

# 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.<sup>1)</sup>

## 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.<sup>2)</sup>

## 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 Maven<sup>3)</sup>, Gradle<sup>4)</sup>
- 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 archivo<sup>5)</sup> 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**

Siga los pasos de instalación del asistente y al finalizar reinicie el IDE, al volver a iniciar el IDE al dar click derecho sobre la raíz de cualquier proyecto saldrá un nuevo menú **SonarLint** lo cuál indica que el proceso se realizó correctamente.

## Lanzar un análisis de código fuente

## Estructuras de Código Fuente

Las siguientes estructuras de código fuente que deben ser documentadas:

- Clases
- Métodos de Clases
- Ventanas
- Funciones y Eventos de Ventanas
- Formularios y Vista Web (html)
- Paquetes de base de datos
- Funciones y Procedimientos de bases de datos

## Tipos de documentación

Se definen los siguientes tipos de documentación:

- **Interna:** Utilizada en el código fuente de las aplicaciones de la compañía. Ver [Estructuras de Código Fuente](#). Esta documentación se caracteriza porque está interna en el código fuente y puede ser consultada directamente consultando las estructuras de código o utilizando una aplicación de generación de documentación técnica según el lenguaje de documentación.
- **Externa:** Utilizada en el código fuente de las aplicaciones de la compañía del proceso de migración. Esta documentación es generada complementaria a la documentación interna. Este proceso ya está definido en la siguiente sección: [Documentación Externa de Microservicios](#)

## Estructura de documentación

## Plantilla Base

A continuación se define la Plantilla Base de documentación que debe ser implementada en las [Estructuras de Código Fuente](#) de las aplicaciones de la compañía

```
/**  
  
Título  
  
Descripción  
  
Retorno  
  
Argumentos  
  
Modo de Uso  
  
Autor  
  
Fecha  
  
Deprecated  
  
Throws  
  
Revision History  
  
Copyright Año ADA, Inc. All rights reserved.  
*/
```

### Consideraciones

- Los nombres de los campos pueden variar según el lenguaje de programación.

### Definición de Columnas

#### Título

Nombre de la [Estructura de Código Fuente](#) que se documentará.

#### Descripción

Describe el objetivo de la [Estructura de Código Fuente](#). Esta descripción debe ser clara y concisa.

## Retorno

Describe el valor o valores de salida de un método. No aplica para definiciones de clases.

## Argumentos

Describe los parámetros de un método.

## Modo de Uso

Este campo se utiliza para presentar un ejemplo de la forma de invocar o ejecutar la [Estructura de Código Fuente](#).

## Autor

Indica quién escribió la [Estructura de Código Fuente](#) que se refiere el comentario. Si son varias personas se escriben los nombres separados por comas. El nombre del autor debe ser escrito según el correo del usuario de la compañía con la siguiente plantilla:

**nombre.apellido@ada.co** Ejemplo: carlos.torres@ada.co el cual es otorgado al empleado al ingresar.

Si es un autor externo se debe registrar el nombre completo más el correo personal.

**Nombres Apellidos, correo@empresa.com** Ejemplo: Pepe Perez, pepe.perez@gmail.com

## Fecha

Este campo registra la fecha inicial de creación de la [Estructura de Código Fuente](#)

## Deprecated

Indica que la [Estructura de Código Fuente](#) es antigua y que no se recomienda su uso porque posiblemente desaparecerá en versiones posteriores o será reemplazada.

## Throws

Sólo aplica para métodos. Describe las excepciones que pueden ser lanzadas por el método.

## Revision History

Esta sección de documentación es utilizada para registrar las evoluciones de la [Estructura de Código Fuente](#). Las evoluciones de código fuente se debe registrar con la siguiente plantilla:

Número de Revisión - Autor evolución - Fecha : Descripción - Autor revision par - Fecha revision par

Ejemplo:

1.0 - carlos.torres@ada.co - 27/07/2021 08:39:40 : Initial version. - edixson.matos@ada.co - 27/07/2021 14:00:05

### Copyright [Año] ADA, Inc. All rights reserved

Este campo se utiliza para registrar los derechos reservados de la [Estructura de Código Fuente](#) se debe diligenciar la columna **Año** según la vigencia de creación.

### Alcance de la plantilla

A continuación se presenta una matriz de alcance de la plantilla base de documentación según la [Estructura de Código Fuente](#)

	Clase	Método Clase	Ventana	Evento/Función Ventana	Formulario/Vista Web	Paquete DB	Funcion/Procedure DB
<b>Título</b>	X	X	X	X	X	X	X
<b>Descripción</b>	X	X	X	X	X	X	X
<b>Retorno</b>		X		X			X
<b>Argumentos</b>		X		X			X
<b>Modo de Uso</b>	X	X	X	X	X	X	X
<b>Autor</b>	X	X	X	X	X	X	X
<b>Fecha</b>	X	X	X	X	X	X	X
<b>Deprecated</b>	X	X	X	X	X	X	X
<b>Throws</b>		X		X			X
<b>Revision History</b>	X	X	X	X	X	X	X
<b>Copyright</b>	X	X	X	X	X	X	X

### Herramientas para generar documentación interna

Según el lenguaje de programación se definen algunas de las herramientas que ayudan a simplificar la generación de la documentación del código fuente:

Lenguaje	Herramienta de Generación de Plantilla	Herramienta para generación de Reporte de Documentación Técnica
Java	<a href="#">Javadoc</a>	<a href="#">Javadoc</a>
Powerbuilder	<a href="#">Autohotkey</a>	<a href="#">PBDoc</a>
Base de Datos	<a href="#">Autohotkey</a>	Manual

[←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=1637081327>

Last update: **2021/11/16 16:48**

