

Análisis de ABGS en SonarQube

Leidy Katherine Calderón Castaño

Camilo Andres Tiria Corredor

SENA

Análisis y Desarrollo de Sistemas de información

Bogotá D.C

3/12/2021



Copyright © 2021

Calle 49ª Bis sur # 10d -20

Absence and Bad Grades Software
Todos los Derechos Reservados

320 447 3192
313 710 3188

Análisis de ABGS en SonarQube

Aprendices:

Leidy Katherine Calderón Castaño

Camilo Andres Tiria Corredor

Instructor:

Graciela Arias

SENA

Análisis y Desarrollo de Sistemas de información

Bogotá D.C

3/12/2021



Copyright © 2021

Calle 49ª Bis sur # 10d -20

Absence and Bad Grades Software
Todos los Derechos Reservados

320 447 3192
313 710 3188



INDICE

Introducción.....	4
¿Qué es SonarQube?	4
¿Por qué usar SonarQube?	4
Información que analiza SonarQube:.....	4
• Bugs y Vulnerabilities:	4
• Code Smells:	4
Instalación de SonarQube:	5
Análisis de ABGS:	11
Conclusión:	12
Bibliografías:	12



Introducción

En este documento, se llevará a cabo el paso a paso de la instalación de SonarQube, así como el manejo de la misma. Se realizará una prueba del código de ABGS, el cual usa un lenguaje de PHP, HTML, Java y estilos CSS, las pruebas se harán en PHP y CSS.

¿Qué es SonarQube?

En la industria de software, uno de los factores más importantes a tener en cuenta es la calidad del código, por lo que es necesario conocer y disponer de herramientas que brinden retroalimentación del estado de nuestro código y la forma en que podríamos mejorarlo, es allí donde SonarQube (o SonarCloud) cumple un rol protagónico dentro del proceso de desarrollo de software, específicamente en integración continua.

¿Por qué usar SonarQube?

Destaca la capacidad de identificar aspectos tales como: código duplicado, código muerto, estándares de codificación, complejidad ciclomática, comentarios, test unitarios y test de integración. De esta manera, nos guía hacia una mejor manera de escribir nuestro código y desarrollar Software.

Información que analiza SonarQube:

- Bugs y Vulnerabilities:

Hacen referencia tanto a puntos de fallo reales o potenciales en el software, como a puntos débiles de seguridad que pueden ser usados como foco de un ataque.

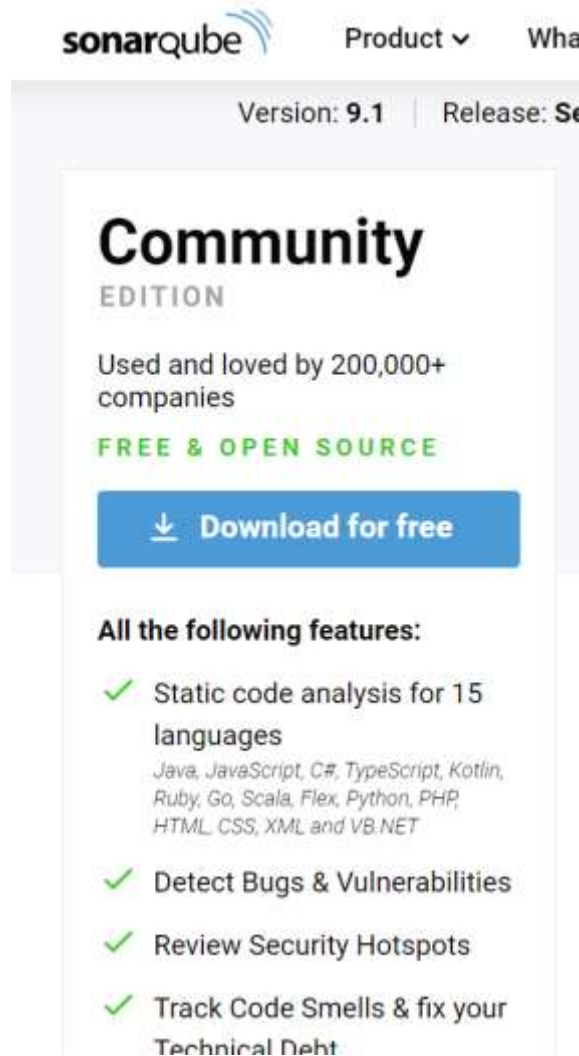
- Code Smells:

Son una serie de síntomas en el código que nos vienen a indicar que tal vez no se están haciendo las cosas de una forma del todo correcta, lo que puede llevar a que haya algún problema a futuro y un problema de trasfondo.

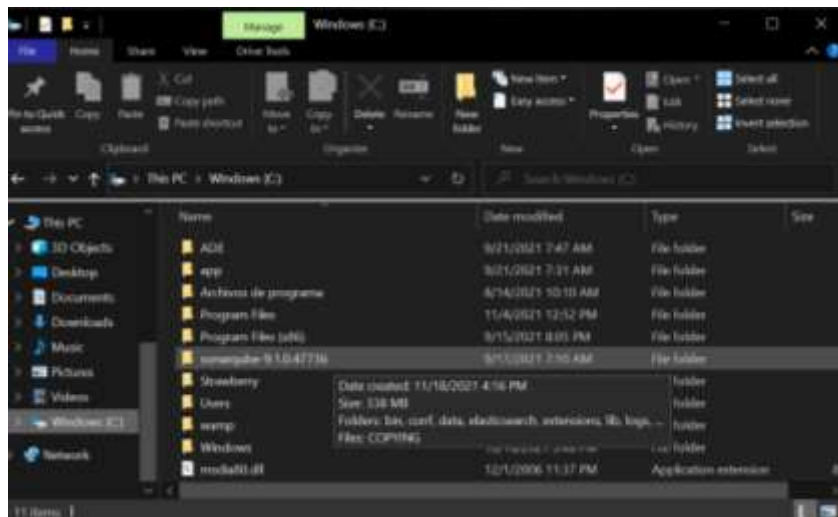
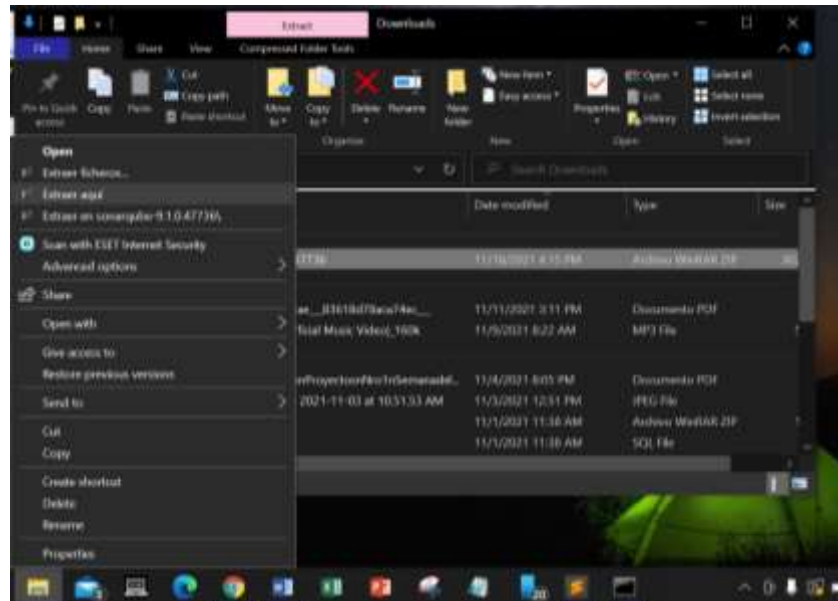


Instalación de SonarQube:

Nos dirigimos a la página oficial de SonarQube y descargamos la edición "Community".



Extraemos los ficheros en el Disco local C.



Copyright © 2021

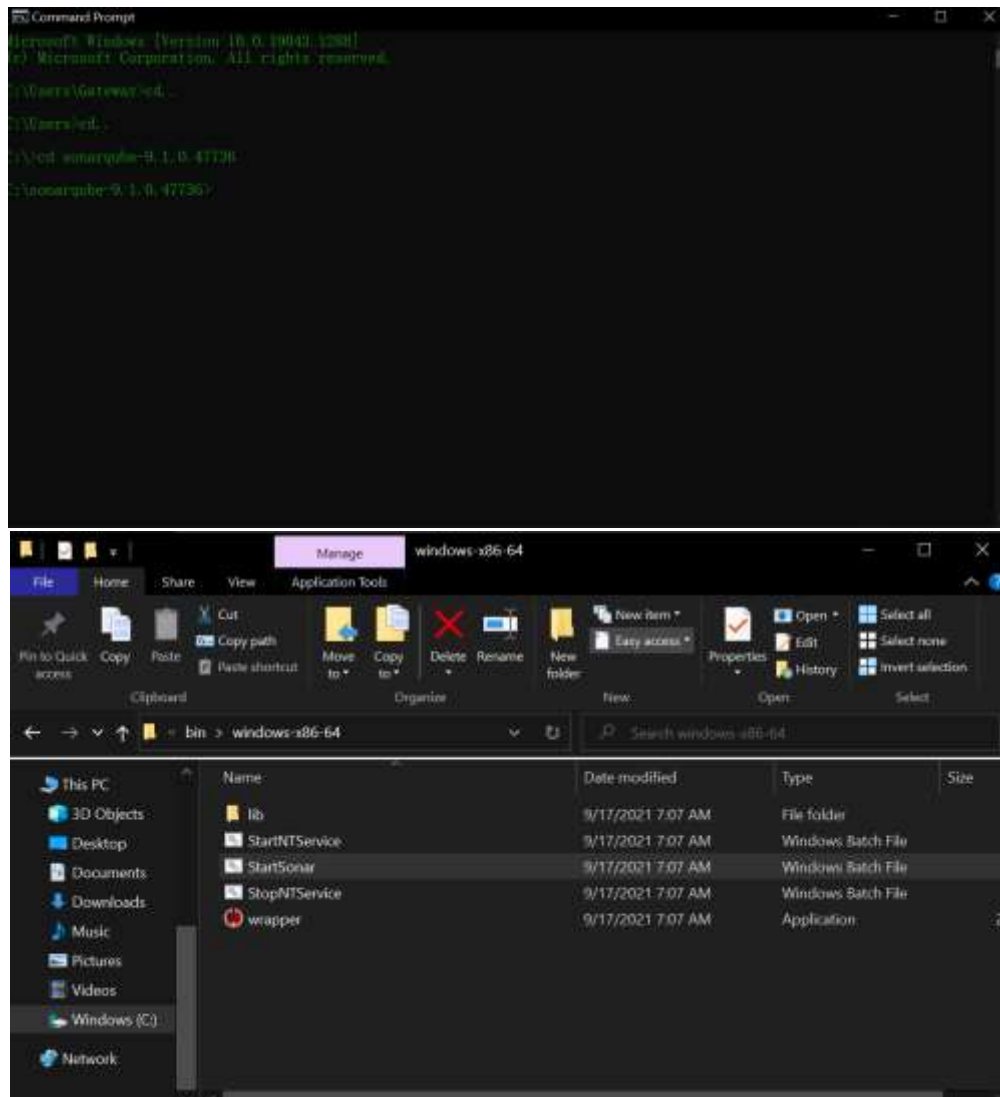
Calle 49ª Bis sur # 10d -20

Absence and Bad Grades Software
 Todos los Derechos Reservados

320 447 3192

313 710 3188

Abrimos el CMD y nos dirigimos a la carpeta de SonarQube-Bin-Windows x86-64 e iniciamos el archivo “StartSonar.”



Aparecerá el ejecutable de SonarQube, el cual se iniciará, esperamos que el proceso finalice exitosamente y nos dirigimos en el navegador al Localhost:9000.


```

Select SonarQube
jvm 1 | 2021.11.18 16:27:43 INFO app[] [o.s.a.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2, logFileName=
Prefix=web]] from [C:\sonarqube-9.1.0.47736]: C:\Program Files\Java\jdk-11.0.11\bin\java -Djava.awt.headless=true -Dfile
encoding=UTF-8 -Djava.io.tmpdir=C:\sonarqube-9.1.0.47736\temp -XX:OmitStackTraceInFastThrow --add-opens=java.base/java
util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java
.rmi/sun.rmi.transport=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.nio=ALL
UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL-UNNAMED --add-ope
ns=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xms512m -Xmx128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonPr
oxyHosts=localhost[127.*][::1] -cp ./lib/sonar-application-9.1.0.47736.jar;C:\sonarqube-9.1.0.47736\lib\jdb\h2-h2-i.4.1
99.jar org.sonar.server.app.WebServer C:\sonarqube-9.1.0.47736\temp\sq-process16254184326896732974properties
jvm 1 | 2021.11.18 16:28:30 INFO app[] [o.s.a.SchedulerImpl] Process[web] is up
jvm 1 | 2021.11.18 16:28:30 INFO app[] [o.s.a.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFileName=
Prefix=ce]] from [C:\sonarqube-9.1.0.47736]: C:\Program Files\Java\jdk-11.0.11\bin\java -Djava.awt.headless=true -Dfile
encoding=UTF-8 -Djava.io.tmpdir=C:\sonarqube-9.1.0.47736\temp -XX:OmitStackTraceInFastThrow --add-opens=java.base/java.u
til=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-ope
ns=java.base/java.nio=ALL-UNNAMED --add-opens=java.management/sun.management=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-ope
ns=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xms512m -Xmx128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonPr
oxyHosts=localhost[127.*][::1] -cp ./lib/sonar-application-9.1.0.47736.jar;C:\sonarqube-9.1.0.47736\lib\jdb\h2-h2-i.4.1
99.jar org.sonar.ce.app.CeServer C:\sonarqube-9.1.0.47736\temp\sq-process3516490543257946092pro
perties
jvm 1 | 2021.11.18 16:28:31 WARN app[] [startup] #####
jvm 1 | 2021.11.18 16:28:31 WARN app[] [startup] Default Administrator credentials are still being used. Make sure to
change the password or deactivate the account.
jvm 1 | 2021.11.18 16:28:31 WARN app[] [startup] #####
jvm 1 | 2021.11.18 16:28:35 INFO app[] [o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | 2021.11.18 16:28:35 INFO app[] [o.s.a.SchedulerImpl] SonarQube is up
  
```

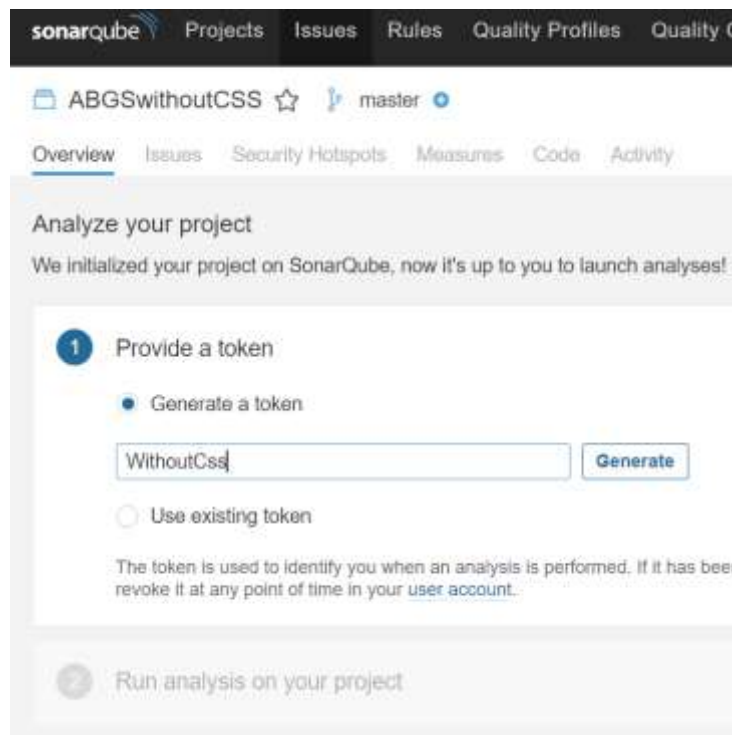
Ingresaremos el usuario y la contraseña los cuales son “admin”, luego de esto crearemos nuestra contraseña nueva.



Luego de ingresar Crearemos un proyecto.

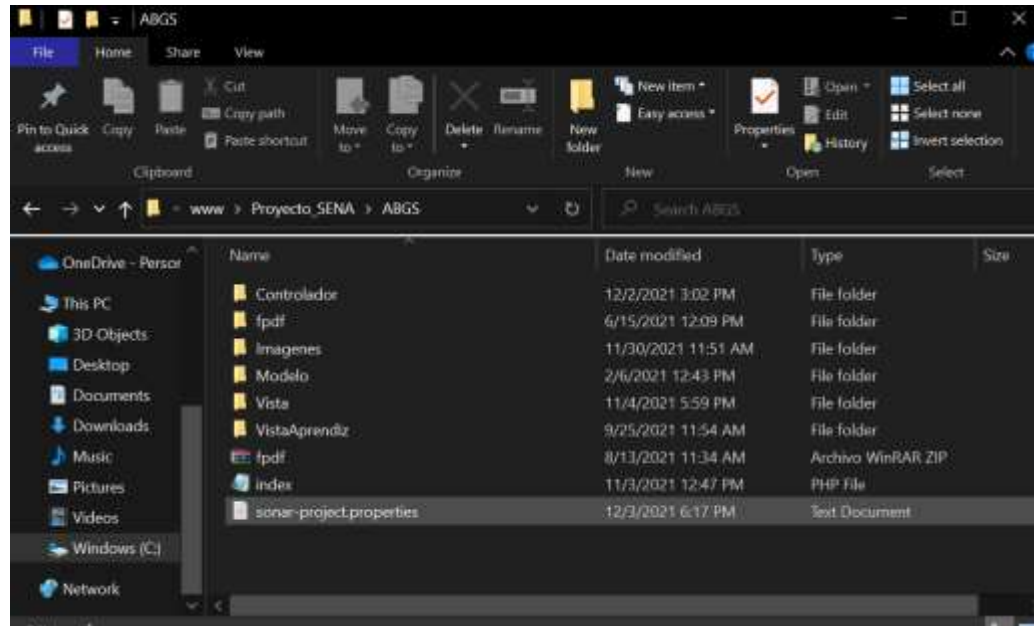


The screenshot shows the 'Create a project' form in SonarQube. The form has a dark header with the SonarQube logo and navigation links: Projects, Issues, Rules, and Quality Profiles. The main title is 'Create a project'. Below it, a note states 'All fields marked with * are required'. There are two required fields: 'Project display name *' and 'Project key *'. Both fields have 'ABGSwithoutCSS' entered and a green checkmark icon to the right. Below the 'Project display name' field, a note says 'Up to 255 characters. Some scanners might override the value you provide.' Below the 'Project key' field, a note says 'The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.' At the bottom of the form is a 'Set Up' button.



The screenshot shows the 'Overview' page for the project 'ABGSwithoutCSS' in SonarQube. The header is dark with the SonarQube logo and navigation links: Projects, Issues, Rules, Quality Profiles, and Quality Checks. Below the header, the project name 'ABGSwithoutCSS' is displayed with a star icon and a 'master' branch indicator. The 'Overview' tab is selected, showing a sub-header 'Analyze your project' and a message 'We initialized your project on SonarQube, now it's up to you to launch analyses!'. Below this, there are two steps: 1. 'Provide a token' and 2. 'Run analysis on your project'. Step 1 is active and shows two options: 'Generate a token' (selected) and 'Use existing token'. Under 'Generate a token', there is a text input field with 'WithoutCss' and a 'Generate' button. Below the input field, a note says 'The token is used to identify you when an analysis is performed. If it has been revoke it at any point of time in your [user account](#).' Step 2 is currently disabled.

Esto nos generará un código, el cual copiaremos y pegaremos en el CMD. Para esto debemos crear una carpeta “.properties” en nuestro proyecto, debe llevar de nombre “sonar-project.properties”.



El documento debe tener los siguientes datos (La última línea es opcional, se usa en caso de querer excluir algo, en este caso, los archivos “.css”):



Nos dirigimos en el CMD a la carpeta donde se encuentra nuestro proyecto con el archivo “.properties” y ejecutamos el código que copiamos con anterioridad.



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\wamp\www\Proyecto_SENA\ABGS>sonar-scanner.bat -D"sonar.projectKey=ABGSwithoutCSS" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.login=ca832fa011282a2a6871bf3be749ene879ed4055"
  
```

Esperamos que se ejecute y al terminar, el localhost:9000 se actualizara automáticamente y nos mostrara los errores y calificaciones de nuestro código.

Análisis de ABGS:

En este caso se puede evidenciar que **ABGS** paso la prueba de código con una vulnerabilidad que hace referencia a la falta de contraseña de la base de datos. Por ende, se mejoró la seguridad y se realizó nuevamente la prueba.

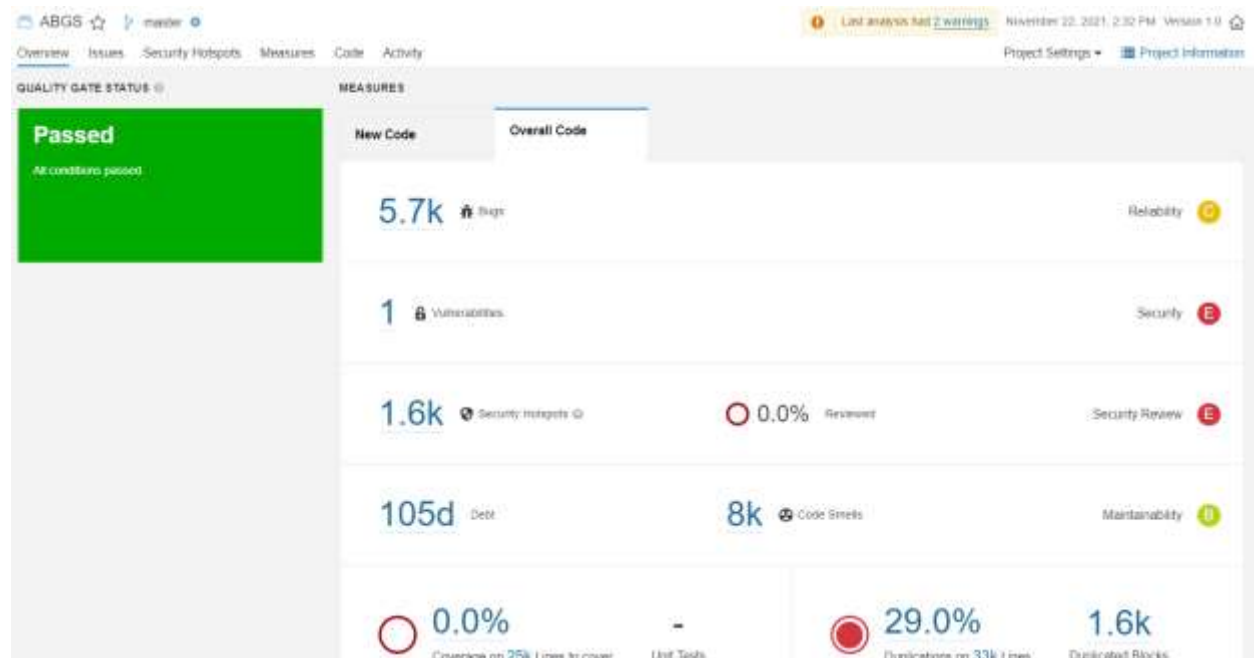
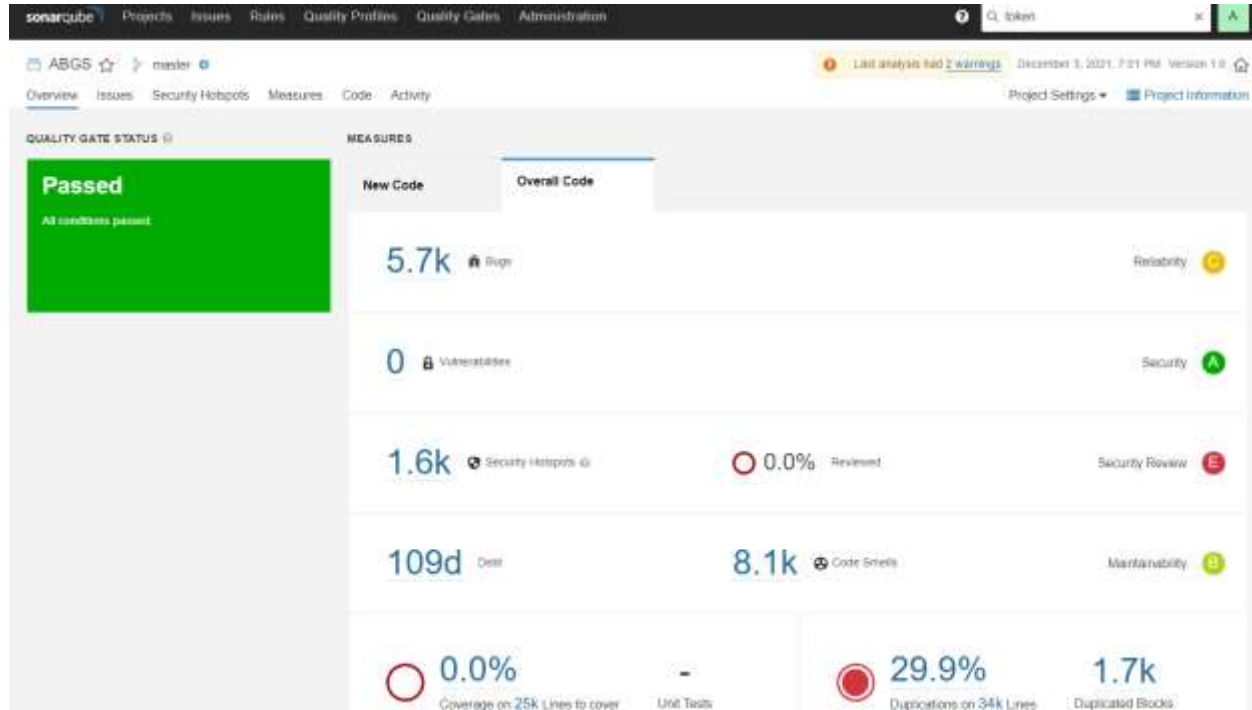


Figure 1 Primer prueba de ABGS

En la siguiente imagen se evidencia la mejora de seguridad al código.



Conclusión:

Al usar un software que pruebe nuestro código de manera automática como SonarQube, podemos evidenciar las fallas o mejoras que le debemos realizar a nuestro software de una manera eficaz, sencilla y rápida, esta evaluación nos sirve para realizar software de una mejor manera, no tiene costos y funciona para muchos lenguajes de programación, es importante tener en cuenta este tipo de softwares para poder evaluar nuestro código y entregar un producto de calidad.

Bibliografías:

Montilla,A.(7 mayo de 2021). SonarQube: Una herramienta útil para verificar la calidad del código.Castor.com.co. Tomado de: <https://castor.com.co/sonarqube-una-herramienta-util-para-verificar-la-calidad-del-codigo/>

Díaz,C.(05 de agosto de 2019). Code smells y deuda técnica. OpenWebinars.net. Tomado de: <https://openwebinars.net/blog/code-smells-y-deuda-tecnica/>

