

# My-Epic-To-Do-List

Proyecto Control 2 – TBD

**Autores:** Byron Gracia, Orlando Solis , Maximiliano Ormeño, Jairo Santi, José Llancamil, Jaime Vadell

## Descripción

El proyecto implica la creación de un sistema de gestión de tareas que consta de dos partes: un frontend desarrollado en Vue.js y un backend basado en una API RESTful desarrollada en Spring. Los usuarios podrán realizar las siguientes acciones en el sistema: crear, editar, eliminar y hacer un seguimiento de sus tareas pendientes. Este sistema permitirá a los usuarios administrar sus listas de tareas de manera efectiva.

## Características

El sistema en cuestión se caracteriza por su funcionalidad de gestión de tareas, que permite a los usuarios llevar un registro de sus tareas pendientes. Para llevar a cabo su implementación, se siguió un enfoque escalonado comenzando por el desarrollo de las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) de las entidades relacionadas con las tareas.

Además, el sistema se compone de dos servidores distintos: uno para el frontend, desarrollado en Vue.js, y otro para el backend, basado en SpringBoot. El servidor de Vue.js se encarga de la interfaz de usuario, mientras que el servidor de SpringBoot se encarga de la lógica detrás de la gestión de tareas y la autenticación de usuarios mediante WebSecurity y JWT.

En el lado del backend, para la gestión de la base de datos, se optó por la utilización de SQL2o. Debido a esta elección, el proyecto no hace uso de un Mapeador Objeto-Relacional (ORM). En lugar de depender de un ORM, SQL2o proporciona una interfaz directa y eficiente para interactuar con la base de datos, lo que permite un control más preciso sobre las consultas y operaciones de base de datos.

La aplicación actualmente ofrece a los usuarios un conjunto de características. Los usuarios pueden iniciar sesión y registrarse de manera segura a través de un sistema de autenticación basado en Web Security y JSON Web Tokens (JWT). Además, pueden gestionar sus tareas de manera eficaz, lo que incluye la capacidad de crear nuevas tareas, realizar ediciones en las existentes y eliminar aquellas que ya han sido completadas o no son relevantes.

# Instrucciones de configuración y despliegue de la aplicación de gestión de tareas, correspondiente al Control 2 de TBD 02-2023

## 1. Prerrequisitos

Se debe contar con las siguientes tecnologías para la ejecución optima del proyecto.

- PgAdmin4
- Node.js
- IDE IntelliJ
- JDK 17
- Postgres 12

## 2. Clonar el Repositorio

Se debe clonar o descargar el proyecto desde GitHub, la URL al repositorio del proyecto es [onlyG19/Control2\\_TBD2023 \(github.com\)](https://github.com/onlyG19/Control2_TBD2023)

## 3. Base de Datos

Se debe crear una base de datos llamada **postgres1**, la cual será poblada con el archivo SQL que se provee en el proyecto (en la carpeta dbScripts), este archivo es dbCreate.sql el cual genera las tablas o entidades necesarias en la base de datos.

La forma de ejecutar el script SQL es a través de pgadmin. Se puede directamente abrir el archivo o copiar el código en el "Query Tool" y ejecutar el script.

Nota: El script debe ejecutarse sobre la base de datos previamente creada, como ya se mencionó esta debe tener el nombre de **postgres1**.

## 4. Backend

Dentro del directorio base del repositorio tenemos una carpeta src, luego debe acceder a la carpeta del **backend** homónima. Idealmente se debe abrir esta carpeta de backend con el IDE IntelliJ.

Una vez abierto el proyecto, en la carpeta src/main/resources/ se tiene un archivo llamado application.properties. Aquí se deben asignar ó reemplazar, de ser necesario, las credenciales de acceso a la base de datos postgres (usuario, contraseña y url de la base de datos).

Nota: Es importante que la configuración de application.properties coincida con el detalle de nuestra base de datos. Además, por defecto el puerto del server backend es 8080.

Para ejecutar el código se debe abrir la carpeta raíz del proyecto con el IDE IntelliJ y abrir el archivo **BackendApplication.java** y presionar el boton de play verde "Run" en el editor.

Notar que antes de ejecutar la aplicación se deben instalar las dependencias. A través del IDE IntelliJ, se pueden instalar automáticamente utilizando el plugin de Maven.

## 5. Frontend

El código fuente de la parte del frontend del proyecto se encuentra en la carpeta src del directorio raíz del repositorio, luego se debe acceder a la carpeta con nombre **frontend**

Antes de poder ejecutar el frontend se deben instalar las dependencias en una consola utilizando el siguiente comando:

```
npm install
```

Luego, para poder ejecutar un servidor web de pruebas para el frontend se utiliza el siguiente código

```
npm run dev
```

Si se ejecutó lo anterior sin problemas, a continuación se podrá visualizar la interfaz gráfica de la aplicación en el navegador donde se encontraran las vistas del sistema. Por defecto se puede acceder al frontend en con la url <http://localhost:3030/>