuarios restringido por validación del Superadministrador**

Todo nuevo usuario que desee acceder al sistema (chofer o administrador) debe ser validado por el Superadministrador. La validación se realiza mediante notificación por correo electrónico. Sin esta validación, no se permite el acceso.

**RB-02. Clientes no interactúan directamente con el sistema**  
El único medio por el cual los clientes pueden interactuar con la empresa para realizar pedidos es mediante el chatbot de WhatsApp. No tienen acceso a la aplicación web ni pueden visualizar funcionalidades internas del sistema.

**RB-03. Un camión y chofer pueden estar asignados a un solo pedido a la vez**  
Para mantener la integridad y eficiencia del proceso logístico, un camión y su chofer no pueden ser asignados a más de un pedido simultáneamente.

**RB-04. Verificación manual del stock antes del despacho**  
Antes de la entrega de un pedido, una persona designada debe validar manualmente si los materiales están completos. Sin esta verificación, el pedido no puede ser despachado.

**RB-05. Asignación de camiones solo por un administrador autorizado**  
La asignación de un camión y chofer para un pedido debe ser realizada exclusivamente por un administrador. No se permite la autoasignación ni decisiones automáticas sin intervención humana (en la situación actual).

**RB-06. Costo de envío depende de la distancia**  
El valor del envío se calcula de forma manual dependiendo de la distancia del lugar de entrega. Sin embargo, existen casos excepcionales donde el envío es gratuito, a criterio del personal autorizado.

**RB-07. No se estima el tiempo de entrega automáticamente**  
Actualmente, no existe una fórmula ni lógica sistematizada para calcular el tiempo estimado de entrega. Esta información es evaluada de forma empírica por el personal encargado.

**RB-08. Confirmación de entrega sin formalización digital**  
La entrega del pedido se considera completada al recibir una fotografía por WhatsApp del pedido entregado. No existe firma digital, geolocalización o evidencia estructurada.

**RB-09. No existe seguimiento en tiempo real del pedido**  
Una vez realizado el pedido, ni el cliente ni el personal pueden hacer seguimiento digital de su ubicación o estado. La única forma de obtener información es mediante llamadas o mensajes directos.

# Requerimientos de interfaces externas

## Interfaces de usuario

El sistema contará con interfaces gráficas diferenciadas para cada tipo de usuario: **Chofer** y **Administrador del Sistema**. Cada una se diseñará de forma intuitiva, moderna y con una experiencia de usuario centrada en la simplicidad, tomando como base el framework visual Roboto y una paleta de colores que transmita seriedad y dinamismo. A continuación, se describen las características de las interfaces de usuario del sistema de gestión de pedidos y entregas para la ferretería FERKONSA S.A., clasificadas por tipo de usuario y área funcional.

**Clasificación por tipo de usuario y áreas del sistema**

**1. Cliente (vía Chatbot)**

* **Área: Chatbot inteligente de pedidos**
  + Medio principal de interacción del cliente.
  + Permite realizar pedidos mediante lenguaje natural.
* **Área: Consulta de estado de pedido**
  + Consulta de estado mediante chatbot
  + Se muestran los siguientes datos:
    - Estado actual (pendiente, en camino, entregado, no entregado)
    - Información del pedido (fecha, productos, valor total)
    - Información de envío (nombre del chofer, camión asignado)

**2. Chofer**

* **Área: Panel de entregas asignadas**
  + Requiere autenticación.
  + Visualiza una lista de entregas pendientes y en curso.
  + Puede actualizar el estado de la entrega (Pendiente, En camino, Entregado, No entregado).
  + Accede a historial de entregas con filtros por fecha o estado.
  + Información por entrega:
    - Código del pedido
    - Cliente y dirección
    - Botón de actualización de estado

**3. Administrador del sistema**

* **Área: Gestión de pedidos y asignaciones**
  + Asigna camiones y choferes a pedidos pendientes.
  + Consulta pedidos por estado, chofer o fecha.
  + Edita datos del pedido en caso de errores detectados.
* **Área: Gestión de camiones**
  + Alta, baja y edición de camiones.
  + Asignación automática o manual de vehículos a entregas.
* **Área: Gestión de usuarios**
  + Edición y eliminación de usuarios choferes o administrativos.
* **Área: Reportes**
  + Generación de reportes financieros y logísticos.
  + Filtros por fecha, chofer, estado del pedido

**Estándares de Interfaz Gráfica (GUI)**

**Guía de estilo y organización visual**

**Paleta de colores:**

* Rojo corporativo: #ff0400 (para botones principales y elementos destacados)
* Blanco: #ffffff (fondo principal para alta legibilidad)
* Gris claro: #b5b5b5 (textos secundarios, bordes y estados inactivos)

**Tipografía:**

* Fuente base: **Roboto** (limpia, legible y moderna)
* Título principal: Roboto 32px, negrita (bold)
* Subtítulo o encabezados intermedios: Roboto 24px, seminegrita
* Texto del cuerpo: Roboto 16px, regular
* Botones: Roboto 18px, seminegrita, texto en mayúsculas

**Botones y componentes comunes:**

* Botones principales: color de fondo #ff0400, texto blanco #ffffff, bordes redondeados
* Botones secundarios: bordes grises #b5b5b5, fondo blanco, texto gris oscuro
* Confirmaciones y mensajes importantes deben mostrarse mediante modales con fondo semitransparente.

**Estructura de pantalla:**

* Encabezado superior fijo con logo de FERKONSA S.A. y nombre de usuario autenticado.
* Menú lateral para el Administrador y Chofer, con navegación jerárquica por áreas.
* Área de contenido con tarjetas o tablas según la función.
* Pie de página con información legal y versión del sistema.

## Interfaces de hardware

El sistema está diseñado para ser accesible y funcional desde diversos dispositivos, asegurando compatibilidad y disponibilidad en múltiples entornos tecnológicos. A continuación, se describen los dispositivos soportados, los protocolos de comunicación utilizados, y la interacción entre el software y el hardware:

**Dispositivos soportados**

* **Computadoras de escritorio y laptops**
  + Acceso completo al sistema mediante navegadores modernos (Chrome, Firefox, Edge).
  + Usado principalmente por el **Administrador del Sistema** para la gestión total de pedidos, usuarios, camiones y reportes financieros.
* **Dispositivos móviles (smartphones y tablets)**
  + Soporte para Android e iOS.
  + Interfaz adaptada (responsive) para facilitar el uso por parte de **choferes**, quienes actualizarán el estado de las entregas.
  + Los **clientes** podrán consultar el estado de su pedido desde su navegador móvil ingresando un código de envío proporcionado vía WhatsApp.
* **Chatbot integrado (WhatsApp)**
  + El **cliente** interactúa con el sistema a través de WhatsApp para:
    - Registrar un pedido
    - Consultar estado de pedido

**Protocolos de comunicación soportados**

* **HTTP/HTTPS**: Para la comunicación segura entre el cliente (navegador o móvil) y el servidor web del sistema.
* **WebSockets**: Para actualizaciones el estado de pedidos en el panel del administrador.
* **REST API**: Para la integración del sistema con el chatbot y el sistema actual de inventario y facturación.
* **WhatsApp Business API** o integración con servicios de terceros (como Twilio o Zenvia) para el chatbot conversacional.
* **SMTP/IMAP/POP3**: Para el envío y recepción de correos electrónicos relacionados con validaciones de usuarios, recuperación de contraseñas y notificaciones del sistema.

## Interfaces de software

El sistema automatizado de gestión de pedidos y entregas estará diseñado para integrarse de forma efectiva con los componentes de software ya existentes en FERKONSA

**Sistema operativo**

* **Recomendado:** Windows Server.
* **Compatibilidad:** El sistema automatizado será compatible con el entorno actual de servidores Windows utilizados por FERKONSA.
* **Interacción:**
  + Se alojará tanto la lógica del sistema como las APIs necesarias.
  + Permite mantener uniformidad y facilidad de administración.

**Plataforma de chatbot (WhatsApp Business API)**

* **Proveedor sugerido:** Twilio, Zenvia, Meta (API oficial).
* **Interacción:**
  + El cliente realiza el pedido a través de WhatsApp.
  + El chatbot captura los datos del pedido y los registra en el sistema.
  + El sistema responde con confirmación y actualizaciones.

**Navegadores web compatibles**

* **Para usuarios administrativos y choferes:**
  + Acceso mediante navegador a una interfaz web.
  + Compatible con: Google Chrome, Microsoft Edge, Firefox.
  + Diseño responsivo para que pueda usarse en PC, tablet o smartphone.

**Librerías y tecnologías**

* **Frontend:**
  + Framework: React.js
  + Estilos: Tailwind CSS
* **Backend:**
  + Lenguaje: Node.js,
  + API RESTful para comunicación entre frontend, chatbot
* **Correo electrónico:**
  + Uso de nodemailer o integración SMTP para:
    - Validación de usuarios.
    - Recuperación de contraseñas.
    - Notificaciones del sistema.

1. **Base de datos**

* **Motor:** Posgresql
* **Tecnologías relacionadas:**
  + Soporte para conexión segura, autenticación y transacciones.

1. **Frontend (Interfaz Web - Panel Administrativo y Choferes)**

* **Framework:** React.
* **Librerías adicionales:**
  + **Tailwind CSS**: Para diseño responsivo y rápido.
  + **React Router**: Para navegación interna.
  + **Zustand o Redux Toolkit**: Para manejo de estado.
  + **Axios**: Para llamadas a API.
  + **React Hook Form + Yup**: Validaciones de formularios.
  + **Lucide o Heroicons**: Para íconos modernos.
* **Distribución y despliegue:** Vite o Next.js si se Requieren funcionalidades SSR (renderizado del lado del servidor)

1. **Backend (APIs y lógica de negocio)**

* **Framework recomendado:** Node.js + Express (con estructura en capas).
* **Base tecnológica:**
  + **Lenguaje:** JavaScript o TypeScript (sugerido por su escalabilidad y mantenibilidad).
  + **Autenticación:** JSON Web Tokens (JWT) para sesiones seguras.
  + **Servicios RESTful:** Para exponer endpoints consumidos por frontend y chatbot.

1. **Plataforma de chatbot**

* **Canal:** WhatsApp Business (API Oficial).
* **Proveedor sugerido:** Twilio, Zenvia o Meta API directamente.
* **Interacción:**
  + Recibe pedidos desde WhatsApp.
  + Retorna confirmaciones, estado de envío y códigos de seguimiento.
* **Lógica:** El backend creado en Node.js expondrá endpoints específicos para procesar mensajes entrantes y devolver respuestas automáticas.
* **El cliente solo usa WhatsApp para pedidos**, sin autenticación en la web.

1. **Compatibilidad con sistemas operativos y dispositivos**

* **Sistema operativo servidor:** Windows Server
* **Navegadores compatibles:** Chrome, Edge, Firefox.
* **Dispositivos compatibles:** PC, tablets, smartphones.
* **Interfaz responsive:** Para que choferes y clientes puedan usar desde móvil.

1. **Servicios y librerías adicionales**

* **Correo electrónico:**
  + SMTP + nodemailer para validaciones y recuperación de contraseñas.
* **Control de versiones:**
  + GitHub / GitLab.
* **Seguridad:**
  + Helmet (cabezeras seguras).
  + CORS controlado.
  + Sanitización de inputs.

## Interfaces de comunicación

El sistema automatizado de gestión de pedidos y entregas Requiere múltiples interfaces de comunicación para garantizar la conectividad, la seguridad de los datos transmitidos, y la interoperabilidad con usuarios, servicios de terceros, y otros componentes del sistema. A continuación, se describen los principales elementos de comunicación implicados:

**1. Protocolo de red**

* **HTTP/HTTPS**: Se utilizará **HTTPS (HTTP seguro)** para todas las comunicaciones entre cliente y servidor, protegiendo la integridad y confidencialidad de los datos mediante **TLS 1.3**.
* **Puerto estándar:** 443 (HTTPS).
* Todas las peticiones RESTful del sistema se harán sobre HTTPS.

**2. Correo electrónico (Email)**

* **Uso:**
  + Validación y activación de cuentas nuevas por parte del administrador.
  + Recuperación de contraseña.
* **Formato del mensaje:**
  + Plantillas dinámicas generadas en backend.
* **Protocolo utilizado:**
  + **SMTP** (Simple Mail Transfer Protocol) mediante nodemailer u otra librería.
  + Compatible con proveedores como Gmail SMTP, Outlook 365 o servicio empresarial propio.

**3. Navegadores web**

* **Navegadores compatibles:**
  + Google Chrome, Mozilla Firefox, Microsoft Edge.
  + Safari en iOS y Chrome en Android para dispositivos móviles.
* **Requisitos:**
  + Soporte para ES6+ y APIs modernas de JavaScript.
  + Visualización responsive y adaptativa mediante Tailwind CSS.

**4. Chatbot – WhatsApp Business API**

* **Canal de entrada para pedidos:** WhatsApp a través de la **API Oficial de WhatsApp Business**.
* **Formato de mensajería:**
  + Mensajes estructurados en texto plano, botones interactivos (si se usan proveedores como Twilio o Zenvia).
  + Envío de confirmaciones automáticas con código de seguimiento.
* **Requisitos de seguridad:**
  + Validación del número y origen de mensajes.
  + Verificación del código de pedido para acceso seguro a información del envío.

**5. Formularios electrónicos**

* **Uso:** Registro de usuarios, creación y edición de pedidos, asignación de camiones y choferes, recuperación de contraseña, generación de reportes.
* **Validaciones:**
  + Validación en frontend con React Hook Form y en backend mediante Express Validator.
  + Protección contra entradas maliciosas con sanitización de datos.

**6. Seguridad en la comunicación**

* **Encriptación:**
  + Toda la comunicación se cifra mediante TLS 1.3.
  + Contraseñas almacenadas con algoritmo de hashing **bcrypt** (salteado y seguro).
* **Autenticación y autorización:**
  + **JWT (JSON Web Tokens)** para acceso de usuarios autenticados.
  + Los tokens se renuevan periódicamente y expiran por seguridad.
* **Política CORS:**
  + Acceso controlado a dominios autorizados únicamente.

**7. Estándares adicionales**

* **JSON:** Formato estándar de intercambio de datos entre frontend, backend y servicios externos.
* **RESTful APIs:** Estructura de servicios web modular, interoperable y fácilmente mantenible.

# Requerimientos no funcionales

**RNF-01. Disponibilidad del sistema**

El sistema web deberá estar disponible al menos el **99% del tiempo**, especialmente en horario laboral de la ferretería, para permitir a los administradores y choferes gestionar pedidos sin interrupciones.

**RNF-02. Escalabilidad**

La aplicación debe ser **escalable** para soportar un aumento de usuarios (administradores y choferes) y pedidos sin degradar el rendimiento. Debe poder ampliarse fácilmente en infraestructura (servidores, base de datos) si en el futuro se incrementa la flota o el volumen de ventas.

**RNF-03. Rendimiento y tiempos de respuesta**

* Las acciones del administrador (como asignar un camión o actualizar estado de un pedido) deben ejecutarse en **menos de 2 segundos**.
* El chatbot debe responder a las interacciones del cliente en **menos de 3 segundos** para asegurar una buena experiencia.

**RNF-04. Seguridad**

* El sistema debe contar con inicio de sesión seguro, utilizando contraseñas encriptadas.
* La validación de nuevos usuarios (choferes y administradores) será obligatoria antes de conceder acceso.
* Solo personal autorizado puede realizar acciones críticas como asignar camiones, modificar pedidos o gestionar usuarios.

**RNF-05. Integridad de datos**

Todos los datos (pedidos, inventario migrado, asignaciones de camiones, estados de entrega) deben mantenerse consistentes y protegidos contra pérdidas o corrupción, incluso en caso de caídas del sistema.

**RNF-06. Compatibilidad e integración**

El sistema deberá ser capaz de integrarse vía API con el sistema actual de inventario y facturación, asegurando la migración correcta de datos y actualización en tiempo real de productos disponibles.

**RNF-07. Usabilidad**

La interfaz debe ser intuitiva para los administradores y choferes, con flujos de trabajo claros para asignación de pedidos, visualización de estados y actualización de entregas, dado que los usuarios pueden no ser expertos en sistemas digitales.

**RNF-08. Mantenibilidad**

La aplicación debe estar diseñada de forma modular para facilitar futuras actualizaciones, correcciones de errores o incorporación de nuevas funcionalidades (por ejemplo, geolocalización de camiones o automatización de cobros de envío).

**RNF-09. Accesibilidad**

La aplicación web debe ser accesible desde dispositivos de escritorio y móviles, permitiendo a choferes actualizar estados de entrega desde sus teléfonos.

**RNF-10. Respaldo de la información**

Se deben realizar copias de seguridad diarias de las bases de datos para prevenir pérdidas de información crítica.

# Otros Requerimientos

**OR-02. Requerimientos de Integración**

El sistema debe exponer y consumir servicios **API RESTful** para sincronizar el inventario del sistema actual de la ferretería.

El sistema debe tener endpoints seguros protegidos mediante **tokens de autenticación** (por ejemplo, OAuth2 o JWT).

**OR-03. Internacionalización (I18n)**

Aunque inicialmente el sistema estará en español, la estructura de la aplicación debe permitir una futura **traducción a otros idiomas** si fuera necesario (por ejemplo, configurando textos y mensajes en archivos de idioma separados).

# Glosario

* **RF:** Requerimientos Funcionales
* **RN:** Reglas de Negocio
* **RNF:** Requerimientos No Funcionales
* **RO:** Otros Requerimientos