

Instalación de SonarQube

Valery Chumpitaz Caycho

Manual de Instalación de SonarQube y Explicación de sus Partes

1. Requisitos previos

Antes de instalar **SonarQube**, asegúrate de tener lo siguiente:

- **Java:** SonarQube requiere Java 11 o superior. Puedes verificar si tienes Java instalado ejecutando el siguiente comando en la terminal:

```
java -version
```

- **Base de datos:** SonarQube usa una base de datos para almacenar los datos. Por defecto, utiliza **H2** (una base de datos en memoria), pero para entornos de producción es recomendable usar bases de datos como **PostgreSQL**, **MySQL**, o **Oracle**.

2. Instalación de SonarQube

Opción 1: Instalación con Docker (Recomendado para entornos locales)

Si tienes **Docker** instalado, esta es una forma rápida de configurar SonarQube.

Abre la terminal o el **Command Prompt** y ejecuta el siguiente comando para descargar e iniciar SonarQube en un contenedor:

```
docker run -d -p 9000:9000 --name sonarqube sonarqube:lts
```

Este comando descarga la versión **LTS (Long-Term Support)** de SonarQube y lo expone en el puerto 9000.

Abre tu navegador y dirígete a:

<http://localhost:9000>

La interfaz de SonarQube debería estar disponible en tu navegador.

La primera vez que accedas a SonarQube, utiliza las credenciales predeterminadas:

Usuario: admin
Contraseña: admin

Opción 2: Instalación manual (sin Docker)

1. Descarga la última versión de SonarQube desde su página oficial:
<https://www.sonarqube.org/downloads/>
2. Extrae el archivo descargado y navega hasta la carpeta extraída.
3. Para iniciar SonarQube, ejecuta el siguiente comando (dependiendo de tu sistema operativo):
 - **En Windows:**
Ejecuta el archivo **StartSonar.bat** en la carpeta **bin/windows-x86-64/**.

En Linux/macOS:

Ejecuta el siguiente comando en la terminal

```
./bin/linux-x86-64/sonar.sh start
```

Accede a la interfaz web de SonarQube desde tu navegador:

<http://localhost:9000>

3. Configuración básica de SonarQube

Crear un Proyecto en SonarQube

Una vez que SonarQube esté en funcionamiento, sigue estos pasos para crear un proyecto y analizarlo:

1. En el dashboard de SonarQube, haz clic en "**Create new project**" (Crear nuevo proyecto).
2. Asigna un **nombre** y un **key** (clave) al proyecto. El **key** es un identificador único para el proyecto, por ejemplo: `com.example:sonar-analysis-demo`.
3. Genera un **token** de acceso para autenticar tu análisis:
 - En el menú superior derecho, haz clic en tu nombre de usuario y selecciona "**My Account**".
 - Ve a la pestaña "**Security**" y genera un nuevo token de acceso.

- Ahora, puedes ejecutar el análisis de SonarQube en tu proyecto usando Maven o Gradle, o cualquier otra herramienta de integración.

Ejemplo de Análisis usando Maven

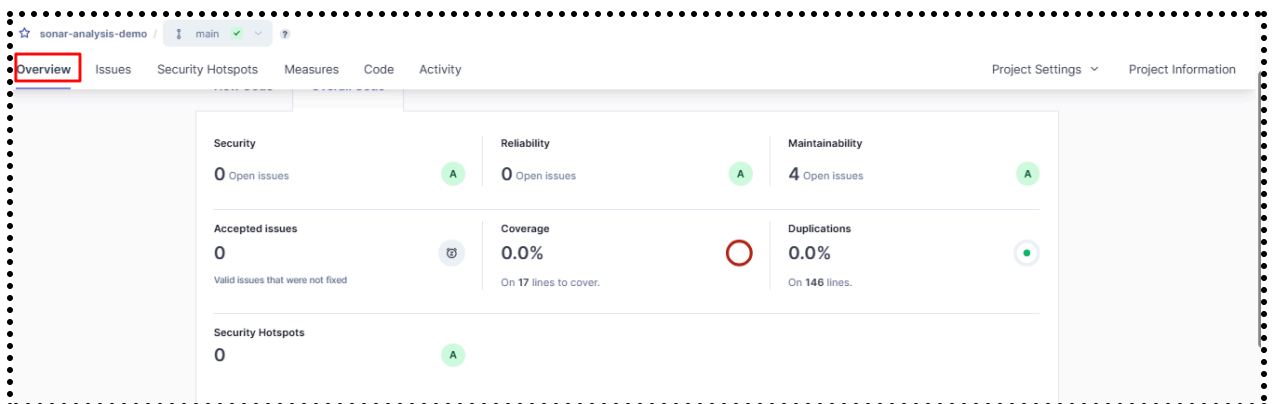
Asegúrate de tener el plugin de SonarQube configurado en tu proyecto Maven. Luego, ejecuta el siguiente comando en la terminal de tu proyecto:

```
mvn sonar:sonar -Dsonar.host.url=http://localhost:9000  
-Dsonar.token=YOUR_TOKEN
```

4. Explicación de las Partes de SonarQube

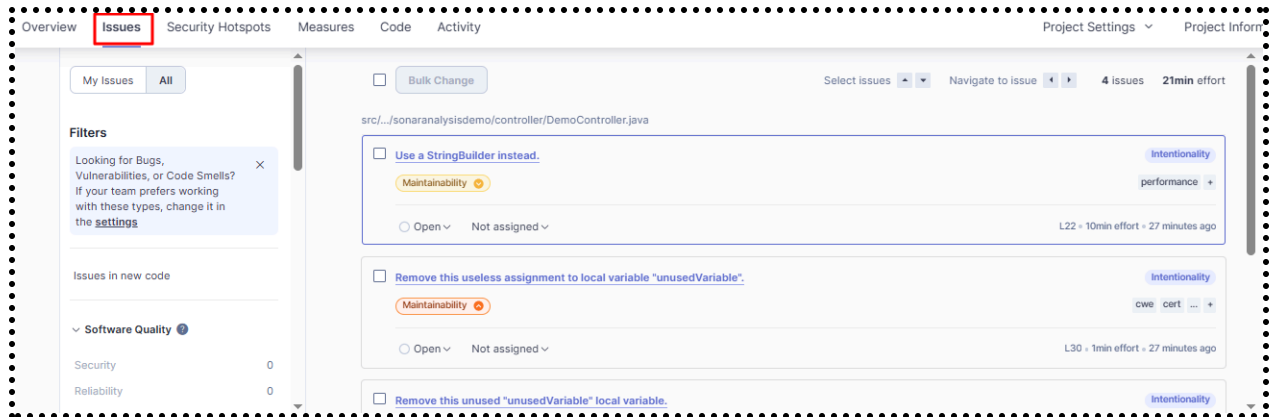
→ Overview (Visión general)

- ◆ Esta sección te proporciona un resumen general del estado de calidad del proyecto. Muestra los indicadores clave como el número de **issues** (problemas), la **cobertura de pruebas**, la **duplicación de código**, el **código sin tests**, etc. Es un panorama rápido que te permite conocer de un vistazo la salud de tu código y cómo se comporta en relación con las métricas de calidad definidas en SonarQube.



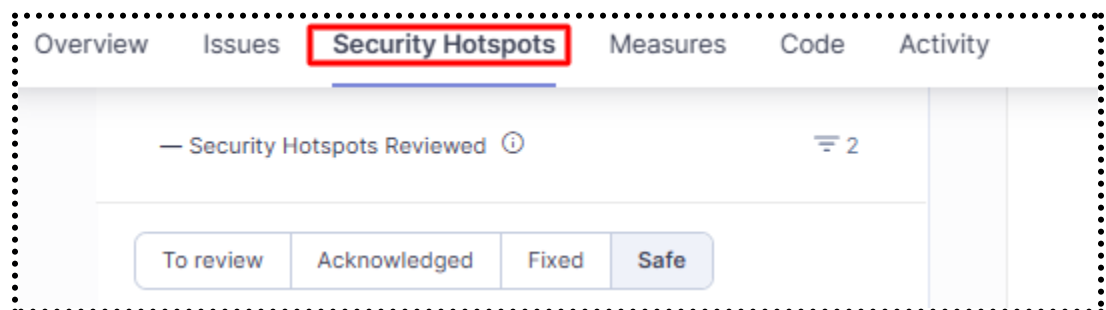
→ Issues (Problemas)

- ◆ Aquí se listan todos los problemas identificados en el código, clasificados por tipos (bugs, vulnerabilidades, code smells, etc.). Los issues pueden ser configurados según su severidad, permitiéndote priorizar las correcciones. SonarQube los categoriza y muestra información sobre cada uno, como el archivo donde se encuentra el problema y las líneas de código afectadas. Puedes filtrar los problemas por tipo, severidad y estado.



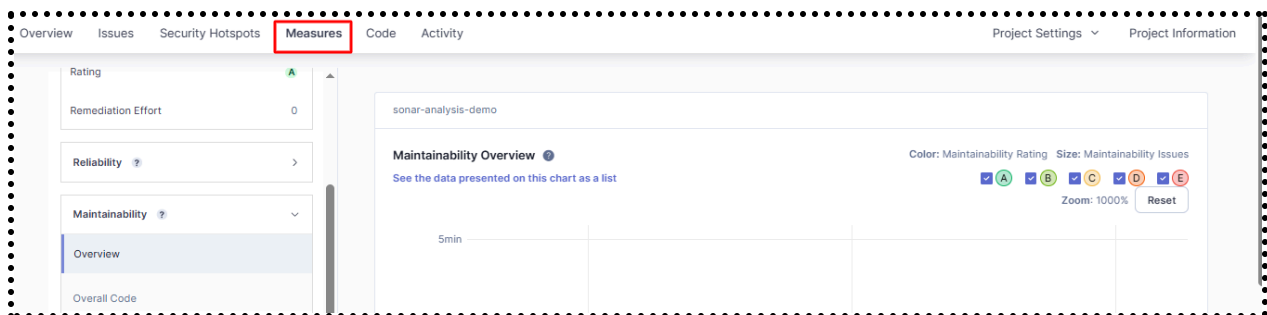
→ Security Hotspots (Puntos críticos de seguridad)

- ◆ Esta sección está destinada a identificar posibles vulnerabilidades de seguridad en el código. SonarQube analiza el código y marca aquellas áreas que podrían ser potencialmente peligrosas si no se manejan correctamente. Los Security Hotspots no son necesariamente errores, pero requieren revisión manual para verificar si realmente representan un riesgo de seguridad. Son importantes para asegurar que el código sea seguro frente a posibles ataques o explotaciones.



→ Measures (Medidas)

- ◆ Las Measures son métricas cuantitativas que se usan para evaluar la calidad del código. Incluye medidas como complejidad ciclomática, cobertura de pruebas, duplicación de código, número de líneas de código, entre otras. Estas métricas te permiten entender la estructura del código y cómo se comporta en términos de legibilidad, mantenimiento y pruebas. SonarQube también muestra tendencias a lo largo del tiempo, lo que te permite observar si la calidad del código está mejorando o empeorando.



→ Code (Código)

- ◆ En esta sección puedes ver el código fuente del proyecto y los detalles de los problemas que se han encontrado en él. SonarQube te permite navegar por las clases, métodos y líneas del código y muestra los issues directamente en el contexto del código. Además, puedes ver las sugerencias de refactorización y buenas prácticas para mejorar la calidad del código. Esto te ayuda a realizar correcciones específicas directamente en el código fuente.

The screenshot shows the SonarQube interface with the 'Code' tab selected. The main area displays a table of code quality metrics for the project 'sonar-analysis-demo'. The table has columns for Lines of Code, Security, Reliability, Maintainability, Security Hotspots, Coverage, and Duplications. The data is as follows:

	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
sonar-analysis-demo	119	0	0	4	0	0.0%	0.0%
src	48	0	0	4	0	0.0%	0.0%
pom.xml	71	0	0	0	0	—	0.0%

→ Activity (Actividad)

- ◆ Esta sección muestra el historial de los análisis realizados sobre el proyecto. Puedes ver cómo ha evolucionado la calidad del código con el tiempo, los problemas que se han resuelto y los nuevos problemas que han aparecido. Es útil para mantener un seguimiento sobre el progreso en términos de calidad y también para identificar patrones de calidad a lo largo del tiempo. Además, muestra las contribuciones del equipo, lo que permite ver la interacción de diferentes desarrolladores con el proyecto.

