

28. SonarQube: El Guardián de la Calidad del Código

FASE 1 — Montar SonarQube en Docker (Ubuntu)

Comprueba Docker

- docker --version

```
ub@forense-ai:~$ docker --version
Docker version 29.2.1, build a5c7197
```

- docker ps

```
ub@forense-ai:~$ sudo docker ps
[sudo] contraseña para ub:
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
NAMES
f78426341423   wordpress:latest "docker-entrypoint.s..." 2 months ago   Up 56 minutes   0.0.0.0:8000->80/tcp, [::]:8000->80/tcp
000->80/tcp, [::]:8000->80/tcp
5d771c5bb40a   mariadb:latest  "docker-entrypoint.s..." 2 months ago   Up 56 minutes   3306/tcp
wp_mariadb
```

- docker compose version

```
ub@forense-ai:~$ docker compose version
Docker Compose version v5.0.2
```

Crear el directorio de SonarQube

- mkdir -p ~/proyectos/sonarqube-docker
- cd ~/proyectos/sonarqube-docker

```
ub@forense-ai:~$ mkdir -p ~/proyectos/sonarqube-docker
ub@forense-ai:~$ cd ~/proyectos/sonarqube-docker
ub@forense-ai:~/proyectos/sonarqube-docker$
```

Restaurar los parámetros del kernel requeridos por SonarQube

- sysctl vm.max_map_count

```
ub@forense-ai:~/proyectos/sonarqube-docker$ sysctl vm.max_map_count
vm.max_map_count = 1048576
```

Si no es 262144:

- sudo sysctl -w vm.max_map_count=262144
- echo "vm.max_map_count=262144" | sudo tee /etc/sysctl.d/99-sonarqube.conf
- sudo sysctl --system

```

ub@forense-ai:~/proyectos/sonarqube-docker$ sudo sysctl -w vm.max_map_count=262144
vm.max_map_count = 262144
ub@forense-ai:~/proyectos/sonarqube-docker$ echo "vm.max_map_count=262144" | sudo tee /etc/sysctl.d/99-sonarqube.conf
vm.max_map_count=262144
ub@forense-ai:~/proyectos/sonarqube-docker$ sudo sysctl --system
* Aplicando /usr/lib/sysctl.d/10-apparmor.conf...
* Aplicando /etc/sysctl.d/10-bufferbloat.conf...
fs.protected_regular = 2
fs.protected_symlinks = 1
vm.max_map_count = 262144

```

Crear docker-compose.yml

- sudo nano docker-compose.yml

```
ub@forense-ai:~/proyectos/sonarqube-docker$ sudo nano docker-compose.yml
```

Pega el siguiente contenido:

```

services:
  db:
    image: postgres:15
    container_name: sonar-db
    environment:
      POSTGRES_USER: sonar
      POSTGRES_PASSWORD: sonar
      POSTGRES_DB: sonarqube
    volumes:
      - sonar_db_data:/var/lib/postgresql/data
    networks:
      - sonar-net
    restart: unless-stopped

  sonarqube:
    image: sonarqube:community
    container_name: sonarqube
    depends_on:
      - db
    ports:
      - "9000:9000"
    environment:
      SONAR_JDBC_URL: jdbc:postgresql://db:5432/sonarqube
      SONAR_JDBC_USERNAME: sonar
      SONAR_JDBC_PASSWORD: sonar
    volumes:
      - sonar_data:/opt/sonarqube/data
      - sonar_extensions:/opt/sonarqube/extensions
      - sonar_logs:/opt/sonarqube/logs
    networks:
      - sonar-net
    restart: unless-stopped

```

```
volumes:  
  sonar_db_data:  
  sonar_data:  
  sonar_extensions:  
  sonar_logs:
```

```
networks:  
  sonar-net:  
    driver: bridge
```

```
GNU nano 7.2                                     docker-compose.yml *
```

```
services:  
  db:  
    image: postgres:15  
    container_name: sonar-db  
    environment:  
      POSTGRES_USER: sonar  
      POSTGRES_PASSWORD: sonar  
      POSTGRES_DB: sonarqube  
    volumes:  
      - sonar_db_data:/var/lib/postgresql/data
```

Arrancar SonarQube

En la misma carpeta:

- docker compose up -d

```
ub@forense-ai:~/proyectos/sonarqube-docker$ sudo docker compose up -d  
[+] up 35/35  
✓ Image sonarqube:community          Pulled          76.3s  
✓ Image postgres:15                  Pulled          20.8s  
✓ Network sonarqube-docker_sonar-net Created         0.1s  
✓ Volume sonarqube-docker_sonar_data Created         0.0s  
✓ Volume sonarqube-docker_sonar_extensions Created         0.0s  
✓ Volume sonarqube-docker_sonar_logs Created         0.0s  
✓ Volume sonarqube-docker_sonar_db_data Created         0.0s  
✓ Container sonar-db                Created         0.3s  
✓ Container sonarqube              Created         0.0s
```

Comprueba que están levantados:

- docker ps

```
ub@forense-ai:~/proyectos/sonarqube-docker$ sudo docker ps  
CONTAINER ID   IMAGE          COMMAND       CREATED          STATUS          PORTS          NAMES  
b1fc15be2edb  sonarqube:community "/opt/sonarqube/dock_..."  About a minute ago  Up About a minute  0.0.0.0:9000->9000/tcp, [::]:9000->9000/tcp  sonarqube  
671c24b0ea38  postgres:15    "docker-entrypoint.s..."  About a minute ago  Up About a minute  5432/tcp        sonar-db  
f78426341423  wordpress:latest "docker-entrypoint.s..."  2 months ago     Up About an hour  0.0.0.0:8000->80/tcp, [::]:8000->80/tcp  wp_site  
5d771c5bb40a  mariadb:latest  "docker-entrypoint.s..."  2 months ago     Up About an hour  3306/tcp        wp_mariadb
```

Ver logs y esperar a que esté listo

- docker logs -f sonarqube

```
ub@forense-ai:~/proyectos/sonarqube-docker$ sudo docker logs -f sonarqube
```

Acceso web y primer login

Abre en el navegador:

- http://localhost:9000

Credenciales por defecto:

- Usuario: admin
- Contraseña: admin

The screenshot shows the SonarQube login interface. On the left, there's a "Welcome to SonarQube" section with fields for "Username" (admin) and "Password" (admin), and a "Log in" button. On the right, a large blue sidebar titled "Sonar" and "SonarQube Server: Your independent code verification layer" provides information about the service, including "Continuous Verification", "DevSecOps Integrated", and "Empowering developers". At the bottom of the sidebar, there are links for "Terms & Conditions", "Community", "Documentation", and "Plugins".

Actualizar contraseña:

Update your password

⚠ This account should not use the default password.

Enter a new password

Old Password *

.....

Password *

Confirm Password *

Update

The screenshot shows the "How do you want to create your project?" page. It features a header with "SonarQube community" and navigation links for "Projects", "Issues", "Rules", "Quality Profiles", "Quality Gates", "Administration", and "More". Below the header, a section asks "How do you want to create your project?". It lists options for importing from various platforms: "Import from Azure DevOps", "Import from Bitbucket Cloud", "Import from Bitbucket Server", "Import from GitHub", and "Import from GitLab", each with a "Setup" button. A question "Are you just testing or have an advanced use-case?" has a "Create a local project" button. At the bottom, a footer states "SonarQube™ technology is powered by SonarSource Sàrl" and "Community Build v26.2.0.119303 - MQR MODE". A promotional sidebar on the right encourages using SonarQube for IDE and provides "Learn More" and "Dismiss" buttons.

FASE 2 — Proyectos de ejemplo + preparación para análisis

1) Estructura de carpetas

- mkdir -p ~/proyectos/grupo01/proyecto-php/src
- mkdir -p ~/proyectos/grupo01/proyecto-java/src

```
ub@forense-ai:~$ mkdir -p ~/proyectos/grupo01/proyecto-php/src  
ub@forense-ai:~$ mkdir -p ~/proyectos/grupo01/proyecto-java/src
```

2) Proyecto PHP con Apache en contenedor

- cd ~/proyectos/grupo01/proyecto-php

```
ub@forense-ai:~$ cd ~/proyectos/grupo01/proyecto-php
```

2.1 Crear archivo PHP con “malos olores”

- sudo nano src/index.php

Pega:

```
<?php  
  
function suma($a, $b){  
    return $a + $b;  
}  
  
function suma2($a, $b){ // duplicación  
    return $a + $b;  
}  
  
$x = 5;  
$y = 10;  
  
if($x == $y){  
    echo "Son iguales";  
} else {  
    echo "No son iguales";  
}  
  
echo "<br>Resultado: " . suma($x, $y);  
  
// Código muerto  
$z = 100;  
  
?>
```

```

GNU nano 7.2                                         src/index.php
<?php

function suma($a, $b){
    return $a + $b;
}

function suma2($a, $b){ // duplicación
    return $a + $b;
}

$x = 5;
$y = 10;

if($x == $y){
    echo "Son iguales";
} else {
    echo "No son iguales";
}

echo "<br>Resultado: " . suma($x, $y);

// Código muerto
$z = 100;

?>

```

2.2 Crear docker-compose del proyecto PHP

- nano docker-compose.yml

```
ub@forense-ai:~/proyectos/grupo01/proyecto-php$ nano docker-compose.yml
```

Contenido:

```

services:
  php-apache:
    image: php:8.2-apache
    container_name: php-grupo01
    ports:
      - "8081:80"
    volumes:
      - ./src:/var/www/html
    restart: unless-stopped

```

```

GNU nano 7.2                                         docker-compose.yml *
services:
  php-apache:
    image: php:8.2-apache
    container_name: php-grupo01
    ports:
      - "8081:80"
    volumes:
      - ./src:/var/www/html
    restart: unless-stopped

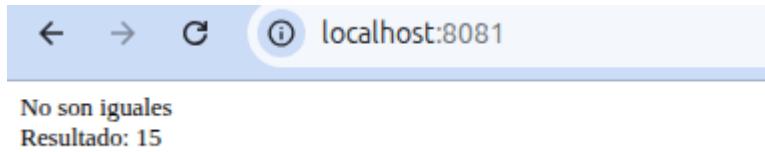
```

Arrancar:

- sudo docker compose up -d

```
ub@forense-ai:~/proyectos/grupo01/proyecto-php$ sudo docker compose up -d
[sudo] contraseña para ub:
[+] up 1/1
✓ Container php-grupo01 Running
```

En navegador: <http://localhost:8081>



3) Crear configuración básica de Sonar para PHP

En la raíz del proyecto PHP:

- nano sonar-project.properties

```
ub@forense-ai:~/proyectos/grupo01/proyecto-php$ nano sonar-project.properties
sonar.projectKey=lab1-php
sonar.projectName=Proyecto PHP Lab1
sonar.projectVersion=1.0
sonar.sources=src
sonar.sourceEncoding=UTF-8
GNU nano 7.2                                     sonar-project.properties *
sonar.projectKey=lab1-php
sonar.projectName=Proyecto PHP Lab1
sonar.projectVersion=1.0
sonar.sources=src
sonar.sourceEncoding=UTF-8
```

4) Proyecto Java simple

- cd ~/proyectos/grupo01/proyecto-java
- nano src/Main.java

```
● ub@forense-ai:~$ cd ~/proyectos/grupo01/proyecto-java
● ub@forense-ai:~/proyectos/grupo01/proyecto-java$ nano src/Main.java
```

Contenido:

```
public class Main {

    public static int suma(int a, int b){
        return a + b;
    }

    public static int suma2(int a, int b){ // duplicación
        return a + b;
    }

    public static void main(String[] args){
        int x = 5;
        int y = 5;
```

```

        if(x == y){
            System.out.println("Iguales");
        }

        int z = 100; // código muerto
        System.out.println(suma(x, y));
    }
}

```

```

GNU nano 7.2                               src/Main.java
public class Main {

    public static int suma(int a, int b){
        return a + b;
    }

    public static int suma2(int a, int b){ // duplicación
        return a + b;
    }

    public static void main(String[] args){
        int x = 5;
    }
}

```

sonar-project.properties para Java

- nano sonar-project.properties

```

ub@forense-ai:~/proyectos/grupo01/proyecto-java$ nano sonar-project.properties
sonar.projectKey=lab1-java
sonar.projectName=Proyecto JAVA Lab1
sonar.projectVersion=1.0
sonar.sources=.
sonar.sourceEncoding=UTF-8

```

```

GNU nano 7.2                               sonar-project.properties *
sonar.projectKey=lab1-java
sonar.projectName=Proyecto JAVA Lab1
sonar.projectVersion=1.0
sonar.sources=.
sonar.sourceEncoding=UTF-8

```

FASE 3 — Primer análisis real con Sonar Scanner

1) Crear Token en SonarQube

El scanner necesita autenticarse. No se usan usuario/contraseña, se usa token.

Paso 1 — Entrar en SonarQube

Navegador: <http://localhost:9000>

Login con tu usuario.

How do you want to create your project?

Do you want to benefit from all of SonarQube Community Build'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.

- Import from Azure DevOps (Setup)
- Import from Bitbucket Cloud (Setup)
- Import from Bitbucket Server (Setup)
- Import from GitHub (Setup)
- Import from GitLab (Setup)

Are you just testing or have an advanced use-case?

Create a local project

Paso 2 — Crear token

Ir a: User → My Account → Security → Generate Token

Nombre sugerido: scanner-grupo01

A Administrator

Profile Security Notifications Projects

My Account / Security

Security

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name	Type	Expires in
scanner-grupo01	Global Analysis Token	30 days

scanner-grupo01

Name	Type	Project	Last use	Created	Expiration
scanner-grupo01	Global		Never	February 13, 2026	March 15, 2026

[Revoke](#)

2) Ejecutar Sonar Scanner desde contenedor

Entra en el directorio del proyecto:

- cd ~/proyectos/grupo01/proyecto-php

```
ub@forense-ai:~/proyectos/grupo01/proyecto-java$ cd ~/proyectos/grupo01/proyecto-php
```

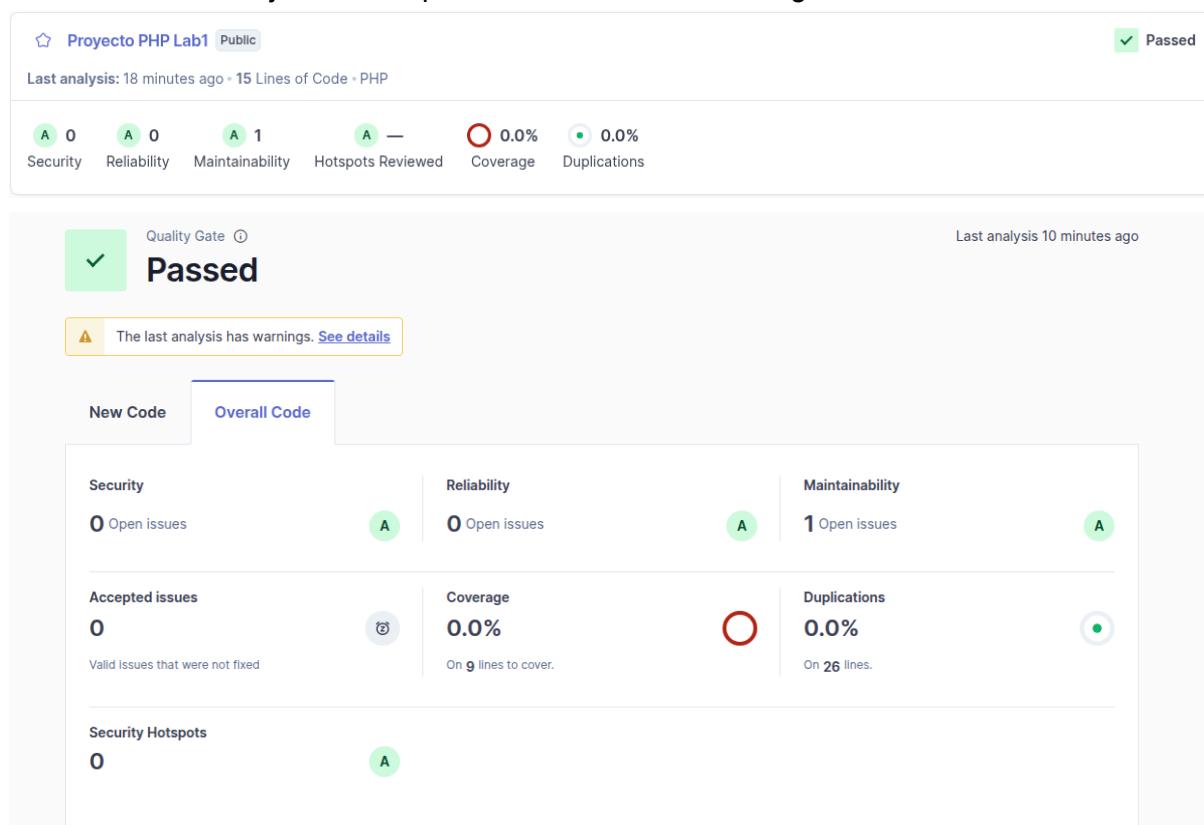
Ejecuta el contenedor del escáner:

```
sudo docker run --rm --network host \
-e SONAR_HOST_URL="http://127.0.0.1:9000" \
-e SONAR_TOKEN="sqa_49db28f546b10ab045d709f112df7a1a3c0ff0db" \
-v "$(pwd):/usr/src" \
-w /usr/src \
sonarsource/sonar-scanner-cli
```

3) Ver el resultado del análisis

Ir a: <http://localhost:9000>

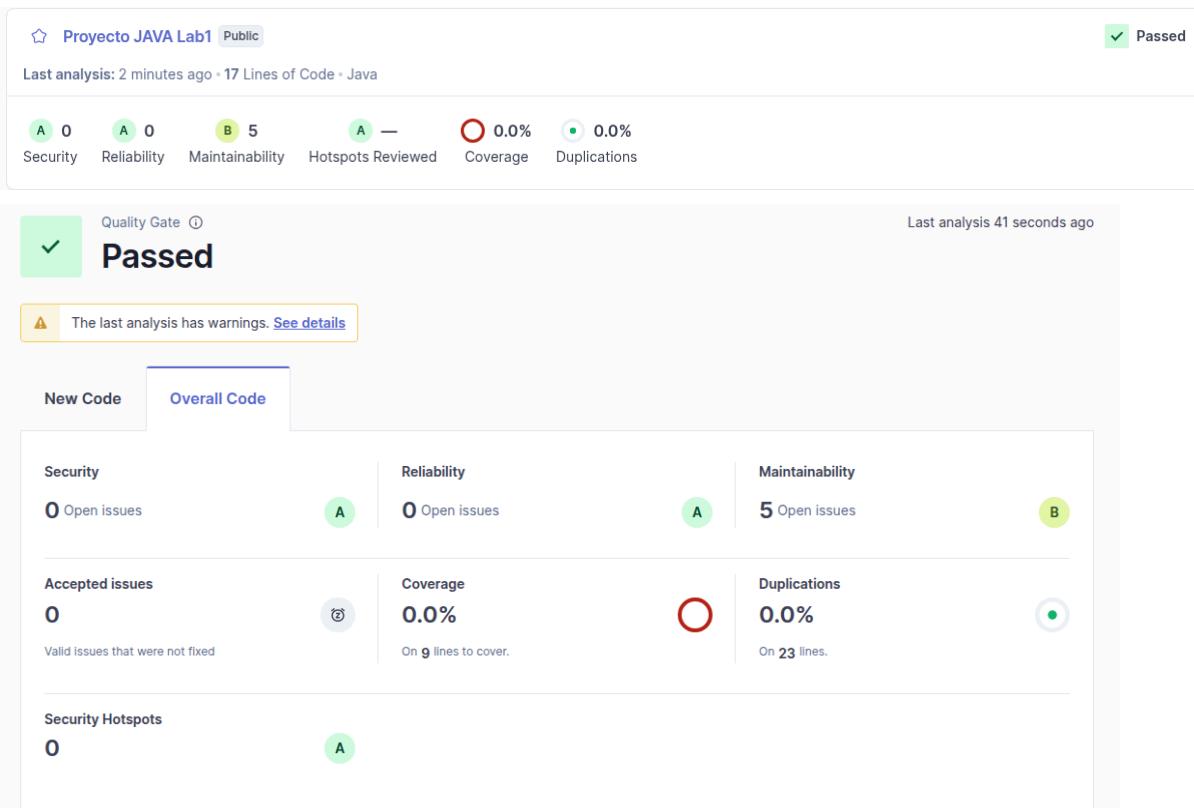
- Bugs: correspondientes a la dimensión “Reliability”, representan errores de codificación que provocan un funcionamiento anómalo del programa. La captura de pantalla muestra un valor de 0, lo que significa que no se han detectado errores.
- Vulnerabilities: correspondientes a la dimensión “Security”, representan los riesgos de seguridad presentes en el código. La captura de pantalla indica un valor de 0, lo que significa que no se han detectado vulnerabilidades de seguridad.
- Code Smells: Correspondiente a la dimensión Maintainability, denota código que se desvía de las mejores prácticas y es difícil de mantener. La captura de pantalla indica un valor de 1, lo que significa la presencia de un olor de código que requiere optimización.
- Maintainability Rating: Una evaluación compuesta basada en métricas como los olores de código. La captura de pantalla muestra una A, lo que indica una buena mantenibilidad general.
- Duplications: La proporción de segmentos de código duplicados es del 0,0 %, lo que indica que no se ha detectado ningún código duplicado.
- Métricas complementarias: Coverage (0,0 %), Hotspots Reviewed (0,0 %) y otros también reflejan otros aspectos de la calidad del código.



4) Analizar también el proyecto Java

- cd ~/proyectos/grupo01/proyecto-java

```
sudo docker run --rm --network host \
-e SONAR_HOST_URL="http://127.0.0.1:9000" \
-e SONAR_TOKEN="sqa_49db28f546b10ab045d709f112df7a1a3c0ff0db" \
-v "$(pwd):/usr/src" \
-w /usr/src \
sonarsource/sonar-scanner-cli
```



Maintainability: Cinco problemas sin resolver, calificados con una B. Esto indica cinco Code Smells, con una calificación de mantenibilidad de B. Se requiere una optimización específica.

Eliminar el código duplicado:

```
public static int suma(int a, int b){  
    return a + b;  
}  
  
//    public static int suma2(int a, int b){ // duplicación  
//        return a + b;  
//    }
```

Código muerto: Eliminar esta variable.

```

// int z = 100; // código muerto
System.out.println(suma(x, y));
}
}

```

Números codificados de forma rígida:

```

public static void main(String[] args){
    int x = 5;
    int y = 5;
}

```

Cambiar a constante

```

1 public class Main {
2
3     private static final int DEFAULT_VALUE = 5;
4
5     public static int suma(int a, int b){
6         return a + b;
7     }
8
9     // public static int suma2(int a, int b){ // duplicación
10    //     return a + b;
11    // }
12
13    public static void main(String[] args){
14        int x = DEFAULT_VALUE;
15        int y = DEFAULT_VALUE;
16
17        if(x == y){
18            System.out.println("Iguales");
19        }
20    }
}

```

Comparación débil:

```

if(x == y){ ←
    System.out.println("Iguales");
}

```

```

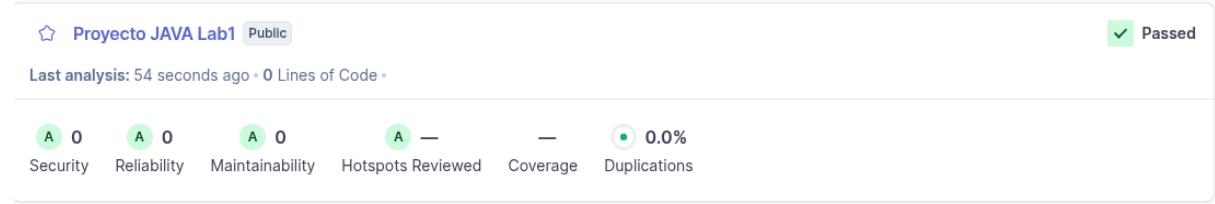
if(x === y){
    System.out.println("Iguales");
}

```

Ejecutando el contenedor del escáner una vez más

```
ub@forense-ai:~/proyectos/grupo01/proyecto-java$ sudo docker run --rm --network host \
-e SONAR_HOST_URL="http://127.0.0.1:9000" \
-e SONAR_TOKEN="sqa_49db28f546b10ab045d709f112df7a1a3c0ff0db" \
-v "$(pwd):/usr/src" \
-w /usr/src \
sonarsource/sonar-scanner-cli
```

Podemos observar que el recuento de Maintainability se ha reducido ahora a cero.



RETO FINAL — Auditoría real con SonarQube

Entra en el directorio raíz.

```
ub@forense-ai:~$ cd NovaDrive
ub@forense-ai:~/NovaDrive$ tree
.
├── db.php
└── index.php
privado
├── clases
│   └── Vehiculo.php
├── controlador
│   ├── cambiar_estado.php
│   ├── Controlador_buscador.php
│   ├── Gestion_Vehiculo.php
│   ├── limpiar_historial.php
│   ├── login.php
│   ├── logout.php
│   └── register.php
└── includes
    ├── footer.php
    ├── header.php
    ├── nav2.php
    └── nav.php
index.php
```

Crear sonar-project.properties

- nano sonar-project.properties

```
ub@forense-ai:~/NovaDrive$ nano sonar-project.properties
```

Contenido:

```
sonar.projectKey=novadrive-php
sonar.projectName=NovaDrive Proyecto Real
sonar.projectVersion=1.0
sonar.sources=
sonar.exclusions=public/img/**,public/fondo/**,public/img_user/**,public/css/**
sonar.sourceEncoding=UTF-8
```

```

GNU nano 7.2
sonar-project.properties *

sonar.projectKey=novadrive-php
sonar.projectName=NovaDrive Proyecto Real
sonar.projectVersion=1.0
sonar.sources=.
sonar.exclusions=public/img/**,public/fondo/**,public/img_user/**,public/css/**■
sonar.sourceEncoding=UTF-8

```

Generar un nuevo token:

sqa_2db86908f682f3588464abb4b10918aede141032

- User → My Account → Security → Generate Token

The screenshot shows the 'Generate Tokens' interface. At the top, there are fields for 'Name' (Enter Token Name), 'Type' (Select Token Type), and 'Expires in' (30 days). A success message says 'New token "novadrive-scan" has been created. Make sure you copy it now, you won't be able to see it again!' Below this, a table lists existing tokens:

Name	Type	Project	Last use	Created	Expiration
scanner-grupo01	Global		1 hour ago	February 13, 2026	March 15, 2026
novadrive-scan	Global		Never	February 13, 2026	March 15, 2026

Realizar un análisis exhaustivo del proyecto.

- cd ~/NovaDrive

Ejecute:

```

sudo docker run --rm --network host \
-e SONAR_HOST_URL="http://127.0.0.1:9000" \
-e SONAR_TOKEN="sqa_2db86908f682f3588464abb4b10918aede141032" \
-v "$(pwd):/usr/src" \
-w /usr/src \
sonarsource/sonar-scanner-cli \
-Dsonar.ws.timeout=120

```

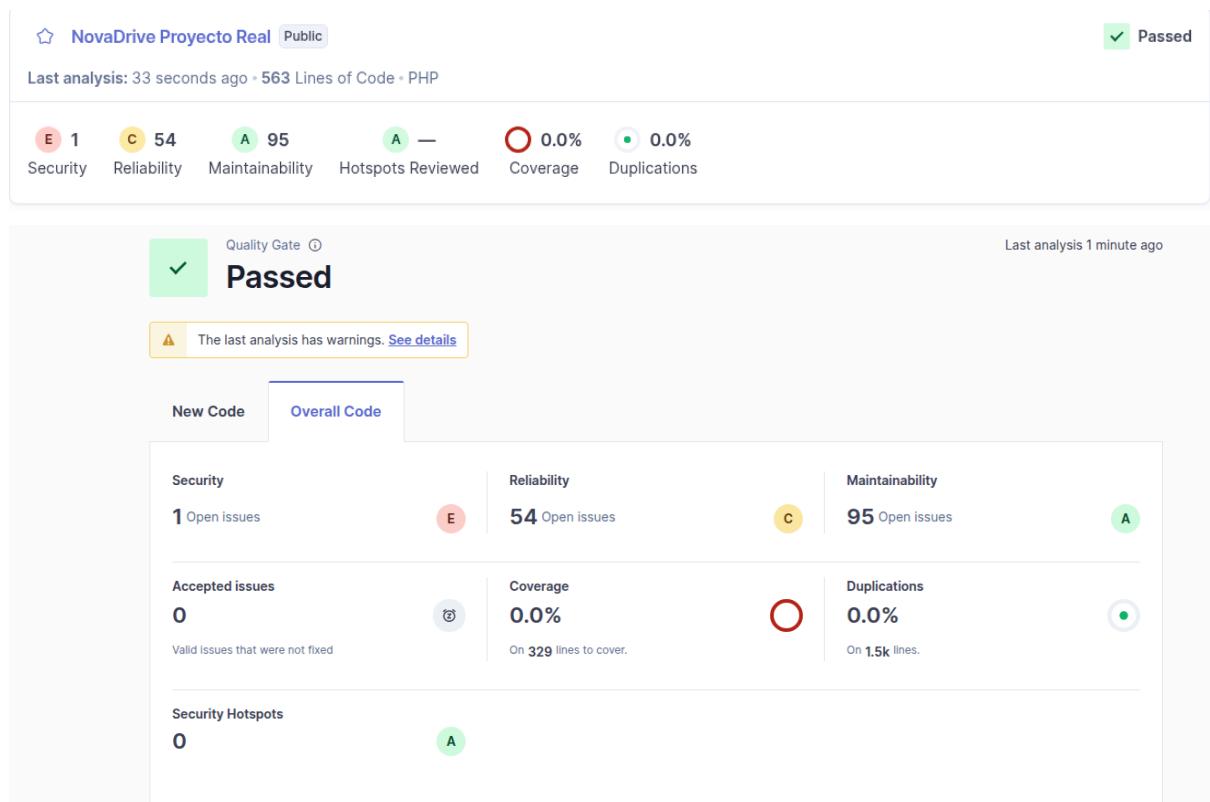
```

ub@forense-ai:~/NovaDrive$ sudo docker run --rm --network host \
-e SONAR_HOST_URL="http://127.0.0.1:9000" \
-e SONAR_TOKEN="sqa_2db86908f682f3588464abb4b10918aede141032" \
-v "$(pwd):/usr/src" \
-w /usr/src \
sonarsource/sonar-scanner-cli \
-Dsonar.ws.timeout=120

```

Ver los resultados del análisis

Navegador: <http://localhost:9000>



Quality Gate (PASSED)

A pesar de las vulnerabilidades de seguridad, los errores y los problemas de código, el proyecto cumple los requisitos mínimos para la puerta de calidad.

Nº Bugs

Total: 54

Medianos: 6

Bajos: 48

Problemas clave:

- Múltiples instancias de “include” en lugar de “include_once”, lo que supone un riesgo de inclusión de archivos duplicados
- El contenido inaccesible en algunas etiquetas <a> compromete la compatibilidad con los lectores de pantalla.
- Las etiquetas de los formularios no están asociadas a los controles; los elementos interactivos no nativos carecen de soporte auxiliar.
- Los elementos interactivos personalizados carecen de compatibilidad con eventos de teclado; las imágenes carecen de atributos alt.

Nº Vulnerabilities

Total: 1

Gravedad: Blocker

Problema específico: La conexión a la base de datos en “db.php” utiliza una contraseña vacía («\$password = “”), lo que supone un riesgo crítico para la seguridad al permitir el acceso directo a la base de datos.

Nº Code Smells

Tipos principales:

- Nombres de variables no estándar.
- Duplicación de código.
- Falta de comentarios.
- Rutas y configuraciones codificadas de forma rígida.

Calificación de mantenibilidad

Calificación: A

A pesar de los 95 problemas de código, la estructura general del código y la mantenibilidad obtienen la calificación más alta, lo que indica una lógica central clara que solo requiere una optimización menor.

Análisis

1. La conexión a la base de datos utiliza una contraseña en blanco (Bug de nivel de bloqueo):

La conexión a la base de datos supone un grave riesgo para la seguridad y requiere una solución inmediata.

Problema técnico

Cuando los usuarios predeterminados de la base de datos MySQL (como root) utilizan contraseñas en blanco, cualquier atacante con acceso al servidor puede iniciar sesión directamente a través del puerto de la base de datos (3306) y manipular todos los datos sin necesidad de autenticación. Además, la ausencia de restricciones de acceso para los usuarios de la base de datos dentro del proyecto amplifica significativamente el riesgo de acceso no autorizado debido a estas contraseñas en blanco.

The screenshot shows a code editor interface with the following details:

- Header: NovaDrive Proyecto Real > /db.php
- Toolbar: Open in IDE, See all issues in this file
- Code Area:

```
1 $servername = "localhost";
2 $username = "root";
3 $password = "";
4 $dbname = "coches_db";
5
6 $conn = new mysqli($servername, $username, $password, $dbname);
7
8 Add password protection to this database.
9
10 ?>
11
12 <!--
13 $servername = "localhost";
14 $username = "NovaDrive";
15 $password = "T9qSE0w414^|";
16 $dbname = "coches_db";
```
- Bottom: Line numbers 1 through 16, a '+' sign, and a '-' sign.

Riesgos potenciales

- **Seguridad:** riesgo extremadamente alto, ya que los atacantes podrían eliminar o manipular directamente los datos de los usuarios y la información de los vehículos, o incluso implantar código malicioso.

- **Cumplimiento normativo:** violación de las normas de seguridad de los datos; si se trata de información real de los usuarios, esto daría lugar a riesgos de incumplimiento de la normativa sobre violación de datos.
- **Repercusión en el negocio:** tras la manipulación de la base de datos, las funciones básicas del proyecto (inicio de sesión de los usuarios, listado de vehículos) quedarían totalmente inoperativas.

Añadir protección con contraseña:

The screenshot shows a comparison between 'Noncompliant code example' and 'Compliant solution'. The non-compliant example uses a plain password string, while the compliant solution uses an environment variable to store the password.

```

Noncompliant code example
$conn = new mysqli($servername, $username, ""); // Noncompliant

Compliant solution
$password = getenv('MYSQL_SECURE_PASSWORD');
$conn = new mysqli($servername, $username, $password);

```

Tras volver a escanear, podemos ver que el problema se ha resuelto.

The screenshot displays the results of a static code analysis for the 'NovaDrive Proyecto Real' project. It shows a 'Passed' status with a green checkmark. Key metrics include Security (A 0), Reliability (C 54), Maintainability (A 95), Hotspots Reviewed (A —), Coverage (0.0%), and Duplications (0.0%). The last analysis was performed 8 seconds ago on 564 Lines of Code in PHP.

2. Se han utilizado múltiples instancias de “include” en lugar de “include_once”:

Archivos como header.php, historial.php y user_home.php utilizan con frecuencia include dentro del proyecto, lo que indica un riesgo de inclusión de archivos duplicados.

The screenshot shows three separate instances of a code inspection warning. Each instance is titled 'Replace "include" with "include_once".' and has a 'Reliability' level of 'Low'. The first instance is labeled 'L2' with a 5min effort and 4 days ago. The second is 'L3' with a 5min effort and 4 days ago. The third is 'L4' with a 5min effort and 4 days ago. All three instances have 'Consistency' and 'No tags' associated with them and are marked as 'Open' with 'Not assigned' status.

Problema técnico

Cada llamada a “include” recarga el archivo. Si se incluye el mismo archivo varias veces, esto puede dar lugar a:

- Definiciones duplicadas de funciones/clases (error PHP: No se puede redeclarar la función);
- Reasignación de variables, anulando la lógica original;

- Conexiones a la base de datos creadas repetidamente, consumiendo recursos del servidor.

Riesgos potenciales

- **Mantenibilidad:** la complejidad de la lógica del código y las inclusiones duplicadas dificultan la localización de errores.
- **Estabilidad:** en escenarios de alta concurrencia, la creación repetida de conexiones a la base de datos puede agotar el grupo de conexiones del servidor, provocando fallos en el proyecto.
- **Rendimiento:** la carga repetida de archivos aumenta la sobrecarga de E/S del servidor, lo que reduce la velocidad de respuesta de la página.

Reemplaza “include” por “include_once”

Noncompliant code example

```
include 'code.php'; //Noncompliant; not a "_once" usage and not conditional
include $user.'_history.php'; // Noncompliant
require 'more_code.php'; // Noncompliant; not a "_once" usage
```

Compliant solution

```
require_once 'code.php';
if (is_member($user)) {
    include_once $user.'_history.php';
}
require_once 'more_code.php';
```

3. Las etiquetas de los formularios no están asociadas a los controles (Bug nivel medio)

La etiqueta <label> no hace referencia al ID del control correspondiente a través del atributo «for» ni encierra el elemento <input>, lo que da lugar a una asociación errónea entre la etiqueta y el campo de entrada.

A form label must be associated with a control.

Reliability Medium Intentionality + accessibility

Open Not assigned L124 · 5min effort = 4 days ago

```
119      <form action=".../controlador/Gestion_Vehiculo.php" method="POST" enctype="multipart/form-data">
120          <input type="hidden" name="action" value="editar">
121          <input type="hidden" name="vehiculo_id" id="edit_id">
122
123          <div class="form-group">
124              <label>Marca</label>
125
126              <input type="text" name="Marca" id="edit_marca" class="form-control" required>
127          </div>
128
129          <div class="form-group">
130              <label>Modelo</label>
131
132              <input type="text" name="Modelo" id="edit_modelo" class="form-control" required>
133          </div>
```

Problema técnico

- **Deficiencia de accesibilidad:** las tecnologías de asistencia, como los lectores de pantalla, no pueden identificar qué campo de entrada corresponde a la etiqueta “Marca”, lo que impide a los usuarios con discapacidad visual comprender la finalidad del campo de entrada.
- **Interrupción de la lógica de interacción:** al hacer clic en la etiqueta “Marca” no se enfoca automáticamente el campo de entrada subyacente, lo que se desvía del comportamiento nativo esperado del formulario.

Riesgos potenciales

- **Riesgos técnicos:** La validación de formularios y las pruebas automatizadas pueden fallar; las puntuaciones de SEO y las métricas de rendimiento de Lighthouse pueden disminuir; las modificaciones de iteraciones posteriores se vuelven propensas a errores.
- **Experiencia del usuario:** Los usuarios normales que hacen clic en la etiqueta no pueden enfocar el campo de entrada, lo que degrada la experiencia operativa.
- **Riesgos de mantenimiento:** La legibilidad del código se deteriora, lo que aumenta los costes de comprensión para los nuevos desarrolladores y complica la adaptación multilingüe.

Resuelva el problema utilizando el método de envolviendo el elemento `<input>` dentro de un elemento `<label>`.

```
Noncompliant code example
<input type="text" />
<label>Favorite food</label>

Compliant solution
<label>
  <input type="text" />
  Favorite food
</label>
```

4. Los elementos interactivos personalizados carecen de atributos de accesibilidad/eventos de teclado (Bug de prioridad media/baja)

- **Avoid non-native interactive elements**
Este `<div>` se utiliza para simular un menú de perfil de usuario en el que se puede hacer clic, pero no es un elemento interactivo nativo (como `<button>`) y carece de atributos de asistencia como `role`, lo que impide que los lectores de pantalla reconozcan su funcionalidad.
- **Add a 'onKeyPress|onKeyDown|onKeyUp' attribute to this <div> tag**
Este elemento solo vincula el evento `onclick` del ratón sin el correspondiente soporte para eventos del teclado, lo que impide que se activen acciones mediante el teclado (teclas Intro/Espacio).

Add a 'onKeyPress|onKeyDown|onKeyUp' attribute to this <div> tag.

Reliability Low

Open Not assigned L35 • 5min effort • 4 days ago

Intentionality accessibility +

Where is the issue? Why is this an issue? Activity More info

```

35 <div class="avatar-box" onclick="toggleUserMenu()">
    Avoid non-native interactive elements. If using native HTML is not possible, add an appropriate role and support for tabbing, mouse, keyboard, and touch inputs to an interactive content element.
</div>
36     
    Add an "alt" attribute to this image.
37     </div>
38

```

Problema técnico

Riesgo de compatibilidad: Las diferencias en la forma en que los navegadores analizan los elementos interactivos personalizados pueden hacer que esta funcionalidad falle en determinados entornos.

Riesgos potenciales

- Mantenibilidad: La lógica de interacción personalizada está fragmentada. Las modificaciones posteriores requieren el manejo simultáneo de eventos del ratón y del teclado, lo que aumenta la complejidad y los costes de mantenimiento del código.

5. Nombres de variables no estándar (code smell)

En la línea 11 del archivo privado/clases/Vehiculo.php, la propiedad privada de la clase \$user_id ha sido marcada como no conforme con las normas de nomenclatura. SonarQube requiere que coincida con la expresión regular `^[a-z][a-zA-Z0-9]*$`, lo que significa:

- El primer carácter debe ser una letra minúscula (a-z).
- Los caracteres siguientes deben ser solo alfanuméricos; no se permiten guiones bajos.

La presencia de un guion bajo (_) en `\$user_id` incumple esta regla.

Rename this field "\$user_id" to match the regular expression `^[a-z][a-zA-Z0-9]*$`.

Maintainability Low

Open Not assigned L11 • 2min effort • 4 days ago

Consistency convention +

```

7     private $ano;
8     private $color;
9     private $precio;
10    private $descripcion;
11    private $user_id;
12
13    // 构造函数 - Constructor
14    public function __construct($marca = "", $imagen = [], $modelo = "", $ano = "", $color = "", $precio = "", $descripcion = "") {
15        $this->setMarca(addslashes($marca));
16        $this->setImagen($imagen);
17        $this->setModelo(addslashes($modelo));
18        $this->setAño(addslashes($ano));
19        $this->setColor(addslashes($color));
20        $this->setPrecio(addslashes($precio));

```

Rename this field "\$user_id" to match the regular expression ^[a-zA-Z0-9]*\$.

Problema técnico

Convenciones de nomenclatura inconsistentes: el proyecto mezcla nombres basados en guiones bajos (por ejemplo, `'\$user_id'`) con los nombres alfanuméricos obligatorios, lo que socava la coherencia general del código.

- **Escasa legibilidad y mantenibilidad:** las convenciones de nomenclatura inconsistentes dificultan que otros desarrolladores comprendan la intención del código, lo que aumenta la dificultad del mantenimiento y la depuración posteriores.
- **Riesgos para la colaboración en equipo:** cuando varios desarrolladores colaboran, la nomenclatura inconsistente aumenta la sobrecarga de comunicación y puede introducir errores lógicos debido a la confusión de los nombres de las variables.

Riesgos potenciales

- **Mantenimiento:** el código se vuelve difícil de leer y modificar, lo que aumenta significativamente los costes de mantenimiento a largo plazo.
- **Estabilidad:** mayor riesgo de errores derivados de la confusión de nombres de variables.
- **Coherencia:** socava el estilo general del código base, contraviniendo las mejores prácticas modernas de desarrollo de PHP.

Solución

1. Cambiar el nombre de la propiedad: Cambiar el nombre de `'\$user_id` por un nombre compatible, como por ejemplo utilizando camelCase como `'\$userId`.
2. Actualizar las referencias globalmente: Actualizar de forma sincronizada todas las referencias a esta propiedad en todo el archivo Vehiculo.php, incluyendo:
 - Asignaciones dentro del constructor.
 - Métodos setter como `setUserId()`.
 - Código relacionado con este campo dentro de las operaciones de la base de datos.

```

class MyClass {
    $my_field;
}

```

Should be replaced with:

```

class MyClass {
    $myField;
}

```