

**FOXTECH**

# INFORME DE CALIDAD EN EL CÓDIGO FUENTE



¿Qué es SonarQube?



Instalación y configuración de SonarQube en Linux



Configuración de SonarQube en el proyecto



Resultado del análisis hecho por SonarQube al código fuente



Ajustes en el código fuente



Conclusión

# ¿Qué es SonarQube?

SonarQube es una plataforma de código abierto para la inspección de la calidad del código a través de diferentes herramientas de análisis de código fuente. Esta plataforma proporciona métricas que ayudan a mejorar la calidad del código.

## Principales métricas de SonarQube

- **Complejidad:** Es una métrica de calidad software basada en el cálculo del número de caminos independientes que tiene nuestro código.
- **Duplicados:** Nos indica el número de bloques de líneas duplicados.
- **Bugs:** Son errores o defectos en el software.
- **Vulnerabilidad:** Son problemas de seguridad en el software y deben ser solucionados de inmediato.
- **Security Hotspot:** Es un fragmento de código sensible a la seguridad, pero es posible que no afecte la seguridad general del software.
- **Mantenibilidad:** Se refiere al recuento total de problemas de Code Smell o malas prácticas implementadas en el proyecto.
- **Tamaño:** Permiten hacerse una idea del volumen del proyecto en términos generales.
- **Pruebas:** Son una forma de comprobar el correcto funcionamiento de una unidad de código y de su integración.

# Instalación y configuración de SonarQube en Linux

## Descargar e instalar openjdk 11 o jdk 11

Java Archive Downloads - Java SE 11

JDK 11

## Descargar SonarQube Community edition

Code Quality and Code Security | SonarQube  
sonarqube

### 1. Extraer el zip en la raíz del disco duro



### 2. Modificar la configuración del archivo wrapper.conf el cual esta en la carpeta conf de sonarqube

*(i)* En la tercera linea del archivo se especifica el path donde esta instalado el openjdk o el jdk.

```
# Path to JVM executable. By default it must be available in PATH.  
# Can be an absolute path, for example:  
wrapper.java.command=/usr/lib/jvm/java-11-openjdk-amd64/bin/java  
#wrapper.java.command=java
```

### 3. Acceder a la carpeta bin, luego a linux-x86-64 y abrir una terminal

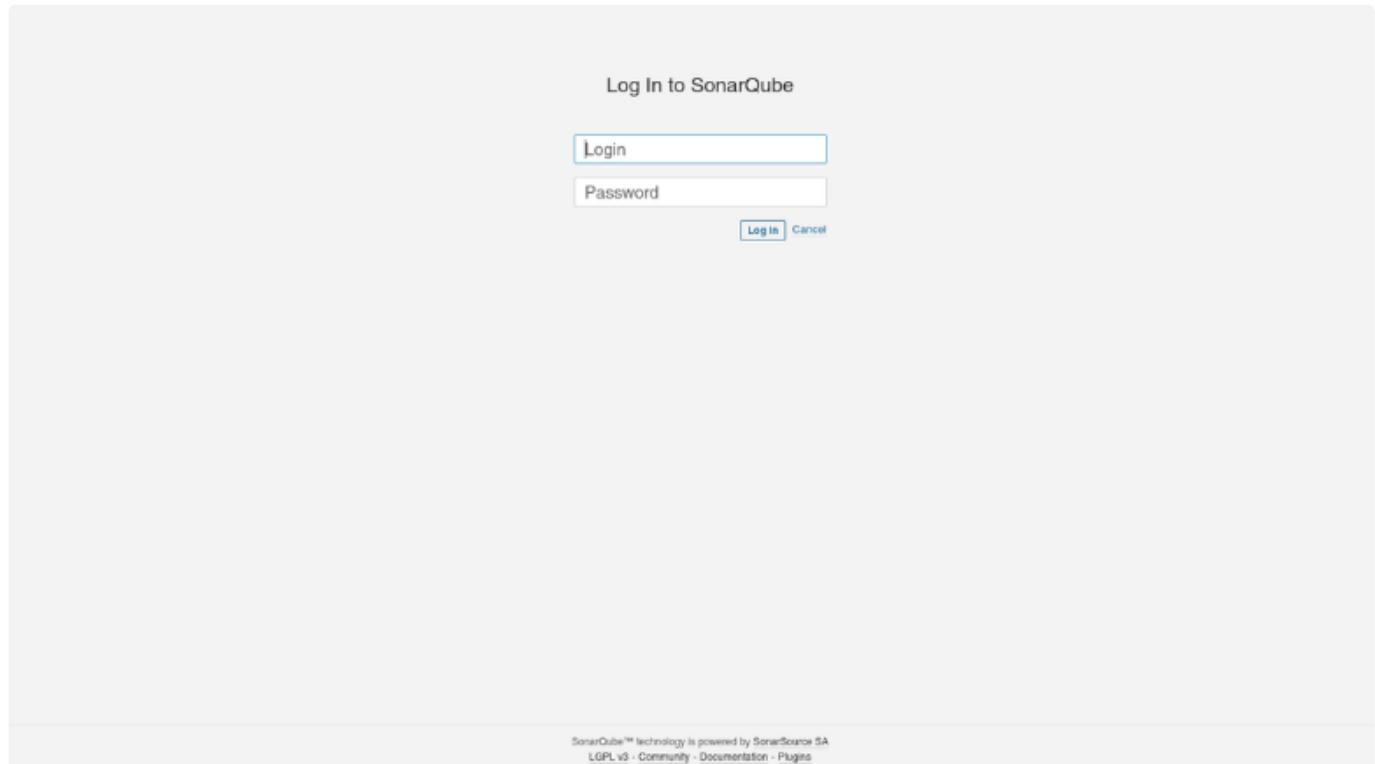
Ejecutar el siguiente comando para iniciar el servidor de SonarQube.

```
sh sonar.sh start
```

### 4. Acceder en el navegador a la interfaz de sonarqube

Se debe colocar en el navegador localhost:9000 o 127.0.0.1:9000.

- ⓘ La primera vez que se accede el usuario es admin y la contraseña admin, después nos pedirá actualizar la contraseña.



# Configuración de SonarQube en el proyecto

## 1. Seleccionar la forma en como se creara el nuevo proyecto en SonarQube

En este caso se seleccionara la opcion “Manually”.

The screenshot shows the SonarQube interface for creating a new project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A search bar and a user icon are also present. Below the navigation, a section asks "How do you want to create your project?". It provides four options: "From Azure DevOps", "From Bitbucket", "From GitHub", and "From GitLab", each with a "Set up global configuration" link. Below these, a note says "Do you want to benefit from all of SonarQube's features (like repository Import and Pull Request decoration)? Create your project from your favorite DevOps platform. First, you need to set up a DevOps platform configuration." Then, it shows a "Manually" option with a double arrow icon and a "Set up global configuration" link.

## 2. Especificar el nombre del nuevo proyecto y el identificador único

The screenshot shows the "Create a project" form. The title is "Create a project". A note says "All fields marked with \* are required". The "Project display name" field is filled with "FoxTech" and has a checkmark. A note below it says "Up to 255 characters. Some scanners might override the value you provide.". The "Project key" field is also filled with "FoxTech" and has a checkmark. A note below it 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.". A "Set Up" button is at the bottom.

## 3. Seleccionar la forma en como se analizara el código del proyecto

En este caso seleccionaremos la opcion “Locally”.

This screenshot is identical to the one above, showing the "Create a project" form with the "Project display name" and "Project key" fields both set to "FoxTech". The "Set Up" button is visible at the bottom.

FoxTech master

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

How do you want to analyze your repository?

Do you want to integrate with your favorite CI? Choose one of the following tutorials.

-  With Jenkins
-  With GitHub Actions
-  With Bitbucket Pipelines
-  With GitLab CI
-  With Azure Pipelines
- Other CI

Are you just testing or have an advanced use-case? Analyze your project locally.

-  Locally

## 4. Se crea o se utiliza un token ya generado

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

FoxTech master

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!

- Provide a token
 

Generate a token

Enter a name for your token  Generate

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.
- Run analysis on your project

Damos clic en generate:

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

FoxTech master

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!

- Provide a token
 

FoxTech: 352a229c62c6aaa253d3c31b2865c21557526b2e 

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.

[Continue](#)
- Run analysis on your project

## 5. Seleccionar la tecnología o el lenguaje de programación que se uso en el proyecto

The screenshot shows the SonarQube project setup interface for a project named 'FoxTech'. Step 1, 'Provide a token', has a green checkmark and the token value 'FoxTech:352a229c62c6aaa353d3c31b2865c21557526b2e'. Step 2, 'Run analysis on your project', asks 'What option best describes your build?' with options: Maven, Gradle, .NET, and Other (for JS, TS, Go, Python, PHP, ...). The 'Other' option is selected.

## 6. Seleccionar el sistema operativo que estamos usando

The screenshot shows the SonarQube project setup interface for a project named 'FoxTech'. It includes steps 1 and 2 from the previous screenshot, plus a new section 'What is your OS?' with options: Linux, Windows, and macOS. The 'Linux' option is selected.

## 7. Descargar SonarScanner

SonarScanner | SonarQube Docs

## 8. Extraer el zip en la raiz del disco duro

## 9. En el archivo .bashrc se agrega el path o la ruta donde esta almacenada la carpeta de SonarScanner

```
PATH="/opt/sonar-scanner-4.6.2.2472-linux/bin:$PATH"
```

## 10. Crear un archivo de configuración de sonarqube en la raiz del proyecto

 El archivo se debe llamar sonar-project.properties

## 11. Agregar la siguiente configuración al archivo creado en el paso anterior

- **sonar.projectKey** = Acá debe ir el nombre del proyecto que especificamos en sonarqube.
- **sonar.projectName** = Acá debe ir el nombre de la carpeta raíz del proyecto que vamos a analizar.

```
# must be unique in a given SonarQube instance
sonar.projectKey=FoxTech
# --- optional properties ---

# defaults to project key
sonar.projectName=FoxTech
# defaults to 'not provided'
sonar.projectVersion=1.0

# Path is relative to the sonar-project.properties file. Defaults to .
sonar.sources=.

# Encoding of the source code. Default is default system encoding
sonar.sourceEncoding=UTF-8

sonar.python.version=3.8
```

## 12. Ejecutar el análisis al proyecto

Después de escoger la tecnología del proyecto y el sistema operativo, además de configurar e instalar el sonarScanner, sonarqube nos genera un código el cual se debe ejecutar en una nueva terminal y la terminal debe estar en la carpeta raíz del proyecto.

The screenshot shows the SonarQube interface for the 'FoxTech' project. At the top, there's a navigation bar with links for Overview, Issues, Security Hotspots, Measures, Code, Activity, Project Settings, and Project Information. Below the navigation, there are tabs for Linux, Windows, and macOS, with 'Windows' selected. A section titled 'Download and unzip the Scanner for Linux' provides instructions to visit the official documentation and add the bin directory to the PATH environment variable. Another section, 'Execute the Scanner', shows a command-line snippet for running the scanner with specific parameters like -Dsonar.projectKey=FoxTech, -Dsonar.sources=., -Dsonar.host.url=http://localhost:9000, and -Dsonar.login=... . To the right of the command is a 'Copy' button. Below these sections, there are links for 'Is my analysis done?', 'Pull Request Decoration', and useful links for 'Branch Analysis' and 'Pull Request Analysis'.

```
COMPUTIDIA ~ > Escritorio > Proyecto formativo django > FoxTech > sonar-scanner \
-Dsonar.projectKey=FoxTech \
-Dsonar.sources=.
-Dsonar.host.url=http://localhost:9000 \
-Dsonar.login=352a229c62c6aaa353d3c31b2865c21557526b2e
```

Si el análisis se realizó de manera correcta nos debe generar el siguiente mensaje en la terminal:

```
INFO: ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard?id=FoxTech
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AX83nEPkwYQgiyCj-nB_
INFO: Analysis total time: 52.514 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 59.664s
INFO: Final Memory: 23M/80M
INFO: -----
```

```
COMPUTIDIA ~ > Escritorio > Proyecto formativo django > FoxTech
```

Pegamos el link que nos generó sonarqube en el navegador para ver el resultado del análisis:

## QUALITY GATE STATUS

## MEASURES

**Passed**

All conditions passed.

## New Code

## Overall Code

6 Bugs

Reliability C

0 Vulnerabilities

Security A

15 Security Hotspots

0.0% Reviewed

Security Review E

1h 21min Debt

13 Code Smells

Maintainability A

0.0%

Coverage on 1.1k Lines to cover

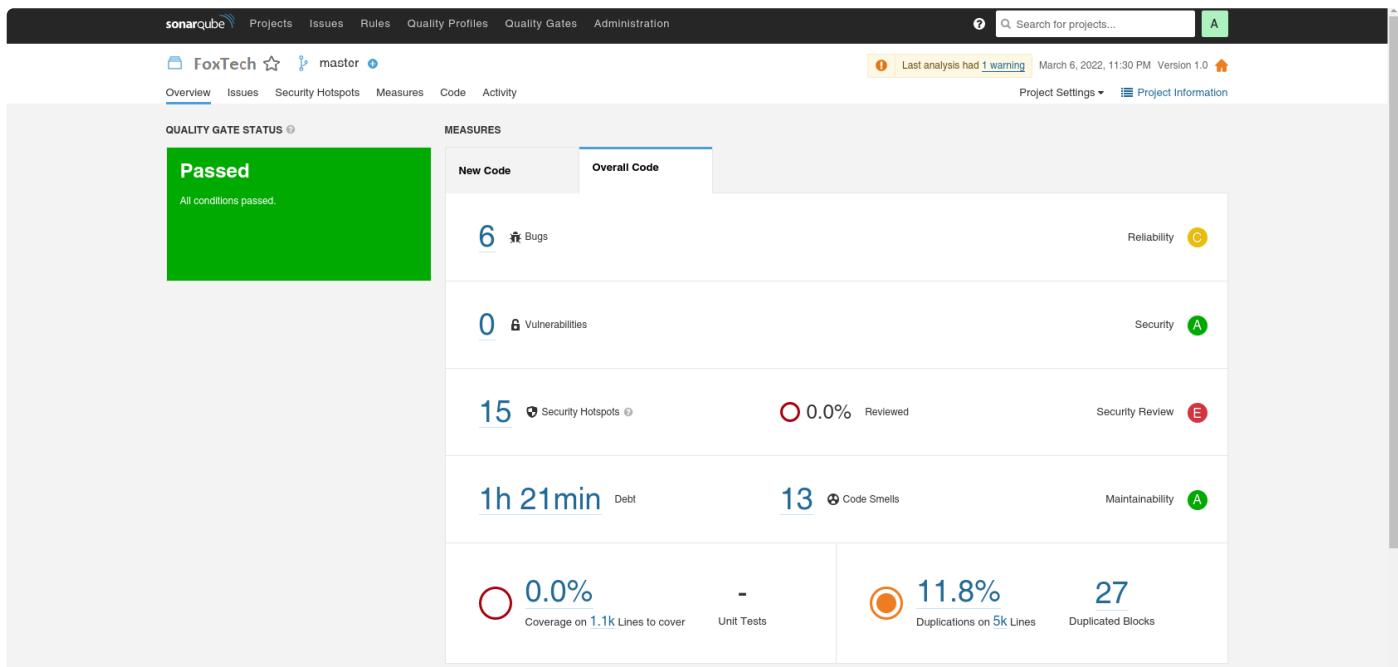
Unit Tests

11.8%

Duplications on 5k Lines

27 Duplicated Blocks

# Resultado del análisis hecho por SonarQube al código fuente



## Resumen del análisis:

- Se encontraron 6 bugs.
- El software tiene 0 vulnerabilidades de seguridad.
- Se encontraron 13 Code smells o malas prácticas.
- Se encontraron 15 fragmentos de código que podrían afectar la seguridad del software.
- Se demoraría aproximadamente 1h 21 minutos para refactorizar el código.
- El 11.8% del código está duplicado.

# Ajustes en el código fuente



Bugs



Security Hotspots



Code smell

# Bugs

The screenshot shows the SonarQube interface for the 'FoxTech' project. The top navigation bar includes 'Overview', 'Issues' (selected), 'Security Hotspots', 'Measures', 'Code', and 'Activity'. A message at the top right indicates 'Last analysis had 1 warning' on March 6, 2022, at 11:30 PM, Version 1.0. The main area displays a list of issues under 'My Issues' and 'All' filters. The first issue is a 'Bug' labeled 'Add a <title> tag to this page. Why is this an issue?'. It has a severity of 'Major', is 'Open', and assigned to 'Not assigned'. The second issue is 'Replace this <> tag by <em>. Why is this an issue?' with a severity of 'Minor', 'Open', and assigned to 'Not assigned'. The third issue is 'Replace this <> tag by <em>. Why is this an issue?' with a severity of 'Minor', 'Open', and assigned to 'Not assigned'. The fourth issue is 'Add a <title> tag to this page. Why is this an issue?' with a severity of 'Major', 'Open', and assigned to 'Not assigned'. The fifth issue is 'Replace this <> tag by <em>. Why is this an issue?' with a severity of 'Minor', 'Open', and assigned to 'Not assigned'. The sixth issue is 'Add an "alt" attribute to this image. Why is this an issue?' with a severity of 'Minor', 'Open', and assigned to 'Not assigned'. The bottom of the list shows a note: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

## Solución bugs 2, 3 y 5.

Para solucionar este bug se le agrega el atributo `aria-hidden` para que la tecnología de asistencia del navegador omita este ícono, según la MDN este atributo ayuda a mejorar la accesibilidad de la página.

aria-hidden - Accessibility | MDN

The screenshot shows a snippet of MDN documentation for the `aria-hidden` attribute. It features a dark header bar with three circular icons (red, yellow, green) and the word 'HTML'. Below the header, the code example is shown:

```
1 <i class="fas fa-info-circle container-link-main-aside__icon" aria-hidden="true"></i>
```

## Solución bugs 1 y 4

Estos bugs son falsos positivos porque se usan para la herencia entre templates, es decir son plantillas por ende no necesitan la etiqueta title.

## Solución bug 6

Para la solución de este bug se agrega el atributo alt para ofrecer un texto alternativo en caso de que el navegador no cargue la imagen.



HTML

```
1 
```

# Security Hotspots

## Solución security hotspots 1 al 14

Para solucionar estos security hotspots se le agrego un decorador a cada una de las vistas para restringir el tipo de petición que pueden recibir, de esta manera se solucionan las posibles vulnerabilidades de seguridad.

Ejemplo #1 de una vista con el decorador agregado:



Python

```
1 from django.views.decorators.http import require_GET
2
3 @require_GET
4 def info(request):
5     return render(request, 'info.html', context={})
```

Ejemplo #2 de una vista con el decorador agregado:



Python

```
1 from django.views.decorators.http import require_http_methods
2
3 @require_http_methods(['GET', 'POST'])
4 def cart_view(request):
5     cart = get_or_create_cart(request)
6
7     return render(request, 'cart/carrito de compras.html', context = {
8         'carrito': cart,
9     })
```



# Code smell

The screenshot shows the SonarQube interface for a project named 'FoxTech'. The 'Issues' tab is selected, displaying 13 code smell issues across several files:

- Carts/models.py:** Rename this variable; It shadows a builtin. Why is this an issue? (Code Smell, Major, Open, Not assigned, 5min effort)
- Carts/views.py:** Remove the unused local variable "cart\_product". Why is this an issue? (Code Smell, Minor, Open, Not assigned, 5min effort)
- Products/migrations/0001\_initial.py:** Define a constant instead of duplicating this literal 'Fecha de modificacion' 3 times. Why is this an issue? (Code Smell, Critical, Open, Not assigned, 6min effort)
- Products/utils.py:** Rename this variable; It shadows a builtin. Why is this an issue? (Code Smell, Major, Open, Not assigned, 5min effort)
- SaleCold/settings.py:** Remove this commented out code. Why is this an issue? (Code Smell, Major, Open, Not assigned, 5min effort)
- Users/forms.py:** Rename field "genero" to prevent any misunderstanding/clash with field "GENERO" defined on line 17. Why is this an issue? (Code Smell, Blocker, Open, Not assigned, 10min effort)
- Users/forms.py:** Define a constant instead of duplicating this literal 'Este campo acepta minimo 4 caracteres y maximo 25 characters.' 6 times. Why is this an issue? (Code Smell, Critical, Open, Not assigned, 12min effort)
- Users/views.py:** Rename function "updateDataUser" to match the regular expression "[a-z\_][a-z0-9\_]\*\$". Why is this an issue? (Code Smell, Major, Open, Not assigned, 10min effort)

Filters on the left include 'Type: CODE SMELL' (selected), 'Severity' (Blocker: 1, Critical: 2, Major: 9), and 'Scope'.

## Solucion code smell 1 y 3

Para solucionar estos code smell se re nombraron dos variables porque estaban usando palabras reservada del lenguaje.

```
Python

1 def random_products(product):
2     productos = Product.objects.exclude(pk=product.id_product)
3     cantidad_productos = abs(Product.objects.all().count() - 7)
4     numero_aleatorio = random.randint(1, cantidad_productos)
5
6     return productos[numero_aleatorio:numero_aleatorio + 6]
```

## Solucion code smell 4

Para solucionar este code smell se elimino una linea de codigo que no se estaba usando.

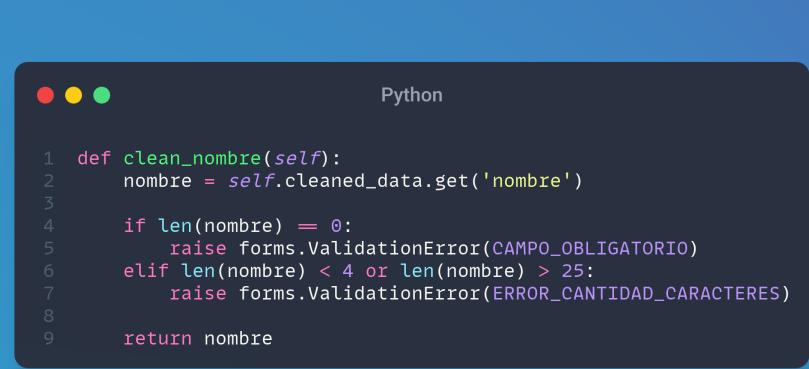
## Solucion code smell 7

Para solucionar este code smell se creo una variable constante al inicio del archivo y luego se utilizo en los lugares donde se repetía la cadena.



```
1 ERROR_CANTIDAD_CARACTERES = 'Este campo acepta minimo 4 caracteres y maximo 25 caracteres.'
```

Ejemplo de una funcion para validar un campo del formulario usando la variable constante:



```
1 def clean_nombre(self):
2     nombre = self.cleaned_data.get('nombre')
3
4     if len(nombre) == 0:
5         raise forms.ValidationError(CAMPO_OBLIGATORIO)
6     elif len(nombre) < 4 or len(nombre) > 25:
7         raise forms.ValidationError(ERROR_CANTIDAD_CARACTERES)
8
9     return nombre
```

## Solucion code smells 10 al 13

La solución de estos tres code smells fue eliminar el código repetido.

# Conclusión

Después de la refactorización al código para solucionar los problemas de calidad, se ejecuta de nuevo el análisis dando un resultado satisfactorio, ya que el código fuente del aplicativo cuenta con calidad.

