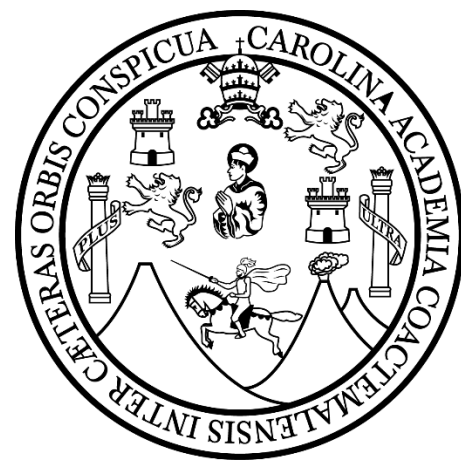


Universidad de San Carlos de Guatemala  
Centro Universitario de Occidente (CUNOC)  
División de Ciencias de la Ingeniería  
Laboratorio de análisis y Diseño de Sistemas 1  
Auxiliar: Pedro Gordillo



## Manual Técnico

### Práctica 1 - Taller Mecánico

Diego José Ávila Estrada	201731120
Hania Mirleth Mazariegos Alonzo	202031953
José Roberto Bautista Rojas	202131284

## Introducción

En el presente manual se detalla que tipo de arquitectura se utilizo para el diseño de la aplicación para el sistema de un taller mecánico, se proporciona información importante para aquellos desarrolladores que deseen explorar el proyecto y entender como funciona la estructura del proyecto, se explica la distribución de paquetes utilizada para darle un orden al proyecto.

Este manual incluye también la información sobre las tecnologías utilizadas tanto en la parte FrontEnd como en el BackEnd.

El objetivo principal de este manual es facilitar la implementación del proyecto de “Taller Mecánico” en diversos entornos de producción, asegurando que los usuarios puedan resolver problemas comunes y realizar tareas de mantenimiento con eficiencia.

## Implementación de Desarrollo FrontEnd y BackEnd

### Implementación Backend

Requisitos de instalación:

1. GIT
2. Maven versión 3.9.x o superior.
3. JDK 21 (Java 21)
4. IDE (cualquier IDE)
5. PostgreSQL

Tecnologías utilizadas:

1. Java 21
2. Spring Framework
3. SpringBoot 3.x.x
4. PostgreSQL

Arquitectura del proyecto:

El proyecto está desarrollado bajo la arquitectura Monolítica, esta arquitectura es un modelo de desarrollo de software donde toda la aplicación (En este caso la aplicación Backend) se construye como un único bloque de código. Por ser un proyecto no tan grande, se decidió que todo se desarrollara en un solo servicio.

Configuración del Proyecto:

Archivo application.yml :

En la raíz del proyecto se encuentra el archivo **application.yml** en el que se encuentra la configuración básica del proyecto, tales como:

1. versión de Java
2. número de puerto en el que opera localmente,
3. conexión JDBC a la base de datos: Aquí se debe de modificar según el usuario y contraseña de la conexión local al SGBD (PostgreSQL)
4. configuración de API para enviar correos Gmail a usuarios.

Archivo POM.xml:

En la raíz del proyecto también se encuentra el archivo pom.xml, el encargado de gestionar todas nuestras dependencias de desarrollo que son necesarias para la implementación del código, se encuentran dependencias claves como:

1. Dependencia de Driver para conexión de PostgreSQL a Java.
2. Dependencia Spring Web para la creación de la REST API.

3. JPA/Hibernate para el manejo de entidades de la base de datos.

Instrucciones de implementación:

1. Ubicar el repositorio GIT donde está el proyecto backend
2. Clonar el repositorio
3. Ubicarse en la rama main para tener los cambios más actualizados y seguros del desarrollo
4. Si se desea implementar nuevas funcionalidades, se aconseja crear una nueva rama a partir de la rama MAIN.
5. Compilar el proyecto ejecutando **mvn clean install**.
6. Iniciar el proyecto con mvn spring-boot:run o en el botón de inicio del IDE.



## Implementación FrontEnd

### Tecnologías Frontend

Nuxt 4: Un framework de código abierto construido sobre Vue.js que simplifica el desarrollo de aplicaciones web con renderizado del lado del servidor (SSR) y generación de sitios estáticos, optimizando el rendimiento y la experiencia del desarrollador.

Vue 3: Un framework progresivo de JavaScript utilizado para construir interfaces de usuario e interfaces de una sola página. Se destaca por su sintaxis intuitiva, un alto rendimiento gracias al Virtual DOM y una reactividad eficiente.

Nuxt UI: Una biblioteca de componentes de interfaz de usuario para Nuxt, que ofrece una amplia gama de elementos listos para usar, como botones, formularios y modales, facilitando el diseño rápido de interfaces con una apariencia profesional y consistente.

Tailwind CSS: Un framework CSS que proporciona clases de utilidad de bajo nivel para estilizar componentes directamente en el marcado HTML. Acelera el proceso de diseño al permitir la creación de interfaces personalizadas sin escribir CSS desde cero.

### Requisitos:

1. Tener instalado NPM
2. Tener instalado NodeJS

### Instrucciones de implementación:

Para poder iniciar el proyecto Frontend, se deben de seguir las siguientes instrucciones:

1. Ubicar el repositorio GIT donde se encuentra el proyecto.
2. Clonar el repositorio a nuestro entorno local.
3. Pegar el archivo .env en la raíz del proyecto
4. Desde la terminal, ejecutar los siguiente comandos:
  - a. npm install
  - b. npm run dev
5. la URL inicial es /login

