

# Workshop Service Management WSM

## Contenido

Introducción .....	3
Alcance .....	3
Funciones.....	3
Gestión de ordenes.....	3
Gestión del inventario .....	3
Gestión de clientes .....	3
Gestión de talleres/sucursales .....	4
Mensajería .....	4
Reporteo .....	4
Cobro, Facturación e Impuestos .....	4
Gestión de empleados / Nomina.....	4
Tecnologías y requerimientos.....	4
Frontend .....	4
Backend.....	4
Base de datos:.....	5
Autenticación: .....	5
CI/CD .....	5
Notificaciones y Emails .....	5
Cloud Platform .....	5
Estructura de la base de datos .....	5
Diagramas de clases .....	5
Diagramas de interacción .....	5

Casos de uso .....	6
Cliente .....	6
Tecnico / Taller .....	6
Administrador Taller .....	6
Super Usuario (Dueño de franquicia) .....	6
Sistema automatizado .....	6
Anexo 1 – Wireframes .....	7
Anexo 2 – Estructura de Base de Datos .....	8
Tablas Principales .....	8
Relaciones Clave .....	10
Optimizaciones .....	10

# Introducción

## Alcance

Esta plataforma busca facilitar las funciones principales de los talleres de servicio con un enfoque en vehículos ligeros como lo son las bicicletas, motocicletas y scooter eléctricos.

Se busca cubrir las necesidades de gestión de los activos, las piezas de recambio, ordenes de servicio e incluso los aspectos internos del taller como pueden ser cobros, administración de personal y nominas.

## Funciones

### Gestión de ordenes

- Recepcion
- Registro multimedia (Fotos, Audio, video)
- Seguimiento
- Notificaciones
- Prioridades

### Gestión del inventario

- Almacen
- Entradas
- Salidas
- Catalogo SKU
- Mínimos

### Gestión de clientes

- Datos personales (Enfoque en privacidad)
- Historial
- Firma digital

## Gestión de talleres/sucursales

- Técnicos
- Checklist (Ingreso, reparaciones, entrega)
- Presupuestos

## Mensajería

### Reporteo

- Ordenes activas
- Ordenes pendientes
- Ordenes finalizadas
- Estado de inventario
- Balance
- SLAs
- Metricos top por modelos, refacciones , fallas, rentabilidad por servicio

## Cobro, Facturación e Impuestos

## Gestión de empleados / Nómina

## Tecnologías y requerimientos

### Frontend:

- Móviles
  - React / Flutter
- Web
  - React.js
- Tablet
  - React Native

### Backend:

- Spring boot (Java)
- Microservicios Docker + Kubernetes
  - Gestion de ordenes
  - Inventario
  - Notificaciones

## Base de datos:

- PostgreSQL
- Alternativa Supabase

## Autenticacion:

- Firebase o Auth0

## CI/CD

- Github Actions

## Notificaciones y Emails

- WebSockets

## Cloud Platform

- Google Cloud Platform GCP or AWS

## Estructura de la base de datos

## Diagramas de clases

## Diagramas de interacción

# Casos de uso

## Cliente

- Registrar una orden de servicio
- Seguir estado de orden
- Consultar historial

## Tecnico / Taller

- Gestionar orden
- Solicitar refacción
- Recoleccion / Entrega

## Administrador Taller

- Supervisar inventario
- Generar reportes
- Asignar recursos

## Super Usuario (Dueño de franquicia)

- Administrar talleres
- Analíticas globales estratégicas

## Sistema automatizado

- Notificaciones
- Sincronizacion
  - Consolidacion de inventario en BD
  - RespalDOS

## Anexo 1 – Wireframes

# Anexo 2 – Estructura de Base de Datos

## Tablas Principales

### 1. usuarios

- id (PK)
- rol (cliente, técnico, admin\_taller, super\_admin)
- nombre, email, teléfono, contraseña\_hash
- taller\_id (FK) → Si es técnico/admin.

### 2. talleres

- id (PK)
- nombre, dirección, teléfono, geo\_coordenadas

### 3. clientes (extiende usuarios)

- usuario\_id (PK, FK a usuarios.id)
- historial\_compras, preferencias\_contacto

### 4. ordenes\_servicio

- id (PK)
- cliente\_id (FK a usuarios.id)
- taller\_id (FK a talleres.id)
- fecha\_creación, estado (recibido/en diagnóstico/terminado/etc.)
- tipo\_servicio (reparación/mantenimiento), prioridad
- descripción\_falla, fotos\_urls (array o tabla separada)

### 5. seguimiento\_ordenes (log de cambios)

- id (PK)
- orden\_id (FK a ordenes\_servicio.id)
- tecnico\_id (FK a usuarios.id)
- estado\_anterior, nuevo\_estado, fecha\_cambio, comentario



## **6. inventario\_refacciones**

- id (PK)
- taller\_id (FK)
- nombre, sku, proveedor, stock\_actual, stock\_minimo
- costo, precio\_venta, ubicacion\_en\_taller

## **7. movimientos\_inventario**

- id (PK)
- refaccion\_id (FK a inventario\_refacciones.id)
- tipo (entrada/salida/ajuste), cantidad
- orden\_id (FK, nullable) → Si se usó en una orden.
- fecha, responsable\_id (FK a usuarios.id)

## **8. checklists\_servicio**

- id (PK)
- orden\_id (FK)
- tecnico\_id (FK)
- items (JSON o tabla separada): [{tarea: "testear motor", completado: bool}]
- firma\_cliente\_url, fecha\_completado

## **9. recolecciones\_entregas**

- id (PK)
- orden\_id (FK)
- tipo (recolección/entrega)
- direccion, geo\_coordenadas, fecha\_hora
- estado (programado/en camino/completado)

## **10. notificaciones**

- id (PK)
- usuario\_id (FK)

- tipo (email/sms/app), contenido, fecha\_envio, leído

## Relaciones Clave

1. ordenes\_servicio → usuarios:
  - Una orden pertenece a un **cliente** y es gestionada por un **taller**.
2. movimientos\_inventario → ordenes\_servicio:
  - Registrar qué refacciones se usaron en cada orden.
3. seguimiento\_ordenes:
  - Tabla de auditoría para rastrear cambios de estado.

## Optimizaciones

- **Índices:**
  - ordenes\_servicio.estado, inventario\_refacciones.sku, usuarios.email.
- **Normalización:**
  - Fotos de órdenes podrían ser una tabla separada ordenes\_fotos si son muchas.
- **Tipo de datos:**
  - geo\_coordenadas: Usar tipo POINT (PostgreSQL) o lat/long separados.
  - fotos\_urls: Array de strings (ej: TEXT[]) o tabla relacionada.