

# Documento de Requerimientos de software

*Diseño e implementación de un sistema automatizado para la gestión de pedidos y entregas mediante un chatbot con machine learning para la empresa FERKONSA S.A.*

***Fecha: 16/04/2025***

## Contenido

Historial de Versiones .....	4
Información del Proyecto.....	4
Aprobaciones .....	4
1. Propósito .....	5
2. Alcance del producto / Software .....	5
3. Referencias .....	6
4. Funcionalidades del producto.....	7
5. Clases y características de usuarios .....	7
6. Entorno operativo .....	8
7. Requerimientos funcionales .....	8
7.1 Inicio de sesión y autenticación de usuarios .....	8
7.2 Registro de pedidos vía chatbot de WhatsApp .....	9
7.3 Asignación de camiones y choferes.....	10
7.4 Gestión de pedidos .....	10
7.5 Historial de pedidos.....	11
7.6 Consulta del estado del pedido .....	11
7.7 Visualización de pedidos.....	12
7.8 Actualización de estado de envío .....	12
7.9 Historial de entregas .....	13
7.10 Reporte financiero por envío .....	13
7.11 Gestión de usuarios y camiones .....	13
8. Reglas de negocio .....	14

9.	Requerimientos de interfaces externas .....	16
9.1.	Interfaces de usuario.....	16
9.2.	Interfaces de hardware .....	19
9.3.	Interfaces de software.....	20
9.4.	Interfaces de comunicación .....	25
10.	Requerimientos no funcionales .....	27
11.	Otros Requerimientos.....	29
12.	Glosario .....	30

## Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
16/04/2025	1	Franklin Satuquinga, Alejandro Yaucen	FERKONSA S.A.	

## Información del Proyecto

Empresa / Organización	FERKONSA S.A.
Proyecto	Diseño e implementación de un sistema automatizado para la gestión de pedidos y entregas mediante un chatbot con machine learning para 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

## 1. Propósito

**Diseño e implementación de un sistema automatizado para la gestión de pedidos y entregas mediante un chatbot con machine learning para la empresa Ferkonsa S.A.**

El propósito de este proyecto es desarrollar un sistema automatizado para la gestión de pedidos y entregas de materiales de construcción en **FERKONSA S.A.**, optimizando los tiempos de entrega y mejorando la comunicación con los clientes. El software contará con un chatbot integrado con WhatsApp para registrar pedidos, verificar la disponibilidad de camiones para el traslado de materiales, y proporcionará una interfaz web donde los clientes podrán visualizar el estado de sus envíos en tiempo real. Además, el sistema permitirá la asignación eficiente de recursos logísticos, optimizando la disponibilidad de camiones y reduciendo el tiempo de entrega. La integración con el sistema actual de inventarios y facturación de Ferkonsa S.A. garantizará la sincronización de datos, lo que permitirá una mayor precisión y eficiencia en la gestión de pedidos.

## 2. Alcance del producto / Software

El sistema **FERKONSA System** permitirá a **FERKONSA S.A.** gestionar pedidos y entregas de materiales de construcción, mejorando la eficiencia y optimizando los tiempos de respuesta al cliente. La aplicación web ofrecerá una gestión automatizada del registro de pedidos a través de un chatbot de WhatsApp, permitirá verificar la disponibilidad de camiones y mostrará a los clientes el estado de los envíos en tiempo real.

### **Objetivo General:**

Desarrollar una plataforma que automatice el proceso de registro de pedidos, verifique la disponibilidad de camiones, y permita el seguimiento de las entregas en tiempo real. Además, integrará el sistema con el actual gestor de inventario y facturación de Ferkonsa S.A., garantizando un flujo de trabajo eficiente.

### **Beneficios:**

- Mejora en la eficiencia de la gestión de pedidos y entregas.
- Optimización de tiempos de respuesta al cliente.
- Automatización del registro de pedidos mediante un chatbot de WhatsApp.
- Mayor transparencia y control sobre el estado de los envíos.
- Reducción de errores manuales en la gestión logística.

### **Objetivos y Metas:**

- Integrar un chatbot para el registro automático de pedidos.
- Desarrollar una interfaz web para que los clientes consulten el estado de sus envíos.
- Verificar la disponibilidad de camiones antes de confirmar el envío.

- Facilitar a los administradores el monitoreo y la gestión de los pedidos en tiempo real.
- Alinear el sistema con los objetivos estratégicos de la empresa, como la satisfacción del cliente y la optimización de recursos logísticos.

### 3. 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 Vizuite Albán

Fecha: 2020

Ubicación: Ecuador

[2] E. R. Vizuite 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: Rodríguez Valdivia, Maria Jessica; Wong Herrera, Josiel Yosiana

Fecha: 2021

Ubicación: Perú

[3] M. J. Rodríguez 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>

## 4. Funcionalidades del producto

El software **FERKONSA Delivery System** incluirá las siguientes funcionalidades principales:

1. Inicio de sesión y autenticación
2. Registro de pedidos vía chatbot de WhatsApp
3. Asignación de camiones y choferes
4. Gestión de pedidos
5. Historial de pedidos
6. Consulta del estado del pedido
7. Visualización de pedidos
8. Actualización de estado de envío
9. Historial de entregas
10. Reporte financiero por envío
11. Gestión de usuarios y camiones

## 5. 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 Final

**Descripción:** Usuarios que realizan pedidos de materiales de construcción. Interactúan principalmente con el sistema a través del chatbot de WhatsApp y la interfaz web para consultar el estado de sus envíos.

**Frecuencia de uso:** Baja a moderada. Generalmente, los clientes interactúan al realizar pedidos y consultar el estado de las entregas.

**Funcionalidades relevantes:** Registro de pedidos, consulta de estado de envío.

**Privilegios de seguridad:** Acceso solo a su historial de pedidos y detalles de envío.

### 2. Administrador Sistema

**Descripción:** Usuario encargado de gestionar todo el sistema: pedidos, usuarios, camiones y reportes.

**Frecuencia de uso:** Alta. Interactúan frecuentemente con el sistema para gestionar y monitorear pedidos.

**Funcionalidades relevantes:** Autenticación y validación de registros, Asignación de recursos logísticos (choferes y camiones), Gestión y supervisión de pedidos, Visualización de historiales, Generación de reportes financieros, Gestión de usuarios y camiones.

**Privilegios de seguridad:** Acceso completo al sistema, modificación de datos, validación de registros, exportación de información.

### 3. Chofer

**Descripción:** Usuarios encargados de realizar las entregas de los pedidos. Interactúan con el sistema principalmente para recibir información sobre los pedidos y las rutas de entrega.

**Frecuencia de uso:** Alta. Los choferes interactúan frecuentemente con el sistema para recibir detalles de los pedidos y confirmaciones de rutas.

**Funcionalidades relevantes:** Visualización de pedidos asignados, seguimiento de rutas de entrega, actualización del estado de la entrega.

**Privilegios de seguridad:** Acceso solo a los pedidos asignados y la información necesaria para realizar las entregas.

## 6. Entorno operativo

Tecnologías

- **Frontend:** React.js + Material-UI.
- **Backend:** Node.js + Express.js.
- **Base de datos:** Oracle Database.
- **Autenticación:** JWT con Express.js.
- **Alojamiento:** AWS o Heroku.

## 7. Requerimientos funcionales

### 7.1 Inicio de sesión y autenticación de usuarios

**Descripción:** Permite a los diferentes usuarios del sistema iniciar sesión de forma segura según sus roles (cliente, administrador, chofer, administrador del sistema). También incluye el proceso de registro de nuevos usuarios, validación por parte del administrador y recuperación de contraseña.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

- El usuario accede a la pantalla de inicio de sesión.
- Ingresa su correo y contraseña.
- El sistema valida las credenciales.
- Si son correctas, redirige al usuario a su panel correspondiente según su rol.
- Si son incorrectas, muestra un mensaje de error.
- Si el usuario no tiene cuenta, puede registrarse ingresando: nombre, apellido, correo electrónico, contraseña y confirmación de contraseña.
- Al registrarse, el sistema envía una notificación al correo del administrador para su validación.
- El administrador aprueba o rechaza el nuevo registro desde su panel.



- Si el usuario olvida su contraseña, puede recuperarla mediante su correo electrónico.

### **Requerimientos funcionales:**

- **RF-01:** El sistema debe permitir el inicio de sesión mediante correo electrónico y contraseña.
- **RF-02:** El sistema debe validar las credenciales en una base de datos segura.
- **RF-03:** El sistema debe utilizar JWT para generar y validar tokens de sesión.
- **RF-04:** El sistema debe redirigir al usuario al módulo correspondiente según su rol.
- **RF-05:** El sistema debe mostrar un mensaje de error si las credenciales son incorrectas.
- **RF-06:** El sistema debe permitir el registro de usuarios ingresando nombre, apellido, correo electrónico, contraseña y confirmación de contraseña.
- **RF-07:** El sistema debe validar que la contraseña cumpla con criterios de seguridad (mínimo 8 caracteres, una mayúscula, una minúscula y un número).
- **RF-08:** El sistema debe verificar que la confirmación de contraseña coincida.
- **RF-09:** El sistema debe enviar una notificación al administrador para validar nuevos registros.
- **RF-10:** El administrador debe poder aprobar o rechazar nuevos usuarios desde el panel de gestión.
- **RF-11:** El sistema debe notificar por correo al usuario una vez aprobado su registro.
- **RF-12:** El sistema debe permitir la recuperación de contraseña mediante un enlace enviado al correo electrónico del usuario.
- **RF-13:** El enlace de recuperación debe tener una validez limitada por razones de seguridad.

## **7.2 Registro de pedidos vía chatbot de WhatsApp**

**Descripción:** Permite a los clientes realizar pedidos desde WhatsApp utilizando un chatbot automatizado.

**Prioridad:** Alta

### **Acciones iniciadoras y comportamiento esperado:**

- El cliente escribe al chatbot desde WhatsApp.
- El chatbot guía al usuario para seleccionar productos, cantidades y dirección de entrega.
- Se confirma el pedido y se registra en el sistema.

### Requerimientos funcionales:

- **RF-14:** El chatbot debe permitir al usuario registrar un pedido mediante lenguaje natural.
- **RF-15:** El chatbot debe solicitar los siguientes datos para el pedido: producto, cantidad y dirección de entrega.
- **RF-16:** El chatbot debe validar que el producto existe y está disponible en inventario.
- **RF-17:** El chatbot debe registrar el pedido en el sistema de gestión de pedidos.
- **RF-18:** El chatbot debe confirmar al usuario que el pedido ha sido recibido correctamente.
- **RF-19:** El chatbot debe poder entender sinónimos o frases comunes para hacer pedidos (por ejemplo: “necesito”, “quiero comprar”, “hazme un pedido”).

### 7.3 Asignación de camiones y choferes

**Descripción:** Permite al administrador asignar camiones y choferes disponibles a cada pedido registrado.

**Prioridad:** Alta

#### Acciones iniciadoras y comportamiento esperado:

El administrador visualiza los pedidos pendientes.

Revisa la disponibilidad de camiones y choferes.

Asigna recursos disponibles al pedido.

#### Requerimientos funcionales:

- **RF-20:** El sistema debe mostrar la lista de camiones y choferes disponibles.
- **RF-21:** El sistema debe permitir asignar un chofer y camión por pedido.
- **RF-22:** El sistema debe bloquear camiones y choferes ya asignados a otros pedidos.

### 7.4 Gestión de pedidos

**Descripción:** Permite a los administradores visualizar, modificar y gestionar los pedidos activos.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

- El administrador accede al panel de pedidos.
- Puede editar datos del pedido, cambiar asignaciones o cancelar si es necesario.

**Requerimientos funcionales:**

- **RF-22:** El sistema debe permitir modificar pedidos antes de ser despachados.
- **RF-23:** El sistema debe permitir cancelar pedidos con justificación.
- **RF-24:** El sistema debe registrar el historial de cambios por pedido.

## 7.5 Historial de pedidos

**Descripción:** Permite a los administradores consultar pedidos anteriores.

**Prioridad:** Media

**Acciones iniciadoras y comportamiento esperado:**

- El administrador accede a la opción "Historial".
- El sistema muestra una lista con los pedidos anteriores y su estado.

**Requerimientos funcionales:**

- **RF-25:** El sistema debe permitir consultar el historial filtrado por fechas.
- **RF-26:** El sistema debe mostrar los detalles completos de cada pedido.

## 7.6 Consulta del estado del pedido

**Descripción:** Permite al cliente verificar el estado actual de su pedido.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

- El cliente accede a la opción de "Estado de pedido" en web.
- El sistema muestra el estado actualizado.

**Requerimientos funcionales:**

- **RF-27:** El sistema debe mostrar el estado del pedido: pendiente, en camino, entregado, no entregado.
- **RF-28:** El sistema debe actualizar este estado conforme se procesa el pedido.

**7.7 Visualización de pedidos**

**Descripción:** Permite a choferes y administradores visualizar los pedidos asignados.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

- El usuario accede al módulo de pedidos.
- Visualiza la lista y detalles de los pedidos.

**Requerimientos funcionales:**

- **RF-29:** El sistema debe mostrar todos los pedidos asignados al chofer logueado.
- **RF-30:** El sistema debe permitir ver detalles completos del pedido (cliente, dirección, productos).

**7.8 Actualización de estado de envío**

**Actualización de estado de envío**

**Descripción:** Permite a los choferes actualizar el estado del pedido conforme avanzan con la entrega.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

- El chofer inicia la entrega y actualiza el estado: pendiente, en camino, entregado, no entregado.

**Requerimientos funcionales:**

- **RF-31:** El sistema debe permitir marcar el estado de un pedido.
- **RF-32:** El sistema debe permitir registrar la entrega con observaciones.

### 7.9 Historial de entregas

**Descripción:** Permite a los choferes y administradores revisar entregas realizadas.

**Prioridad:** Media

**Acciones iniciadoras y comportamiento esperado:**

- El usuario accede al módulo de historial de entregas.
- El sistema muestra una lista con entregas previas.

**Requerimientos funcionales:**

- **RF-33:** El sistema debe mostrar entregas anteriores con filtros por fecha o estado.
- **RF-34:** El sistema debe mostrar nombre del cliente, productos, chofer y fecha de entrega.

### 7.10 Reporte financiero por envío

**Descripción:** Permite al administrador generar reportes financieros vinculados a cada pedido (costo, ingresos, envío).

**Prioridad:** Media

**Acciones iniciadoras y comportamiento esperado:**

- El administrador selecciona un pedido.
- El sistema genera un reporte detallado del mismo.

**Requerimientos funcionales:**

- **RF-35:** El sistema debe calcular y mostrar el total del pedido incluyendo costos de envío.
- **RF-36:** El sistema debe permitir exportar el reporte en PDF.

### 7.11 Gestión de usuarios y camiones

**Descripción:** Permite al Administrador del Sistema gestionar usuarios y camiones en el sistema.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

- El administrador accede al módulo de gestión.
- Puede crear, editar, desactivar o eliminar usuarios y camiones.
- Al registrar un nuevo usuario, se notifica al administrador general para su aprobación.

**Requerimientos funcionales:**

- **RF-37:** El sistema debe permitir registrar nuevos usuarios ingresando nombre, apellido, correo electrónico, contraseña, confirmación de contraseña y rol.
- **RF-38:** El sistema debe enviar una notificación al administrador para validar el nuevo registro.
- **RF-39:** El sistema debe permitir al administrador aprobar o rechazar nuevos usuarios desde el panel de gestión.
- **RF-40:** El sistema debe permitir agregar camiones con características como capacidad y número de placa.
- **RF-41:** El sistema debe validar datos duplicados o incompletos para usuarios y camiones.
- **RF-42:** El sistema debe permitir desactivar cuentas de usuario o camiones fuera de uso sin eliminarlos permanentemente.

## 8. Reglas de negocio

**RN-01: Roles de usuario y permisos**

- El sistema distingue al menos cuatro roles principales: Cliente, Chofer y Administrador del Sistema.
- Cada rol solo podrá acceder a las funcionalidades autorizadas según su perfil:

- **Cliente:** Puede registrar pedidos y consultar estados
- **Chofer:** Puede ver pedidos asignados, actualizar estado de entrega y consultar historial de entregas.
- **Administrador:** Puede asignar camiones y choferes, gestionar pedidos, generar reportes financieros y gestionar usuarios y camiones.

### **RN-02: Validación y aprobación de registros de usuario**

- Todo nuevo usuario registrado debe ser validado por el Administrador del Sistema a través de un correo de notificación antes de que pueda iniciar sesión.

### **RN-03: Recuperación de contraseña**

- Si un usuario olvida su contraseña, podrá recuperarla mediante una solicitud con su correo electrónico registrado. Se enviará un enlace temporal para restablecer la contraseña.

### **RN-04: Disponibilidad de recursos**

- Un camión y chofer solo pueden ser asignados a un pedido a la vez, y no pueden ser seleccionados si ya están ocupados con otra entrega en progreso.

### **RN-05: Control de estados del pedido**

- Los pedidos deben seguir una secuencia de estados válida: pendiente → en camino → entregado o no entregado.
- No se puede cambiar el estado de un pedido a "entregado" si no ha pasado previamente por los estados intermedios.

### **RN-06: Estimación de entrega**

- Toda estimación de tiempo de entrega debe considerar la distancia, la disponibilidad de camiones y pedidos pendientes.

### **RN-07: Integridad de los datos**

- Todos los formularios deben validar campos obligatorios y verificar que los datos ingresados no estén duplicados (ej. correo, número de placa de camión).
- No se permitirá guardar registros incompletos o inválidos.

## 9. Requerimientos de interfaces externas

### 9.1. Interfaces de usuario

El sistema contará con interfaces gráficas diferenciadas para cada tipo de usuario: **Cliente, 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 y Portal de Consulta)

- **Área: Chatbot inteligente de pedidos**
  - Medio principal de interacción del cliente.
  - Permite realizar pedidos mediante lenguaje natural.
  - Informa el código de seguimiento.
  - Detecta productos disponibles según el inventario.
- **Área: Consulta de estado de pedido**
  - No requiere inicio de sesión.
  - El cliente accede al estado del pedido ingresando un código único de envío.
  - 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**
  - Validación de nuevos registros (excepto clientes que no requieren login).
  - Edición y eliminación de usuarios choferes o administrativos.
- **Área: Reportes**

- Generación de reportes financieros y logísticos.
- Filtros por fecha, chofer, estado del pedido o ciudad.

### 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.

#### Migración de datos desde el sistema actual

- Se desarrollará una interfaz de importación que asegure:

- Migración de datos históricos de pedidos, camiones y usuarios (excepto clientes).
- Validación de integridad antes de cargar los datos al nuevo sistema.
- Reporte de inconsistencias o datos omitidos.
- Respaldo de datos antiguos antes de la migración definitiva.

### 9.2. 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
    - Recibir el código de seguimiento

#### 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.

### Interacciones entre el software y el hardware

- **Base de datos migrada:** El sistema migrará los datos del sistema actual de la ferretería, lo cual incluye información de productos, stock, clientes frecuentes, y vehículos existentes. Este proceso Requiere acceso temporal al hardware actual del servidor para extracción y transformación de datos.

### 9.3. 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 S.A., especialmente con el ERP actual y su base de datos. A continuación, se describen las principales interfaces de software:

#### Sistema ERP actual: Silver ERP de S-Innovatec

- **Tipo:** Sistema ERP empresarial en uso actualmente.
- **Interacción:**
  - El nuevo sistema se conectará directamente con el ERP para:
    - Consultar disponibilidad de productos y stock.
    - Sincronizar datos de clientes y productos.
    - Enviar los pedidos generados por chatbot o el sistema hacia el ERP.
  - La integración puede realizarse mediante:
    - **Web services o API del ERP** (si están disponibles).
    - Acceso a **vistas o procedimientos almacenados** en la misma base de datos Oracle.

#### Base de datos

- **Motor:** Oracle Database.
- **Interacción:**

- El sistema automatizado compartirá el **mismo motor de base de datos Oracle**, ya sea mediante un esquema propio o compartido con el ERP.
- Esto facilita la interoperabilidad directa y evita procesos de migración o duplicación de datos.
- Se establecerán **políticas de integridad y control de acceso** para mantener la seguridad de la información.
- Las operaciones de lectura y escritura se harán mediante **procedimientos almacenados y consultas optimizadas**.

### 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, asociándolos con un código de seguimiento.
  - 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 o Bootstrap.
- **Backend:**
  - Lenguaje: Node.js, Laravel o Java.
  - Conectividad con Oracle mediante drivers compatibles (por ejemplo, oracledb en Node.js o oci8 en PHP).
  - API RESTful para comunicación entre frontend, chatbot y sistema ERP.
- **Correo electrónico:**
  - Uso de nodemailer o integración SMTP para:
    - Validación de usuarios.
    - Recuperación de contraseñas.
    - Notificaciones del sistema.

### 1. Integración con ERP: Silver ERP (S-Innovatec)

- **Tipo:** ERP existente de la empresa.
- **Interacción:**
  - Consultas de inventario y stock disponibles.
  - Lectura y escritura de pedidos.
  - Sincronización de datos de clientes, productos y facturación.
- **Método de integración:**
  - Si el ERP expone APIs, se usarán para integración directa.

- Si no las expone, se accederá mediante vistas SQL o procedimientos almacenados compartidos en la base Oracle.

### 2. Base de datos

- **Motor:** Oracle Database.
- **Interacción:**
  - Comunicación directa mediante controladores Oracle actualizados.
  - Consultas optimizadas, uso de procedimientos almacenados y triggers donde sea necesario.
- **Tecnologías relacionadas:**
  - Soporte para conexión segura, autenticación y transacciones.

### 3. 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)

### 4. 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.

### 5. 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.

### 6. 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.

### 7. 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.

### 9.4. 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.

## 10. Requerimientos no funcionales

### 10.1. Seguridad

- **RNF 1:** El sistema debe cifrar todas las comunicaciones usando el protocolo **HTTPS con TLS 1.3**.
- **RNF 2:** Los usuarios deben autenticarse mediante credenciales seguras. Las contraseñas serán almacenadas con **bcrypt**.
- **RNF 3:** Se utilizarán **tokens JWT** para manejar sesiones autenticadas de forma segura.
- **RNF 4:** Los datos sensibles estarán protegidos contra ataques comunes como **XSS, CSRF, e inyección SQL**.
- **RNF 5:** Se debe realizar una validación exhaustiva en frontend y backend.
- **RNF 6:** Solo usuarios validados por el Administrador pueden acceder al sistema.

### 10.2. Rendimiento

- **RNF 7:** El sistema debe soportar al menos **100 usuarios concurrentes** sin degradación perceptible del servicio.
- **RNF 8:** Las respuestas del servidor a peticiones comunes no deben superar los **2 segundos**.
- **RNF 9:** El sistema debe ser capaz de manejar al menos **100 pedidos diarios**.

### 10.3. Usabilidad

- **RNF 10:** La interfaz debe ser **intuitiva y fácil de usar**, siguiendo principios de usabilidad para sistemas web.
- **RNF 11:** El sistema será **responsive**, accesible desde computadoras de escritorio, laptops, tablets y smartphones.
- **RNF 12:** El cliente no Requiere autenticación para consultar el estado de su pedido: solo deberá ingresar el **código de envío proporcionado** vía WhatsApp.

### 10.4. Mantenibilidad

- **RNF 13:** El código debe estar **modularizado y documentado** para facilitar futuras modificaciones.
- **RNF 14:** Debe ser posible actualizar componentes del sistema (librerías, dependencias, etc.) sin afectar la funcionalidad principal.
- **RNF 15:** Se utilizarán prácticas de desarrollo como **control de versiones (Git)** y pruebas automatizadas.

### 10.5. Portabilidad

- **RNF 16:** El sistema web será accesible desde **todos los navegadores modernos** y dispositivos móviles.
- **RNF 17:** La aplicación debe ejecutarse correctamente en servidores con **Windows Server**, tal como lo hace el sistema actual.
- **RNF 18:** El backend y frontend deben poder ser desplegados fácilmente en **ambientes locales y en la nube** (Ej. Azure, AWS, o servidores propios).

### 10.6. Compatibilidad

- **RNF 19:** El sistema debe integrarse con el actual ERP de la empresa **Silver ERP** (de S-Innovatec) mediante una interfaz de interoperabilidad basada en **servicios REST** o **exportación/importación de datos estructurados** (CSV/JSON/XML).
- **RNF 20:** La base de datos se mantendrá en **Oracle**, asegurando compatibilidad con los datos existentes.
- **RNF 21:** Compatible con **WhatsApp Business API** para pedidos mediante chatbot.

### 10.7. Disponibilidad

- **RNF 22:** El sistema debe tener una **disponibilidad mínima del 99.5%** mensual.
- **RNF 23:** Las tareas de mantenimiento se realizarán fuera del horario laboral para evitar interrupciones.

### 10.8. Escalabilidad

- **RNF 24:** La arquitectura debe permitir escalar horizontal o verticalmente tanto el frontend como el backend.
- **RNF 25:** Se debe prever la posibilidad de implementar **microservicios** en el futuro para módulos independientes (como reportes financieros o seguimiento de entregas).

## 11. Otros Requerimientos

### 11.1 Requerimientos de base de datos

- **RO-01:** El sistema debe integrarse con el motor de base de datos **Oracle**, manteniendo la estructura de datos esencial del sistema actual **Silver ERP**.
- **RO-02:** Se debe permitir la **migración de datos históricos** (clientes, productos, camiones, pedidos, entregas) desde el sistema actual al nuevo sistema.
- **RO-03:** Se deben implementar **procedimientos almacenados** y **vistas** para consultas frecuentes, reportes y validaciones.

- **RO-04:** La integridad referencial y las restricciones (ej. claves foráneas, unicidad) deben mantenerse estrictamente.
- **RO-05:** Se deben realizar **copias de seguridad automáticas** diarias con opción de restauración rápida en caso de fallo.

### 11.2. Requerimientos legales y normativos

- **RO-06:** El sistema debe cumplir con las normativas de **protección de datos personales** vigentes en Ecuador (como las establecidas por la **Ley Orgánica de Protección de Datos Personales**).
- **RO-07:** Toda la información recolectada será almacenada de forma segura y solo se usará con fines operativos y logísticos.

### 11.3. Requerimientos de empaquetado y despliegue

- **RO-08:** El sistema debe poder ser **empaquetado como contenedores Docker** para facilitar el despliegue en entornos controlados.

## 12. Glosario

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