

# ¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.  
 Ese apoyo que siempre quiso tener...

## 1. Desarrollo de componentes y proyectos a medida



## 2. Auditoría de código y recomendaciones de mejora

## 3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



## 4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,  
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)  
 Gestor de contenidos (Alfresco)  
 Aplicaciones híbridas

Tareas programadas (Quartz)  
 Gestor documental (Alfresco)  
 Inversión de control (Spring)

Control de autenticación y  
 acceso (Spring Security)  
 UDDI  
 Web Services  
 Rest Services  
 Social SSO  
 SSO (Cas)

JPA-Hibernate, MyBatis  
 Motor de búsqueda empresarial (Solr)  
 ETL (Talend)

Dirección de Proyectos Informáticos.  
 Metodologías ágiles  
 Patrones de diseño  
 TDD

BPM (jBPM o Bonita)  
 Generación de informes (JasperReport)  
 ESB (Open ESB)



» Estás en: [Inicio](#) [Tutoriales](#) [Sonar Runner: Analizar proyectos sin Maven en cualquier lenguaje](#)



Rubén Aguilera Díaz-Heredero

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática, especialidad en Ingeniería del Software

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)



## Catálogo de servicios Autentia



Síguenos a través de:



## Últimas Noticias

» ¡¡¡Terrakas 1x04 recién salido del horno!!!

» Estreno Terrakas 1x04: "Terraka por un día"

» Nuevos cursos de gestión de la configuración en IOS y Android

» La regla del Boy Scout y la Oxidación del Software

» Autentia conquista los Alpes

[Histórico de noticias](#)

## Últimos Tutoriales

» [Talend. Lectura y tratamiento de base de datos Mysql.](#)

» [Lectura y tratamiento de ficheros XML con Talend](#)

» [Desplegando una aplicación en Cloud Foundry con Maven](#)

» [Primeros pasos en Android \(II\)](#)

» [Eclipse Juno, la versión 4.2 de Eclipse](#)

## Últimos Tutoriales del

Fecha de publicación del tutorial: 2012-09-12

Tutorial visitado 1 veces [Descargar en PDF](#)

# Sonar Runner: Analizar proyectos sin Maven en cualquier lenguaje

## 0. Índice de contenidos.

- 1. Entorno
- 2. Introducción
- 3. Configuración inicial
- 4. Analizando el código Java de la aplicación
- 5. Analizando el código Javascript de la aplicación
- 6. Conclusiones

## 1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Mac Book Pro 17" (2,6 Ghz Intel Core i7, 8 GB DDR3)
- Sistema Operativo: Mac OS X Snow Leopard 10.6.4
- Sonar 3.2
- Sonar Runner 1.4
- Javascript Plugin 1.0

## 2. Introducción

En adictos tenemos numerosos tutoriales que hablan de Sonar. Una magnífica herramienta que nos da una idea de la calidad de nuestra aplicación haciendo un análisis estático del código Java para proyecto con Maven. Hasta ahora si querías analizar el código Java de un proyecto sin Maven, tenías que "mavenizar" el proyecto como se mostraba en este [tutorial](#).

Ahora con Sonar Runner no solo podemos analizar el código de proyecto sin Maven si no que además podemos analizar código que no sea exclusivamente Java, como por ejemplo, código Javascript e incluso Cobol.

[Aquí](#) podéis encontrar el listado completo de lenguajes soporta actualmente en forma de plugins, alguno de ellos comerciales.

## 3. Configuración inicial

Este tutorial vamos a ver los pasos que hay que dar para poder analizar el código Javascript y Java de un proyecto sin Maven. Lo primero que tenemos que hacer es arrancar una instancia de sonar. En este tutorial os explicamos como hacerlo.

Una vez arrancada la instancia de Sonar en local, que por defecto será accesible desde la URL <http://localhost:9000>, es momento de descargar Sonar Runner de esta [URL](#) donde también se os muestra como instalarlo y configurarlo.

Los pasos a seguir son sencillo. Descomprimos el .zip en algún directorio de nuestro sistema de archivos y establecemos la variable de sistema SONAR\_RUNNER\_HOME apuntando a este directorio. Luego actualizamos el PATH del sistema añadiendo SONAR\_RUNNER\_HOME/bin.

En caso de no tener la configuración por defecto de Sonar, deberemos editar el fichero SONAR\_RUNNER\_HOME/conf/sonar-runner.properties y modificar las variables en función de nuestra instalación de Sonar. Para el caso de un Sonar que este instalado en un servidor llamado "preproduccion" con una base de datos MySQL y corriendo en el puerto 9000 esta sería la configuración de las variables.

```
view plain print ?
01. #---- Default Sonar server
02. sonar.host.url=http://preproduccion:9000
03.
04. #---- PostgreSQL
05. #sonar.jdbc.url=jdbc:postgresql://localhost/sonar
06. #sonar.jdbc.driver=org.postgresql.Driver
```

```

07.
08. #---- MySQL
09. sonar.jdbc.url=jdbc:mysql://preproduccion:3306/sonar?useUnicode=true&characterEncoding=utf8
10. sonar.jdbc.driver=com.mysql.jdbc.Driver
11.
12. #---- Oracle
13. #sonar.jdbc.url=jdbc:oracle:thin:@localhost/XE
14. #sonar.jdbc.driver=oracle.jdbc.driver.OracleDriver
15.
16. #---- Global database settings
17. sonar.jdbc.username=root
18. sonar.jdbc.password=admin
19.
20. #---- Default directory layout
21. #sources=src/main/java
22. #tests=src/test/java
23. #binaries=target/classes
24.
25. sonar.sourceEncoding=UTF-8
26.

```

## Autor

- » Desplegando una aplicación en Cloud Foundry con Maven
- » Database MessageSource: obtener los literales de una base de datos
- » Primeros pasos con Cloud Foundry
- » Balanceando la carga de MySQL con HAProxy
- » Montar un cluster de MySQL

## 4. Analizando el código Java de la aplicación

Con la configuración anterior ya estamos en disposición de poder analizar código Java de proyectos que no tienen que estar gestionados con Maven. Para ello, en la raíz del proyecto vamos a crear un fichero que obligatoriamente se tiene que llamar sonar-project.properties con el siguiente contenido:

```

view plain print ?
01. #Se puede equiparar con el groupId de Maven (Obligatorio)
02. sonar.projectKey=my:project
03. #Se puede equiparar con el artifactId de Maven. Es el nombre que se va a mostrar el dashboard de Sonar
04. sonar.projectName=My project
05. #Se puede equiparar a la versión de Maven. (Obligatorio)
06. sonar.projectVersion=1.0
07.
08. #Indicamos todos los directorios donde haya código Java. (Obligatorio)
09. sources=srcDir1,srcDir2
10.
11. #Indicamos todos los directorios donde haya tests (Opcional)
12. tests=testDir1,testDir2
13.
14. #Indicamos los directorios donde se almacenan los binarios del proyecto (Opcional)
15. binaries=binDir
16.
17. #Indicamos la ruta de las librerías y directorios que contengan .class (Opcional)
18. libraries=path/to/library.jar,path/to/classes/dir
19.
20.
21. # Indicamos el lenguaje que queremos analizar (Por defecto, Java)
22. #sonar.language=cobol
23.
24. # Podemos indicar otros parámetros
25. my.property=value

```

Basta con abrir un terminal, situarnos en el directorio raíz del proyecto y ejecutar "sonar-runner". Una vez finalizado el proceso podremos ver nuestro proyecto analizado en el dashboard de nuestro Sonar.

## 5. Analizando el código Javascript de la aplicación

Muchos proyectos, Maven o no, contienen mucho código Javascript que podemos analizar con Sonar. Para ello hay que instalar el plugin "Javascript Plugin". Esto se puede realizar desde el "Update Center" de nuestro Sonar.

Para ello nos tenemos que logar como administradores, haciendo click en "Login" e introduciendo el usuario "admin" y la contraseña "admin" que son las credenciales del usuario administrador que viene por defecto en la instalación de Sonar.

Acto seguido hacemos click en "Configuration" y después en "Update Center"; con lo que el sistema nos mostrará una pantalla con el listado de plugins instalados. En la pestaña "Available Plugins" podemos instalar el plugin para código Javascript.

## Últimas ofertas de empleo

2011-09-08

 Comercial - Ventas - MADRID.

2011-09-03

 Comercial - Ventas - VALENCIA.

2011-08-19

 Comercial - Compras - ALICANTE.

2011-07-12

 Otras Sin catalogar - MADRID.

2011-07-06

 Otras Sin catalogar - LUGO.

Home Configuration Administrator » Log out

Quality Profiles  
My Profile  
My Filters  
Manual Metrics  
Manual Rules  
Default Dashboards

**SECURITY**  
Users  
Groups  
System Administrators  
Roles

**SYSTEM**  
General Settings  
Backup  
Bulk Deletion  
System Info  
**Update Center**

Update Center

Installed Plugins Available Plugins Plugin Updates System Updates

Additional Languages

<b>ABAP</b>	Enable analysis and reporting on ABAP projects
<b>C</b>	Enable analysis and reporting on C projects.
<b>C++</b>	Enable analysis and reporting on C++ projects.
<b>Cobol</b>	Enable analysis and reporting on Cobol projects.
<b>Delphi</b>	Enables analysis of Delphi projects into Sonar
<b>Drools</b>	Analyze Drools files.
<b>Flex</b>	Enables analysis of ActionScript projects into Sonar.
<b>Groovy</b>	Enables analysis of Groovy projects into Sonar.
<b>JavaScript</b>	Enables analysis of JavaScript projects into Sonar. License: GNU LGPL 3 Author: Eriks Nukis Links: <a href="#">Homepage</a> <a href="#">Issue Tracker</a> Version: 1.0 (Jun 7, 2012) <input type="button" value="Install"/>
<b>Natural</b>	Enable analysis and reporting on Natural projects.
<b>PHP</b>	Sonar PHP Plugin is set of tools that brings PHP support to sonar. It relies on Depend, PHPMD, PHP_CodeSniffer and PHPUnit
<b>PL/SQL</b>	Enable analysis and reporting on PL/SQL projects.
<b>Python</b>	Enable analysis and reporting on python projects.
<b>Sonar C++ Plugin</b>	C++ Plugin for Sonar
<b>Visual Basic</b>	Enable analysis and reporting on Visual Basic projects.
<b>Web</b>	Analyze web code.
<b>Xml</b>	Analyze XML and XHTML documents.

Una vez instalado el plugin, tenemos que reiniciar Sonar para poder trabajar con él. Ahora en el directorio raíz de nuestro proyecto, renombramos el fichero anterior a "sonar-project-java.properties" y creamos una copia de ese fichero a la que nombramos "sonar-project-js.properties" estableciendo el parámetro "sonar.language=js", indicando los directorios que contienen código Javascript en nuestra aplicación ("sources") y cambiando el "sonar.projectName" ya que se va a mostrar como un proyecto independiente del anterior.

Ahora para poder seleccionar uno u otro fichero de propiedades tenemos que ejecutar el Sonar Runner con el parámetro -Dproject.settings. Por tanto, ejecutando "sonar-runner -Dproject.settings=sonar-project-js.properties" arrancamos el análisis del código Javascript, finalizado el proceso podemos ver el resultado en el dashboard de nuestro Sonar y podremos ver las métricas y las violaciones que ha detectado.

## 6. Conclusiones

Sonar es una herramienta que ha sabido evolucionar y de una manera sencilla e intuitiva nos permite analizar proyectos de cualquier tipo soportando la mayoría de lenguajes actuales.

Cualquier duda o sugerencia en la zona de comentarios.

Saludos.

### A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

### Por favor, vota +1 o compártelo si te pareció interesante

Share |

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» [Regístrate](#) y accede a esta y otras ventajas «



Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

IMPULSA

Impulsores

Comunidad

¿Ayuda?

----- 0 personas han traído clicks a esta página  
sin clicks + + + + + + + +

powered by [karmacray](#)

Copyright 2003-2012 © All Rights Reserved | Texto legal y condiciones de uso | Banners | Powered by Autentia | Contacto

W3C XHTML 1.0

W3C CSS

XML RSS

XML RTDM