



Lab. Análisis y Diseño de Sistemas 1

Manual tecnico - Taller Mecanico

por

Rudy Alessandro Reyes Oxlaj	202031213
Jose Luis Baltazar Pu Sam	201731370
Eiler Rigoberto Gómez Figueroa	202031693

1 de Septiembre de 2025



MANUAL TÉCNICO - SISTEMA DE TALLER MECÁNICO

Este manual técnico proporciona la información necesaria para comprender, instalar y mantener el sistema de gestión de taller mecánico. El sistema está compuesto por un frontend desarrollado en React y un backend desarrollado en Node.js con Express.

CONTENIDO

1. Introducción
2. Arquitectura general
3. Requerimientos de hardware
4. Requerimientos de software
5. Guía de instalación
6. Estructura del proyecto
7. Documentación de la API
8. Modelo y Base de datos



INTRODUCCIÓN AL SISTEMA DE TALLER MECÁNICO

PROPÓSITO DEL SISTEMA

El Sistema de Taller Mecánico es una aplicación web completa diseñada para gestionar todas las operaciones de un taller automotriz. El sistema permite administrar clientes, vehículos, servicios, trabajos, repuestos, facturación y personal técnico, ofreciendo una solución integral para la gestión eficiente del taller.

FUNCIONALIDADES PRINCIPALES

El sistema incluye las siguientes funcionalidades principales:

1. Gestión de usuarios y roles

- Administradores, empleados, especialistas técnicos, clientes y proveedores
- Control de acceso basado en roles
- Autenticación de doble factor

2. Gestión de vehículos

- Registro y seguimiento de vehículos
- Historial de servicios y mantenimientos por vehículo

3. Gestión de servicios

- Registro de ingreso de vehículos
- Asignación de trabajos a empleados y especialistas
- Seguimiento del estado de los servicios
- Diagnósticos y pruebas técnicas



4. Gestión de inventario

- Control de repuestos y materiales
- Gestión de proveedores
- Solicitudes y aprobación de uso de repuestos

5. Facturación y pagos

- Generación de facturas
- Registro de pagos
- Consulta de saldos

6. Módulo de clientes

- Seguimiento de servicios
- Solicitudes de servicios adicionales
- Cotizaciones
- Calificación de servicios

7. Reportes operativos y financieros

- Reportes de trabajos realizados
- Reportes de ingresos y egresos
- Historial de mantenimientos
- Calificaciones de servicio

VISIÓN TÉCNICA GENERAL

El sistema está desarrollado utilizando una arquitectura cliente-servidor, con las siguientes tecnologías principales:

- **Frontend:** React.js, Bootstrap, Axios
- **Backend:** Node.js, Express
- **Base de datos:** MySQL
- **Autenticación:** JWT (JSON Web Tokens)
- **API:** REST

La aplicación está diseñada siguiendo el patrón MVC (Modelo-Vista-Controlador) para una mejor organización y mantenimiento del código.

PÚBLICO OBJETIVO

Este manual técnico está dirigido a:

- Desarrolladores de software
- Administradores de sistemas
- Personal de soporte técnico
- Cualquier persona encargada de la instalación, configuración o mantenimiento del sistema

El documento asume conocimientos básicos de desarrollo web, bases de datos relacionales y administración de servidores.

REQUERIMIENTOS DEL SISTEMA

REQUERIMIENTOS DE HARDWARE

Servidor (Entorno de producción)

Para un taller mecánico de tamaño pequeño a mediano (5-20 usuarios concurrentes):

- **Procesador:** CPU de 4 núcleos, 2.5 GHz o superior
- **Memoria RAM:** Mínimo 8 GB, recomendado 16 GB
- **Almacenamiento:** Mínimo 50 GB de espacio libre, preferiblemente SSD
- **Conectividad:** Conexión a Internet de al menos 10 Mbps de subida y bajada
- **Respaldo:** Sistema de respaldo para la base de datos

Para un taller mecánico grande (más de 20 usuarios concurrentes):

- **Procesador:** CPU de 8 núcleos, 3.0 GHz o superior
- **Memoria RAM:** Mínimo 16 GB, recomendado 32 GB
- **Almacenamiento:** Mínimo 100 GB de espacio libre, preferiblemente SSD
- **Conectividad:** Conexión a Internet de al menos 20 Mbps de subida y bajada
- **Respaldo:** Sistema de respaldo automático para la base de datos

Estaciones de trabajo (Clientes)

- **Procesador:** CPU de 2 núcleos, 2.0 GHz o superior
- **Memoria RAM:** Mínimo 4 GB
- **Almacenamiento:** 5 GB de espacio libre
- **Pantalla:** Resolución mínima de 1366 x 768 pixeles
- **Conectividad:** Conexión a la red local o Internet

Dispositivos móviles (opcional)

El sistema es accesible desde dispositivos móviles que cumplan con:

- **Sistema operativo:** Android 9.0+ o iOS 13+
- **Navegador:** Chrome, Safari o Firefox actualizado
- **Pantalla:** Mínimo 5 pulgadas con resolución HD
- **Conectividad:** WiFi o datos móviles 4G/5G

REQUERIMIENTOS DE SOFTWARE

Servidor

- **Sistema Operativo:**
 - ◆ Linux (Ubuntu 20.04 LTS o superior, Debian 11 o superior)
 - ◆ Windows Server 2016 o superior
 - ◆ macOS Catalina o superior
- **Entorno de ejecución:**
 - ◆ Node.js versión 16.x o superior
 - ◆ npm versión 8.x o superior
- **Base de Datos:**
 - ◆ MySQL versión 8.0 o superior
 - ◆ MariaDB 10.5 o superior (alternativa)

→ **Servidor web(opcional para producción):**

- ◆ Nginx 1.18 o superior (recomendado)
- ◆ Apache 2.4 o superior

→ **Otras dependencias:**

- ◆ OpenSSL para certificados SSL
- ◆ Git para control de versiones y despliegue

Cliente (Navegadores compatibles)

→ **Desktop:**

- ◆ Google Chrome (versión 90 o superior)
- ◆ Mozilla Firefox (versión 88 o superior)
- ◆ Microsoft Edge (versión 90 o superior)
- ◆ Safari (versión 14 o superior)

→ **Móvil:**

- ◆ Chrome para Android (versión 90 o superior)
- ◆ Safari para iOS (versión 14 o superior)

Entorno de desarrollo

→ **IDE recomendado:**

- ◆ Visual Studio Code con las siguientes extensiones:
- ◆ ESLint



- ◆ Prettier
- ◆ React Developer Tools
- ◆ SQLTools

REST Client

→ Herramientas adicionales:

- ◆ Postman o Insomnia para pruebas de API
- ◆ MySQL Workbench para gestión de base de datos
- ◆ Git para control de versiones

CONSIDERACIONES ADICIONALES

Seguridad

- Firewall configurado para permitir únicamente los puertos necesarios (típicamente 80, 443, 3306)
- Certificado SSL válido para comunicaciones HTTPS
- Políticas de backup y recuperación implementadas
- Actualizaciones de seguridad regulares en todos los componentes

Redes

- Conexión a Internet estable
- Configuración de red que permita el acceso a los puertos necesarios
- DNS configurado correctamente si se utiliza un nombre de dominio



Otros

- Cuenta de correo electrónico SMTP para envío de notificaciones
- Espacio de almacenamiento adicional para copias de seguridad
- Plan de mantenimiento para actualizaciones regulares



GUÍA DE INSTALACIÓN DEL SISTEMA DE TALLER MECÁNICO

Esta guía proporciona los pasos detallados para instalar y configurar correctamente el sistema de Taller Mecánico en entornos de desarrollo y producción.

PREPARACIÓN DEL ENTORNO

Requisitos Previos

Antes de comenzar la instalación, asegúrese de tener instalado:

- **Node.js (v16.x o superior)**: [Descargar Node.js](https://nodejs.org/)
- **MySQL (v8.0 o superior)**: [Descargar MySQL](https://dev.mysql.com/downloads/)
- **Git**: [Descargar Git](https://git-scm.com/downloads)

Verificación de requisitos

Ejecute los siguientes comandos para verificar que las herramientas están instaladas correctamente:

bash

```
node --version
```

```
npm --version
```

```
git --version
```

```
mysql --version
```



INSTALACIÓN DEL BACKEND

Clonar el Repositorio

bash

```
git clone https://github.com/yourusername/Practica1AyDTallerMecanico.git
```

```
cd Practica1AyDTallerMecanico/backend
```

1. Instalar Dependencias

bash

```
npm install
```

Configurar Variables de Entorno

1. Copie el archivo de ejemplo de variables de entorno:

bash

```
cp .env.example .env
```

2. Abra el archivo `.env` y configure los parámetros según su entorno:

- DB_HOS: Host de la base de datos (por defecto `localhost`)
- DB_USER: Usuario de la base de datos
- DB_PASSWORD: Contraseña del usuario de la base de datos
- DB_NAME: Nombre de la base de datos

3. Ejecutar Migraciones y Sembrar Datos

bash

```
npx sequelize-cli db:migrate
```

```
npx sequelize-cli db:seed:all
```

4. Iniciar el Servidor

bash

```
npm start
```

El backend debería estar corriendo en `http://localhost:3000` (puerto por defecto).

INSTALACIÓN DEL FRONTEND

1. Construir el proyecto:

bash

```
cd ../frontend
```

```
npm install
```

```
npm run build
```

2. Configurar el archivo de entorno:

Copiar el archivo `.env.example` a `.env`

bash

```
cp .env.example .env
```

Configurar la URL del backend en el archivo `.env`:

env

```
VITE_API_URL=http://localhost:3000
```

3. Iniciar el servidor de desarrollo:

bash

```
npm run dev
```

El frontend debería estar corriendo en `http://localhost:5173` (puerto por defecto).

ACCESO A LA APLICACIÓN

- ➔ Interfaz Web: Acceda a la aplicación a través de un navegador web en `http://localhost:5173`
- ➔ API REST: La API está disponible en `http://localhost:3000/api`

SOLUCIÓN DE PROBLEMAS COMUNES

→ Problemas de conexión a la base de datos:

- ◆ Verifique que el servidor MySQL esté corriendo
- ◆ Asegúrese de que las credenciales de la base de datos en el archivo `.env` sean correctas

→ Errores al iniciar el servidor:

- ◆ Revise los mensajes de error en la consola
- ◆ Asegúrese de haber instalado todas las dependencias correctamente

→ Problemas con puertos:

- ◆ Si los puertos por defecto están en uso, actualice los archivos de configuración y los comandos de inicio con los nuevos puertos

ARQUITECTURA GENERAL DEL SISTEMA DE TALLER MECÁNICO

El sistema de Taller Mecánico implementa una arquitectura cliente-servidor, siguiendo un patrón de diseño MVC (Modelo-Vista-Controlador). La aplicación está dividida en dos componentes principales: el frontend (cliente) y el backend (servidor).

COMPONENTES DE LA ARQUITECTURA

Frontend (Cliente)

- Tecnología principal: React.js (biblioteca JavaScript para construir interfaces de usuario)
- Framework de diseño: Bootstrap 5 para diseño responsive
- Gestión de estado: Estados locales de React y contexto (AuthContext)
- Comunicación con backend: Axios (cliente HTTP)
- Enrutamiento: React Router Dom

El frontend está organizado en componentes reutilizables y sigue un enfoque modular para facilitar el mantenimiento y la escalabilidad. La aplicación utiliza un diseño responsive que se adapta a diferentes tamaños de pantalla.

Backend (Servidor)

- Tecnología principal: Node.js con Express.js
- Patrón arquitectónico: MVC (Modelo-Vista-Controlador)
- ORM: Sequelize para la comunicación con la base de datos
- Autenticación: JWT (JSON Web Tokens) y sistema opcional de autenticación de dos factores
- Seguridad: Bcrypt para hashing de contraseñas

El backend expone una API RESTful que es consumida por el frontend. Cada ruta está asociada a un controlador específico que maneja la lógica de negocio y la comunicación con la base de datos.

Base de Datos

- Sistema de gestión: MySQL
- Modelado: Relacional
- ORM: Sequelize (mapeo objeto-relacional)

La base de datos está estructurada en múltiples tablas relacionadas que representan las entidades principales del sistema (usuarios, vehículos, servicios, repuestos, etc.).

FLUJO DE DATOS

Autenticación:

- El usuario introduce credenciales en la interfaz de React
- La solicitud se envía al servidor Node.js
- El servidor verifica las credenciales y genera un token JWT
- El token se almacena en el cliente para autenticar futuras solicitudes

Operaciones CRUD:

- El usuario interactúa con la interfaz React
- React genera una solicitud HTTP con Axios incluyendo el token JWT
- El servidor valida el token, procesa la solicitud y consulta/actualiza la base de datos
- El servidor devuelve una respuesta JSON
- React actualiza la interfaz de usuario con los datos recibidos

SEGURIDAD

La arquitectura implementa varias capas de seguridad:

- Autenticación: Sistema JWT con opción de autenticación de dos factores
- Autorización: Control de acceso basado en roles (RBAC)
- Encriptación: Contraseñas hasheadas con bcrypt
- Protección de rutas: Middleware de autenticación para proteger endpoints
- Validación: Validación de datos de entrada en el servidor

ESCALABILIDAD

La arquitectura permite la escalabilidad horizontal y vertical:

- Horizontal: Se pueden añadir más instancias del servidor para distribuir la carga
- Vertical: Se pueden aumentar los recursos de los servidores existentes

La separación clara entre frontend y backend facilita el escalado independiente de cada componente según las necesidades.

ESTRUCTURA DE DIRECTORIOS DEL BACKEND

```
\backend
├── config
├── Controller
├── index.js
├── migrations
├── Model
└── node_modules
```

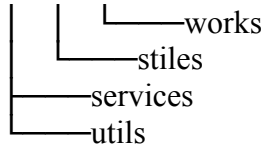


- package-lock.json
- package.json
- README.md
- Ruters
- seeders
- sequelize.config.js
- services

ESTRUCTURA DE DIRECTORIOS DEL FRONTEND

frontend:.

- node_modules
- src
 - auth
 - pages
 - admin
 - invoices
 - styles
 - proveedores
 - reports
 - styles
 - requests
 - styles
 - services
 - styles
 - vehicles
 - client
 - comments
 - invoices
 - profile
 - services
 - vehicles
 - employee
 - proveedor
 - specialist
 - comments
 - diagnostics
 - parts
 - profile
 - recommendations
 - styles
 - support
 - tests



DOCUMENTACIÓN DE LA API DEL SISTEMA DE TALLER MECÁNICO

Este apartado proporciona información sobre los endpoints disponibles en la API del sistema, organizados por módulos funcionales.

BASE URL

Todos los endpoints comienzan con: <http://localhost:3000/api>

ÍNDICE

1. Autenticación y Gestión de Usuarios
2. Gestión de Servicios
3. Gestión de Especialistas
4. Facturación
5. Solicitudes de Cliente
6. Funcionalidades de Cliente
7. Reportes

1. AUTENTICACIÓN Y GESTIÓN DE USUARIOS

Autenticación

POST /personas/login



- Descripción: Inicia sesión de usuario
- Requiere Autenticación: No

POST /personas/autenticar-codigo-verificacion

- Descripción: Verifica el código de autenticación de dos factores
- Requiere Autenticación: No

POST /personas/logout

- Descripción: Cierra la sesión del usuario
- Requiere Autenticación: Sí

POST /personas/recuperar-contrasena

- Descripción: Solicita la recuperación de contraseña
- Requiere Autenticación: No

POST /personas/validar-codigo-recuperacion

- Descripción: Valida el código de recuperación de contraseña
- Requiere Autenticación: No

PUT /personas/cambiar-contrasena

- Descripción: Cambia la contraseña del usuario
- Requiere Autenticación: Sí

PUT /personas/cambiar-autenticacion

- Descripción: Activa o desactiva la autenticación de dos factores
- Requiere Autenticación: Sí

Gestión de Usuarios

GET /management/users

- Descripción: Obtiene todos los usuarios
- Requiere Autenticación: Sí (admin)

GET /management/users/user/:id

- Descripción: Obtiene un usuario por ID
- Requiere Autenticación: Sí (admin)

POST /management/users

- Descripción: Crea un nuevo usuario
- Requiere Autenticación: Sí (admin)

PUT /management/users/:id

- Descripción: Actualiza un usuario existente
- Requiere Autenticación: Sí (admin)

DELETE /management/users/:id

- Descripción: Elimina (desactiva) un usuario
- Requiere Autenticación: Sí (admin)

GET /management/users/roles

- Descripción: Obtiene todos los roles disponibles
- Requiere Autenticación: Sí (admin)

POST /management/users/roles

- Descripción: Crea un nuevo rol
- Requiere Autenticación: Sí (admin)

GET /management/users/specialists

- Descripción: Obtiene todos los especialistas
- Requiere Autenticación: Sí (admin)

GET /management/users/areas

- Descripción: Obtiene todas las áreas de especialistas
- Requiere Autenticación: Sí (admin)

GET /management/users/tipos

- Descripción: Obtiene todos los tipos de técnicos
- Requiere Autenticación: Sí (admin)

POST /management/users/asignar-especializacion

- Descripción: Asigna una especialización a un usuario
- Requiere Autenticación: Sí (admin)

2. GESTIÓN DE SERVICIOS

GET /servicios/vehicles_with_client

- Descripción: Obtiene todos los vehículos con información del cliente
- Requiere Autenticación: Sí (admin/empleador)

GET /servicios/tipo_mantenimiento

- Descripción: Obtiene todos los tipos de mantenimiento
- Requiere Autenticación: Sí

POST /servicios/tipo_mantenimiento

- Descripción: Crea un nuevo tipo de mantenimiento
- Requiere Autenticación: Sí (admin)

POST /servicios/registro_servicio_vehiculo

- Descripción: Registra un nuevo servicio de vehículo
- Requiere Autenticación: Sí (admin/empleados)

GET /servicios/obtener_servicios

- Descripción: Obtiene todos los servicios registrados
- Requiere Autenticación: Sí

PUT /servicios/cambiar_estado_servicio

- Descripción: Cambia el estado de un servicio
- Requiere Autenticación: Sí (admin/empleados)

PUT /servicios/actualizar_servicio

- Descripción: Actualiza información de un servicio
- Requiere Autenticación: Sí (admin/empleados)

GET /servicios/empleados

- Descripción: Obtiene todos los empleados
- Requiere Autenticación: Sí (admin)

GET /servicios/especialistas

- Descripción: Obtiene todos los especialistas
- Requiere Autenticación: Sí (admin)

POST /servicios/asignar_trabajo



- Descripción: Asigna un trabajo a un empleado/especialista
- Requiere Autenticación: Sí (admin)

GET /servicios/trabajos_empleados/:id

- Descripción: Obtiene todos los trabajos asignados a un empleado específico
- Requiere Autenticación: Sí

GET /servicios/trabajos_servicio/:id

- Descripción: Obtiene todos los trabajos asignados a un servicio específico
- Requiere Autenticación: Sí

PUT /servicios/cambiar_empleado_trabajo

- Descripción: Cambia el empleado asignado a un trabajo
- Requiere Autenticación: Sí (admin)

3. GESTIÓN DE ESPECIALISTAS

GET /especialistas/trabajos_asignados/:id

- Descripción: Obtiene los trabajos asignados a un especialista
- Requiere Autenticación: Sí (especialista)

PUT /especialistas/actualizar_trabajo/:id

- Descripción: Actualiza el estado o información de un trabajo asignado
- Requiere Autenticación: Sí (especialista)

GET /especialistas/historial_vehiculo/:id

- Descripción: Obtiene el historial de servicios de un vehículo

- Requiere Autenticación: Sí (especialista)

POST /especialistas/agregar_diagnostico

- Descripción: Registra un diagnóstico para un trabajo
- Requiere Autenticación: Sí (especialista)

POST /especialistas/agregar_detalle_diagnostico/:id

- Descripción: Añade detalles a un diagnóstico existente
- Requiere Autenticación: Sí (especialista)

GET /especialistas/diagnosticos_especialista/:id

- Descripción: Obtiene todos los diagnósticos realizados por un especialista
- Requiere Autenticación: Sí (especialista)

POST /especialistas/agregar_prueba_tecnica

- Descripción: Registra una prueba técnica realizada
- Requiere Autenticación: Sí (especialista)

POST /especialistas/agregar_resultado_prueba/:id

- Descripción: Registra el resultado de una prueba técnica
- Requiere Autenticación: Sí (especialista)

GET /especialistas/pruebas_tecnicas_especialista/:id

- Descripción: Obtiene todas las pruebas técnicas realizadas por un especialista
- Requiere Autenticación: Sí (especialista)

POST /especialistas/agregar_propuesta_solucion

- Descripción: Registra una propuesta de solución
- Requiere Autenticación: Sí (especialista)

GET /especialistas/soluciones_prueba/:id

- Descripción: Obtiene las soluciones propuestas para un resultado de prueba
- Requiere Autenticación: Sí (especialista)

POST /especialistas/agregar_comentarios_vehiculo

- Descripción: Registra comentarios sobre un vehículo
- Requiere Autenticación: Sí (especialista)

GET /especialistas/comentarios_asignacion/:id

- Descripción: Obtiene los comentarios de una asignación de trabajo
- Requiere Autenticación: Sí (especialista/admin)

POST /especialistas/agregar_recomendacion_vehiculo

- Descripción: Registra recomendaciones para un vehículo
- Requiere Autenticación: Sí (especialista)

GET /especialistas/recomendaciones_asignacion/:id

- Descripción: Obtiene las recomendaciones para una asignación de trabajo
- Requiere Autenticación: Sí (especialista/admin)

POST /especialistas/crear_solicitud_apoyo

- Descripción: Crea una solicitud de apoyo técnico
- Requiere Autenticación: Sí (especialista)

GET /especialistas/solicitudes_apoyo_especialista/:id



- Descripción: Obtiene las solicitudes de apoyo de un especialista
- Requiere Autenticación: Sí (especialista)

POST /especialistas/responder_solicitud_apoyo

- Descripción: Responde a una solicitud de apoyo
- Requiere Autenticación: Sí (especialista)

GET /especialistas/solicitudes_repuestos

- Descripción: Obtiene las solicitudes de repuestos
- Requiere Autenticación: Sí (especialista/admin)

POST /especialistas/aceptar_repuesto

- Descripción: Acepta una solicitud de repuesto
- Requiere Autenticación: Sí (especialista/admin)

GET /especialistas/usuarios_especialistas

- Descripción: Obtiene todos los usuarios especialistas
- Requiere Autenticación: Sí (admin)

4. FACTURACIÓN

POST /facturas/generar

- Descripción: Genera una nueva factura para un servicio completado
- Requiere Autenticación: Sí (admin)

GET /facturas/listar

- Descripción: Lista todas las facturas registradas

- Requiere Autenticación: Sí (admin)

POST /facturas/registrar-pago

- Descripción: Registra un pago para una factura
- Requiere Autenticación: Sí

GET /facturas/listar-pagos/:id_factura

- Descripción: Lista todos los pagos de una factura específica
- Requiere Autenticación: Sí

GET /facturas/consultar-saldo/:id_factura

- Descripción: Consulta el saldo pendiente de una factura
- Requiere Autenticación: Sí

GET /facturas/obtener-servicios-completados

- Descripción: Obtiene todos los servicios completados listos para facturar
- Requiere Autenticación: Sí (admin)

5. SOLICITUDES DE CLIENTE

GET /request-client/servicios_adicionales

- Descripción: Obtiene todas las solicitudes de servicios adicionales
- Requiere Autenticación: Sí (admin)

GET /request-client/tipo_mantenimiento

- Descripción: Obtiene todos los tipos de mantenimiento disponibles
- Requiere Autenticación: Sí

POST /request-client/aceptar_servicio_adicional

- Descripción: Acepta una solicitud de servicio adicional
- Requiere Autenticación: Sí (admin)

POST /request-client/rechazar_servicio_adicional/:id

- Descripción: Rechaza una solicitud de servicio adicional
- Requiere Autenticación: Sí (admin)

GET /request-client/cotizaciones_servicios_precio/:id

- Descripción: Obtiene las cotizaciones de un cliente específico
- Requiere Autenticación: Sí (admin)

POST /request-client/agregar_trabajo_cotizacion

- Descripción: Agrega un trabajo a una cotización
- Requiere Autenticación: Sí (admin)

POST /request-client/enviar_cotizacion/:id_registro_cotizacion

- Descripción: Envía una cotización al cliente
- Requiere Autenticación: Sí (admin)

6. FUNCIONALIDADES DE CLIENTE

GET /client/mis_vehiculos/:id

- Descripción: Obtiene todos los vehículos registrados de un cliente
- Requiere Autenticación: Sí (cliente)

GET /client/mis_servicios/:id



- Descripción: Obtiene todos los servicios activos de un cliente
- Requiere Autenticación: Sí (cliente)

PUT /client/autorizar_servicio/:id

- Descripción: Autoriza un servicio por parte del cliente
- Requiere Autenticación: Sí (cliente)

PUT /client/no_autorizar_servicio/:id

- Descripción: No autoriza un servicio por parte del cliente
- Requiere Autenticación: Sí (cliente)

GET /client/detalle_servicios/:id

- Descripción: Obtiene los detalles completos de los servicios de un vehículo
- Requiere Autenticación: Sí (cliente)

GET /client/servicios_con_comentarios/:id

- Descripción: Obtiene los servicios con comentarios de seguimiento
- Requiere Autenticación: Sí (cliente)

POST /client/comentarios_seguimiento

- Descripción: Agrega un comentario de seguimiento a un servicio
- Requiere Autenticación: Sí (cliente)

GET /client/servicios_adicionales/:id

- Descripción: Obtiene los servicios adicionales solicitados por un cliente
- Requiere Autenticación: Sí (cliente)

POST /client/servicios_adicionales



- Descripción: Solicita un servicio adicional
- Requiere Autenticación: Sí (cliente)

GET /client/tipos_mantenimiento

- Descripción: Obtiene los tipos de mantenimiento disponibles
- Requiere Autenticación: Sí (cliente)

GET /client/cotizaciones_servicios/:id

- Descripción: Obtiene las cotizaciones de servicios del cliente
- Requiere Autenticación: Sí (cliente)

POST /client/solicitar_cotizacion

- Descripción: Solicita una cotización para un servicio
- Requiere Autenticación: Sí (cliente)

GET /client/facturas/:id

- Descripción: Obtiene todas las facturas del cliente
- Requiere Autenticación: Sí (cliente)

POST /client/pagar_factura

- Descripción: Registra el pago de una factura por el cliente
- Requiere Autenticación: Sí (cliente)

POST /client/calificar_servicio

- Descripción: Califica un servicio recibido
- Requiere Autenticación: Sí (cliente)

7. REPORTES

Reportes Operativos

GET /reportes/trabajos_por_periodo

- Descripción: Obtiene un reporte de trabajos realizados por período
- Requiere Autenticación: Sí (admin)
- Parámetros: startDate, endDate, status (opcional)

GET /reportes/historial_mantenimiento/:vehicleId

- Descripción: Obtiene el historial de mantenimiento para un vehículo específico
- Requiere Autenticación: Sí (admin)

GET /reportes/trabajos_completados

- Descripción: Obtiene un reporte de los trabajos completados
- Requiere Autenticación: Sí (admin)
- Parámetros: startDate, endDate, mechanicId (opcional)

GET /reportes/vehiculos

- Descripción: Obtiene todos los vehículos para reportes
- Requiere Autenticación: Sí (admin)

Reportes Financieros

GET /reportes/ingresos_gastos

- Descripción: Obtiene un reporte de ingresos y egresos por período
- Requiere Autenticación: Sí (admin)
- Parámetros: startDate, endDate

GET /reportes/gastos_proveedor



- Descripción: Obtiene un reporte de gastos por proveedor
- Requiere Autenticación: Sí (admin)
- Parámetros: startDate, endDate, providerId (opcional)

GET /reportes/proveedores

- Descripción: Obtiene todos los proveedores para reportes
- Requiere Autenticación: Sí (admin)

Reportes de Inventario

GET /reportes/partes_usadas

- Descripción: Obtiene un reporte de uso de repuestos por período
- Requiere Autenticación: Sí (admin)
- Parámetros: startDate, endDate

GET /reportes/partes_mas_usadas

- Descripción: Obtiene un reporte de los repuestos más usados por tipo de vehículo
- Requiere Autenticación: Sí (admin)
- Parámetros: brand (opcional), model (opcional), startDate (opcional), endDate (opcional)

Reportes de Clientes

GET /reportes/clientes

- Descripción: Obtiene todos los clientes para reportes
- Requiere Autenticación: Sí (admin)

GET /reportes/cliente_historial/:clientId

- Descripción: Obtiene el historial de servicios para un cliente específico



- Requiere Autenticación: Sí (admin)

GET /reportes/cliente_calificaciones

- Descripción: Obtiene un reporte de calificaciones de servicios
- Requiere Autenticación: Sí (admin)
- Parámetros: startDate (opcional), endDate (opcional), minRating (opcional), maxRating (opcional)