# Índice

# 

[**Índice 0**](#_heading=h.2et92p0)

[**Introducción 1**](#_heading=h.17dp8vu)

[**Objetivos 2**](#_heading=h.26in1rg)

[General 2](#_heading=h.lnxbz9)

[Específicos 2](#_heading=h.35nkun2)

[Nombre Del Proyecto 3](#_heading=h.1ksv4uv)

[Descripción General de la Aplicación Web 3](#_heading=h.44sinio)

[Responsable del proyecto 3](#_heading=h.2jxsxqh)

[Palabras Clave 4](#_heading=h.z337ya)

[Requerimientos Técnicos 4](#_heading=h.3j2qqm3)

[Herramientas a utilizar 5](#_heading=h.tnj11mdvit39)

[Node.js 5](#_heading=h.dp62wk1bpxsr)

[MySQL 6](#_heading=h.3tvdrgn3iyr6)

[React 8](#_heading=h.8si32gveasaj)

[Git 8](#_heading=h.ruae5kxj60ih)

[RapidAPI Client 9](#_heading=h.l380797v7su4)

[Herramientas y lenguajes utilizados para el desarrollo. 10](#_heading=h.1y810tw)

[Diagramas 11](#_heading=h.4i7ojhp)

[Diagrama de Arquitectura 11](#_heading=h.3fwokq0)

[Diagrama Base de datos 12](#_heading=h.3tbugp1)

[Diagrama de Tablas:Diagrama de Tablas PitShop Manager.png 12](#_heading=h.t4c64lykr51m)

[Diagrama ER: Diagrama ER-PitStop-Manager.png 12](#_heading=h.qc67mdci5ysc)

[Modelo Conceptual: Modelo conceptual - PitStop-Manager 13](#_heading=h.nmf14n)

[Modelo Lógico: Modelo Logico-PitShop Manager.png 13](#_heading=h.37m2jsg)

[Modelo Físico: Modelo Fisico-PitStop-Manager.png 13](#_heading=h.utvijyj2qr8n)

[Instalación Del Sistema 14](#_heading=h.46r0co2)

[**Casos de Uso 14**](#_heading=h.4zektzjtqvim)

[Estructura del proyecto 15](#_heading=h.111kx3o)

[Estructura del Backend 15](#_heading=h.3l18frh)

[Estructura del Frontend 16](#_heading=h.2zbgiuw)

[Despliegue de la Aplicación 17](#_heading=h.3ygebqi)

[Despliegue 17](#_heading=h.2dlolyb)

[Backend 17](#_heading=h.sqyw64)

[Frontend 17](#_heading=h.3cqmetx)

# 

# 

# Introducción

El presente manual técnico documenta el diseño, desarrollo e implementación de la aplicación web PitStop Manager, un sistema orientado a la gestión integral de talleres mecánicos. La herramienta surge como respuesta a la necesidad de modernizar y optimizar procesos que comúnmente se realizan de forma manual y desorganizada, tales como la administración de inventarios, la asignación de trabajos, la facturación y la atención al cliente.

El sistema ha sido diseñado bajo un enfoque modular y multiusuario, permitiendo a cada perfil (administrador, empleado, especialista, cliente y proveedor) interactuar con las funcionalidades que corresponden a sus responsabilidades. Asimismo, se incorporan mecanismos de seguridad como autenticación multifactor, recuperación de contraseñas y control de accesos, con el objetivo de resguardar la información sensible del taller y de los clientes.

Este manual constituye una guía técnica que describe los componentes del sistema, los objetivos de su implementación, la estructura de la aplicación y las tecnologías empleadas, sirviendo como referencia para su mantenimiento, escalabilidad y mejora continua.

# 

## 

# Objetivos

## General

* Desarrollar una aplicación web que permita gestionar de manera eficiente las operaciones principales de un taller mecánico, optimizando la atención al cliente, la administración de recursos, el control de trabajos y la generación de reportes administrativos.

## Específicos

* Diseñar una interfaz gráfica amigable que facilite la interacción entre los distintos perfiles de usuario.
* Implementar funciones de autenticación, doble factor de seguridad y recuperación de credenciales.
* Desarrollar módulos para la gestión de clientes, vehículos, empleados, especialistas y proveedores.
* Controlar de manera eficiente el inventario de repuestos y su relación con los trabajos mecánicos.
* Incorporar la facturación digital y el control de pagos, tanto de clientes como a proveedores.
* Implementar un módulo de reportes que permita generar información consolidada para la toma de decisiones.
* Favorecer el trabajo colaborativo y la integración de procesos dentro del entorno del taller.

# Nombre Del Proyecto

**PitStop Manager**

# Descripción General de la Aplicación Web

PitStop Manager es una aplicación web de gestión de talleres mecánicos que centraliza las principales operaciones en un entorno digital seguro y accesible.

El sistema se estructura en torno a cinco roles principales:

* Administrador: Tiene control total sobre la gestión del taller, incluyendo clientes, vehículos, trabajos, inventarios, facturación y reportes.
* Empleado: Ejecuta los trabajos asignados, registra avances y notifica necesidades adicionales.
* Especialista: Atiende intervenciones técnicas específicas, emite diagnósticos detallados y recomendaciones.
* Cliente: Consulta en tiempo real el estado de su vehículo, autoriza servicios, realiza pagos y descarga facturas.
* Proveedor: Administra pedidos, entrega repuestos, emite facturas y actualiza disponibilidad de productos.

El sistema incluye además un módulo de reportes que facilita la generación de estadísticas clave como ingresos, egresos, historial de clientes, uso de repuestos y productividad de los empleados. Todo ello con el respaldo de funciones avanzadas de seguridad y control de accesos.

# Responsable del proyecto

| Jeffrey Kenneth Menéndez Castillo  Brandon Josué Pinto Méndez  William Alexander Miranda Santos |
| --- |

# Palabras Clave

* **PitStop Manager:** Plataforma de gestión y manejo de talleres mecánico.
* **Node.js v18.19.1:** Entorno de ejecución para el backend.
* **MySQL**: Base de datos relacional.
* **JWT**: Sistema de autenticación basado en tokens.
* **React:** Framework para el frontend.
* **RapidAPI Client:** Extensión para consumo de API.
* **Git:** Sistema de control de versiones.
* **GitHub:** Plataforma para alojar repositorios de Git.
* **Gestión de talleres mecánicos:** Se refiere al conjunto de procesos administrativos, técnicos y operativos que permiten organizar y controlar las actividades de un taller automotriz.
* **Aplicación web:** Software accesible a través de navegadores de internet, desarrollado para ejecutarse en un servidor y ofrecer funcionalidades a múltiples usuarios en tiempo real.
* **PitStop Manager:** Nombre de la aplicación web desarrollada en el marco de este proyecto académico.
* **Inventario de repuestos:** Módulo del sistema que centraliza el registro, control y actualización de los repuestos disponibles en el taller.
* **Control de trabajos:** Funcionalidad que permite dar seguimiento a las tareas de mantenimiento o reparación de los vehículos.
* **Facturación electrónica:** Proceso mediante el cual se generan, emiten y almacenan comprobantes fiscales en formato digital, garantizando su validez legal.
* **Usuarios multirol:** Característica del sistema que permite la existencia de distintos perfiles de usuario (administrador, empleado, especialista, cliente y proveedor), cada uno con permisos y funcionalidades específicas de acuerdo con sus responsabilidades dentro del taller.
* **Reportes administrativos:** Módulo orientado a la generación de información consolidada sobre las operaciones del taller.
* **Autenticación y seguridad:** Conjunto de mecanismos que garantizan la protección de la información y la integridad del sistema.
* **Optimización de procesos:** Acción de mejorar la eficiencia y efectividad de los procedimientos del taller mediante la digitalización y automatización.

# Requerimientos Técnicos

**Procesador: core i3 gen o superior**

**Memoria RAM: 6 Gb**

**Espacio de almacenamiento: 15 Gb**

**Sistema Operativo**

**Otros:** Se requiere de la instalación de las siguientes herramientas.

# Herramientas a utilizar

## Node.js

Simulador de JavaScript para la creación del backend y las APIs REST.

**Instalación en Linux (Distribuciones basadas en Ubuntu/Debian):**

1. **Instalar Node.js usando nvm (Node Version Manager):**

* **Instala nvm:**
* curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.5/install.sh | bash
* **Luego, reinicia el terminal o ejecuta:**
* source ~/.bashrc
* **Instala la versión deseada de Node.js:**
* nvm install node 18

1. **Verifica la instalación:**

* node -v
* npm -v

**Instalación en Mac:**

1. **Usar Homebrew (recomendado):**

* Asegúrate de tener Homebrew instalado. Si no, instálalo con:
* /bin/bash -c "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh>)"
* Instala Node.js:
* brew install node

1. **Verifica la instalación:**

* node -v
* npm -v

**Instalación en Windows:**

1. **Descargar el instalador:**

* Ve al sitio web oficial de Node.js y descarga el instalador de la versión LTS (Long Term Support).

1. **Ejecutar el instalador:**

* Abre el archivo .msi descargado y sigue las instrucciones del asistente de instalación. Asegúrate de que la opción "Add to PATH" esté marcada.

1. **Verificar la instalación:**

* Abre cmd o PowerShell y ejecuta:
* node -v
* npm -v

## MySQL

Base de datos relacional para la gestión de datos como usuarios, productos y pedidos.

**Instalación en Linux (Distribuciones basadas en Ubuntu/Debian):**

1. **Instalar MySQL Server:**

* sudo apt install mysql-server

1. **Iniciar el servicio de MySQL:**

* sudo systemctl start mysql

1. **Habilitar el inicio automático de MySQL al arrancar el sistema:**

* sudo systemctl enable mysql

1. **Asegurar la instalación de MySQL:**

* sudo mysql\_secure\_installation

1. **Verificar el estado de MySQL:**

* sudo systemctl status mysql

**Instalación en Mac:**

1. **Usar Homebrew:**

* Instala MySQL con Homebrew:
* brew install mysql

1. **Iniciar el servicio de MySQL:**

* brew services start mysql

1. **Verificar la versión de MySQL:**

* mysql --version

1. **Configurar la seguridad de MySQL:**

* mysql\_secure\_installation

**Instalación en Windows:**

1. **Descargar el instalador:**

* Ve a la página de descargas de MySQL y descarga el instalador (generalmente el "MySQL Installer for Windows").

1. **Ejecutar el instalador:**

* Abre el instalador descargado y sigue los pasos del asistente. Puedes optar por la instalación "Developer Default" o personalizada.
* Durante la configuración, elige el tipo de servidor (e.g., Standalone MySQL Server), y establece la contraseña de root.

1. **Configurar y verificar la instalación:**

* Asegúrate de que el servicio de MySQL esté configurado para iniciarse automáticamente.
* Después de la instalación, abre cmd o PowerShell y verifica la versión:
* mysql --version

1. **Acceder a MySQL:**

* Accede a MySQL con
* mysql -u root -p
* Ingresa la contraseña que configuraste durante la instalación.

## 

## React

Framework/biblioteca utilizada para la creación de interfaces de usuario en aplicaciones web.

La instalación de React en Linux, Mac y Windows es prácticamente la misma, ya que se basa en Node.js y npm (o yarn/pnpm).

**Requisitos Previos**

* Node.js y npm deben estar instalados en tu sistema. Puedes verificarlo ejecutando:
* node -v
* npm -v

**Instalación en Linux**, **Mac** y **Windows:**

1. **Abrir la terminal.**
2. **Crear un nuevo proyecto React con Vite:**

* npm create vite@latest my-app

1. **Durante la instalación, selecciona:**

* Nombre del proyecto.
* Framework: React.
* Variante: JavaScript o TypeScript.

1. **Entrar al proyecto:** - cd my-app
2. **Instalar dependencias** - npm install
3. **Ejecutar la aplicación:** - npm run dev

## Git

Sistema de control de versiones para gestionar el código.

**Instalación en Linux (Distribuciones basadas en Ubuntu/Debian):**

1. **Instalar Git:**

* sudo apt install git

1. **Verificar la instalación:**

* git --version

**Instalación en Mac:**

1. **Usar Homebrew:**

* Instala Git con Homebrew:
* brew install git

1. **Verificar la instalación:**

* git --version

**Instalación en Windows:**

1. **Descargar el instalador:**

* Ve al sitio oficial de Git y descarga el instalador de Windows.

1. **Ejecutar el instalador:**

* Abre el archivo .exe descargado y sigue los pasos del asistente de instalación.
* Durante la instalación, se recomienda seleccionar las opciones predeterminadas a menos que tengas requisitos específicos (e.g., elegir tu editor de texto preferido para Git).

1. **Verificar la instalación:**

* Abre cmd, PowerShell o Git Bash y ejecuta:
* git --version

## RapidAPI Client

RapidAPI Client es una extensión para gestionar APIs dentro de editores de código como Visual Studio Code (VS Code).

**Instalación en VS Code:**

1. **Abrir Visual Studio Code**.
2. **Ir a la sección de extensiones**:

* Haz clic en el ícono de **Extensiones** en la barra lateral izquierda o usa el atajo Ctrl + Shift + X (Windows/Linux) o Cmd + Shift + X (Mac).

1. **Buscar la extensión**:

* En la barra de búsqueda, escribe RapidAPI Client.

1. **Instalar la extensión**:

* Haz clic en el botón **Instalar** junto a la extensión **RapidAPI Client** (publicada por RapidAPI Inc.).

1. **Reiniciar VS Code** (si es necesario):

* Algunas veces, es posible que necesites reiniciar VS Code para que la extensión se cargue correctamente.

# Herramientas y lenguajes utilizados para el desarrollo.

1. **Node.js:**

Node.js es un entorno de ejecución de JavaScript de código abierto y multiplataforma que permite a los desarrolladores ejecutar JavaScript en el servidor. Utiliza el motor V8 de Google y es ideal para construir aplicaciones de red escalables y de alta concurrencia, como servidores web y APIs.

1. **MySQL:**

MySQL es un sistema de gestión de bases de datos relacional de código abierto. Es ampliamente utilizado para almacenar y gestionar datos en aplicaciones web y empresariales. Es conocido por su fiabilidad, facilidad de uso y capacidad de integrarse con una variedad de lenguajes de programación y plataformas.

1. **React:**

React es una biblioteca de JavaScript desarrollada por Facebook para construir interfaces de usuario de manera eficiente y modular. Se basa en el concepto de componentes reutilizables, lo que facilita el desarrollo y mantenimiento de aplicaciones web modernas. React permite actualizar y renderizar de forma óptima los elementos de la interfaz cuando cambian los datos, utilizando un sistema llamado Virtual DOM.

1. **Git:**

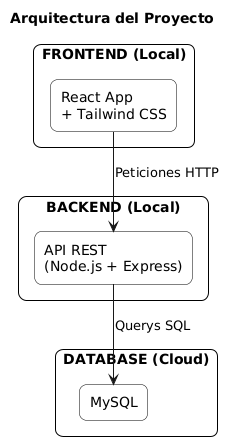
Git es un sistema de control de versiones distribuido que permite a los desarrolladores gestionar y rastrear cambios en el código fuente de un proyecto. Es fundamental para el trabajo colaborativo en proyectos de software, ya que facilita la coordinación y el manejo de versiones del código.

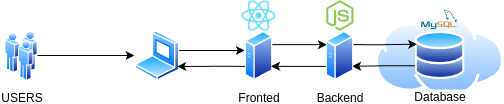
1. **RapidAPI Client:**

RapidAPI Client es una extensión para editores de código, como Visual Studio Code, que permite a los desarrolladores explorar, probar y gestionar APIs directamente desde el entorno de desarrollo. Ofrece una forma eficiente de interactuar con servicios web y probar solicitudes HTTP sin necesidad de herramientas externas.

# Diagramas

### Diagrama de Arquitectura

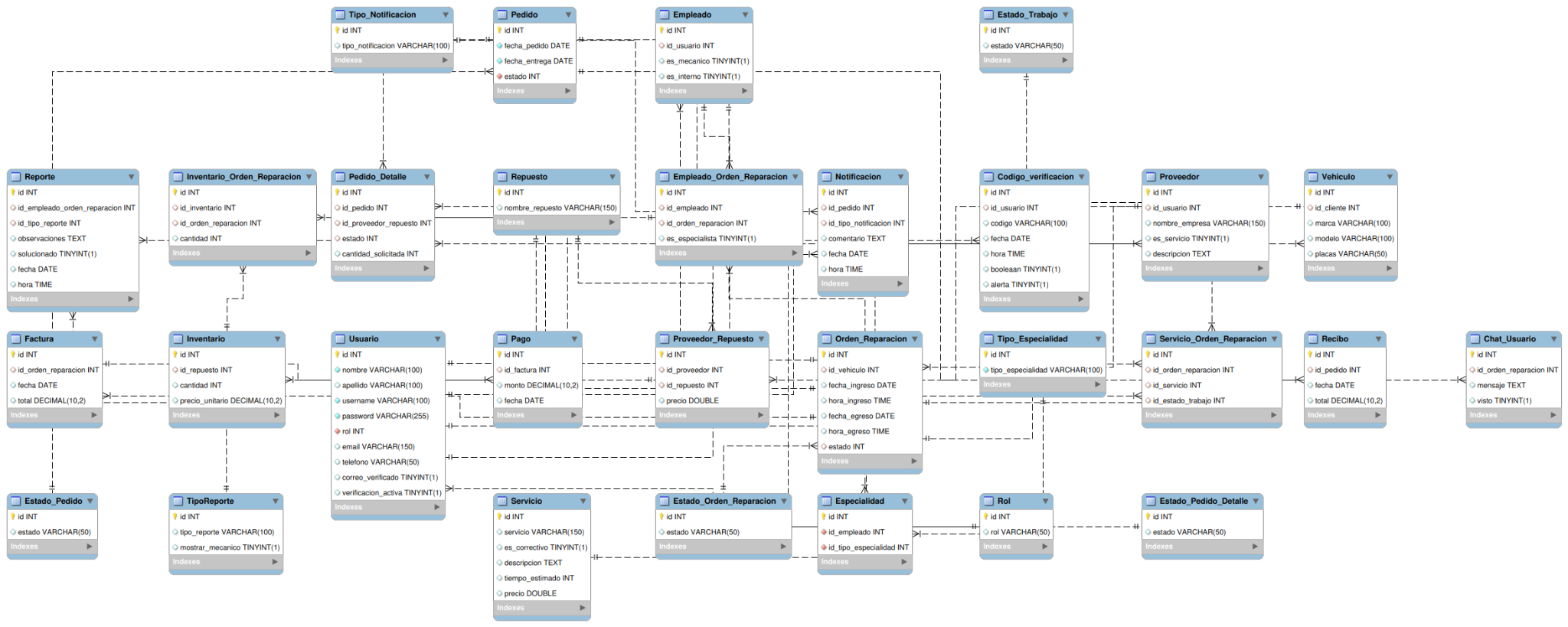




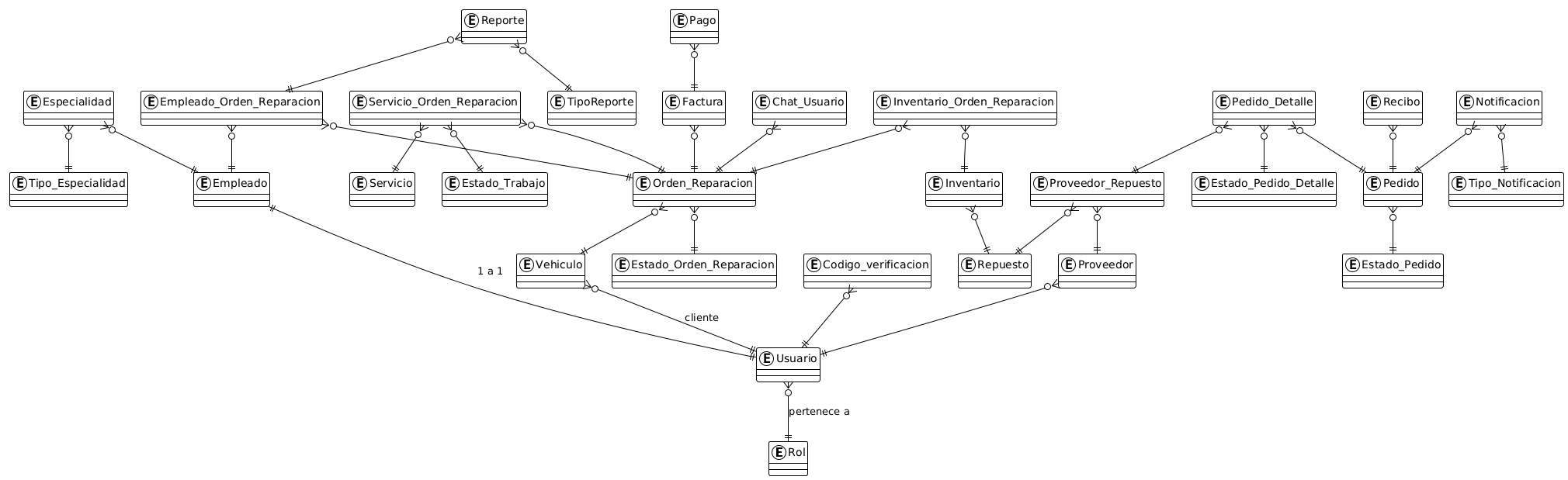
## Diagrama Base de datos

### Diagrama de Tablas:[Diagrama de Tablas PitShop Manager.png](https://drive.google.com/file/d/18_AA45Nk0IZj1ox6N_uQS-NAylEgkr7R/view?usp=drive_link)

### Diagrama ER: [Diagrama ER-PitStop-Manager.png](https://drive.google.com/file/d/1ntn1fvnZsiPK8KpQGws3OU21-XCNGxy6/view?usp=drive_link)



### Modelo Conceptual: [Modelo conceptual - PitStop-Manager](https://drive.google.com/file/d/1xI83xhF_mElXDmLQ8tqBnJ-b7TRNJR9f/view?usp=drive_link)



### Modelo Lógico: [Modelo Logico-PitShop Manager.png](https://drive.google.com/file/d/1M8vgJdiUKzkWnSvQoxL_5mT_NymeJasY/view?usp=drive_link)

### Modelo Físico: [Modelo Fisico-PitStop-Manager.png](https://drive.google.com/file/d/1ojzWZFPUTCiyEZGtBISJKBknZAsgtDlF/view?usp=drive_link)

# Instalación Del Sistema

**Clonar el repositorio desde Github**

***git clone https://github.com/Pinto324/Backend-PitStopManager.git***

**Instalar dependencias para Backend**

***cd Proyecto/Backend***

***npm install***

**Instalar dependencias para Frontend**

***cd Proyecto/Fronted***

***npm install***

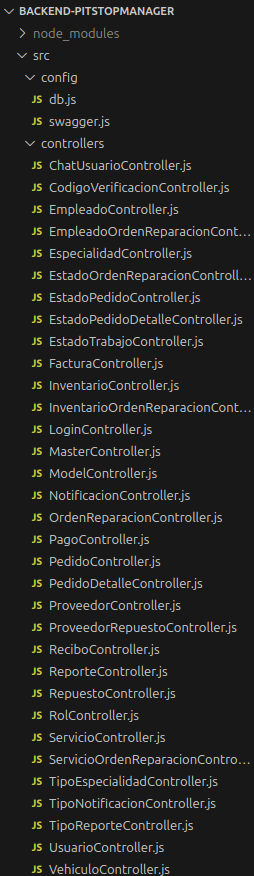
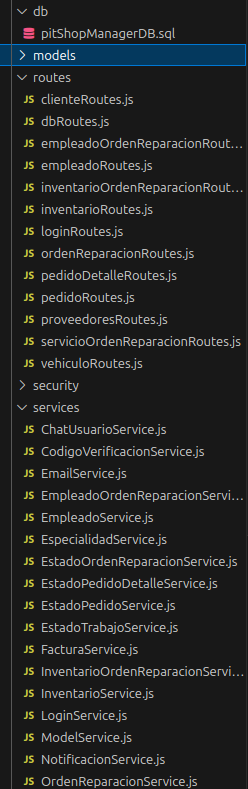
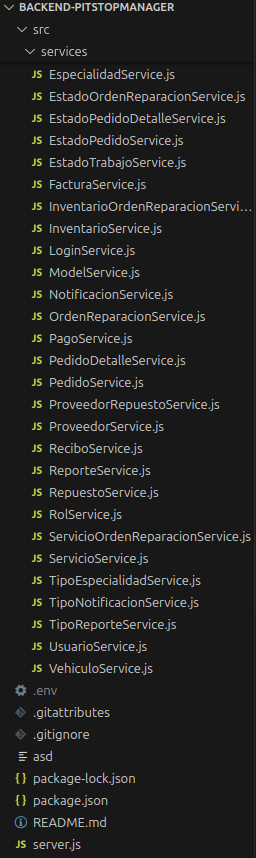
# Casos de Uso

[***CASOS DE USO.pdf***](https://drive.google.com/file/d/17rFvdFNLCFy6TukzyHJVepg1pjfLx94W/view?usp=drive_link)

## 

# Estructura del proyecto

## Estructura del Backend



## Estructura del Frontend

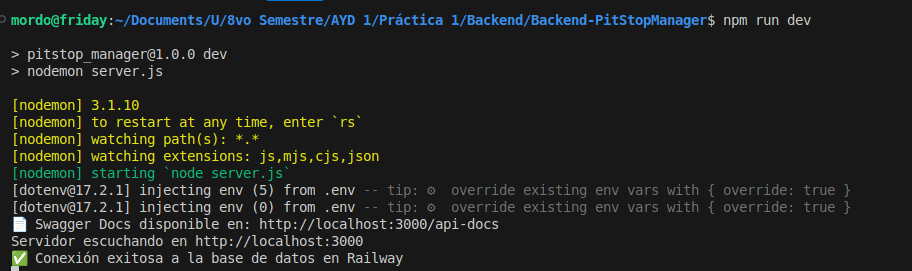
## 

# Despliegue de la Aplicación

## Despliegue

Para la ejecución de la aplicación, ingrese en Visual Studio Code donde se clonó el repositorio del proyecto y siga las siguientes directrices:Para la ejecución de la aplicación, ingrese en Visual Studio Code donde se clonó el repositorio del proyecto y siga las instrucciones siguientes:

### Backend

* **Instalar dependencias:**
* npm install
* **Desplegar la aplicación:**
* npm run dev

Se debería desplegar el siguiente mensaje desde consola para asegurar que se está ejecutando correctamente

### Frontend

* **Instalar dependencias:**
* npm install
* **Desplegar la aplicación:**
* npm run dev/prod (ambiente utilizado)

