INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

Manual

Desarrollador

Tecnologías de la Información

Implementación de una herramienta open source para el monitoreo de bases de datos MariaDB,

Autor

Ángel Paul Patiño Diaz

Índice de contenido

Introducción

Este manual del desarrollador proporciona la información técnica necesaria para comprender, instalar, configurar y mantener la herramienta de monitoreo implementada. El documento sirve como guía para el personal técnico encargado del mantenimiento y la administración del sistema, detallando los aspectos clave de su infraestructura, dependencias y funcionamiento.

Objetivos

Este manual tiene como finalidad asistir al personal técnico en la instalación, configuración y mantenimiento de la herramienta de monitoreo. Proporciona una guía detallada sobre el proceso de instalación, describe los archivos clave del sistema y ofrece información esencial para garantizar una configuración adecuada y un soporte eficiente de la herramienta.

Requerimientos de tecnologías

Para garantizar una instalación correcta y el funcionamiento óptimo de la herramienta de monitoreo, el sistema debe cumplir con los siguientes requisitos:

- Docker
- Java JDK 17
- Node 20.14.0
- Angular CLI v19.1.6

Especificaciones Técnicas Mínimas

Estas especificaciones permitirán ejecutar los servicios con un rendimiento aceptable en entornos de desarrollo pequeños:

- CPU: 4 núcleos (Intel Core i5 de 6ª generación o AMD Ryzen 3)
- RAM: 8 GB
- Almacenamiento: 50 GB SSD
- **Sistema Operativo:**
    - Linux (Ubuntu 20.04+)
    - Windows 10/11 Pro (con WSL2)
    - macOS

- Docker: Docker Desktop o Docker Engine (v20+)
- Java: JDK 17+ (para Spring Boot)
- Node.js: v18+ (para React)
- Navegador: Google Chrome o Edge actualizado

Especificaciones Técnicas Recomendadas

Para obtener un mejor rendimiento en desarrollo y pruebas en múltiples contenedores, es recomendable cumplir con las siguientes especificaciones:

- CPU: 8 núcleos (Intel Core i7 de 10ª generación o AMD Ryzen 7)
- RAM: 16 GB o más
- Almacenamiento: 100 GB SSD NVMe
- Sistema Operativo:
    - Linux (Ubuntu 22.04+)
    - Windows 11 Pro (con WSL2)
    - macOS con chip M1/M2
- Docker: Docker Engine con soporte para Compose v2
- Java: JDK 21+
- Node.js: v20+
- Extras: GPU para aceleración en frontend (opcional), monitoreo con herramientas como htop, ncdu y netdata.

1. Instalaciones

1.1. Instalar de Docker

Para gestionar contenedores, es necesario instalar *Docker Desktop*, una herramienta que facilita la administración y ejecución de contenedores. Para ello acceda al sitio web oficial de Docker y realice la descarga de dicho programa.

Fig. 1 Descarga de Docker Desktop desde el sitio oficial

Una vez descargado e instalado Docker Desktop, ejecútelo y configure las opciones según sus necesidades. Si no requiere ajustes específicos, puede mantener la configuración predeterminada y continuar con el proceso de instalación.

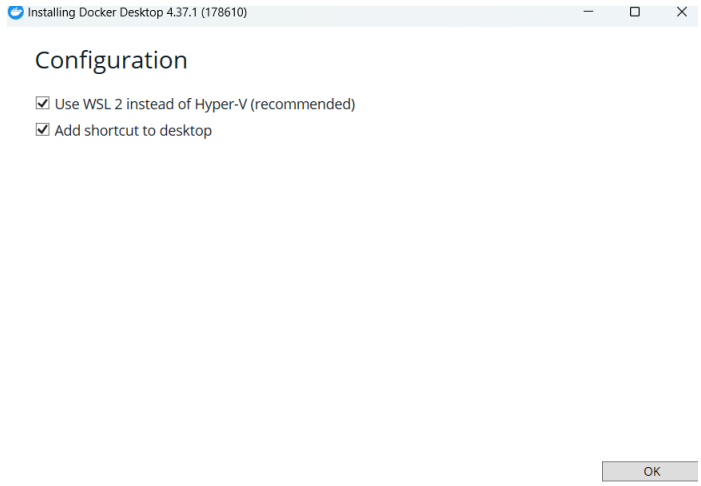

Fig. 2 Opciones de configuración para la instalación de docker desktop.

Una vez finalizada la instalación, aparecerá una ventana con la opción de cerrar el instalador. Al abrir Docker Desktop por primera vez, se mostrará una pantalla con los términos de licencia, los cuales deben aceptarse para acceder al software.

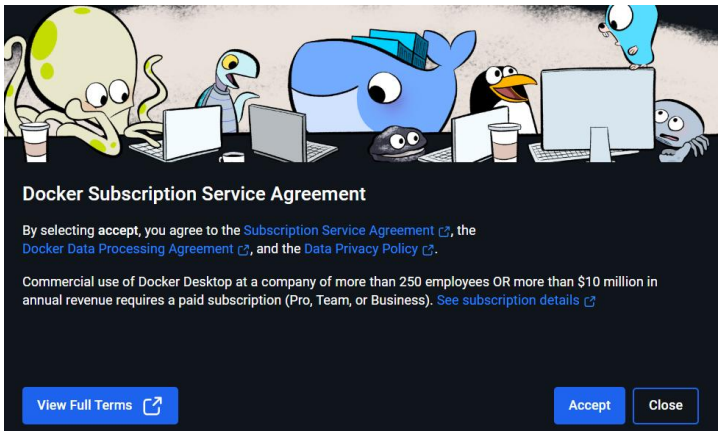

Fig. 3 Aceptar licencia de docker

A continuación, Docker Desktop ofrece la opción de registrarse mediante Gmail, GitHub o una cuenta de Docker. Este paso es opcional, pero se recomienda iniciar sesión para acceder a funciones adicionales y facilitar la gestión de contenedores.

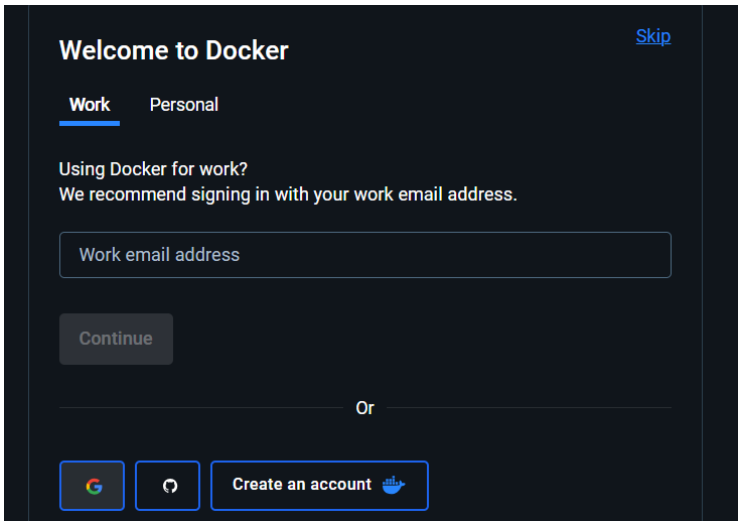

Fig. 4 Registrar dirección de correo electrónico

Después del registro opcional, se presentan algunas preguntas sobre el uso que se le dará a la herramienta. Una vez completado este paso, se podrá acceder al entorno de Docker Desktop y comenzar a gestionar contenedores.

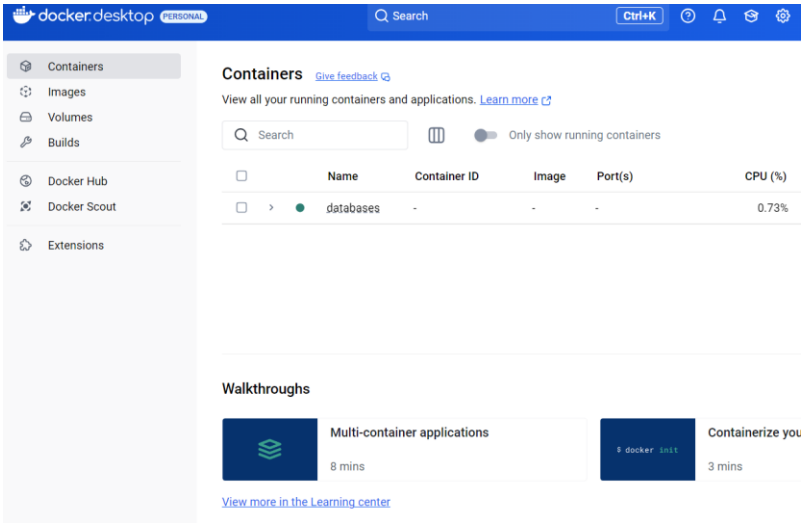

Fig. 5 Entorno de Docker Desktop

1.2.JAVA JDK 17

Para instalar Java JDK 17, debe acceder al sitio web oficial de Oracle y descargar el instalador correspondiente al sistema operativo.

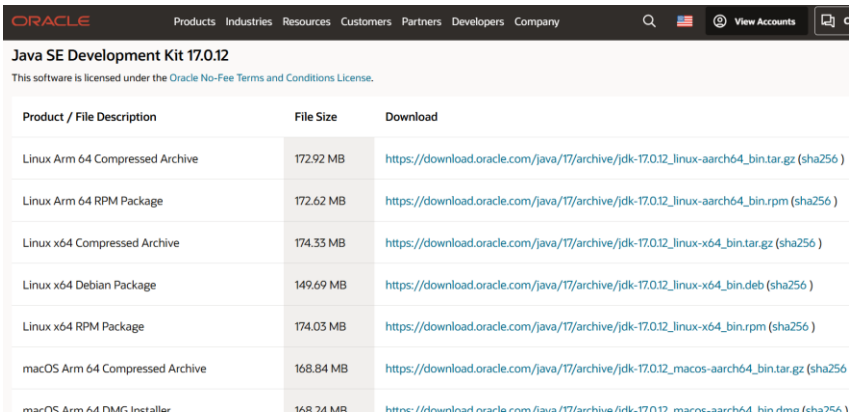

Fig. 6 Sitio oficial para la descarga del JDK 17

Una vez descargado el instalador, ejecute el archivo y seleccione la **ruta de destino** donde desea instalar **Java JDK 17**. Si no se desea cambiar, se puede dejar la ruta predeterminada. Luego haga clic en "Siguiente" y luego en "Instalar" para completar el proceso de instalación. Una vez finalizada la instalación, haga clic en "Close" para cerrar el instalador.

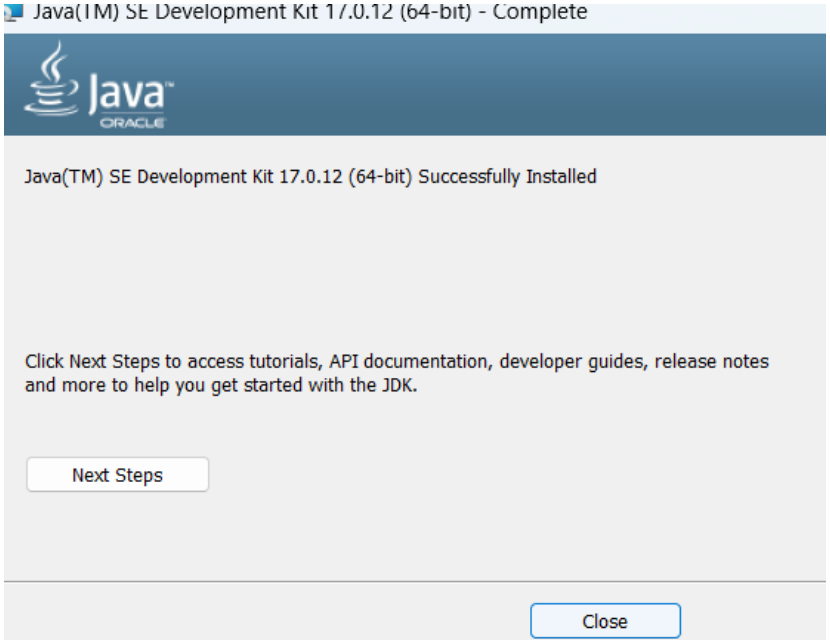

Fig. 7 Instalación de JDK

Para confirmar que Java JDK 17 se ha instalado correctamente, abra la terminal o símbolo del sistema y ejecute el siguiente comando:

```
java -version
```

Para evitar problemas de localización del JDK por parte de las aplicaciones, es necesario configurar la variable de entorno JAVA_HOME. Para ello se debe abrir el panel de control y dirigirse a la opción sistema, en la ventana emergente seleccionar la opción de editar variables

de entorno, esto le permitirá acceder a la sección donde puede añadir o editar las variables de entorno del sistema.

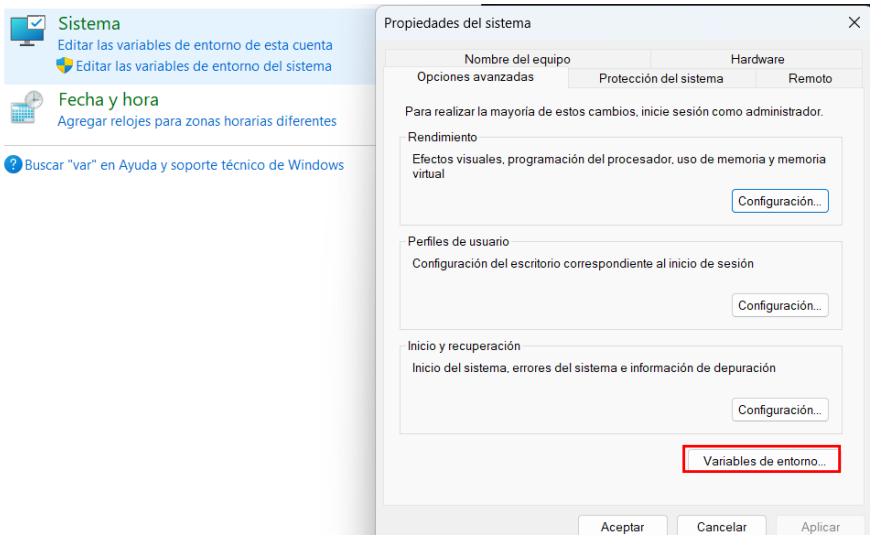

Fig. 8 Propiedades del sistema

Una vez dentro de las Variables de entorno, localice la sección Variables del sistema y seleccione la variable Path, luego haga clic en Editar, en la ventana de edición, haga clic en Nuevo y agregue la ruta del JDK 17, que generalmente se encuentra en '*C:\Program Files\Java\jdk-17\bin*'

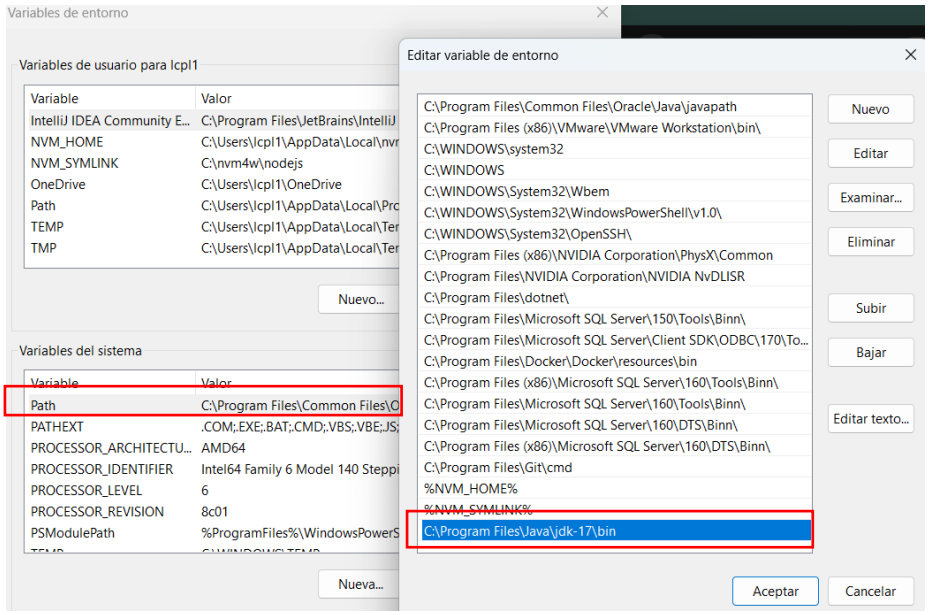

Fig. 9 Añadir la ruta del jdk al path

Luego, en la misma ventana de variables del sistema, haga clic en *Nueva* y en el campo *Nombre de la variable*, ingrese JAVA HOME; en el campo Valor de la variable, agregue la ruta donde se instaló el JDK 17, que generalmente es *C:\Program Files\Java\jdk-17*

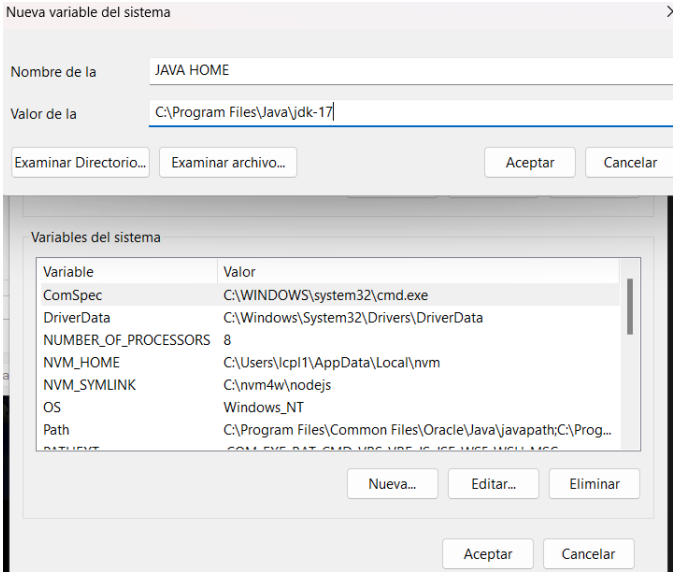

Fig. 10 Nueva variable del sistema JAVA HOME

1.3.Node Version Manager

Es recomendable tener instalado Node Version Manager (NVM) para gestionar múltiples versiones de Node.js en una misma computadora. Esto facilita el cambio entre diferentes versiones de Node según sea necesario para distintos proyectos. La descarga de NVM para Windows se puede realizar desde el siguiente repositorio de GitHub: https://github.com/coreybutler/nvm-windows/releases. Aunque no cuenta con un sitio oficial, es mantenido por colaboradores que actualizan regularmente el proyecto.

Fig. 11 Instalación de NVM desde github

Una vez descargado el instalador, ejecute el archivo y acepte los términos y condiciones de la licencia para continuar con la instalación, posteriormente seleccione la ruta de instalación donde desea que se instale Node Version Manager (NVM). Puede dejar la ruta predeterminada o elegir una personalizada.

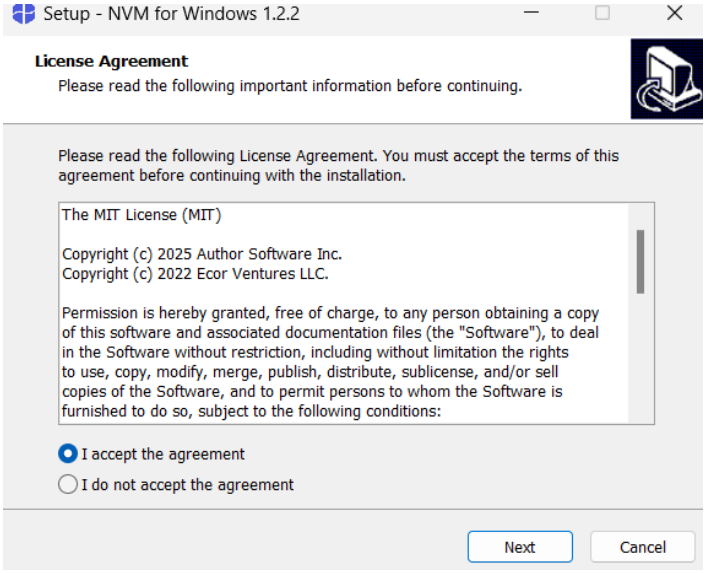

Fig. 12 Aceptación de términos y condiciones de NVM

Durante el proceso de instalación, se pedirá un correo electrónico. Este paso es opcional, por lo que puede dejar el campo en blanco si lo prefiere. Si no desea proporcionar un correo, simplemente puede aceptar la opción por defecto y continuar con la instalación hasta que se complete.

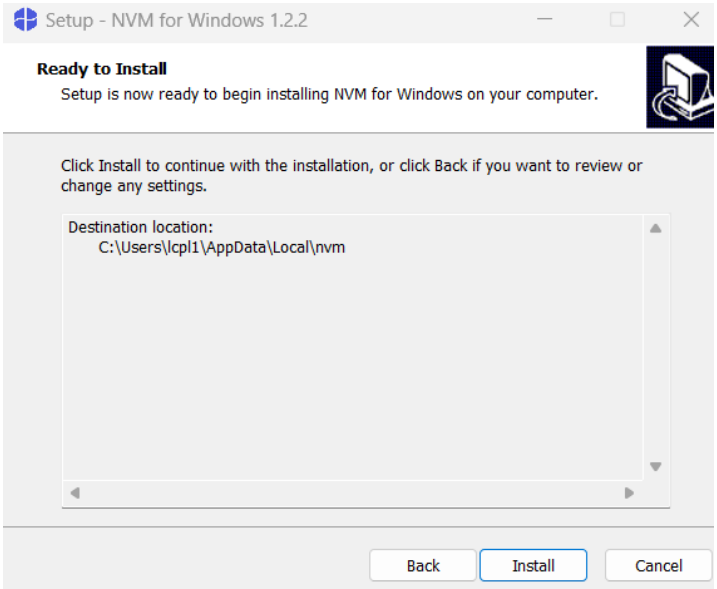

Fig. 13 Instalación de NVM para Windows

Una vez finalizada la instalación, abra la consola de comandos y verifique que Node Version Manager (NVM) se haya instalado correctamente ejecutando el siguiente comando:

```
nvm --versión
```

Para instalar Node.js versión 20.14.0, ejecute en la consola el comando:

```
nvm install 20.14.0
```

Una vez instalada, para utilizar esa versión de Node.js, ejecute:

```
nvm use 20.14.0
```

1.4.Instalar Angular CLI

Para instalar Angular CLI globalmente en su sistema, abra la consola de comandos y ejecute el comando:

```
npm install -g @angular/cli.
```

Una vez completada la instalación, verifique que Angular CLI se haya instalado correctamente ejecutando el comando

```
ng version
```

2. Estructura

El proyecto está estructurado en dos grandes partes: Backend (servidor) y Frontend (interfaz de usuario). Además, incluye configuraciones para Docker, documentación, y archivos esenciales para la gestión del proyecto con Git y Node.js.

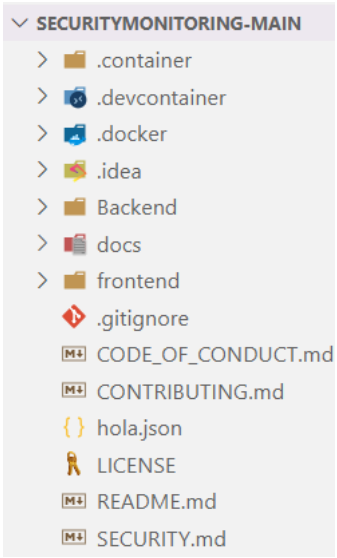

Fig. 14 Estructura general del proyecto

2.1.Front-End

La carpeta de esta sección contiene la estructura del código del cliente en este proyecto, desarrollado con Vite y Tailwind CSS. En la carpeta de node_modules se encuentran todas las dependencias instaladas. La carpeta public almacena archivos estáticos como imágenes, íconos y el archivo favicon.ico.

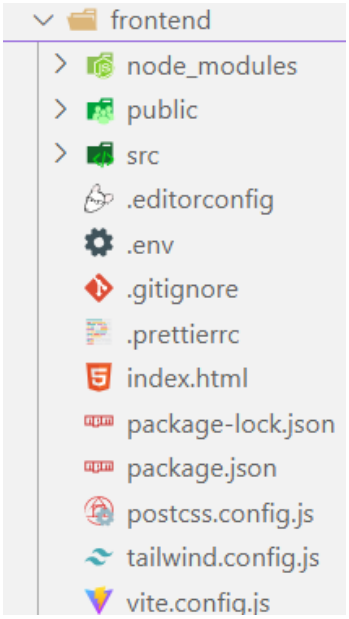

Fig. 15 Estructura del front-end

La carpeta **assets/css** centraliza los estilos de librerías y componentes utilizados en el frontend, permitiendo personalizar la apariencia y mejorar la experiencia de usuario.

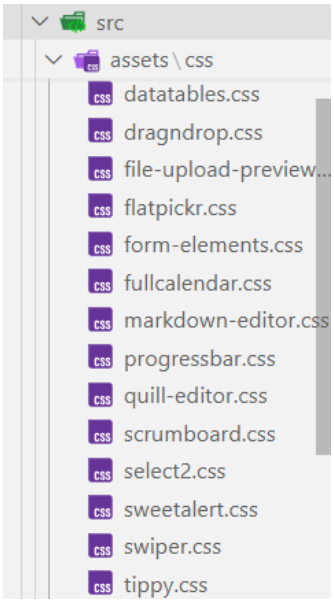

Fig. 16 Carpeta assets\css

En la carpeta de componentes se encuentran los diferentes componentes o módulos que forman parte del proyecto, como

- **Alerts:** Este directorio contiene los archivos relacionados con las alertas o notificaciones del sistema.
- **Icon:** Este archivo contiene los íconos o elementos gráficos utilizados en la aplicación.
- **Layouts:** Este directorio almacena los diferentes diseños o plantillas de la interfaz de usuario.
- **Switch:** Este es un componente funcional de React que representa un interruptor o switch. Recibe dos props: isOn y handleToggle.

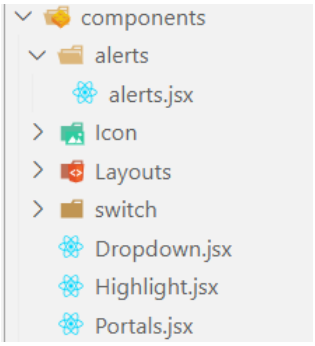

Fig. 17 Carpeta de Componentes

La carpeta **core** constituye la base del proyecto e incluye la carpeta **config**, donde se encuentra el archivo de configuración del entorno de la aplicación. Además, la carpeta **dto** almacena los objetos utilizados para la transferencia de datos entre las diferentes capas del sistema. Por otro lado, la carpeta **models** contiene los modelos de datos que estructuran la información de la aplicación.

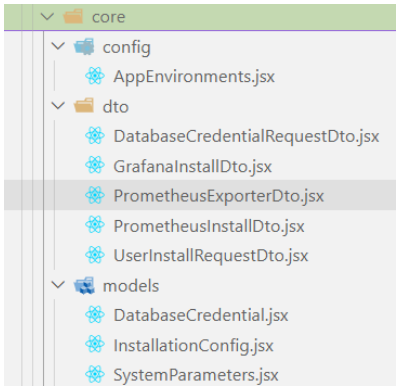

Fig. 18 Carpeta Core

El directorio pages organiza las principales vistas de la aplicación. Incluye subdirectorios como Activate (activación de cuentas), Authentication (gestión del inicio de sesión), password (cambio de contraseña), profile (información del usuario) y users (gestión de usuarios).

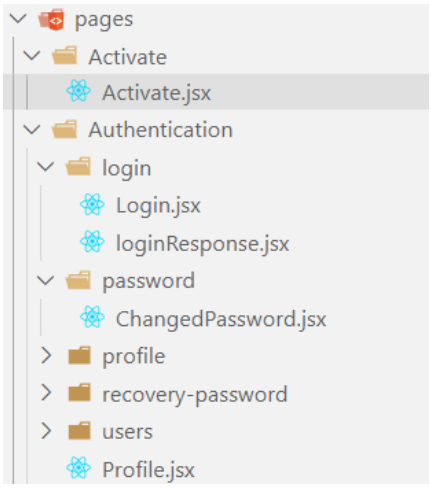

Fig. 19 Carpeta pages

La carpeta **routes** gestiona y configura las rutas de la aplicación, incluyendo la lógica de autenticación y autorización. La carpeta **store** administra el estado global de la aplicación, con un enfoque en la configuración del tema, permitiendo la personalización de apariencia y comportamiento.

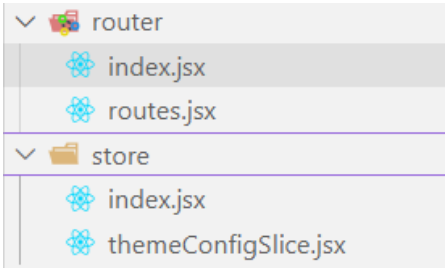

Fig. 20 Carpeta router

2.2.Backend

La carpeta contiene diversas carpetas y archivos clave como: **.idea**, que almacena configuraciones de IntelliJ IDEA, **.mvn** incluye archivos de Maven Wrapper, y **database** gestiona scripts de base de datos. **src** alberga el código fuente, mientras que **target** guarda los archivos compilados. Entre los archivos esenciales, **.env** define variables de entorno, y **Dockerfile** junto a **entrypoint.sh** configuran y ejecutan el contenedor Docker. **pom.xml** gestiona dependencias de Maven, y **README.md** proporciona documentación. Además,

External Libraries contiene bibliotecas externas, y **Scratches and Consoles** almacena archivos temporales y consolas de depuración.

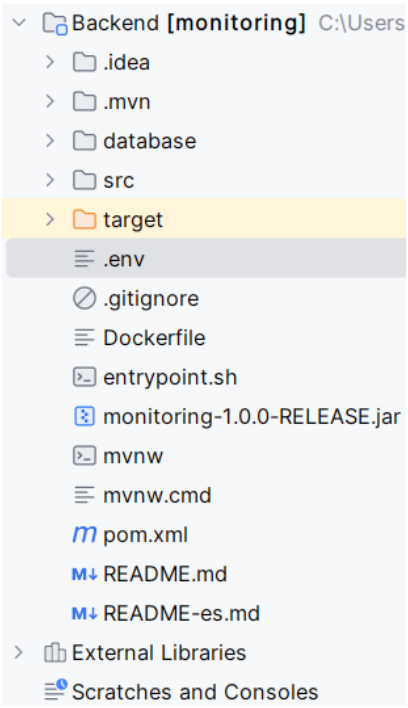

Fig. 21 Estructura general del backend

3. Instalación del Aplicativo

Para instalar la aplicación, acceda al repositorio denominado SecurityMonitoring y proceda con la descarga del proyecto.

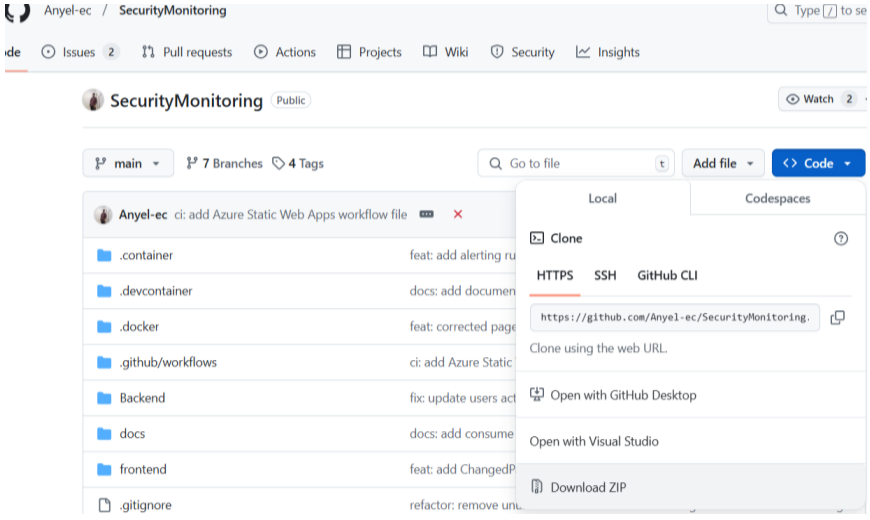

Fig. 22 Repositorio del aplicativo

3.1.Frontend

Una vez descargado el proyecto, abra la carpeta del repositorio en su editor de texto de preferencia. En este caso, puede utilizar Visual Studio Code para facilitar la edición y gestión del código. Para hacerlo, simplemente seleccione "Abrir carpeta" desde el menú de Visual Studio Code y navegue hasta la carpeta del proyecto.

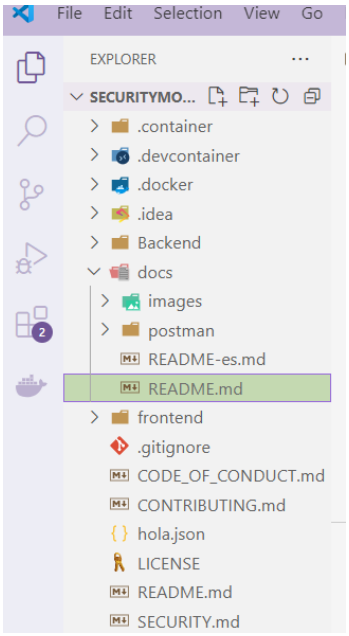

Fig. 23 Acceso al código fuente desde Visual Studio Code.

3.1.1. Arhivo .env

Dentro de la carpeta frontend se deberá configurar el archivo .env en el cual se definen las variables de entorno utilizadas en la configuración del frontend de la aplicación, incluyendo la URL base, el puerto de ejecución y datos específicos de la organización.

1. **Configuración de la Aplicación**
   - **VITE_BASE_URL**: Define la URL base de la API o backend con la que interactúa la aplicación frontend. En este caso, apunta a http://localhost:8080 durante el desarrollo.
   - **PORT**: Especifica el puerto en el que se ejecutará la aplicación frontend (3100).
2. **Configuración de la Identidad de la Aplicación**
   - **REACT_APP_ORG**: Define el nombre de la organización asociada a la aplicación (Security-Monitoring).

- **REACT_APP_NAME_APP**: Variable reservada para definir el nombre de la aplicación.

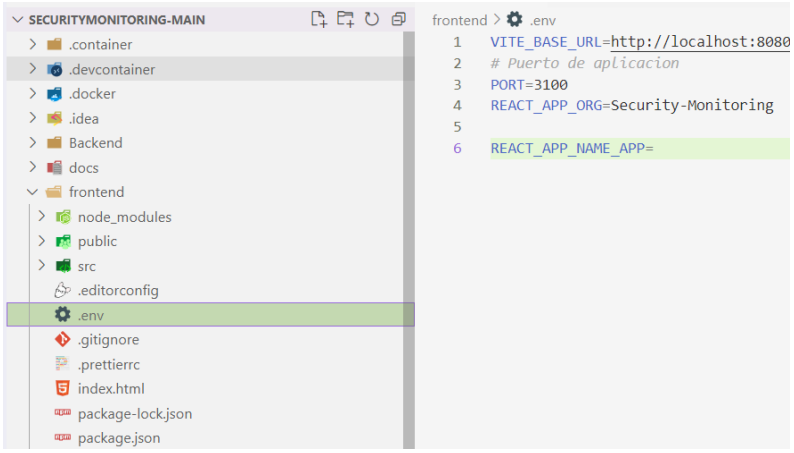

Fig. 24 Configuración de archivo .env del frontend

A continuación, navegue hasta la carpeta frontend dentro del proyecto y descargue todas las dependencias necesarias ejecutando el comando *npm i*. Este comando instalará todas las dependencias definidas en el archivo **package.json**, permitiendo que la aplicación esté lista para su ejecución.

Fig. 25 Instalar dependencias en el frontend

Finalmente, para ejecutar la aplicación, ejecute el comando *npm run dev* para poder acceder al aplicativo a través de su dirección IP local en el navegador.

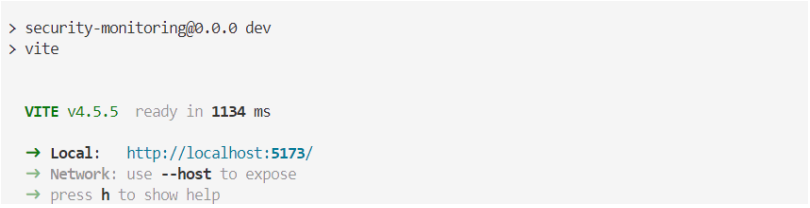

Fig. 26 Ejecución de la aplicación.

3.2.Configurar Docker para las bases de datos

En esta sección también se debe configurar el archivo .env ubicado en .docker/databases/.env que contiene variables de entorno utilizadas para definir credenciales de acceso, nombres de bases de datos y puertos para PostgreSQL, MariaDB y MongoDB.

Fig. 27 Configuración del archivo .env de las bd

Una vez configurado el archivo .env, proceda a levantar los contenedores, para ello, navegue hasta la carpeta donde se encuentra el archivo docker-compose.yml y ejecute el comando:

docker-compose up

3.3.Backend

Para acceder a la parte del backend, abra la carpeta backend del proyecto. En este caso, se utilizará IntelliJ IDEA como editor. Al abrir la carpeta, IntelliJ comenzará automáticamente a descargar las dependencias necesarias para el proyecto, según lo especificado en el archivo de configuración del backend (por ejemplo, pom.xml para proyectos Maven o build.gradle para proyectos Gradle).

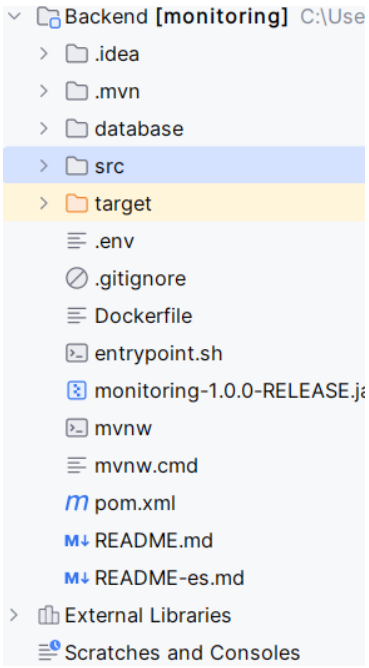

Fig. 28 Abrir el backend

3.3.1. Archivo .env

Para poder ejecutar el backend es necesario configurar el archivo .env, el cual contiene variables de entorno utilizadas para configurar distintos aspectos de la aplicación, incluyendo la base de datos, seguridad y alertas por correo electrónico.

1. **Configuración de la Base de Datos**
    - B**D_USERNAME y BD_PASSWORD**: Credenciales de acceso a la base de datos.
2. **Configuración de Seguridad**
    - **SECRET_KEY_AES**: Clave secreta para cifrado AES, que debe ser una cadena aleatoria de 32 bytes.
    - **SECRET_KEY_JWT**: Clave secreta utilizada para la generación y validación de tokens JWT.
    - **PUBLIC_KEY_RSA y PRIVATE_KEY_RSA**: Claves pública y privada en formato PEM para cifrado RSA.
    - **DEFAULT_EMPTY_PASSWORD_VALUE**: Valor predeterminado que se asigna cuando un campo de contraseña está vacío.
3. **Configuración de Alertas por Correo Electrónico (SMTP)**
    - **ALERT_SMTP_HOST**: Dirección y puerto del servidor SMTP utilizado para el envío de correos electrónicos.
    - **ALERT_SMTP_FROM**: Dirección de correo electrónico desde la cual se enviarán las alertas.
    - **ALERT_SMTP_USER y ALERT_SMTP_PASSWORD**: Credenciales de autenticación para el servidor SMTP.
    - **ALERT_SMTP_TO**: Dirección de correo electrónico que recibirá las alertas generadas por la aplicación.

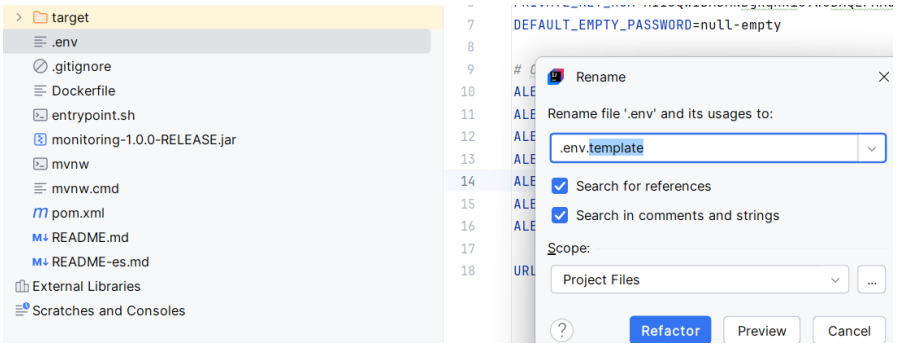

Fig. 29 Cambiar nombre de .env.template solo a .env

3.3.2. Java JDK

Para ejecutar el backend, es necesario configurar el JDK en la versión 17. Configure el entorno de desarrollo para utilizar esta versión.

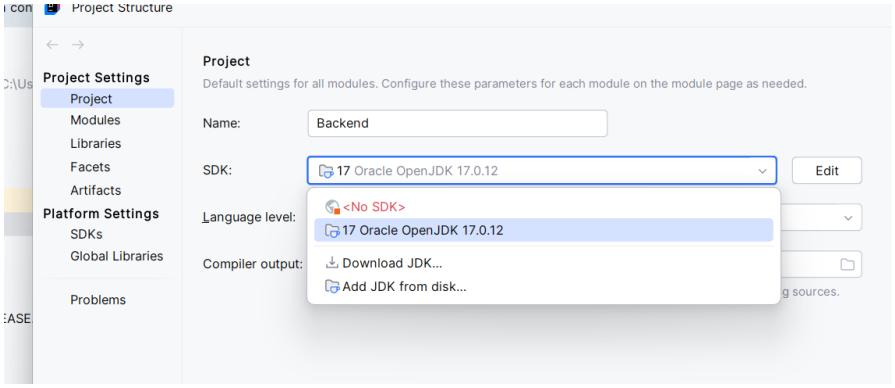

Fig. 30 Añadir JDK correspondiente

4. Ejecución de la Aplicación

Para ejecutar la aplicación, es necesario iniciar tanto el frontend como el backend. Para el frontend, simplemente navegue a la carpeta **frontend** y ejecute el siguiente comando: ***npm run dev*** y después debe dirigirse a la dirección local proporcionada.

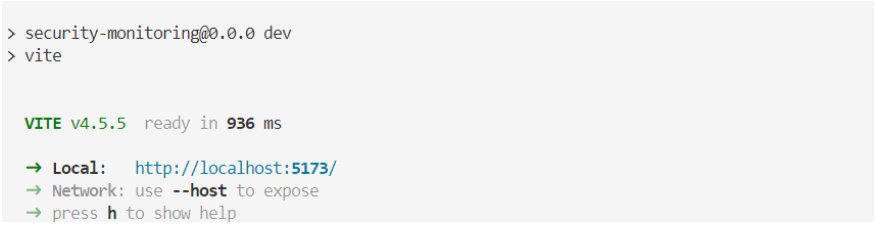

Fig. 31 Ejecución del frontend

En el caso del backend, debe dirigirse a la carpeta **backend** y dentro de ella, buscar y correr el archivo ejecutable **BackendApplication.java**.

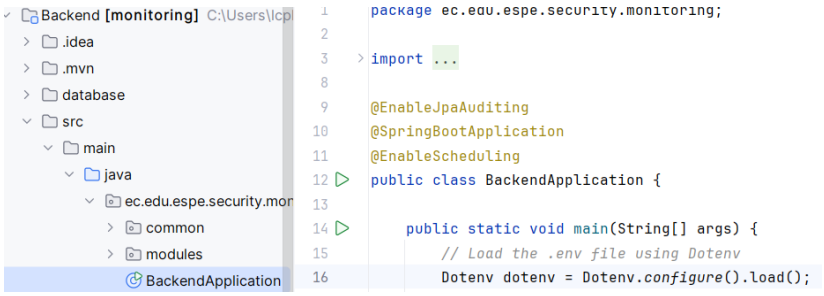

Fig. 32 Ejecución del backend

Con estos pasos, tendrá acceso total y exitoso a la herramienta de monitoreo ingresando a través de la dirección IP local proporcionada en el frontend y aprovechar las funcionalidades disponibles en el lado del backend.

Fig. 33 Security Monitoring en ejecución