

Herramientas de Calidad del Producto Software y Documentación. SonarQube, Maven y Doxygen.

Francisco Javier García González
alu0101030562@ull.edu.es

6. Configure algunos de los plugins soportados por Maven. Comente que permiten los mismos.

Una vez generado nuestro fichero pom para trabajar en el entorno Maven, instalaremos plugins para mejorar nuestro proyecto. Para ello, instalaremos los plugins *Checkstyle* para el correcto uso de guías de estilo y desarrollo de código. *PMD* y *CMD* que es una herramienta de análisis de código para generar un reporte con los resultados de nuestro código. *JXR* (Source Xref y Test Source Xref) para generar informes con líneas de código específicas que si se usan con el plugin PMD hace referencia a errores encontrados en el código. *Surefire Report* analiza los test y los representa mediante DOXIA. *Jdepend* genera un reporte con las métricas del código. Con *TagList* generamos una lista con las etiquetas que hayamos puesto en nuestro código. Con *JavaDoc* generamos documentación de API en formato HTML a partir de nuestro código y con *Dependency-check* tendremos más seguridad en nuestro código.

7. Con la ayuda del Maven y el SonarQube haga un análisis del proyecto. Realice las configuraciones que estime necesarias para evaluar el proyecto.

Una vez puestos los plugins en el fichero POM y habiéndolo comprobado en MAVEN pasaremos nuestro proyecto a SonarQube para tener un análisis de nuestro código. Una vez en SonarQube podemos ver como nuestro proyecto tiene 1 Security Hotspots, 6 bugs y más de 100 Code Smells.

8. Identifique en el proyecto los problemas principales de seguridad. Solucione alguno de los proyectos de seguridad del proyecto.

El único error de seguridad que nos encontramos es el de generar un numero aleatorio y, para arreglar esto, haremos lo siguiente:

Lo primero será hacer un import en nuestro fichero java de la clase `SecureRandom`.

```
import java.security.SecureRandom;
```

Ahora, crearemos un objeto que genere un número aleatorio mediante la clase `SecureRandom` llamada `aleatorio` y la declararemos donde queremos generar ese número.

```
public int aleatorySelectionRCL(int maxTRCL) {  
    SecureRandom aleatorio = new SecureRandom();  
    int low = 0;  
    int high = maxTRCL;  
    int posSelected = aleatorio.nextInt(high-low) + low;  
    return posSelected;  
}
```

9. Documente el proyecto con la ayuda de Doxygen. En la documentación del código fuente debe incluir elementos de descripción abreviada y elementos de descripción detallada. Donde se destaquen las clases, métodos principales de su código y otros elementos principales del desarrollo software. Empleé si lo considera necesario el asistente que posee la herramienta 2 Doxygen para facilitar la configuración y/o mantenimiento de la generación de la documentación del proyecto.

A la hora de usar Doxygen, el documento HTML con la información generada por el Doxygen se encuentra en la carpeta [ExpositoTOP/Documentation/html/index.html](#). Ahí, podemos ver todas las clases con sus métodos con una pequeña descripción de la función que hace y los parámetros que tiene además de lo que retorna en caso de retornar algo.