



Sapresis

CONFIGURACIÓN DEL ENTORNO DE DESARROLLO Y PRUEBAS DE LA APLICACIÓN SAPRESIS

Evidencia: GA8-220501096-AA1-EV02

ABSTRACT

Se documenta el ambiente de desarrollo necesario para la ejecución de las pruebas unitarias de la aplicación Sapresis.

Mauricio Alberto Monroy Calle

Tecnología en Análisis y Desarrollo de Software - 2721455



3/10/2024





Contenido

1	Introducción	2
2	Requisitos de instalación.....	3
3	Requisitos del Sistema	4
4	Información sobre la versión y dependencias técnicas	5
4.1	Versión de la aplicación.....	5
4.2	Versión del JDK y Maven.....	5
4.3	Dependencias principales.....	5
4.4	Entorno de pruebas	6
5	Configuración del entorno de pruebas	7
5.1	Descripción del ambiente de ejecución.....	7
5.2	Ejecución y pruebas recomendadas	7





1 Introducción

El presente documento tiene como objetivo proporcionar las pautas y los lineamientos necesarios para configurar el entorno de desarrollo y de pruebas de la aplicación Sapresis. En él se detallan las instrucciones de instalación, los requisitos del sistema, las configuraciones específicas del entorno y las herramientas recomendadas para ejecutar las pruebas de manera eficaz.

Este documento está dirigido tanto a desarrolladores como a otras personas interesadas en la implementación y el mantenimiento de la aplicación, y ofrece una guía técnica clara para garantizar la correcta ejecución de las pruebas y la funcionalidad del sistema. Además, sirve como complemento del Acta de Aplicación de Pruebas, en la que se describen los resultados obtenidos durante el proceso de pruebas de la aplicación.



2 Requisitos de instalación

Para una óptima aplicación de las pruebas, se recomienda el uso de las siguientes herramientas:

- **IDE:** Se puede utilizar cualquier entorno de desarrollo compatible con Java, aunque se sugiere **IntelliJ IDEA** o **Apache NetBeans** debido a su integración fluida con Maven y Spring Boot. Estas herramientas proporcionan un entorno robusto para el desarrollo y la depuración de aplicaciones Java.
- **Base de datos:** La aplicación requiere acceso a una base de datos **MySQL**. Esto se puede configurar de tres maneras:
 - **Local:** instalar MySQL en la máquina de desarrollo.
 - **Servidor externo:** conectar la aplicación a una instancia de MySQL alojada en un servidor remoto.
 - **Docker:** usando una imagen de MySQL para Docker, lo que permite una gestión más sencilla del entorno sin necesidad de instalación directa en el sistema. Se recomienda configurar las credenciales de acceso y el puerto según las necesidades del proyecto.

Además, es importante asegurarse de que la configuración de la base de datos en el archivo `application.properties` o `application.yml` esté correctamente ajustada, especificando los detalles de la conexión (URL, usuario, contraseña, y cualquier parámetro adicional de configuración).





3 Requisitos del Sistema

JDK: es necesario contar con la versión 11 o superior del **Java Development Kit (JDK)** para ejecutar la aplicación y sus pruebas. Sin embargo, para un rendimiento óptimo y la mejor compatibilidad con las últimas características de Java, se recomienda utilizar las versiones **21** o **23**, que incluyen mejoras de rendimiento y seguridad.

Maven: es indispensable tener **Apache Maven** instalado para gestionar las dependencias del proyecto y ejecutar las pruebas localmente. Maven facilita la automatización de la compilación, empaquetado y prueba del código. Se recomienda asegurarse de tener la variable de entorno `MAVEN_HOME` correctamente configurada y comprobar que el comando `mvn` esté disponible en la terminal.



4 Información sobre la versión y dependencias técnicas

4.1 Versión de la aplicación

La versión actual de **Sapresis** utilizada para las pruebas es la **v1.0.0**. Esta versión incluye todas las funcionalidades descritas en la documentación del proyecto, incluyendo la autenticación basada en JWT, las operaciones CRUD de las entidades principales y las configuraciones de seguridad de Spring Security.

4.2 Versión del JDK y Maven

- **JDK:** La aplicación se ha desarrollado y probado utilizando la versión **11** del JDK, pero es compatible con versiones superiores. Se recomienda utilizar la versión más reciente del JDK (por ejemplo, **21** o **23**) para asegurar un rendimiento óptimo y compatibilidad con nuevas características.
- **Maven:** Para la gestión de dependencias y la ejecución de pruebas, se ha utilizado **Apache Maven** en su versión **3.9.7**. El proyecto está configurado para generar los informes de prueba y otros artefactos utilizando los plugins de Maven.

4.3 Dependencias principales

La configuración de las dependencias del proyecto se gestiona a través del archivo `pom.xml` de **Maven**. Algunas de las dependencias más relevantes son:

- **Spring Boot:** 3.3.2 (Framework principal)
- **JUnit 5:** Para la ejecución de pruebas unitarias.
- **Spring Security:** Para la gestión de la autenticación y autorización.
- **MySQL Connector:** Para la conexión con la base de datos MySQL.





- **Lombok:** Para reducir el código repetitivo en las clases (anotaciones como `@Getter`, `@Setter`, etc.).

El archivo `pom.xml` del repositorio de la aplicación contiene la lista completa de dependencias y sus versiones exactas.

4.4 Entorno de pruebas

El entorno de pruebas se ha configurado con las siguientes versiones de software:

- **Sistema operativo:** Se recomienda **Windows 10** o superior, **Ubuntu 20.04** o superior, o cualquier distribución de Linux compatible con **Java**.
- **Base de datos:** Se ha utilizado **MySQL 8.0.34** para la base de datos de prueba, pero es posible utilizar otras versiones compatibles con el conector **MySQL Connector Java**.



5 Configuración del entorno de pruebas

5.1 Descripción del ambiente de ejecución

Para la ejecución de las pruebas de **Sapresis**, se ha optado por utilizar un entorno local. Aunque la aplicación también se ha desplegado en un servidor externo, este presenta limitaciones importantes, ya que se trata de un servicio gratuito con recursos limitados, como una baja capacidad de procesamiento y memoria. Estas limitaciones provocan tiempos de respuesta elevados entre peticiones y, por tanto, dificultan la fluidez de las pruebas. Por ello, el entorno local se presenta como la opción óptima para realizar las pruebas de manera eficiente y aprovechar al máximo los recursos disponibles del sistema en el que se ejecutan.

5.2 Ejecución y pruebas recomendadas

La ejecución de las pruebas de **Sapresis** puede llevarse a cabo utilizando diferentes herramientas y métodos. A continuación, se detallan las opciones más recomendadas, tanto para futuros desarrolladores y para verificar el correcto funcionamiento en entornos de desarrollo:

- **Ejecución desde un IDE (IntelliJ o NetBeans).** La opción más sencilla para llevar a cabo las pruebas es a través de un entorno de desarrollo integrado (IDE), como **IntelliJ IDEA** o **Apache NetBeans**. Estos IDEs proporcionan una configuración predefinida para ejecutar aplicaciones basadas en Java, lo que facilita la gestión de perfiles específicos y la ejecución de pruebas automatizadas.

En **IntelliJ**, por ejemplo, se puede ejecutar la aplicación directamente, lo que permite visualizar los resultados detallados de las pruebas y *logs* a través de la consola de salida. Además, este IDE se puede integrar con **Maven**, lo que facilita la ejecución de comandos como `mvn test` para realizar pruebas unitarias o de integración.

- **Ejecución desde la terminal (Maven).** Para aquellos que prefieran ejecutar las pruebas desde una terminal, **Maven** es la herramienta clave. Mediante comandos como `mvn test`, es posible





realizar pruebas unitarias y generar informes detallados. Los *logs* y los resultados se mostrarán directamente en la terminal o en archivos generados por **Maven**. Este método es ideal para entornos de desarrollo más controlados o automatizados.

- **Pruebas complementarias con Postman.** Además de las pruebas unitarias y de servicio realizadas en el entorno de desarrollo, se recomienda el uso de **Postman** para la validación de los controladores y las peticiones HTTP en la API de Sapresis. Las pruebas en **Postman** permiten verificar que las peticiones CRUD (Crear, Leer, Actualizar y Eliminar) funcionan correctamente y que el sistema responde como se espera ante diferentes escenarios. La integración de pruebas con **Postman** también es útil para simular flujos de usuario y comprobar la interacción con el sistema en su totalidad.
- **Informes y resultados de las pruebas.** Los resultados de las pruebas ejecutadas se pueden complementar con la documentación generada por **Maven**, utilizando el comando `mvn site`, que genera un informe detallado sobre las pruebas realizadas, el estado del proyecto, y otros datos relevantes. Estos informes incluyen detalles y estructura de las dependencias usadas en la prueba.

Para acceder a la documentación de la API y replicar las pruebas mencionadas, se puede consultar el enlace a la documentación pública de [Sapresis](#) o revisar el archivo adjunto en el repositorio del proyecto de [GitHub](#) en la sección de [Documentación de la API](#). También se puede acceder a la documentación generada por Maven en el siguiente enlace: [Project Documentation](#).

