

Fábrica - Análisis de Código Fuente

Esta sección esta dedicada a la definición del proceso de análisis de código fuente que se debe implementar en los proyectos de las aplicaciones de la compañía. Para realizar este proceso se ha elegido la tecnología Sonar la cual se divide en los siguientes componentes:

SonarQube

SonarQube® es una herramienta de revisión automática de código para detectar errores, vulnerabilidades y olores de código en su código. Puede integrarse con su flujo de trabajo existente para permitir la inspección continua del código en las ramas de su proyecto y solicitudes de extracción.¹⁾

SonarLint

SonarLint® es una extensión IDE gratuita y de código abierto que identifica y le ayuda a solucionar problemas de calidad y seguridad mientras codifica. Como un corrector ortográfico, SonarLint garabatea fallas y proporciona comentarios en tiempo real y una guía clara de corrección para entregar un código limpio desde el principio.²⁾

Integrar SonarQube/SonarLint en los proyectos de la fábrica

Consideraciones previas

- En el alcance inicial solo se soporta el lenguaje java
- Solo se soportan los gestores de dependencias Maven³⁾, Gradle⁴⁾
- Antes de subir un análisis al servidor SonarQube debe existir el proyecto. Este debe ser solicitado al administrador carlos.torres.

Nota

La configuración de la integración SonarQube/SonarLint se realizará en entorno window 10 con maven 3.x

Configuración Global del repositorio

Lo primero que se debe hacer es ir a las configuraciones globales del gestor de dependencias en el usuario en sesión del sistema operativo. En el caso de window esta ruta por lo general se encuentra en C:\Users\Usuario\.m2\wrapper\dists\apache-maven[Versión Maven instalada]\[directorio hash del empaquetado]\apache-maven-[Versión maven instalada]\conf**settings.xml**. El archivo que debemos

ubicar es el archivo **settings.xml** y en el debemos realizar la siguiente configuración:

Sección pluginGroups

Ubique la sección pluginGroups y copie el siguiente texto como se muestra en el siguiente código de ejemplo

```
<pluginGroups>
  <!-- pluginGroup
    | Specifies a further group identifier to use for plugin lookup.
  <pluginGroup>com.your.plugins</pluginGroup>
  -->
  <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>
</pluginGroups>
```

Nota

Si existen más plugins registrado agregue la línea

<pluginGroup>org.sonarsource.scanner.maven</pluginGroup> antes del final de la sección **</pluginGroups>**

Sección profiles

Ubique la sección **profiles** y al final copie el siguiente perfil como se muestra en el siguiente código de ejemplo

```
<profile>
  <id>sonar</id>
  <activation>
    <activeByDefault>true</activeByDefault>
  </activation>
  <properties>
    <!-- Optional URL to server. Default value is http://localhost:9000 -->
    <sonar.host.url>
      http://10.1.40.171:9000
    </sonar.host.url>
  </properties>
</profile>
```

Para finalizar guarde el archivo⁵⁾ y con eso ya estará configurado el maven para lanzar análisis en el servidor SonarQube.

Instalar SonarLint en el IDE

Se asume que el usuario tiene el IDE Eclipse o Spring Tools Suite. Para instalar el SonarLint siga los siguientes pasos:

1. Abrir el IDE
2. Seleccionar el workspace del proyecto que se analizará
3. Ir al menú **Help**→**Eclipse MarketPlace**
4. Ingresar en el campo **Find** el texto *sonarlint*, aparece en el listado una final con el nombre **SonarLint 7.1** o posterior
5. Click en el botón **Install**

Conectar SonarLint con el Servidor SonarQube: Primera Vez, se crea la conexión

Para conectar el SonarLint con el Servidor SonarQube siga los siguientes pasos:

1. Abrir el IDE
2. Seleccionar el workspace del proyecto que se analizará
3. Click derecho sobre la raíz del proyecto que se analizará.
4. Seleccionar en el menú **SonarLint** la opción **Bind to SonarQube or SonarCloud**
5. Se pide que se cree una nueva conexión. Seleccionamos **SonarQube** y click en el botón **Siguiente**.
6. En el siguiente panel ingrese la **URL** del servidor SonarQube la cual es: **<http://10.1.40.171:9000>** y click en el botón **Siguiente**.
7. En el siguiente panel seleccione la opción **Token** y click en el botón **Siguiente**.
8. En el siguiente panel ingrese el **token** que será utilizado en el análisis y click en el botón **Siguiente**. Esta información debe ser suministrada por el líder del equipo. No genere un nuevo token ya que ese análisis no será considerado en la línea base del proyecto.
9. Por último el proceso hace un test de la conexión y si todo es correcto pasa al panel donde se configura el nombre de la conexión se recomienda dar el siguiente nombre **SonarQubeServer** ya que esa será la conexión que se utilizará en los posteriores análisis, click en el botón **Siguiente**.
10. En el siguiente panel active las notificaciones del servidor y click en el botón **Siguiente** y con eso se finaliza la configuración de la conexión.
11. En el siguiente paso, se selecciona el proyecto al cual se le realizará el análisis, click en el botón **Siguiente**.
12. Al llegar a este paso previamente debe existir el proyecto en el servidor SonarQube, de esta forma el sistema automáticamente lo selecciona y con eso se finaliza el proceso de configuración.

Conectar SonarLint con el Servidor SonarQube: Conexión previa

Para conectar el SonarLint con el Servidor SonarQube siga los siguientes pasos:

1. Abrir el IDE
2. Seleccionar el workspace del proyecto que se analizará
3. Click derecho sobre la raíz del proyecto que se analizará.

4. Seleccionar en el menú **SonarLint** la opción **Bind to SonarQube or SonarCloud**
5. Seleccione la conexión existente **SonarQubeServer** creada la primera vez y click en el botón **Siguiente**.
6. En el siguiente paso, se selecciona el proyecto en el servidor SonarQube al cual se le realizará el análisis debe existir el proyecto. sino existe se debe solicitar al administrador.
7. Por último click en Finalizar para terminar la conexión del proyecto local al proyecto del SonarQube.

Lanzar un análisis de código fuente local

Para lanzar un análisis local del SonarLint siga los siguientes pasos:

1. Abrir el IDE
2. Seleccionar el workspace del proyecto que se analizará
3. Click derecho sobre la raíz del proyecto que se analizará.
4. Seleccionar en el menú **SonarLint** la opción **Analyze** para analizar todo el proyecto o **Analyze changed files** para analizar sólo los archivos modificados.
5. Dar click en **Aceptar** o **Ok** y el SonarLint iniciará el análisis local.
6. Al finalizar se visualizará una pestaña **SonarLint Report** en la zona inferior del IDE con los resultados.

Lanzar un análisis de código fuente centralizado

Antes de lanzar un análisis de código fuente centralizado debe solicitar el comando de configuración del proyecto SonarQube este comando debe ser solicitado al líder de equipo de desarrollo y este lo obtendrá del administrador al momento de solicitar la creación de un nuevo proyecto de análisis de código fuente.

1. Abrir el IDE
2. Seleccionar el workspace del proyecto que se analizará
3. Click derecho sobre la raíz del proyecto que se analizará.
4. Seleccionar en el menú **Run As...** la opción **Maven Build...** para desplegar la ventana de configuración de lanzamiento del proyecto.
5. Ingresar el comando de configuración del proyecto en el campo Goals. Es cual es similar al siguiente: ***clean verify sonar:sonar -Dsonar.projectKey=[nombre del proyecto] -Dsonar.host.url=http://10.1.40.171:9000 -Dsonar.login=[token del equipo de desarrollo]***
6. Si todo ha sido registrado correctamente se inicia el análisis centralizado del código fuente del proyecto. Al finalizar el análisis será cargado en el servidor y podrá ser consultado.

[←Volver atras](#)

1)
<https://docs.sonarqube.org/latest/>

2)
<https://www.sonarlint.org/>

3)
<https://maven.apache.org/>

4)

<https://gradle.org/>

5)

En algunas configuraciones solicitará usuario administrador

From:

<http://wiki.adacsc.co/> - **Wiki**

Permanent link:

<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:sourcecodeanalyzer&rev=1637086244>

Last update: **2021/11/16 18:10**

