



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto Unidad II**  
**“Mejoramiento de la Aplicación”**

Curso: Calidad y Pruebas de Software

Docente: Ing. Cuadros Quiroga, Patrick Jose

**Integrantes:**

<b>Albert Apaza Ccalle</b>	<b>(2021071075)</b>
<b>Ricardo Cutipa Gutierrez</b>	<b>(2021069827)</b>
<b>Jesus Huallpa Maron</b>	<b>(2021071085)</b>
<b>Erick Churacutipa Blas</b>	<b>(2020067578)</b>

**Tacna – Perú**  
**2024**

## **Resumen**

El sistema de gestión de inventario y reservas de SOSA e Hijas S.A.C. enfrenta desafíos críticos relacionados con la seguridad de los datos y la calidad del código. En un entorno empresarial dinámico y competitivo, la protección de la información sensible y la optimización del rendimiento son fundamentales para garantizar la eficiencia operativa y la confianza del cliente. Para abordar estos problemas, se propone la implementación de herramientas avanzadas de análisis de código como SonarQube y Snyk. Estas herramientas permitirán realizar una evaluación exhaustiva del código base, identificando y corrigiendo vulnerabilidades de seguridad, así como mejorando la estructura y legibilidad del código. El objetivo es asegurar un despliegue más seguro y robusto del sistema, reduciendo así el riesgo de exposición a amenazas cibernéticas y errores operativos, mientras se fortalece la estabilidad y la escalabilidad del sistema en su conjunto. Este enfoque proactivo no solo mejorará la seguridad del sistema, sino que también posicionará a SOSA e Hijas S.A.C. como un líder en la gestión eficiente y segura de inventarios y reservas en su sector.

## **Abstract**

The inventory and booking management system of SOSA e Hijas S.A.C. faces critical challenges related to data security and code quality. In a dynamic and competitive business environment, safeguarding sensitive information and optimizing performance are crucial for operational efficiency and customer trust. To address these issues, the implementation of advanced code analysis tools such as SonarQube and Snyk is proposed. These tools will enable a comprehensive assessment of the codebase, identifying and rectifying security vulnerabilities, while improving code structure and readability. The goal is to ensure a safer and more robust deployment of the system, reducing the risk of exposure to cyber threats and operational errors, while enhancing overall system stability and scalability. This proactive approach not only enhances system security but also positions SOSA e Hijas S.A.C. as a leader in efficient and secure inventory and booking management within its industry.

## **1. Antecedentes o introducción**

El "Sistema de Alquiler de Equipos para Minería y Construcción Civil" de SOSA e Hijas S.A.C. enfrenta desafíos significativos en términos de seguridad de datos y calidad del código. Las amenazas cibernéticas y los errores de software pueden comprometer la integridad de la información y afectar la experiencia del usuario. La implementación de herramientas avanzadas de análisis de código como SonarQube y Snyk se presenta como una solución crucial para identificar y mitigar estos riesgos, fortaleciendo la infraestructura tecnológica y garantizando operaciones más seguras y eficientes.

## **2. Título**

Sistema de Alquiler de Equipos para Minería y Construcción Civil para la empresa SOSA e Hijas S.A.C.

## **3. Autores**

- Albert Apaza Ccalle (2021071075)
- Erick Churacutipa Blas (2020067578)
- Ricardo Cutipa Gutiérrez (2021069827)
- Jesus Huallpa Maron (2021071085)

El equipo de desarrollo está conformado por estudiantes del séptimo ciclo de la carrera de Ingeniería de Sistemas de la Universidad Privada de Tacna. Cada miembro aporta habilidades y conocimientos específicos para llevar a cabo la implementación del proyecto.

## **4. Planteamiento del problema**

### **4.1. Problema**

El sistema de gestión de inventario y reservas de SOSA e Hijas S.A.C. enfrenta desafíos críticos relacionados con la seguridad de los datos y la calidad del código. Estos desafíos incluyen la presencia de vulnerabilidades de seguridad, prácticas de codificación deficientes y un rendimiento subóptimo del sistema. Sin abordar adecuadamente estos problemas, el sistema corre el riesgo de enfrentar brechas de seguridad significativas, errores operativos frecuentes y una experiencia del usuario insatisfactoria. La falta de una estrategia robusta de análisis de código podría dificultar la detección temprana de problemas, incrementando así los costos y la complejidad en etapas posteriores del desarrollo y mantenimiento.

### **4.2. Justificación**

La implementación de herramientas avanzadas de análisis de código como SonarQube y Snyk es fundamental para abordar los desafíos identificados en el sistema de SOSA e Hijas S.A.C. Estas herramientas permitirán una evaluación exhaustiva del código base, identificando vulnerabilidades de seguridad y áreas de mejora en la calidad del código. Al mejorar la seguridad y eficiencia del sistema, se garantiza una experiencia de usuario más segura y estable. Además, reducirá el riesgo de exposición a posibles amenazas cibernéticas y errores de funcionamiento, fortaleciendo así la confianza de los usuarios y la robustez del sistema en su conjunto.

#### **4.3. Alcance**

El proyecto se centrará en la implementación y configuración de SonarQube y Snyk para realizar un análisis detallado del código del sistema de gestión de inventario y reservas de SOSA e Hijas S.A.C. Se identificarán y corregirán vulnerabilidades de seguridad críticas, así como prácticas de codificación deficientes que afecten la estabilidad y el rendimiento del sistema. Además, se capacitará al equipo de desarrollo en el uso efectivo de estas herramientas, proporcionando recursos y orientación para maximizar su impacto. Se documentará exhaustivamente el proceso de análisis y mejora para asegurar un mantenimiento continuo y eficiente del sistema en el futuro.

### **5. Objetivos**

#### **5.1. General**

El objetivo general de este proyecto es mejorar la seguridad, calidad del código y eficiencia operativa del sistema de gestión de inventario y reservas de SOSA e Hijas S.A.C. mediante la implementación de herramientas avanzadas de análisis de código como SonarQube y Snyk. Se busca asegurar un despliegue más seguro y robusto del sistema, mejorando así la experiencia del usuario y reduciendo el riesgo de vulnerabilidades y errores operativos.

#### **5.2. Específicos**

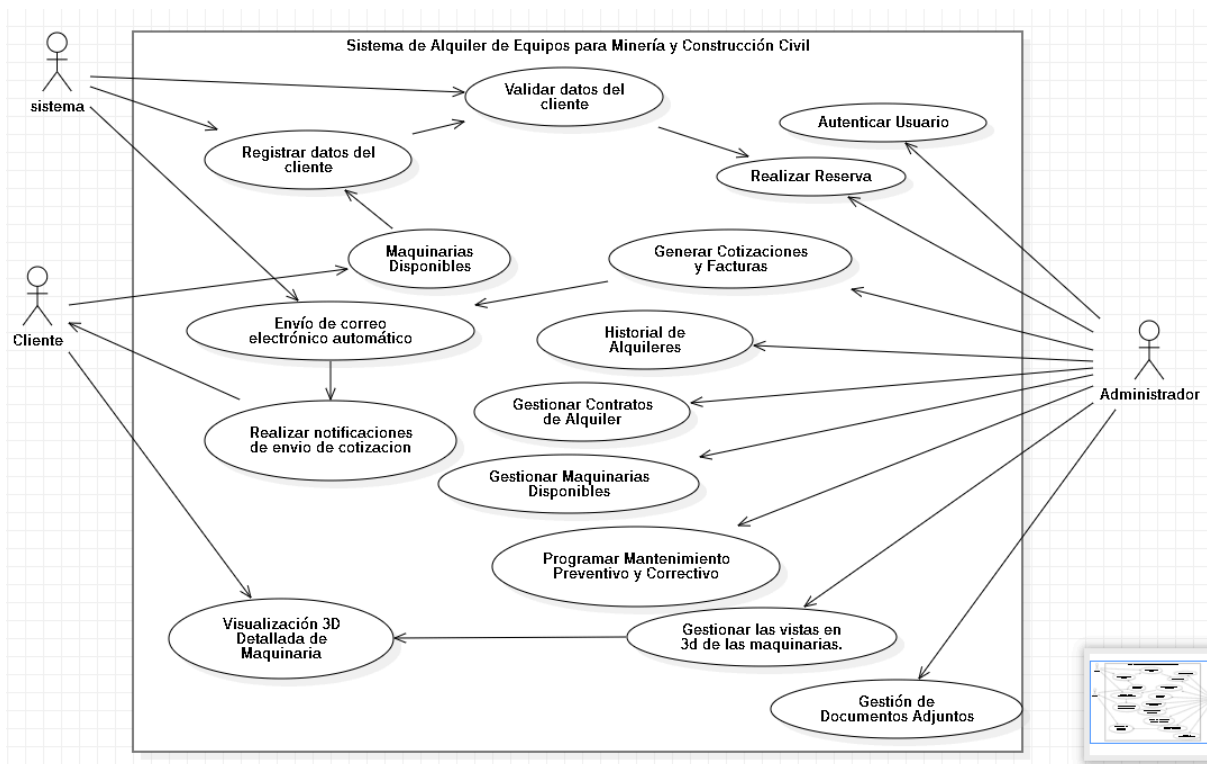
- Identificar y corregir vulnerabilidades de seguridad presentes en el código del sistema utilizando SonarQube y Snyk.
- Mejorar la calidad del código mediante la detección y corrección de prácticas de codificación deficientes y patrones de diseño problemáticos.

- Capacitar al equipo de desarrollo en el uso efectivo de SonarQube y Snyk, proporcionando orientación y recursos adecuados.
- Documentar de manera exhaustiva el proceso de análisis y mejora del sistema para facilitar su mantenimiento continuo y el uso eficiente de las herramientas de análisis de código.

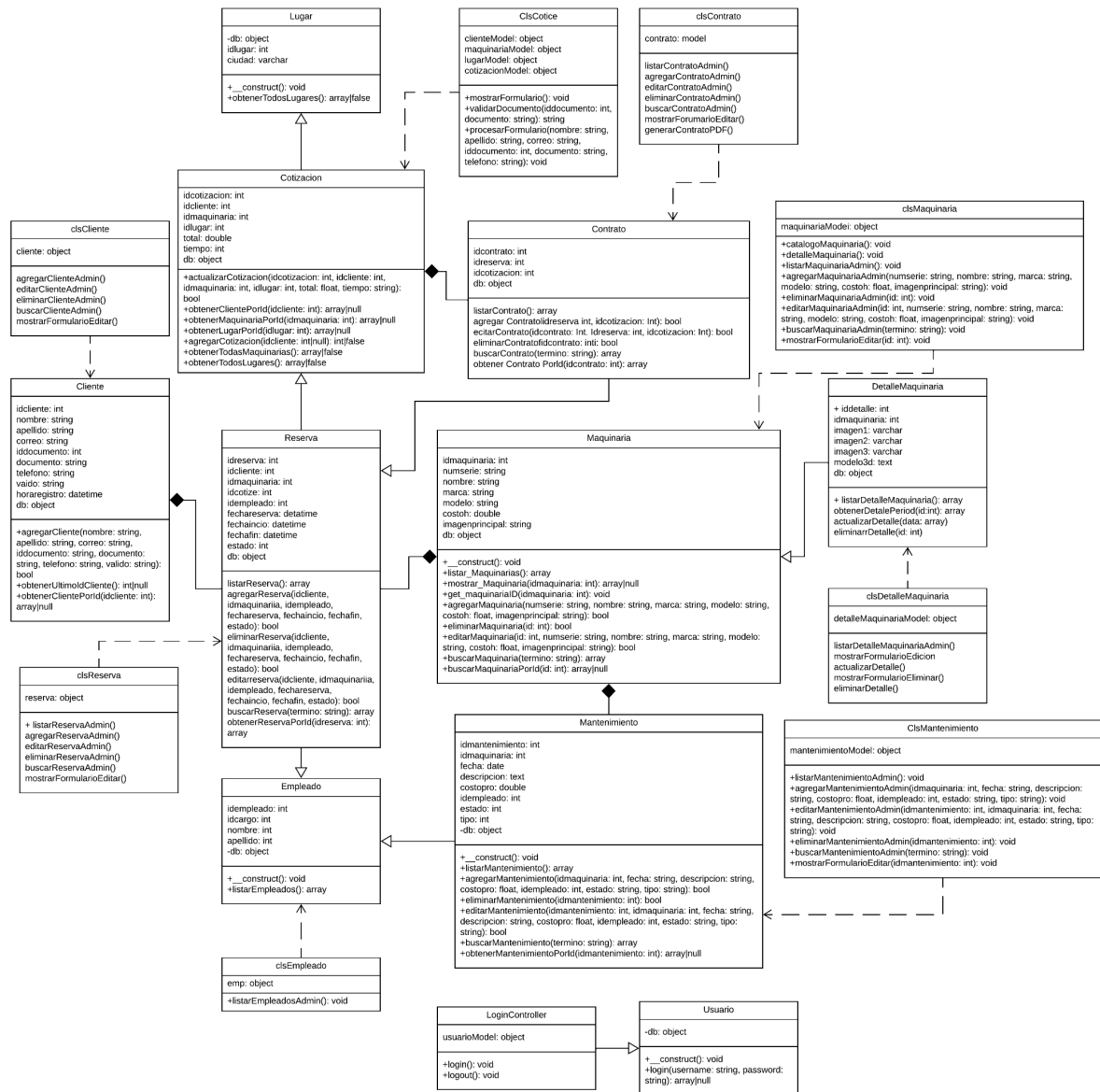
## 6. Referentes teóricos

**Diagramas de Casos de Uso, Diagrama de Clases, Diagrama de Componentes y Arquitectura.**

### Diagrama de Casos de Uso



**Enlace:** <https://lucid.app/publicSegments/view/62c80f97-2e70-4e2b-a926-d429a700f8a9/image.png>





## Diagrama de Componentes y Arquitectura:

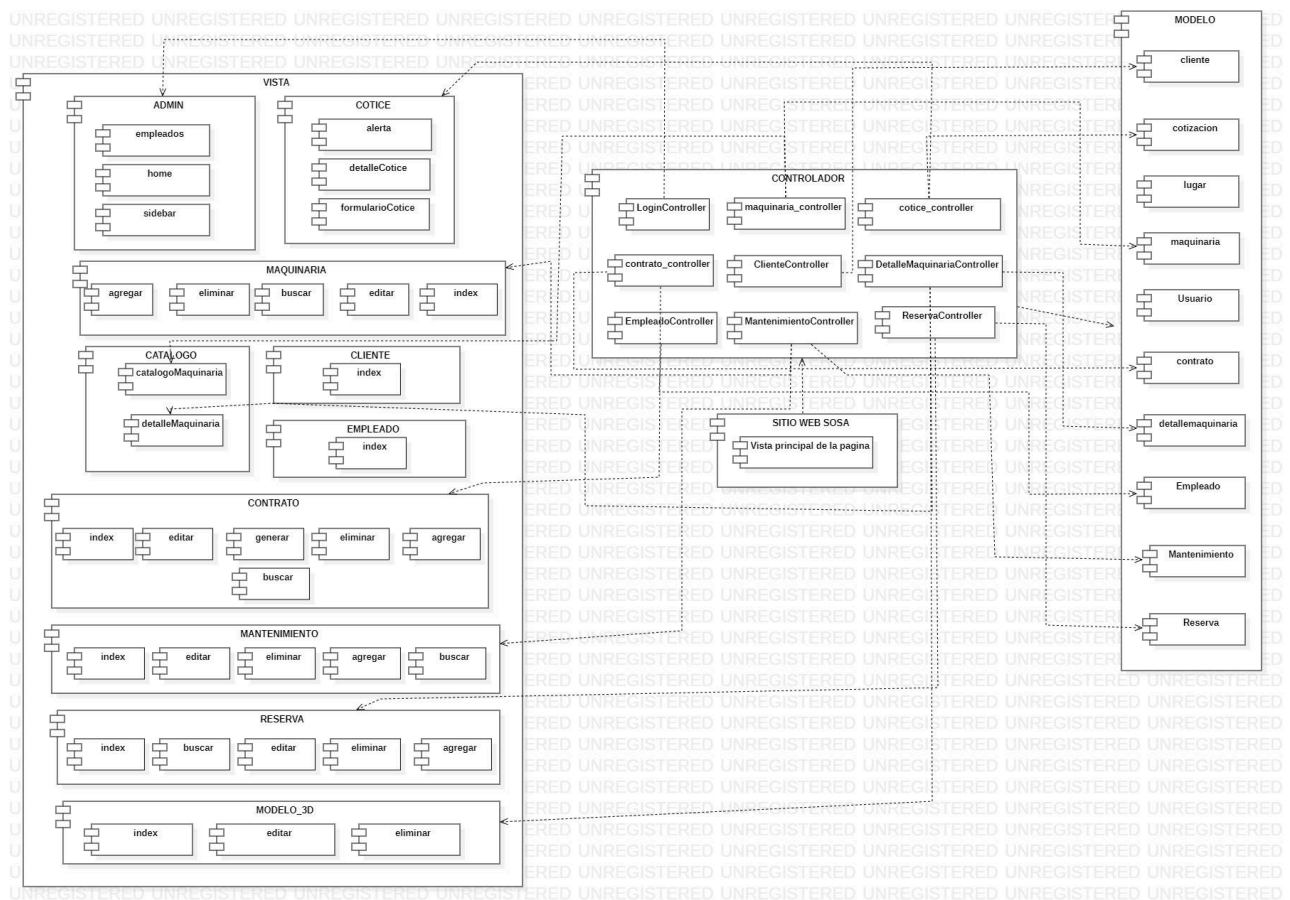


Diagrama: <https://ibb.co/NKBMrqv>

## 7. Desarrollo de la propuesta

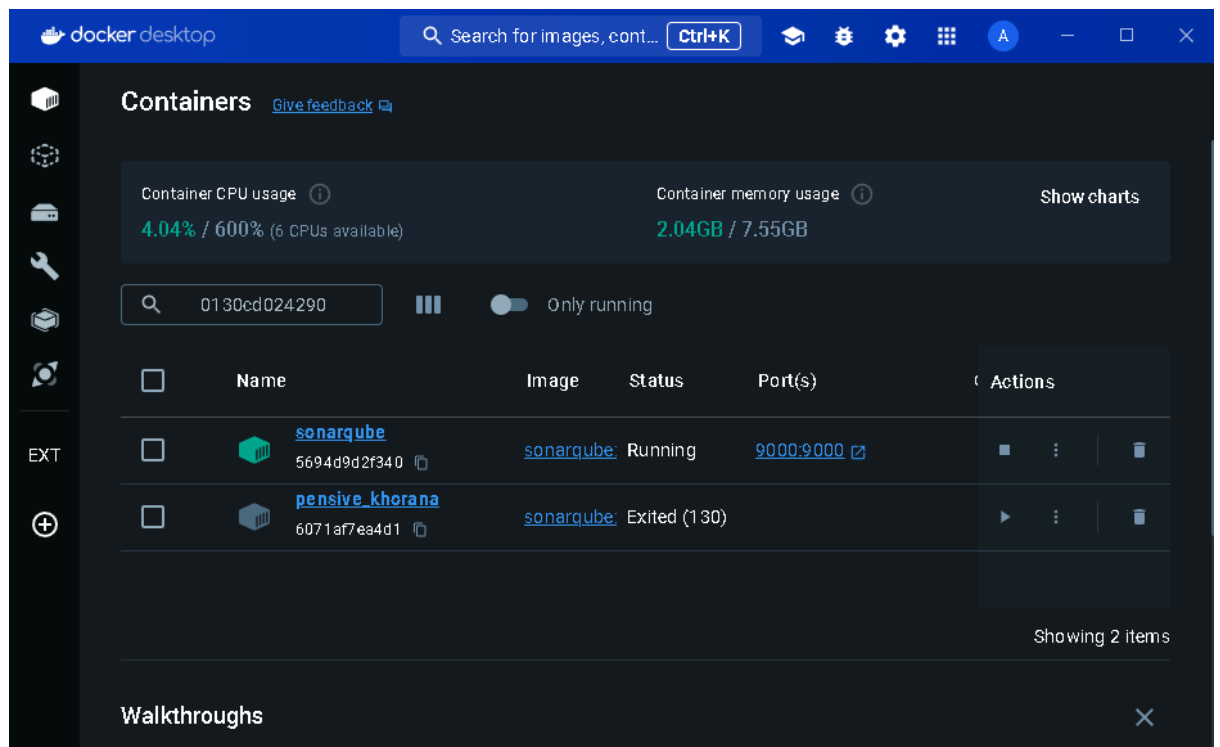
### 7.1. Tecnología de información

#### SonarQube

SonarQube es una plataforma de análisis de código abierto que identifica y corrige problemas de calidad en el código fuente, mejorando así su mantenibilidad, seguridad y eficiencia. Permite realizar análisis estático de código, encontrar vulnerabilidades, errores y malas prácticas, y proporciona informes detallados para facilitar la mejora continua del desarrollo de software.

#### Configuración del Docker:

"Para empezar, es esencial instalar Docker y SonarQube en tu entorno de desarrollo. Además, asegúrate de configurar correctamente SonarQube para su posterior uso."



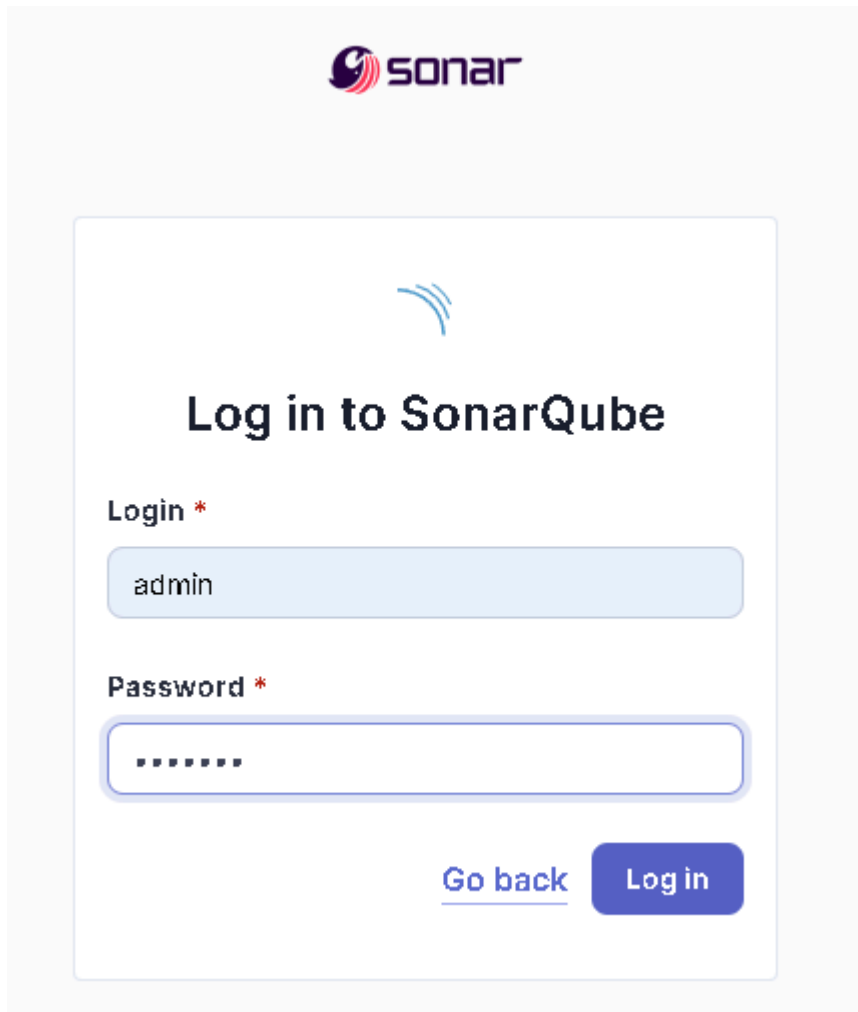
**Image x:** en esta imagen se puede apreciar el docker desktop y el contenedor de sonarqube configurado correctamente.  
**Creación Propia**

"Una vez que el contenedor de SonarQube esté en funcionamiento, accedemos a través de la URL localhost:9000."



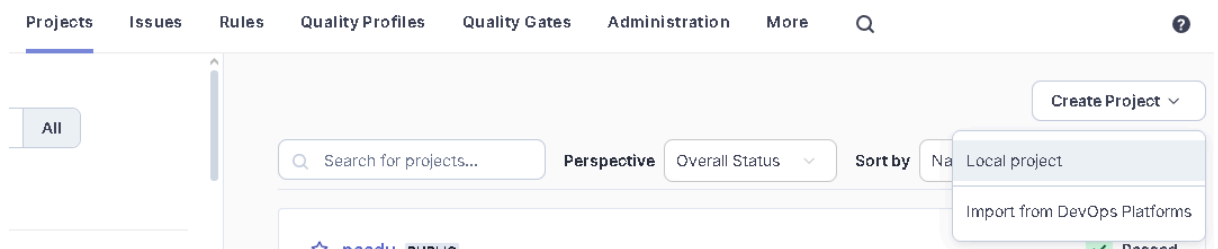
**Image x:** en esta imagen se puede apreciar cómo ingresamos a sonarqube.  
**Creación Propia**

"Ingresamos las credenciales de administrador y la contraseña correspondiente."



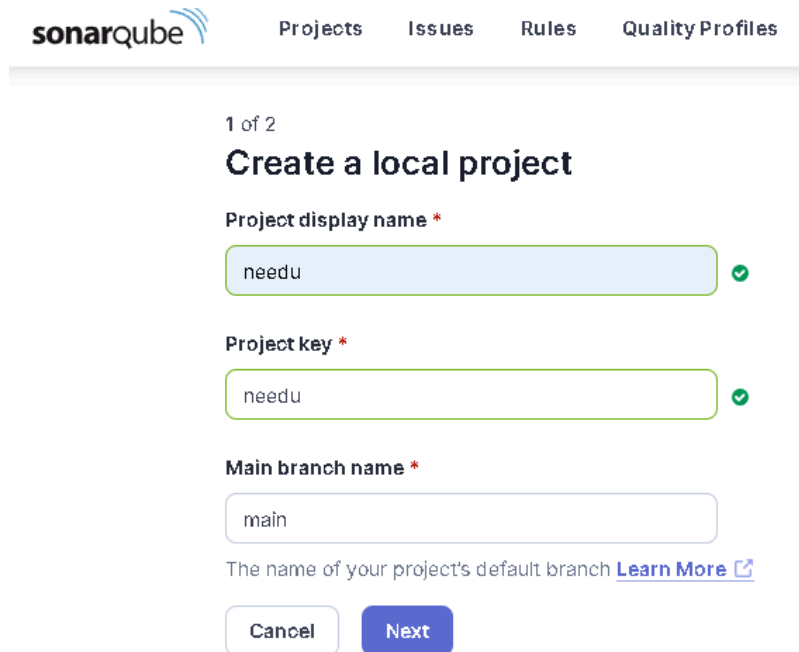
**Image x:** en esta imagen sonarqube nos pide el usuario y contraseña para ingresar.  
**Creación Propia**

"Posteriormente, en la sección de proyectos, seleccionamos la opción 'Proyecto local'."



**Image x:** en esta imagen se puede apreciar cómo se crea un nuevo proyecto para analizar con sonarqube.  
**Creación Propia**

"A continuación, asignamos un nombre al proyecto, su clave de proyecto y el nombre de la rama principal por defecto."



The screenshot shows the SonarQube interface for creating a local project. At the top, the SonarQube logo is on the left, and navigation links for 'Projects', 'Issues', 'Rules', and 'Quality Profiles' are on the right. Below the navigation bar, the page is titled '1 of 2' and 'Create a local project'. The form contains three required fields: 'Project display name \*' with the value 'needu', 'Project key \*' with the value 'needu', and 'Main branch name \*' with the value 'main'. Each of the first two fields has a green checkmark icon to its right. Below the 'Main branch name' field, there is a text label 'The name of your project's default branch' followed by a 'Learn More' link with an external link icon. At the bottom of the form, there are two buttons: 'Cancel' and 'Next'.

**Image x:** en esta imagen se puede apreciar que para crear un proyecto nos piden cosas como el nombre , key del proyecto.  
**Creación Propia**

"En el siguiente paso, elegimos la opción 'Usar la configuración global'."

2 of 2

## Set up project for Clean as You Code

The new code definition sets which part of your code will be considered in project, enabling you to follow the Clean as You Code methodology. Learn more

### Choose the baseline for new code for this project

☒ Use the global setting

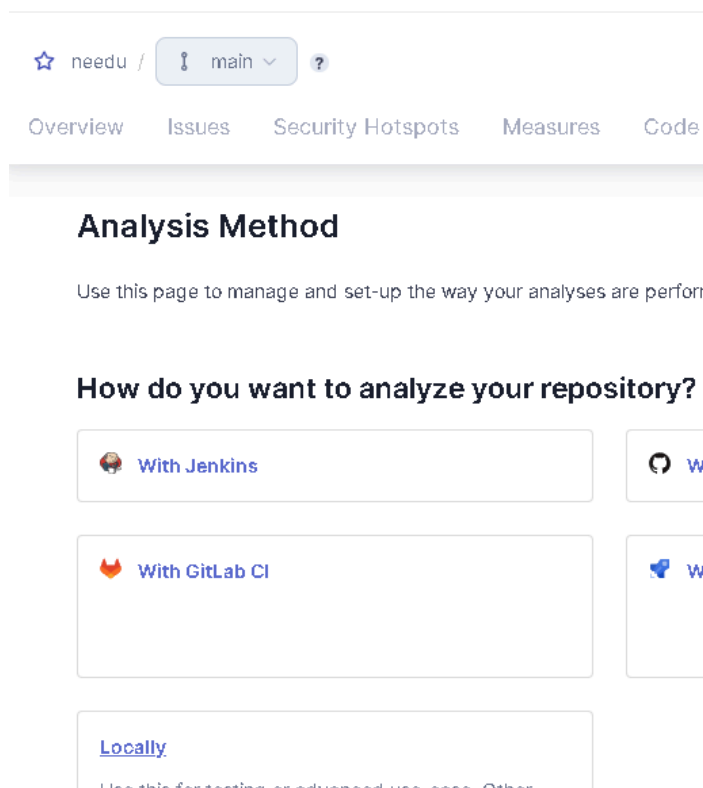
#### Previous version

Any code that has changed since the previous version is considered in project. Recommended for projects following regular versions or releases.

**Image x:** en esta imagen se puede apreciar las configuraciones disponibles para analizar el proyecto en sonarqube.

**Creación Propia**

"En el siguiente paso, seleccionamos la opción 'Localmente'."



**Image x:** en esta imagen se puede apreciar los diferentes repositorios para analizar un proyecto en sonarqube.

**Creación Propia**

"Generamos un token de acceso y guardamos el código proporcionado."

**Image x:** en esta imagen se puede apreciar cómo se crea un token para analizar el proyecto manualmente en sonarqube.

### **Creación Propia**

"Seleccionamos la opción 'Other' (para JavaScript, TypeScript, Go, Python, PHP, etc.), y luego la opción de Windows. Copiamos el comando proporcionado y lo guardamos."

## 2 Run analysis on your project

What option best describes your build?

Maven Gradle .NET Other (for JS, TS, Go, Python, PHP, ...)

What is your OS?

Linux Windows macOS

Download and unzip the Scanner for Windows

Visit the [official documentation of the Scanner](#) to download the latest version, and add the `bin` directory to the `%PATH%` environment variable

Execute the Scanner

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder.

```
sonar-scanner.bat -D"sonar.projectKey=needu" -D"sonar.sources=" -D"sonar.host.url=http://localhost:9000" -D"sonar.token=sqp_1d0cf22e4de9cd4c5601fa83d7495a3c4a0f4db9"
```

Copy

**Image x:** en esta imagen se puede apreciar como correr el análisis de nuestro proyecto sonarqube.

### Creación Propia

"En la raíz del proyecto seleccionado, creamos un archivo llamado 'sonar-project.properties'. En este archivo, proporcionamos información clave como el nombre del proyecto, el token de acceso, la ubicación de los archivos a analizar, entre otros."

```
sonar-project.properties
# Configuración del proyecto
sonar.projectKey=needu
sonar.projectName=needu
sonar.projectVersion=1.0
# Rutas de código fuente
sonar.sources=C:/Users/OVALTECH/Dropbox/Mi PC (KONOR-RA9)/Downloads/needu
# Configuración del servidor de SonarQube
sonar.host.url=http://localhost:9000
sonar.login=sqp_debd174778d983b6cfbbc6684b6b302d0fdf1931
sonar.java.binaries=build/web/WEB-INF/classes
```

**image x:** en esta imagen se puede apreciar como se configuro el archivo

"sonar-project.properties"

### Creación Propia

"Luego, ejecutamos el comando que habíamos copiado previamente."

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

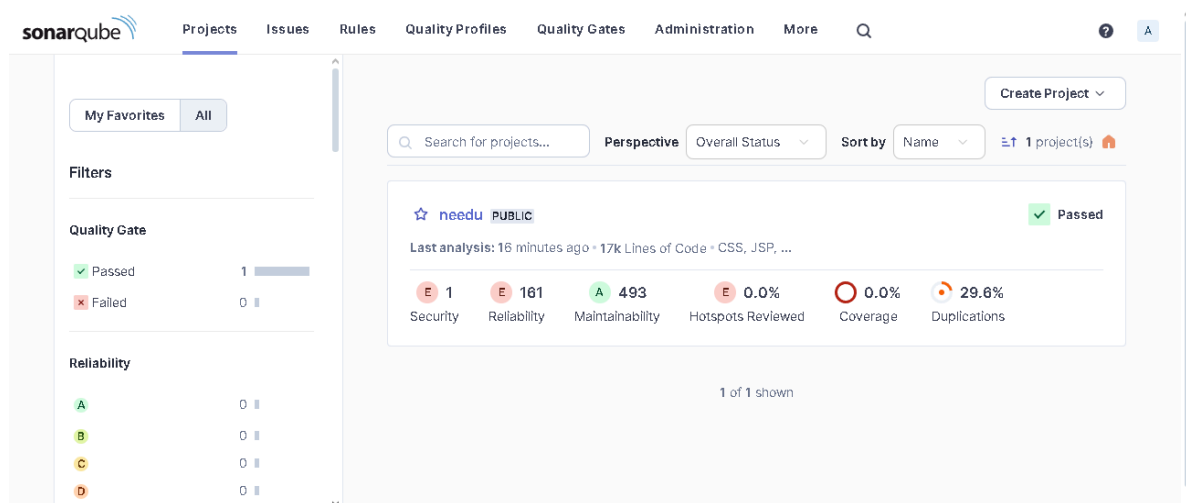
PS C:\Users\OVALTECH\Dropbox\Mi PC (KONOR-RA9)\Downloads\needu> sonar-scanner.bat -D"sonar.proj
ectKey=needu" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.token=sqp_d
ebd174778d983b6cfbbc6684b6b302d0fdf1931"
INFO: Scanner configuration file: C:\calidad\sonar\sonar-scanner-5.0.1.3006-windows\bin\..\conf
\sonar-scanner.properties
INFO: Project root configuration file: C:\Users\OVALTECH\Dropbox\Mi PC (KONOR-RA9)\Downloads\ne
edu\sonar-project.properties
INFO: SonarScanner 5.0.1.3006
INFO: Java 17.0.7 Eclipse Adoptium (64-bit)
INFO: Windows 10 10.0 amd64
INFO: User cache: C:\Users\OVALTECH\sonar\cache
INFO: Analyzing on SonarQube server 10.5.0.89998
INFO: Default locale: "es_ES", source code encoding: "UTF-8"
INFO: Load global settings
INFO: Load global settings (done) | time=71ms
INFO: Server id: 147B411E-AY8eGUue-j4jMHQL3XP6

SonarLint focus: overall code
```

*image: en esta imagen ejecutamos el comando para comenzar a escanear el proyecto*

### **creación propia**

"¡Listo! Ahora deberías poder ver los resultados del análisis de nuestro proyecto 'needu' en la plataforma de SonarQube."

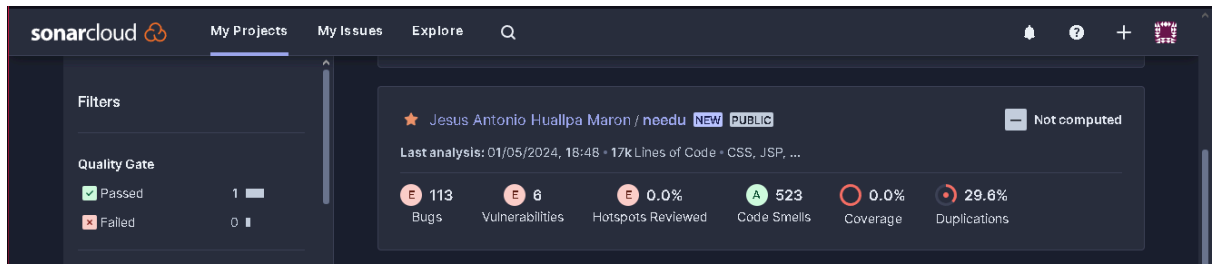


*image: en la imagen se aprecia el resultado del análisis en sonarqube*

### **creación propia**

también se realizó el análisis en sonarcloud para ver la diferencia en este caso como nos centramos en las vulnerabilidades podemos ver una clara diferencia entre sonarcloud con 6 vulnerabilidades y sonarqube con 1 sola vulnerabilidad en security





*image: en la imagen se aprecia el análisis del proyecto en sonarcloud*

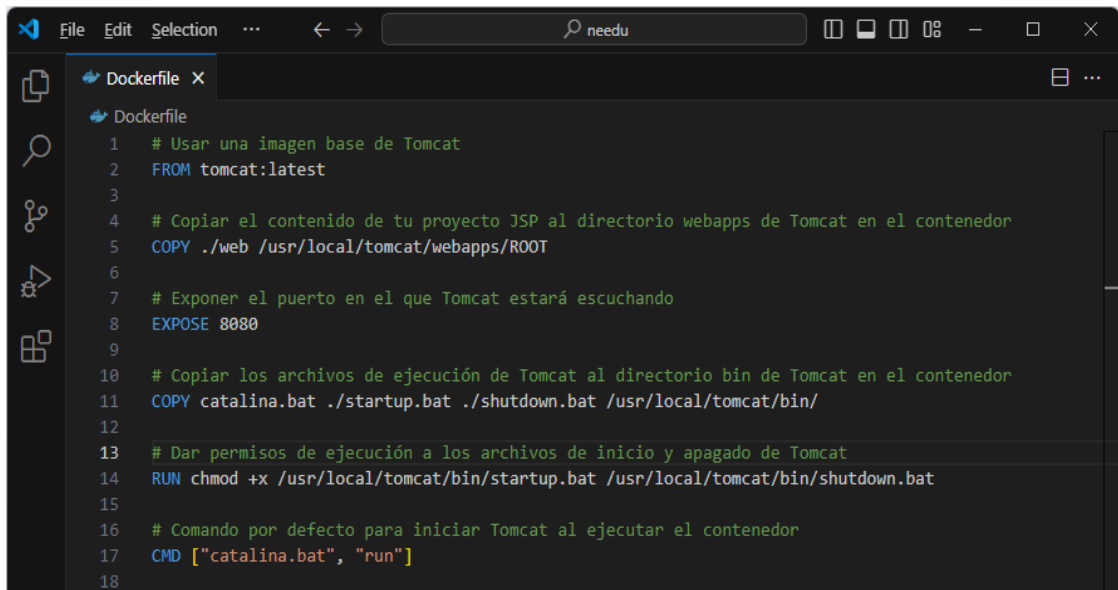
### **creación propia**

## **Snyk:**

Comenzamos creando una cuenta en Snyk para realizar análisis de seguridad en nuestra aplicación. Utilizamos Snyk para identificar vulnerabilidades en nuestras dependencias y para garantizar que nuestro código sea seguro. Además, utilizamos la integración de Snyk con Docker para analizar las imágenes de Docker que creamos para nuestra aplicación.

## **Configuración del entorno Docker:**

Para asegurarnos de que nuestra aplicación se ejecute de manera segura en un entorno Docker, creamos un Dockerfile. En este Dockerfile, utilizamos una imagen base de Tomcat y copiamos el contenido de nuestro proyecto JSP al directorio webapps de Tomcat en el contenedor. Además, copiamos los archivos de ejecución de Tomcat al directorio bin de Tomcat en el contenedor y le dimos los permisos adecuados.



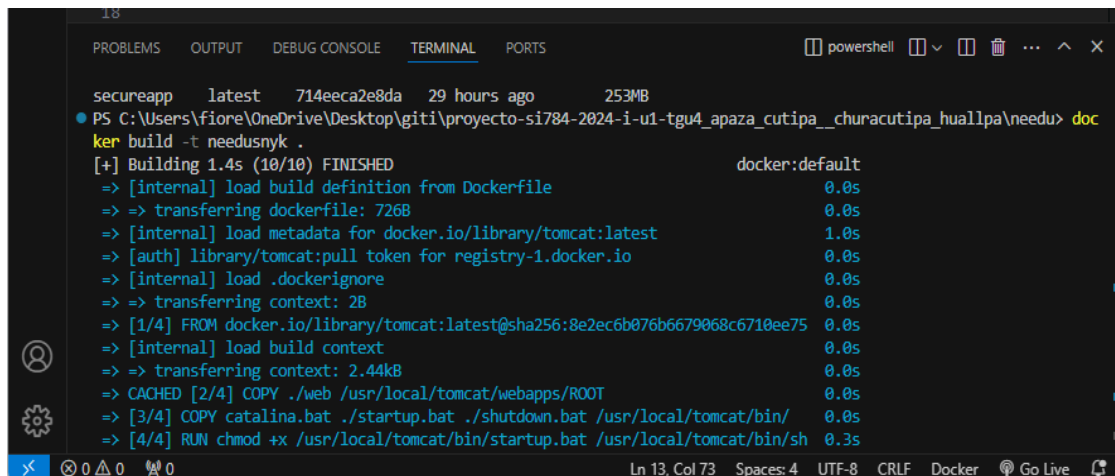
```
1 # Usar una imagen base de Tomcat
2 FROM tomcat:latest
3
4 # Copiar el contenido de tu proyecto JSP al directorio webapps de Tomcat en el contenedor
5 COPY ./web /usr/local/tomcat/webapps/ROOT
6
7 # Exponer el puerto en el que Tomcat estará escuchando
8 EXPOSE 8080
9
10 # Copiar los archivos de ejecución de Tomcat al directorio bin de Tomcat en el contenedor
11 COPY catalina.bat ./startup.bat ./shutdown.bat /usr/local/tomcat/bin/
12
13 # Dar permisos de ejecución a los archivos de inicio y apagado de Tomcat
14 RUN chmod +x /usr/local/tomcat/bin/startup.bat /usr/local/tomcat/bin/shutdown.bat
15
16 # Comando por defecto para iniciar Tomcat al ejecutar el contenedor
17 CMD ["catalina.bat", "run"]
18
```

***Image x:** El siguiente Dockerfile describe el proceso de construcción de la imagen "needusnyk", la cual se utiliza para desplegar una aplicación web desarrollada en JSP sobre un contenedor de Tomcat.*

### **Creación Propia**

## **Creación del contenedor**

docker build -t needusnyk



```
secureapp latest 714eeca2e8da 29 hours ago 253MB
PS C:\Users\Fiore\OneDrive\Desktop\giti\proyecto-si784-2024-i-u1-tgu4_apaza_cutipa_churacutipa_huallpa\needu> docker build -t needusnyk .
[+] Building 1.4s (10/10) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 726B 0.0s
=> [internal] load metadata for docker.io/library/tomcat:latest 1.0s
=> [auth] library/tomcat:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/4] FROM docker.io/library/tomcat:latest@sha256:8e2ec6b076b6679068c6710ee75 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 2.44kB 0.0s
=> CACHED [2/4] COPY ./web /usr/local/tomcat/webapps/ROOT 0.0s
=> [3/4] COPY catalina.bat ./startup.bat ./shutdown.bat /usr/local/tomcat/bin/ 0.0s
=> [4/4] RUN chmod +x /usr/local/tomcat/bin/startup.bat /usr/local/tomcat/bin/sh 0.3s
```

***Image x:** En la terminal de Visual Studio Code, desde el directorio del proyecto donde reside el Dockerfile, se ejecuta el comando `docker build -t needusnyk ..` Esto construye una imagen Docker llamada "needusnyk" utilizando las instrucciones especificadas en el Dockerfile presente en ese directorio. Una vez completada la construcción, esta imagen estará lista para implementar la aplicación.*

### **Creación Propia**

## **Análisis de seguridad de la aplicación sin contenedor Docker:**

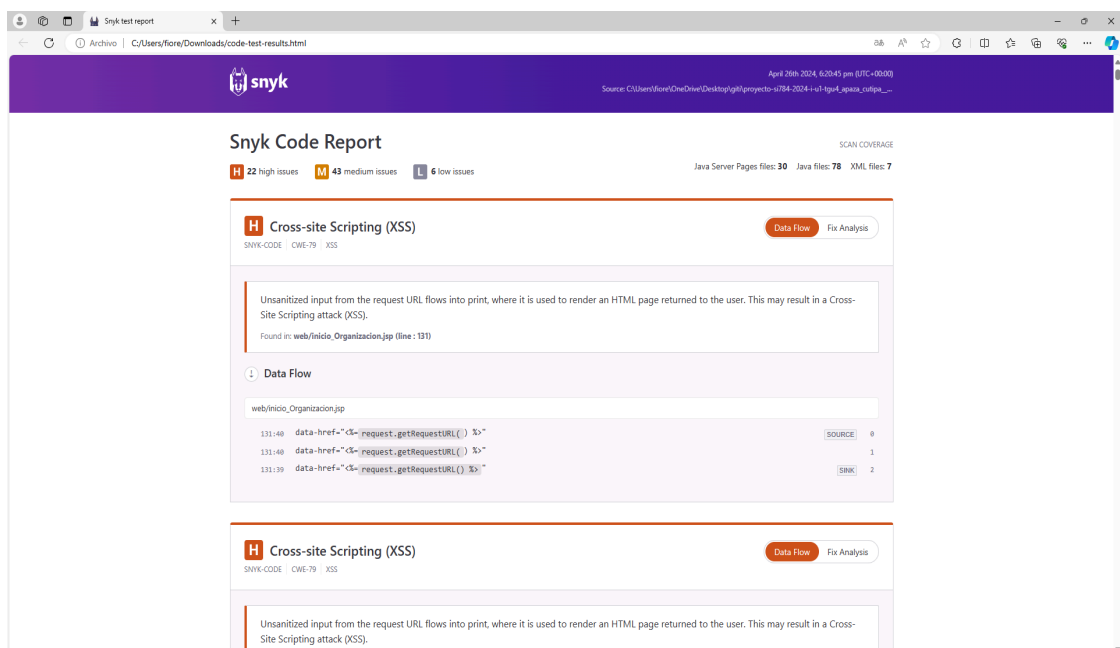
Utilizamos Snyk para realizar un análisis de seguridad de nuestra aplicación..

Ejecutamos el comando :

```
snyk-win.exe code test --json
```

Con este comando se analizará nuestro código y luego pasamos los resultados al comando :

```
snyk-to-html -o code-test-results.html
```



**Image x:** Salida del archivo html, creado a partir del comando presentado con anterioridad, se visualiza las recomendaciones sobre vulnerabilidades, de nuestro proyecto .

### **Creación Propia**

Para generar un informe HTML con los resultados del análisis de seguridad.

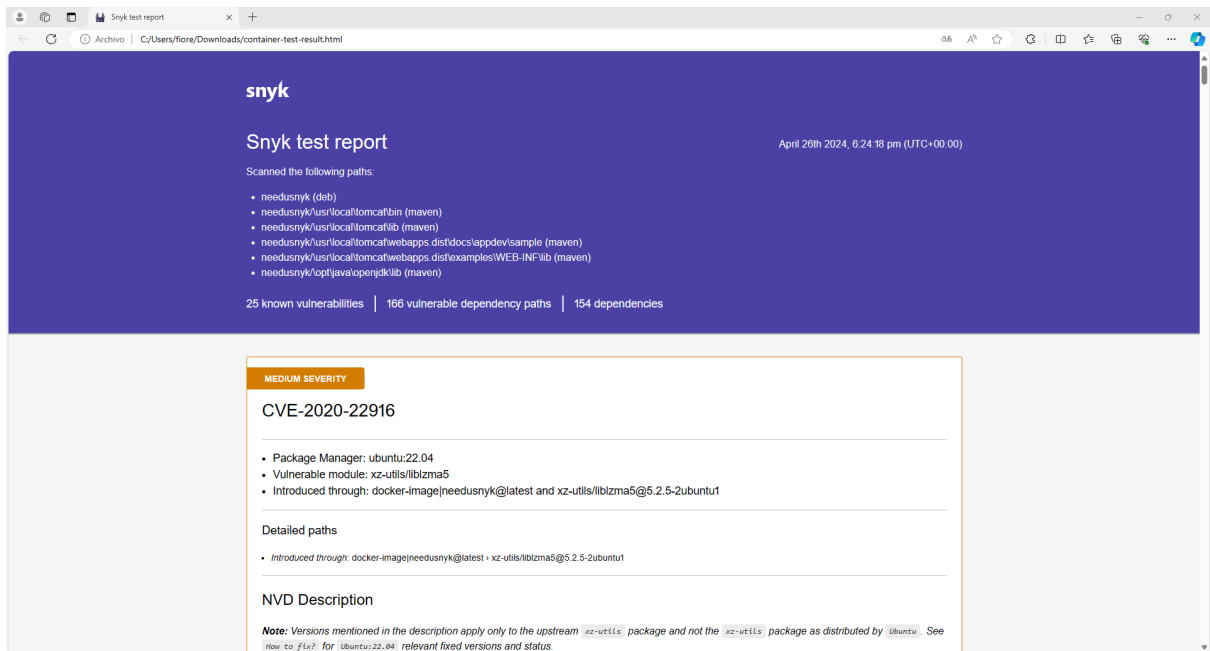
### **Análisis de seguridad de la aplicación con contenedor Docker:**

Utilizando la imagen creada con docker, se realizara un análisis de seguridad de nuestra aplicación con contenedor Docker.

Ejecutamos el comando

```
snyk-win.exe container test needusnyk --json | snyk-to-html -o container-test-result.html
```

Para analizar la imagen de Docker que creamos anteriormente y luego pasamos los resultados al comando:



**Image x:**Salida del reporte de seguridad por Snyk, donde se escaneó el proyecto "needusnyk", revelando 25 vulnerabilidades en 154 dependencias. Destacan CVE-2020-22916 (Gravedad: MEDIA, sin solución para xz-utils en Ubuntu:22.04) y un Open Redirect (Gravedad: MEDIA en wget).

**Creación Propia**

7.2. Metodología, técnicas usadas

Quality Gate Status:

El error clave es almacenar datos sensibles de autenticación junto con el código fuente de la aplicación. Esto expone los secretos a personas no autorizadas, lo que puede llevar a pérdidas financieras si se utilizan para acceder a servicios de pago y a una degradación de la seguridad de la aplicación, permitiendo que los atacantes tomen el control de la misma.

```
sonar-project.properties 1  Conexión.java 4 x
src > java > Config > Conexión.java
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package Config;
6
7  /**
8   *
9   * @author HP
10  */
11  import java.sql.*;
12
13  public class Conexión {
14      Connection con = null;
15
16      public Conexión() {
17          try {
18              Class.forName("com.mysql.cj.jdbc.Driver");
19              //con = DriverManager.getConnection("jdbc:mysql://localhost:3308/db_needu", "root", "");
20              con = DriverManager.getConnection("jdbc:mysql://161.132.47.59:3306/db_needu", "erick", "1234qwer");
21          } catch (ClassNotFoundException | SQLException e) {
22              //
23          }
24      }
25
26      public Connection getConnection() {
27          return con;
28      }
29  }
```

**image :** en esta imagen se puede observar la vulnerabilidad que según sonarqube

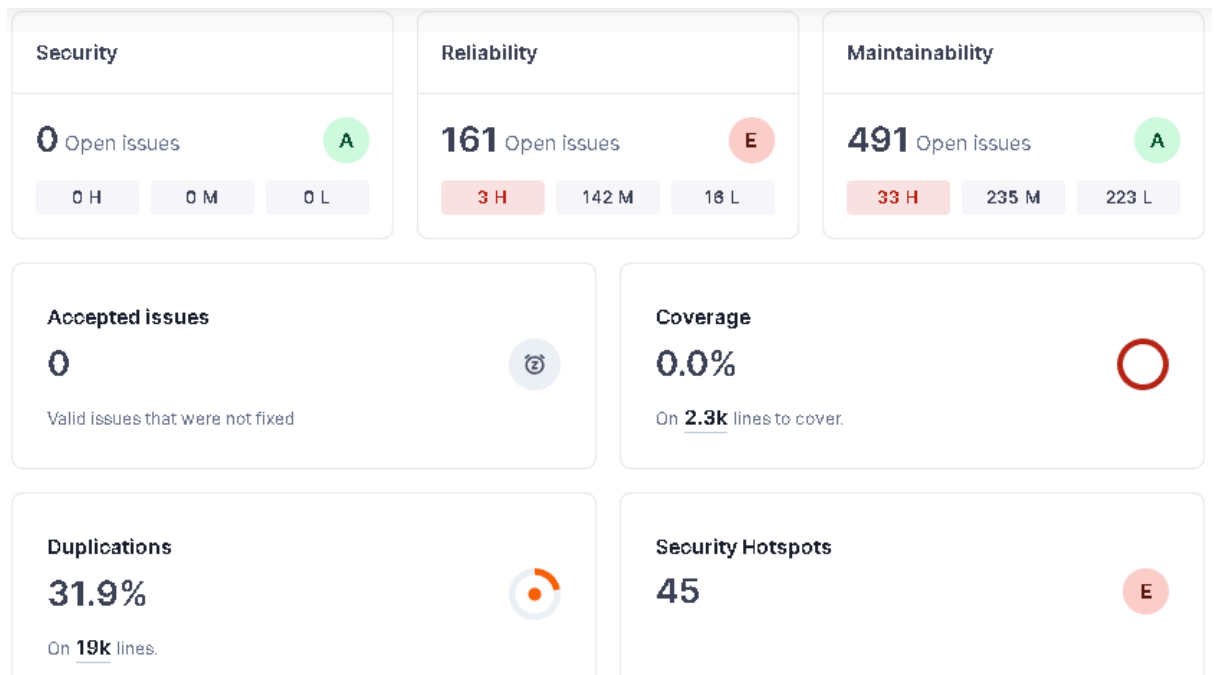
se encuentra en el archivo `conexion.java`

### creación propia

luego de corregir la vulnerabilidad en security y hacer otro análisis al proyecto



una vista mas detallada del proyecto analizado en sonarqube



luego hice un segundo análisis sin la carpeta build para que baje el código duplicado



luego quise bajar más el porcentaje de código duplicado y realice los siguientes cambios , agregue “package org.example;” abajo del otro package eso en todos los archivos.

```
ControladorRegistro.java M x
NeeduCloud > needu > src > java > ControladorRegistro.java
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Comment.java to edit this template
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Comment.java to edit this template
4   */
5  package Controlador;
6  package org.example;
```

luego agregue lang="en" en <html>

```
<!DOCTYPE html>  
<html lang="en">  
<head>
```

Por último solo quedaba la identificación de los fragmentos de código duplicados dentro de un conjunto de archivos. Esto puede lograrse mediante técnicas de análisis estático de código o herramientas de comparación de archivos.

Una vez identificados los fragmentos duplicados, se procede a crear un archivo nuevo que servirá como repositorio para almacenar únicamente estos fragmentos de código únicos. Este archivo actuará como una biblioteca centralizada de código compartido.

A continuación, se lleva a cabo la eliminación de los fragmentos duplicados de los archivos originales. Esto puede lograrse mediante la refactorización manual del código o mediante el uso de scripts automatizados que buscan y eliminan duplicaciones.

Después de eliminar los fragmentos duplicados, se actualizan los archivos originales con las versiones limpias del código. Esto implica reemplazar las instancias de los fragmentos duplicados con referencias al archivo nuevo que contiene los códigos únicos consolidados.

Una vez que todos los archivos han sido limpiados y actualizados, el sistema puede invocar el archivo nuevo como una biblioteca de código compartido. Esto permite un acceso eficiente a los fragmentos de código necesarios en lugar de tener que buscar y mantener duplicaciones dispersas en múltiples archivos.

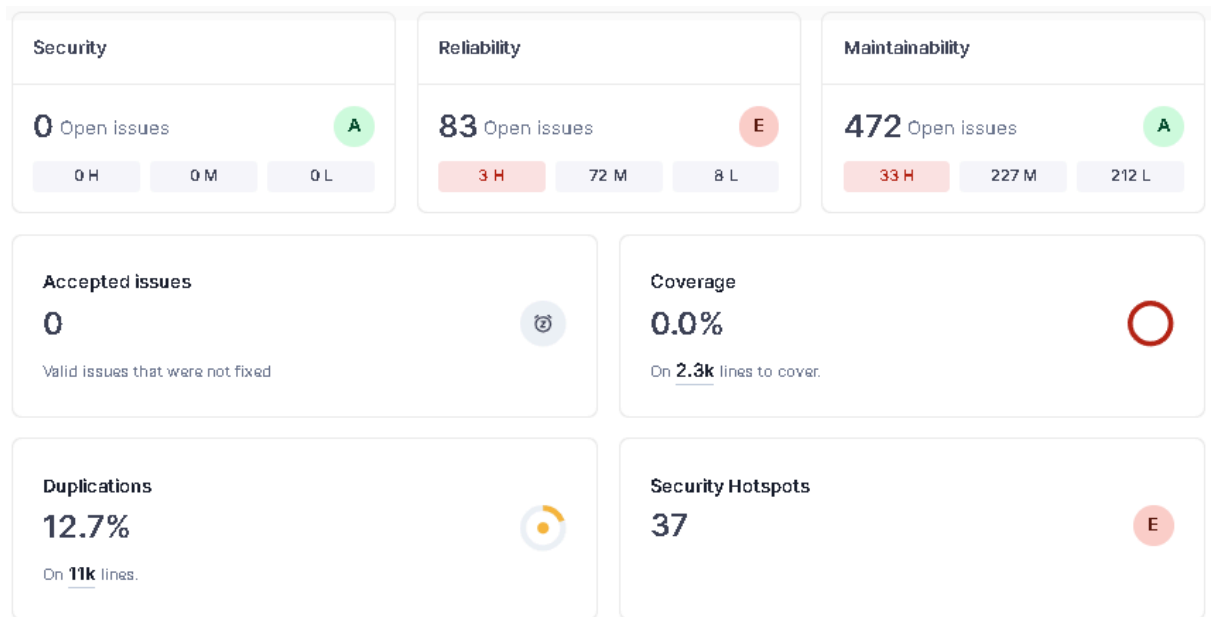
aquí unos ejemplos:

aquí cree el archivo common.js con el código que se repetía en 4 archivos









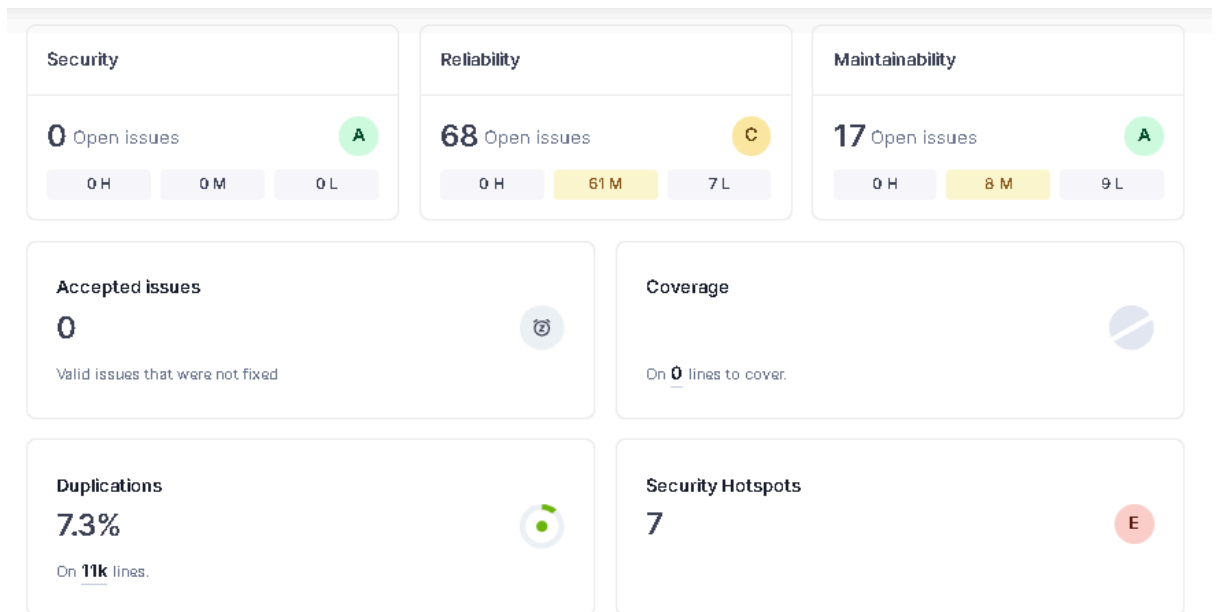
con menos porcentaje de código duplicado



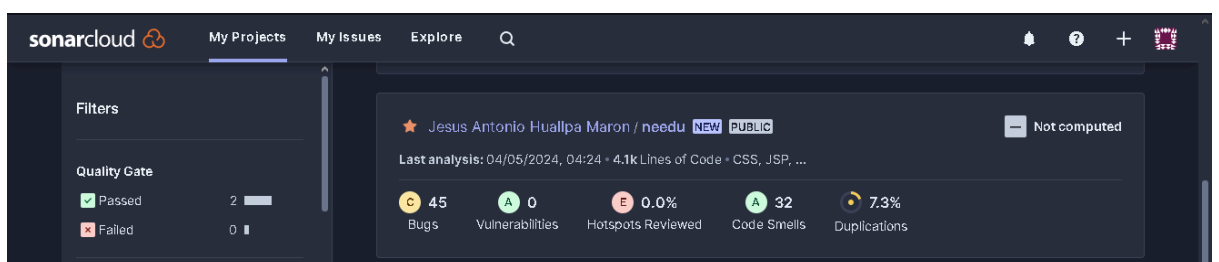
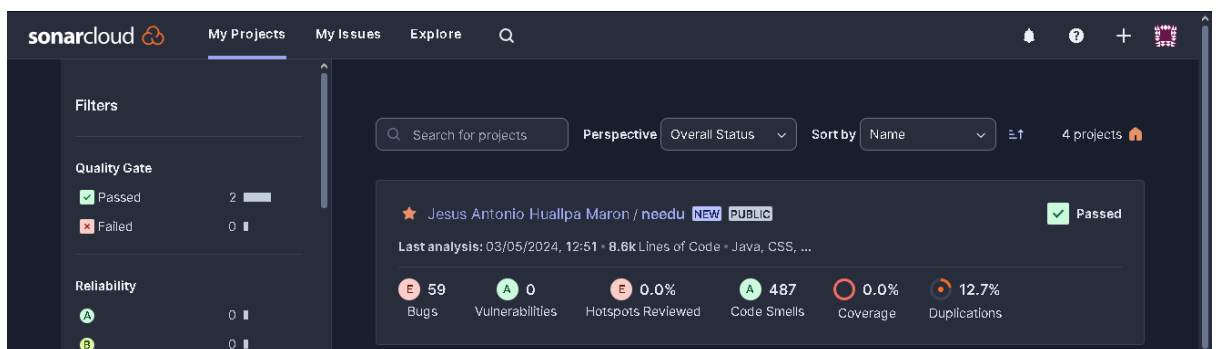
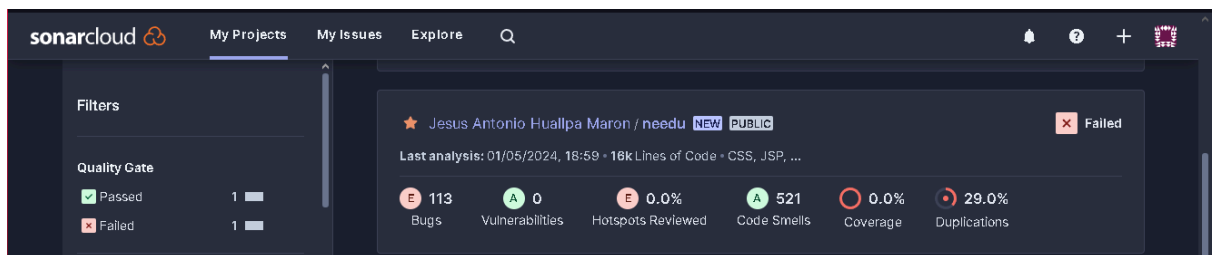
**image:** en la siguiente imagen se puede observar el proyecto sin la vulnerabilidad en security sonarqube.

**creación propia**

una vista mas detallada del proyecto analizado en sonarqube



el análisis en sonarcloud sin la vulnerabilidad, se puede ver una diferencia en sonarcloud y sonarqube



*image: en la siguiente imagen se muestra el analisis del proyecto en sonarcloud*

## **Snyk Code Report:**

**Alto:**

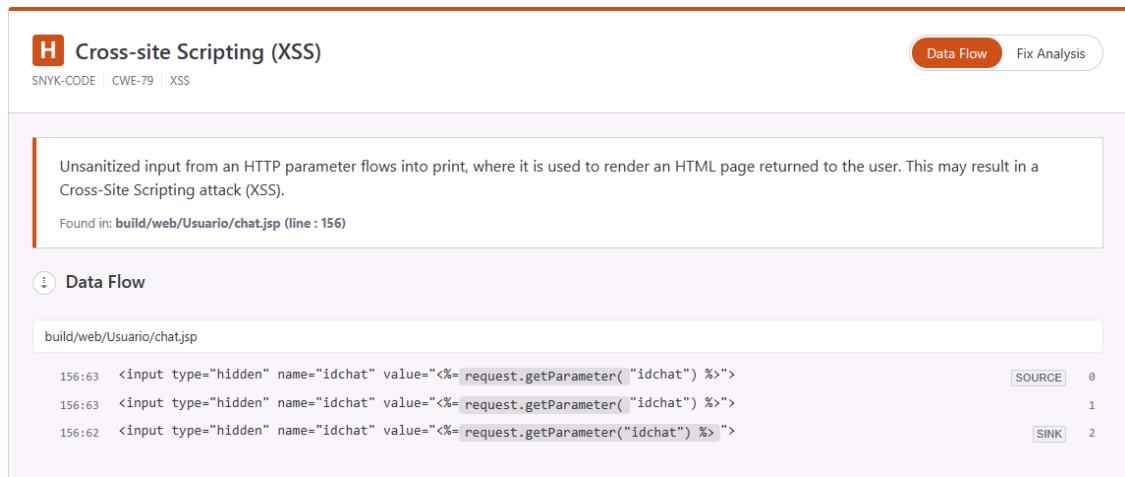
**Tipo de Vulnerabilidad:** Cross-site Scripting (XSS)

**CWE(Common Weakness Enumeration):** CWE-79

Descripción: La vulnerabilidad se encuentra en el archivo chat.jsp ubicado en la carpeta build/web/Usuario. La entrada no sanitizada de un parámetro HTTP llamado "idchat" fluye directamente a la salida de una etiqueta <input> en la línea 156 del archivo chat.jsp. Esta entrada no se filtra ni se codifica adecuadamente antes de ser utilizada para renderizar una página HTML devuelta al usuario. Como resultado, un atacante podría inyectar código JavaScript malicioso a través del parámetro "idchat", lo que podría conducir a un ataque de Cross-Site Scripting (XSS).

## **Solución:**

Se añadió la función `fn:escapeXml()` al código para garantizar que los datos proporcionados por el usuario y utilizados para generar el HTML estén correctamente escapados. Esto ayuda a prevenir posibles ataques XSS (Cross-Site Scripting) al asegurar que cualquier código malicioso incrustado en los datos de entrada no se ejecute en el navegador del usuario.



**Image x:** Snyk code report, que muestra los Problemas altos, indicando la dirección, el tipo de vulnerabilidad, del código.  
**Creación Propia**



**Image x:** Se reemplazó request.getParameter() con la función ``${fn:escapeXml(request.getParameter())}`` para asegurar que los datos del parámetro obtenidos del usuario estén correctamente escapados y prevenir posibles ataques XSS.  
**Creación Propia**



**Image x:** De igual manera para el resto de contenido, que aumentaban en uno, la cantidad de problemas. Ya con este cambio se puede prevenir posibles ataques XSS.  
**Creación Propia**

# Snyk Code Report



0 high issues



15 medium issues



6 low issues

*Image x: Snyk code report, se mitigaron los Problemas altos  
Creación Propia*

## Medio:

**Tipo de Vulnerabilidad:** Cross-site Scripting (XSS)

**CWE(Common Weakness Enumeration):** CWE-79

Descripción: La vulnerabilidad de Cross-site Scripting (XSS) se encuentra en el archivo admin.jsp, ubicado en la carpeta build/web. La entrada no sanitizada del parámetro HTTP "idchat" fluye directamente a la salida de una etiqueta <input> en la línea 156 del archivo chat.jsp. Esta entrada no se filtra ni se codifica adecuadamente antes de ser utilizada para renderizar una página HTML devuelta al usuario. Como resultado, un atacante podría inyectar código JavaScript malicioso a través del parámetro "idchat", lo que podría conducir a un ataque de Cross-Site Scripting (XSS). Se recomienda implementar medidas de seguridad, como la sanitización y la codificación de la entrada del usuario, para mitigar esta vulnerabilidad y proteger la aplicación contra ataques XSS.

## Solución:

Para mitigar la vulnerabilidad de Cross-site Scripting (XSS) utilizando la función fn en JSP, puedes seguir estos pasos:

Uso de la función fn:escapeXml: Esta función ayuda a escapar los caracteres especiales en XML, como <, >, &, ", y ', que son comunes en ataques XSS. Debes

aplicar esta función a cualquier dato dinámico que se incluya en la salida HTML para evitar que se interprete como código HTML.

**M**

**Cross-site Scripting (XSS)**

SNYK-CODE | CWE-79 | XSS

Data Flow

Fix Analysis

Unsanitized input from a database flows into print, where it is used to render an HTML page returned to the user. This may result in a Cross-Site Scripting attack (XSS).

Found in: **build/web/admin.jsp** (line : 85)

Data Flow

src/java/ModeloDAO/OportunidadDAO.java

75:47

opo.put("nombreOrganizacion", rs.getString( "nombreOrganizacion"));

SOURCE

0

75:47

opo.put("nombreOrganizacion", rs.getString( "nombreOrganizacion"));

1

75:17

opo. put("nombreOrganizacion", rs.getString("nombreOrganizacion"));

2

76:17

lista. add(opo);

3

82:16

return lista;

4

build/web/admin.jsp

79:43

List<Oportunidad> oportunidades = oportunidadDao.listarOportunidades( );

5

82:30

for (Oportunidad opo : oportunidades) {

6

85:29

<h2><%= opo. getTitulo() %></h2>

7

85:29

<h2><%= opo.getTitulo( ) %></h2>

8

85:28

<h2><%= opo.getTitulo() %> </h2>

SINK

9

**Image x:** Se visualiza el contenido de Cross-site Scripting XSS, indicando la ubicación del problema.  
**Creación Propia**

```

<div class='oportunidad'>
<h2><%= opo.getTitulo() %></h2>
<p>Descripción: <%= opo.getDescripcion() %></p>
<p>Fecha: <%= opo.getFecha() %></p>

<!-- Botón de compartir en Facebook -->
<div class="fb-share-button"
data-href="<%= org.apache.commons.lang.StringEsc
data-layout="button_count"
data-size="small"
data-description="<%= opo.getDescripcion()
<a target="_blank" href="https://www.faceb
</div>

```

*Image x: El cambio realizado en el código implica sustituir la llamada directa a `opo.getTitulo()` por un encapsulador `fn`.*

### **Creación Propia**

#### **Bajo:**

**Tipo de Vulnerabilidad:** Uso de credenciales codificadas (Hardcoded Credentials).

**CWE (Common Weakness Enumeration):** CWE-798.

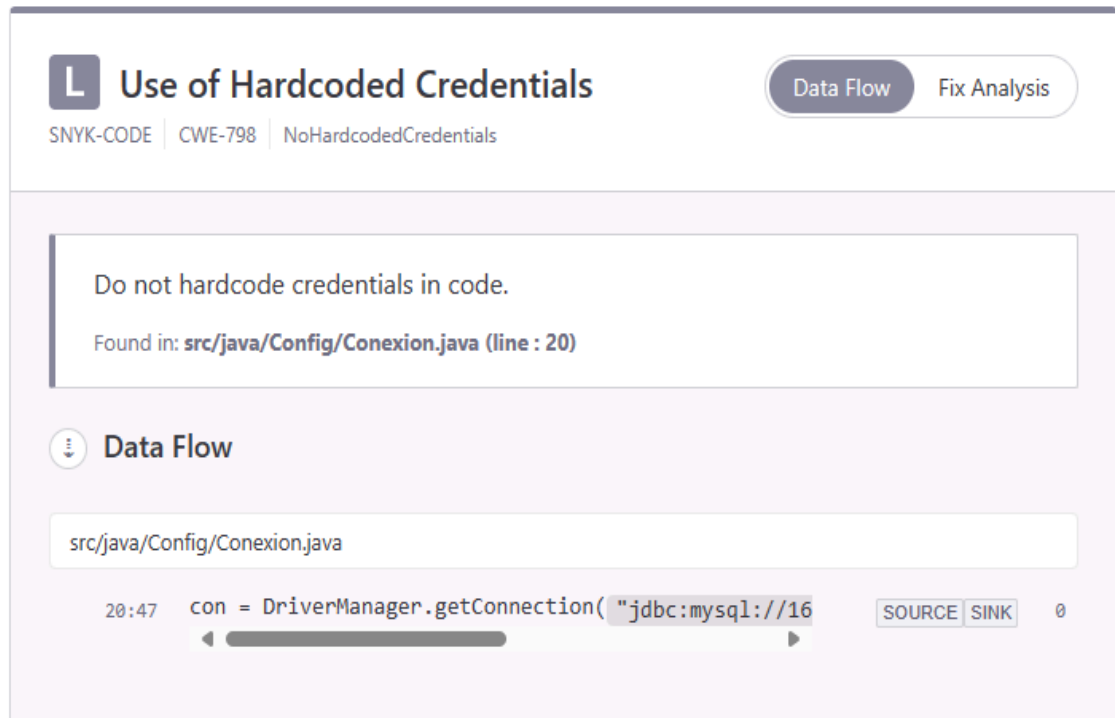
**Descripción:** Esta vulnerabilidad se encuentra en los archivos `pdf.jsp` y `Conexion.java`. En el archivo `pdf.jsp`, se utiliza `DriverManager.getConnection` para establecer una conexión a una base de datos MySQL, y las credenciales de usuario ("root") y contraseña ("") están codificadas directamente en el código. Lo mismo ocurre en el archivo `Conexion.java`, donde se establece una conexión con una base de datos MySQL utilizando credenciales codificadas ("erick" como nombre de usuario y "1234qwer" como contraseña). Este tipo de práctica representa un riesgo de seguridad, ya que las credenciales pueden ser fácilmente comprometidas si el código es accesible para personas no autorizadas.

#### **Solución:**

Para mitigar esta vulnerabilidad, se recomienda no codificar las credenciales directamente en el código fuente. En su lugar, se pueden utilizar métodos seguros



para manejar las credenciales, como almacenarlas de forma segura en archivos de configuración externos o utilizar herramientas de gestión de secretos. Además, es importante restringir el acceso al código fuente que contiene las credenciales para evitar exposiciones no deseadas.



**Image x:**Use of Hardcoded Credentials, se muestra la ubicación del archivo Conexion.java, el cual posee actualmente los datos escritos dentro del mismo archivo, lo cual genera una vulnerabilidad.

**Creación Propia**

```
public class Conexion {
    Connection con = null;

    public Conexion() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            //con = DriverManager.getConnection("jdbc:mysql://localhost:3308/db_needu", "root", "");
            con = DriverManager.getConnection("jdbc:mysql://161.132.47.59:3306/db_needu", "erick", "1234qwer");
        } catch (ClassNotFoundException | SQLException e) {
        }
    }

    public Connection getConnection() {
        return con;
    }
}
```

**Image x:**Así se visualiza el contenido del archivo Conexion.Java, antes de ser estructurado correctamente.

**Creación Propia**

```
src > java > ⚙ .properties
1  # Archivo de configuración de la base de datos
2
3  # URL de la base de datos
4  db.url=jdbc:mysql://161.132.47.59:3306/db_needu
5
6  # Nombre de usuario de la base de datos
7  db.username=erick
8
9  # Contraseña de la base de datos
10 db.password=1234qwer
11
```

*Image x:* Se procedió a establecer las credenciales dentro de un archivo .properties, el cual poseía una pequeña configuración de base de datos.

#### **Creación Propia**

```
src > java > ! .yml
1  # Archivo de configuración de la base de datos
2
3  database:
4    url: "jdbc:mysql://161.132.47.59:3306/db_needu"
5    username: "erick"
6    password: "1234qwer"
7
```

*Image x:* Se establecieron credenciales de base de datos en dos tipos de archivos de configuración. En primer lugar, se empleó un archivo .properties para almacenar la URL, nombre de usuario y contraseña. Posteriormente, se configuraron las mismas credenciales en un archivo YAML (.yml), asegurando así la correcta conexión con la base de datos.

#### **Creación Propia**

```
public class Conexion {
    private Connection con;

    public Conexion() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Cargar las propiedades desde el archivo de configuración
            Properties prop = new Properties();
            try (InputStream input = new FileInputStream("config.properties")) {
                prop.load(input);
            }

            // Obtener las credenciales del archivo de configuración
            String url = prop.getProperty("db.url");
            String username = prop.getProperty("db.username");
            String password = prop.getProperty("db.password");

            // Establecer la conexión con la base de datos
            con = DriverManager.getConnection(url, username, password);
        } catch (ClassNotFoundException | SQLException | IOException e) {
            e.printStackTrace();
        }
    }

    public Connection getConnection() {
        return con;
    }
}
```

**Image x:** Una vez creadas las credenciales en el archivo Conexion.java, el contenido de la base de datos ya no es visible directamente en el código. En su lugar, se accede a las propiedades correspondientes utilizando **Creación Propia**

8. Cronograma (personas, tiempo, otros recursos) Basado en las observaciones que la herramienta SonarQube les informará sobre la aplicación, a fin de reducir la deuda técnica, vulnerabilidades, fallas, etc. a 0.

M23																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
-----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

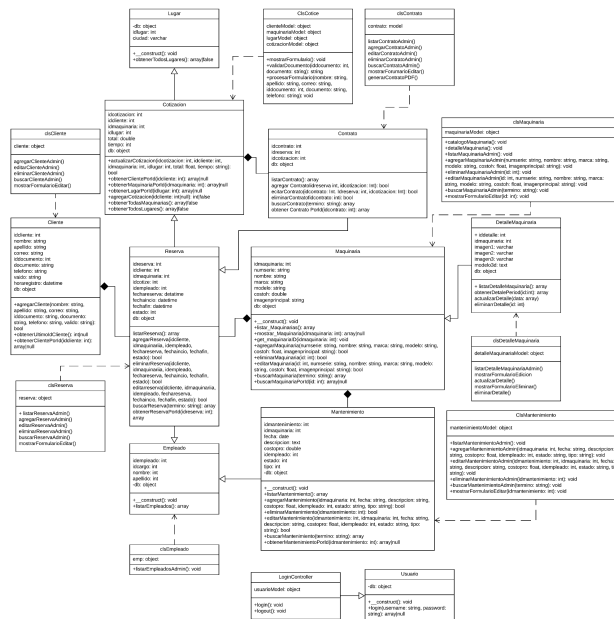
### 9.1 Diagrama de Arquitectura de la aplicación

El diagrama de arquitectura de software ilustra la organización de un sistema de gestión de maquinaria y servicios. Se estructura en tres capas principales: Vista, Controlador y Modelo, además de un componente para el sitio web.

- VISTA:**
  - ADMIN:** empleados, home, sidebar.
  - COTICE:** alerta, detalleCotice, formularioCotice.
  - MAQUINARIA:** agregar, eliminar, buscar, editar, index.
  - CATALOGO:** catalogoMaquinaria, detalleMaquinaria.
  - CLIENTE:** index.
  - EMPLEADO:** index.
  - CONTRATO:** index, editar, generar, eliminar, agregar, buscar.
  - MANTENIMIENTO:** index, editar, eliminar, agregar, buscar.
  - RESERVA:** index, buscar, editar, eliminar, agregar.
  - MODELO\_3D:** index, editar, eliminar.
- CONTROLADOR:**
  - LoginController
  - maquinaeria\_controller
  - cotice\_controller
  - contrato\_controller
  - ClienteController
  - DetalleMaquinariaController
  - EmpleoController
  - MantenimientoController
  - ReservaController
- MODELO:**
  - cliente
  - cotizacion
  - lugar
  - maquinaeria
  - Usuario
  - contrato
  - detallemaquinaria
  - Empleado
  - Mantenimiento
  - Reserva
- SITIO WEB SOSA:**
  - Vista principal de la pagina

Las flechas indican las dependencias y flujos de datos entre los componentes, mostrando cómo las vistas interactúan con los controladores y cómo estos manipulan los datos del modelo.

## 9.2 Diagrama de Clases de la aplicación



**Enlace:** <https://lucid.app/publicSegments/view/62c80f97-2e70-4e2b-a926-d429a700f8a9/image.png>

9.3 Métodos de pruebas implementados para coberturar la aplicación

a) Pruebas Unitarias (cobertura de al menos 70% de código - los métodos más importantes)

- Reporte de cobertura de pruebas

Primer reporte sin análisis de código

C:\Users\IHP\Videos\cobeage88porciento\src / (Dashboard)

	Code Coverage								
	Lines			Functions and Methods			Classes and Traits		
Total		0.00%	0 / 2631		0.00%	0 / 103		0.00%	0 / 19
■ CONTROLLER		0.00%	0 / 164		0.00%	0 / 46		0.00%	0 / 8
■ DB		0.00%	0 / 8		0.00%	0 / 1		0.00%	0 / 1
■ MODEL		0.00%	0 / 328		0.00%	0 / 56		0.00%	0 / 10
■ VIEW		0.00%	0 / 1622		n/a	0 / 0		n/a	0 / 0
📄 admin.php		0.00%	0 / 378		n/a	0 / 0		n/a	0 / 0
📄 comprobacion.php		0.00%	0 / 5		n/a	0 / 0		n/a	0 / 0
📄 email.php		0.00%	0 / 31		n/a	0 / 0		n/a	0 / 0
📄 index.php		0.00%	0 / 35		n/a	0 / 0		n/a	0 / 0
📄 login.php		0.00%	0 / 29		n/a	0 / 0		n/a	0 / 0
📄 prueba.php		0.00%	0 / 21		n/a	0 / 0		n/a	0 / 0
📄 salir.php		0.00%	0 / 5		n/a	0 / 0		n/a	0 / 0
📄 seguridad.php		0.00%	0 / 5		n/a	0 / 0		n/a	0 / 0

Legend

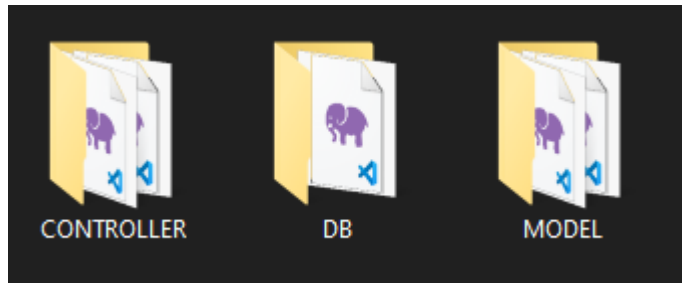
Low: 0% to 50%    Medium: 50% to 90%    High: 90% to 100%

Generated by php-code-coverage 11.0.3 using PHP 8.2.4 and PHPUnit 11.2.2 at Fri Jun 21 0:39:28 CEST 2024.

## Test Creados

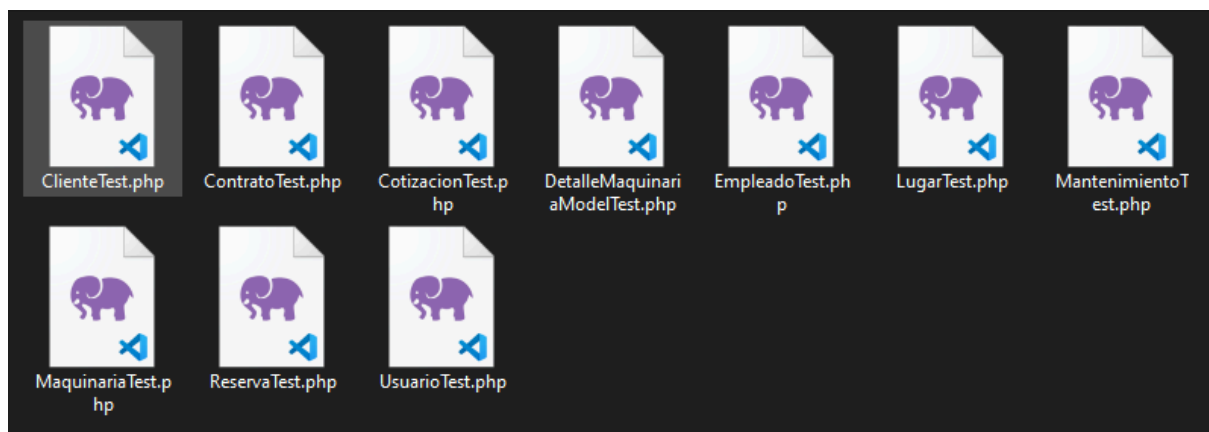
### Métodos de Pruebas (PHPUNIT)

Principales archivos que contiene Funciones , métodos, Clases (MVC)



Para realizar los test se crean los siguientes test, nombre del modelo con el fin test para identificarlo.

### TEST PARA MODELO



### Prueba Unitaria de Cliente

Modelo (Cliente)

Tests (ClienteTest)

**Namespace:** Define el espacio de nombres Tests\Model para organizar el código y evitar colisiones de nombres.

**Importaciones:**

`use PHPUnit\Framework\TestCase;` Importa la clase base TestCase de PHPUnit, utilizada para crear y ejecutar pruebas unitarias.

`use App\Model\Cliente;` Importa la clase Cliente del espacio de

**nombres App\Model, que se va a probar.**

```
namespace Tests\Model;

use PHPUnit\Framework\TestCase;

use App\Model\Cliente;
```

**testListarClientes: Prueba que el método listarClientes devuelve un array.**

```
public function testListarClientes() {
    $clientes = $this->cliente->listarClientes();
    $this->assertIsArray($clientes);
}
```

**testAgregarCliente: Prueba que se puede agregar un cliente correctamente y verifica que los datos del cliente agregado coincidan con los datos ingresados.**

```
public function testAgregarCliente() {
    $nombre = "Juan";
    $apellido = "Perez";
    $correo = "juan@example.com";
    $iddocumento = "DNI";
    $documento = "12345678";
    $telefono = "987654321";

    $resultado = $this->cliente->agregarCliente($nombre,
    $apellido, $correo, $iddocumento, $documento, $telefono);
    $this->assertTrue($resultado);

    $clienteAgregado =
    $this->cliente->obtenerClientePorId($this->cliente->obtenerUltimoIdCliente());
    $this->assertEquals($nombre, $clienteAgregado['nombre']);
    $this->assertEquals($apellido, $clienteAgregado['apellido']);
    $this->assertEquals($correo, $clienteAgregado['correo']);
}
```

**testEditarCliente: Prueba que se puede editar un cliente existente y verifica que los datos editados se actualicen correctamente.**

```
public function testEditarCliente() {
    $idcliente = 3;
```

```

    $nombre = "Nuevo Nombre";
    $apellido = "Nuevo Apellido";
    $correo = "nuevo@example.com";
    $iddocumento = "DNI";
    $documento = "87654321";
    $telefono = "123456789";

    $resultado = $this->cliente->editarCliente($idcliente,
    $nombre, $apellido, $correo, $iddocumento, $documento, $telefono);
    $this->assertTrue($resultado);

    $clienteEditado =
    $this->cliente->obtenerClientePorId($idcliente);
    $this->assertEquals($nombre, $clienteEditado['nombre']);
    $this->assertEquals($apellido, $clienteEditado['apellido']);
    $this->assertEquals($correo, $clienteEditado['correo']);
}

```

**testEliminarCliente: Prueba que se puede eliminar un cliente y verifica que el cliente ya no exista en la base de datos.**

```

public function testEliminarCliente() {
    $idcliente = 235;

    $resultado = $this->cliente->eliminarCliente($idcliente);
    $this->assertTrue($resultado);

    $conexion = Conectar::conexion();
    $query = "SELECT * FROM tbcliente WHERE idcliente = ?";
    $stmt = $conexion->prepare($query);
    $stmt->bind_param("i", $idcliente);
    $stmt->execute();
    $stmt->store_result();

    $this->assertEquals(0, $stmt->num_rows);

    $stmt->close();
    $conexion->close();
}

```

**testObtenerClienteExistentePorId: Prueba que se puede obtener un cliente existente por su ID y verifica que los datos obtenidos son correctos.**



```
public function testObtenerClienteExistentePorId() {
    $idclienteExistente = 1;

    $cliente =
$this->cliente->obtenerClientePorId($idclienteExistente);
    $this->assertIsArray($cliente);
    $this->assertNotEmpty($cliente);

    $this->assertArrayHasKey('idcliente', $cliente);
    $this->assertArrayHasKey('nombre', $cliente);
    $this->assertArrayHasKey('apellido', $cliente);
    $this->assertArrayHasKey('correo', $cliente);
    $this->assertArrayHasKey('iddocumento', $cliente);
    $this->assertArrayHasKey('documento', $cliente);
    $this->assertArrayHasKey('telefono', $cliente);
}
```

**testBuscarCliente: Prueba que el método buscarCliente devuelve un array al buscar un cliente por un término específico.**

```
public function testBuscarCliente() {
    $termino = "1";

    $clientesEncontrados =
$this->cliente->buscarCliente($termino);
    $this->assertIsArray($clientesEncontrados);
}
```

**tearDown: Método especial que se ejecuta después de cada prueba para limpiar el entorno de prueba.**

```
protected function tearDown(): void {
    parent::tearDown();
}
```

Resultado del test

Code Coverage for C:\Users\HP

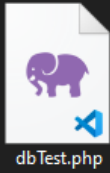
ArchivoC:/Users/HP/Videos/cobeage88porciento/coverage-report/MODEL/Cliente.php.html

C:\Users\HP\Videos\cobeage88porciento\src / MODEL / Cliente.php

	Code Coverage									
	Lines			Functions and Methods				Classes and Traits		
Total	<div></div>	100.00%	46 / 46	<div></div>	100.00%	7 / 7	CRAP	<div></div>	100.00%	1 / 1
Cliente	<div></div>	100.00%	46 / 46	<div></div>	100.00%	7 / 7	11	<div></div>	100.00%	1 / 1
__construct	<div></div>	100.00%	1 / 1	<div></div>	100.00%	1 / 1	1			
listarClientes	<div></div>	100.00%	5 / 5	<div></div>	100.00%	1 / 1	2			
agregarCliente	<div></div>	100.00%	7 / 7	<div></div>	100.00%	1 / 1	2			
editarCliente	<div></div>	100.00%	7 / 7	<div></div>	100.00%	1 / 1	2			
eliminarCliente	<div></div>	100.00%	7 / 7	<div></div>	100.00%	1 / 1	1			
buscarCliente	<div></div>	100.00%	11 / 11	<div></div>	100.00%	1 / 1	2			
obtenerClientePorId	<div></div>	100.00%	8 / 8	<div></div>	100.00%	1 / 1	1			

```
1  <?php
2  namespace App\Model;
3  use App\Controller\clsCliente;
4  use DB\Conectar;
5  use mysqli;
6
7  class Cliente {
8      private $db;
9
10     public function __construct() {
11         $this->db = Conectar::conexion();
12     }
13
14     public function listarClientes() {
15         $clientes = array();
16         $consulta = $this->db->query("SELECT * FROM tbcliente");
17         while ($fila = $consulta->fetch_assoc()) {
18             $clientes[] = $fila;
19         }
20         return $clientes;
21     }
```

## **TEST PARA DB**



### **Prueba Unitaria de db(conexión de la base de datos)**

Base de datos (db)

Tests (dbTest)

El namespace Tests\DB organiza el código dentro de la estructura de pruebas.  
PHPUnit\Framework\TestCase se usa como clase base para las pruebas unitarias.  
DB\Conectar es la clase que contiene el método conexion, que se va a probar.

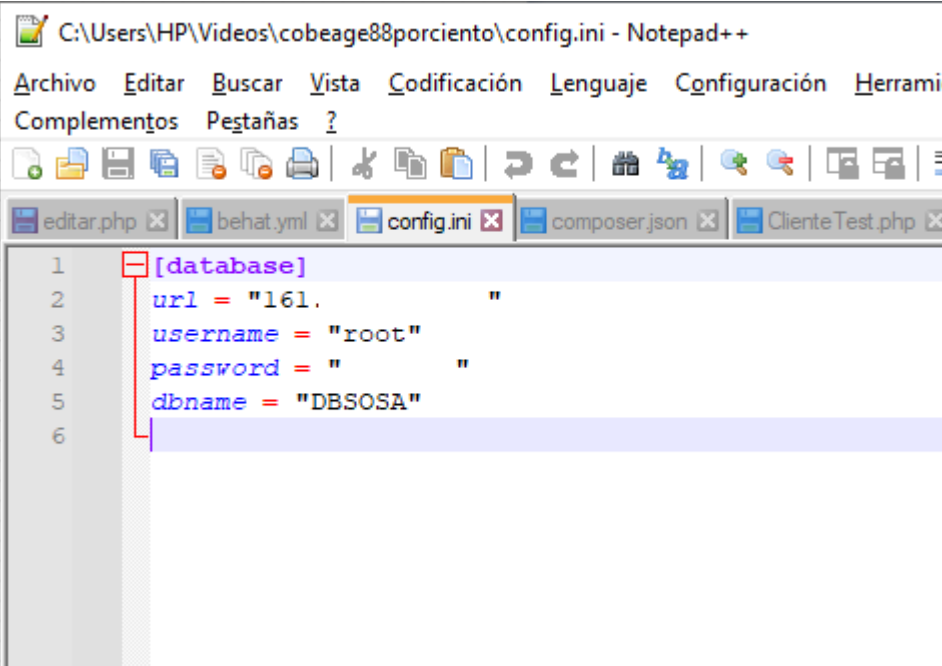
```
namespace Tests\DB;  
use PHPUnit\Framework\TestCase;  
use DB\Conectar;
```

Esta clase de prueba en PHPUnit verifica que la conexión a la base de datos se establece correctamente mediante la clase Conectar. La prueba asegura que la conexión devuelta es una instancia de mysqli.

```
class dbTest extends TestCase {  
    public function testConexion() {  
        $conexion = Conectar::conexion();  
        $this->assertInstanceOf(\mysqli::class, $conexion);  
    }  
}
```

## Sanitización y Validación:









- **Sanitización:** Verifica que los datos de configuración (como las credenciales de acceso a la base de datos) estén protegidos y sean seguros para su uso en la aplicación.



The screenshot shows a Notepad++ window titled "C:\Users\HP\Videos\cobeage88porciento\config.ini - Notepad++". The menu bar includes "Archivo", "Editar", "Buscar", "Vista", "Codificación", "Lenguaje", "Configuración", and "Herramientas". The toolbar contains various icons for file operations and editing. The tab bar shows several open files: "editar.php", "behat.yml", "config.ini", "composer.json", and "ClienteTest.php". The "config.ini" file is active, displaying the following configuration:

```
1 [database]
2 url = "161."
3 username = "root"
4 password = " "
5 dbname = "DBSOSA"
6
```

RESULTADO DEL TEST

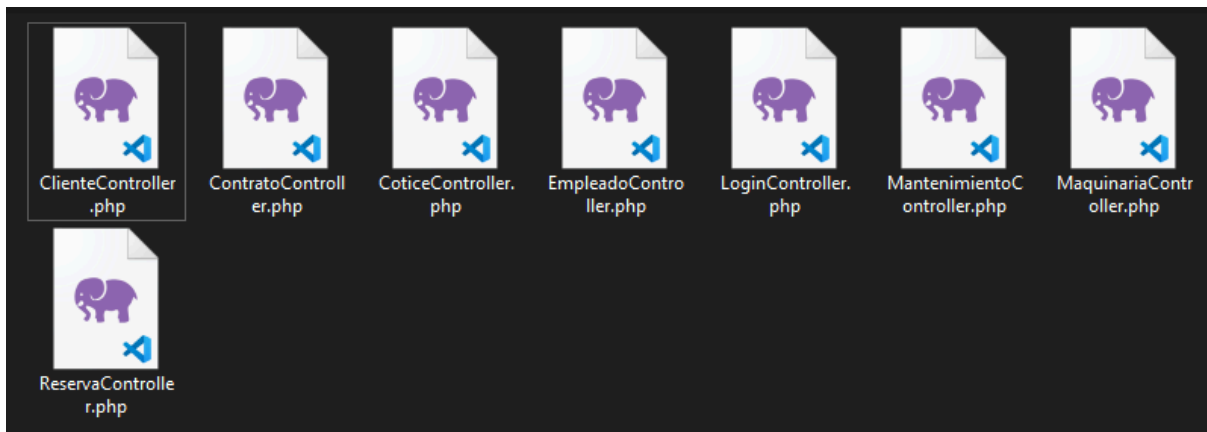
	Code Coverage									
	Lines			Functions and Methods				Classes and Traits		
Total		100.00%	8 / 8		100.00%	1 / 1	<u>CRAP</u>		100.00%	1 / 1
Conectar		100.00%	8 / 8		100.00%	1 / 1	1		100.00%	1 / 1
conexion		100.00%	8 / 8		100.00%	1 / 1	1			

```

1  <?php
2  namespace DB;
3
4  use mysqli;
5
6  class Conectar {
7      public static function conexion() {
8          // Cargar las propiedades desde el archivo de configuración
9          $config = parse_ini_file(__DIR__ . '/../../config.ini', true);
10
11          // Obtener las credenciales del archivo de configuración
12          $host = $config['database']['url'];
13          $username = $config['database']['username'];
14          $password = $config['database']['password'];
15          $dbname = $config['database']['dbname'];
16
17          // Establecer la conexión con la base de datos
18          $conexion = new mysqli($host, $username, $password, $dbname);
19
20          // Verificar si la conexión fue exitosa
21
22          // Establecer el conjunto de caracteres a utf8
23          $conexion->set_charset('utf8');
24
25          return $conexion;
26      }
27  }
28  ?>

```

## **TEST PARA CONTROLLER**



### **Prueba Unitaria de Controlador (Interacción entre vista y modelo)**

Controller (ContratoController)

Tests (ContratoControllerTest)

#### **testListarContratoAdmin()**

Esta prueba verifica que el método `listarContratoAdmin` del controlador `ContratoController` devuelve correctamente una lista de contratos obtenida del modelo `Contrato`. Se utiliza un mock del modelo para simular la respuesta esperada de la base de datos y se verifican las aserciones sobre los datos devueltos por el controlador.

```
public function testListarContratoAdmin() {
    // Creamos un mock del modelo Contrato
    $mockModel = $this->getMockBuilder(Contrato::class)
        ->disableOriginalConstructor()
        ->getMock();

    // Configuramos el comportamiento esperado del mock para
    listar contratos
    $mockModel->expects($this->once())
        ->method('listarContrato')
        ->willReturn([
            ['id' => 1, 'nombre' => 'Cliente A'],
            ['id' => 2, 'nombre' => 'Cliente B']
        ]);

    // Creamos una instancia del controlador inyectando el mock
    del modelo
    $controller = new ContratoController($mockModel);
}
```

```

// Ejecutamos el método del controlador que queremos probar
$contratos = $controller->listarContratoAdmin();

// Verificamos que el método devuelve los contratos esperados
$this->assertCount(2, $contratos);
$this->assertEquals(['id' => 1, 'nombre' => 'Cliente A'],
$contratos[0]);
$this->assertEquals(['id' => 2, 'nombre' => 'Cliente B'],
$contratos[1]);
}

```

#### **testAgregarContratoAdmin()**

**Esta prueba verifica que el método agregarContratoAdmin del controlador ContratoController llama correctamente al método agregarContrato del modelo Contrato con los parámetros adecuados. Se utiliza un mock del modelo para comprobar que la interacción entre el controlador y el modelo es correcta.**

```

public function testAgregarContratoAdmin() {
    // Creamos un mock del modelo Contrato
    $mockModel = $this->getMockBuilder(Contrato::class)
        ->disableOriginalConstructor()
        ->getMock();

    // Configuramos el mock para el método agregarContrato
    $mockModel->expects($this->once())
        ->method('agregarContrato')
        ->with(1, 1); // Verificamos los parámetros

    // Creamos una instancia del controlador inyectando el mock
    del modelo
    $controller = new ContratoController($mockModel);

    // Ejecutamos el método del controlador que queremos probar
    $controller->agregarContratoAdmin(1, 1);

    // No necesitamos verificar el resultado porque el método solo
    imprime un mensaje
    // Podríamos verificar que se imprima el mensaje correctamente
    usando output buffering
}

```

## **testEditarContratoAdmin()**

Esta prueba verifica que el método `editarContratoAdmin` del controlador `ContratoController` llama correctamente al método `editarContrato` del modelo `Contrato` con los parámetros adecuados. Utiliza un mock del modelo para asegurar que la interacción entre el controlador y el modelo sea correcta y se pasen los parámetros esperados.

```
public function testEditarContratoAdmin() {
    // Creamos un mock del modelo Contrato
    $mockModel = $this->getMockBuilder(Contrato::class)
        ->disableOriginalConstructor()
        ->getMock();

    // Configuramos el mock para el método editarContrato
    $mockModel->expects($this->once())
        ->method('editarContrato')
        ->with(1, 1, 1); // Verificamos los parámetros

    // Creamos una instancia del controlador inyectando el mock
    del modelo
    $controller = new ContratoController($mockModel);

    // Ejecutamos el método del controlador que queremos probar
    $controller->editarContratoAdmin(1, 1, 1);

    // No necesitamos verificar el resultado porque el método solo
    imprime un mensaje
    // Podríamos verificar que se imprima el mensaje correctamente
    usando output buffering
}
```

## **testEliminarContratoAdmin()**

Esta prueba verifica que el método `eliminarContratoAdmin` del controlador `ContratoController` llama correctamente al método `eliminarContrato` del modelo `Contrato` con el ID del contrato a eliminar y maneja adecuadamente la eliminación. Utiliza un mock del modelo para simular la eliminación exitosa y verificar que se llame con el parámetro correcto.



```

public function testEliminarContratoAdmin() {
    // Creamos un mock del modelo Contrato
    $mockModel = $this->getMockBuilder(Contrato::class)
        ->disableOriginalConstructor()
        ->getMock();

    // Configuramos el mock para el método eliminarContrato
    $mockModel->expects($this->once())
        ->method('eliminarContrato')
        ->with(1)
        ->willReturn(true); // Simulamos que la eliminación
fue exitosa

    // Creamos una instancia del controlador inyectando el mock
del modelo
    $controller = new ContratoController($mockModel);

    // Ejecutamos el método del controlador que queremos probar
    $controller->eliminarContratoAdmin(1);

    // No necesitamos verificar el resultado porque el método solo
redirige o imprime un mensaje
    // Podríamos verificar el redireccionamiento o mensaje usando
output buffering
}

```

## testBuscarContratoAdmin()

Esta prueba verifica que el método buscarContratoAdmin del controlador ContratoController llama correctamente al método buscarContrato del modelo Contrato con el término de búsqueda adecuado. Utiliza un mock del modelo para verificar que se llame con el parámetro esperado y no verifica el resultado directo ya que el método del controlador podría solo cargar una vista.

```

public function testBuscarContratoAdmin() {
    // Creamos un mock del modelo Contrato
    $mockModel = $this->getMockBuilder(Contrato::class)
        ->disableOriginalConstructor()
        ->getMock();

    // Configuramos el comportamiento esperado del mock para
buscar contrato

```

```

$mockModel->expects($this->once())
    ->method('buscarContrato')
    ->with('termino'); // Verificamos el parámetro

// Creamos una instancia del controlador inyectando el mock
del modelo
$controller = new ContratoController($mockModel);

// Ejecutamos el método del controlador que queremos probar
$controller->buscarContratoAdmin('termino');

// No necesitamos verificar el resultado porque el método solo
carga una vista
// Podríamos verificar la carga de la vista usando output
buffering
}

```

## testMostrarFormularioEditar()

Esta prueba verifica que el método mostrarFormularioEditar del controlador ContratoController llama correctamente al método obtenerContratoPorId del modelo Contrato con el ID del contrato a editar. Utiliza un mock del modelo para verificar que se llame con el ID esperado y no verifica el resultado directo ya que el método del controlador podría solo cargar una vista.

```

public function testMostrarFormularioEditar() {
    // Creamos un mock del modelo Contrato
    $mockModel = $this->getMockBuilder(Contrato::class)
        ->disableOriginalConstructor()
        ->getMock();

    // Configuramos el mock para el método obtenerContratoPorId
    $mockModel->expects($this->once())
        ->method('obtenerContratoPorId')
        ->with(1); // Verificamos el parámetro

    // Creamos una instancia del controlador inyectando el mock
del modelo
    $controller = new ContratoController($mockModel);

    // Ejecutamos el método del controlador que queremos probar

```

```

$controller->mostrarFormularioEditar(1);





















// No necesitamos verificar el resultado porque el método solo
carga una vista
// Podríamos verificar la carga de la vista usando output
buffering
}

```

## RESULTADO DEL TEST CONTROLLER

Code Coverage for C:\Users\HP\Videos\cobeage88porciento\coverage-report/CONTROLLER/ContratoController.p...

C:\Users\HP\Videos\cobeage88porciento\src / CONTROLLER / ContratoController.php

	Code Coverage									
	Lines			Functions and Methods				Classes and Traits		
Total		100.00%	14 / 14		100.00%	7 / 7	CRAP		100.00%	1 / 1
ContratoController		100.00%	14 / 14		100.00%	7 / 7	8		100.00%	1 / 1
__construct		100.00%	1 / 1		100.00%	1 / 1	1			
listarContratoAdmin		100.00%	2 / 2		100.00%	1 / 1	1			
agregarContratoAdmin		100.00%	2 / 2		100.00%	1 / 1	1			
editarContratoAdmin		100.00%	2 / 2		100.00%	1 / 1	1			
eliminarContratoAdmin		100.00%	3 / 3		100.00%	1 / 1	2			
buscarContratoAdmin		100.00%	2 / 2		100.00%	1 / 1	1			
mostrarFormularioEditar		100.00%	2 / 2		100.00%	1 / 1	1			













```

1  <?php
2  namespace App\Controller;
3
4  use App\Model\Contrato;
5  use FPDF;
6
7  class ContratoController {
8
9      private $modeloContrato;
10
11     public function __construct(Contrato $modeloContrato) {
12         $this->modeloContrato = $modeloContrato;
13     }
14
15     public function listarContratoAdmin() {
16         try {
17             $datos = $this->modeloContrato->listarContrato();
18             // Aquí normalmente se devolverían los datos o se pasarían a una vista
19             return $datos;
20         } finally {
21
22






















```

## **TEST DE COBERTURA CON 92.23% DE LOS MÉTODOS**

Reporte final de Cobertura solo considerando Funciones, métodos que estén dentro de una clase

	Code Coverage								
	Lines			Functions and Methods			Classes and Traits		
Total		80.00%	400 / 500		92.23%	95 / 103		84.21%	16 / 19
■ CONTROLLER		39.02%	64 / 164		82.61%	38 / 46		62.50%	5 / 8
■ DB		100.00%	8 / 8		100.00%	1 / 1		100.00%	1 / 1
■ MODEL		100.00%	328 / 328		100.00%	56 / 56		100.00%	10 / 10

Considerando las líneas de código con las vistas que no contiene metodo

C:\Users\HP\Videos\cobeage88porciento\src / (Dashboard)									
	Code Coverage								
	Lines			Functions and Methods			Classes and Traits		
Total		15.20%	400 / 2631		92.23%	95 / 103		84.21%	16 / 19
■ CONTROLLER		39.02%	64 / 164		82.61%	38 / 46		62.50%	5 / 8
■ DB		100.00%	8 / 8		100.00%	1 / 1		100.00%	1 / 1
■ MODEL		100.00%	328 / 328		100.00%	56 / 56		100.00%	10 / 10
■ VIEW		0.00%	0 / 1622		n/a	0 / 0		n/a	0 / 0
📄 admin.php		0.00%	0 / 378		n/a	0 / 0		n/a	0 / 0
📄 comprobacion.php		0.00%	0 / 5		n/a	0 / 0		n/a	0 / 0
📄 email.php		0.00%	0 / 31		n/a	0 / 0		n/a	0 / 0
📄 index.php		0.00%	0 / 35		n/a	0 / 0		n/a	0 / 0
📄 login.php		0.00%	0 / 29		n/a	0 / 0		n/a	0 / 0
📄 prueba.php		0.00%	0 / 21		n/a	0 / 0		n/a	0 / 0
📄 salir.php		0.00%	0 / 5		n/a	0 / 0		n/a	0 / 0
📄 seguridad.php		0.00%	0 / 5		n/a	0 / 0		n/a	0 / 0

Legend

Low: 0% to 50%    Medium: 50% to 90%    High: 90% to 100%

**b) Pruebas de aceptación basadas en Desarrollo Guiado por el Comportamiento una por cada caso de uso o historia de usuario.**

BDD (Desarrollo Guiado por el Comportamiento, por sus siglas en inglés) es una metodología de desarrollo de software que se centra en la colaboración entre desarrolladores, testers y partes interesadas (stakeholders) no técnicas como analistas de

negocio o usuarios finales. El objetivo principal de BDD es asegurar que el software se desarrolle desde la perspectiva del comportamiento del sistema, lo cual implica definir el comportamiento esperado antes de comenzar a implementar cualquier código.

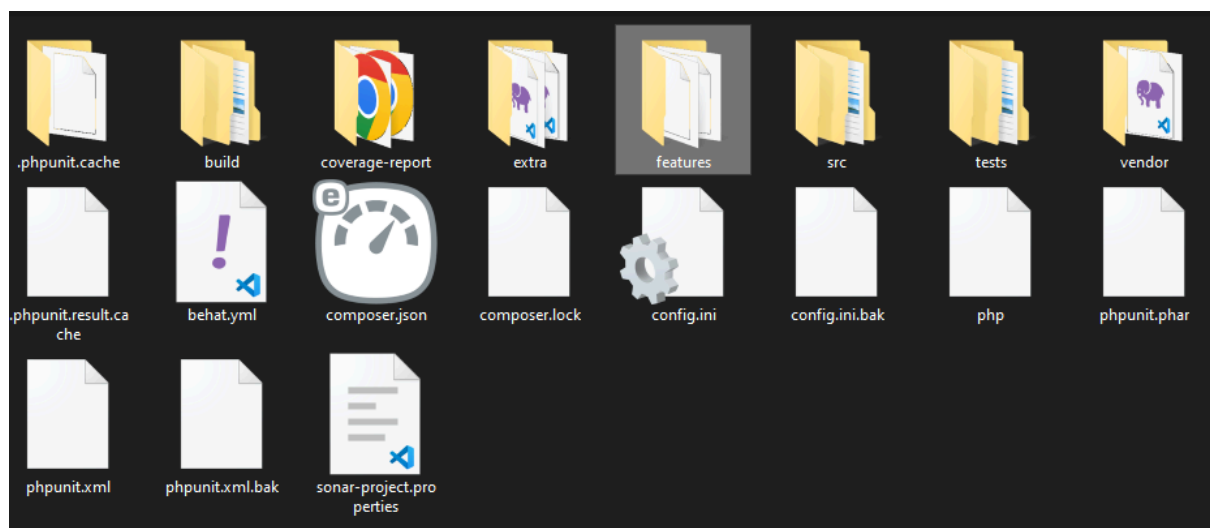
**Given (Dado):** Describe el estado inicial o contexto antes de que ocurra la acción que estamos probando.

**When (Cuando):** Describe la acción o evento que queremos probar.

**Then (Entonces):** Describe el resultado esperado después de que ocurra la acción.

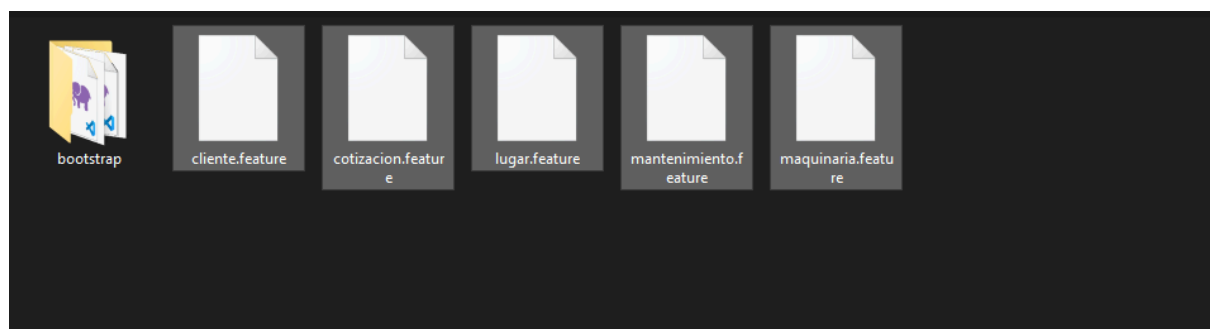
Estos escenarios son escritos en colaboración con los stakeholders y son utilizados para guiar tanto el desarrollo como las pruebas. Los pasos de BDD aseguran que el equipo entienda claramente lo que se está construyendo y que todos tengan una visión común del comportamiento esperado del sistema.

### **TEST PARA LOS ESCENARIOS (FEATURE)**

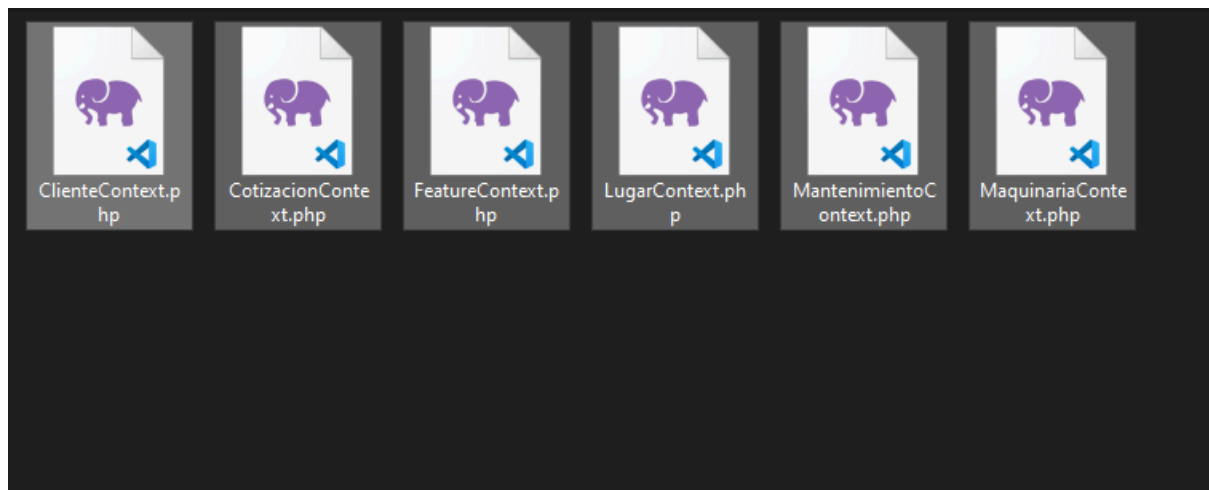


#### Creación de Features

Aquí se redactará las acciones GIVE WHEN THEN



## Carpeta Bootstrap



## Escenario para cotización (feature)

Scenario: Create a new quotation

**Given** I have a Cotizacion instance

**When** I add a new quotation for client with idcliente 1

**Then** the quotation should be successfully added

Scenario: Update an existing quotation

**Given** I have a Cotizacion instance

And there is a quotation with idcotizacion 430

**When** I update the quotation with idcotizacion 430, setting idcliente to 2, idmaquinaria to 567, idlugar to 4, total to 1500.50, and tiempo to 5

**Then** the quotation with idcotizacion 430 should be successfully updated

Scenario: Get client details for quotation

**Given** I have a Cotizacion instance

And there is a client with idcliente 1

**When** I fetch the details of the client with idcliente 1

**Then** I should receive the details of the client

Scenario: Get machinery details for quotation

**Given** I have a Cotizacion instance

And there is a machinery with idmaquinaria 567

**When** I fetch the details of the machinery with idmaquinaria 567

**Then** I should receive the details of the machinery

Scenario: Get location details for quotation

**Given** I have a Cotizacion instance

And there is a location with idlugar 4

**When** I fetch the details of the location with idlugar 4  
**Then** I should receive the details of the location

Scenario: Get all available machinery for quotation  
**Given** I have a Cotizacion instance  
**When** I fetch all available machinery  
**Then** I should receive a list of all machinery

Scenario: Get all available locations for quotation  
**Given** I have a Cotizacion instance  
**When** I fetch all available locations  
**Then** I should receive a list of all locations

### **Contexto donde se realizan los escenarios (CotizacionContext)**

#### **Métodos de Definición de Contexto (@Given)**

- **iHaveACotizacionInstance:** Verifica que la instancia de Cotizacion se haya creado correctamente.
- **thereIsAQuotationWithIdcotizacion:** Asegura que exista una cotización con el idcotizacion especificado en la base de datos, eliminando cualquier cotización existente y luego insertando una nueva.

#### **Métodos de Acción (@When)**

- **iAddANewQuotationForClientWithIdcliente:** Agrega una nueva cotización para un cliente específico (idcliente).
- **iUpdateTheQuotationWithIdcotizacionSetting:** Actualiza una cotización existente (idcotizacion) con nuevos valores de cliente, maquinaria, lugar, total y tiempo.

#### **Métodos de Verificación (@Then)**

- **theQuotationShouldBeSuccessfullyAdded:** Verifica que una cotización haya sido agregada exitosamente, asegurando que el último insert\_id sea mayor que cero.

- **theQuotationWithIdcotizacionShouldBeSuccessfullyUpdated:** Confirma que una cotización haya sido actualizada correctamente, asegurándose de que los detalles actualizados estén presentes en la base de datos.
- **thereIsAClientWithIdcliente:** Comprueba que exista un cliente con el idcliente especificado.
- **iShouldReceiveTheDetailsOfTheClient, iShouldReceiveTheDetailsOfTheMachinery, iShouldReceiveTheDetailsOfTheLocation:** Verifica que los detalles del cliente, maquinaria y lugar se hayan recuperado correctamente.
- **iShouldReceiveAListOfAllMachinery, iShouldReceiveAListOfAllLocations:** Confirma que se haya recibido una lista de todas las máquinas y lugares disponibles.

## Resultado del Escenario

```
C:\Windows\System32\cmd.exe - php vendor/bin/behat

Feature: Quotation Management
  As a user,
  I want to manage quotations in the system
  So that I can view, add, edit, and delete quotations, as well as retrieve client, machinery, and location details.

Scenario: Create a new quotation
  Given I have a Cotizacion instance
  When I add a new quotation for client with idcliente 1
  Then the quotation should be successfully added

Scenario: Update an existing quotation
  Given I have a Cotizacion instance
  And there is a quotation with idcotizacion 431
  When I update the quotation with idcotizacion 431, setting idcliente to 2, idmaquinaria to 567, idlugar to 4, total to 1500.50, and tiempo to 5
  Then the quotation with idcotizacion 431 should be successfully updated

  Failed to update quotation.
  Failed asserting that null is true.

Scenario: Get client details for quotation
  Given I have a Cotizacion instance
  When I fetch the details of the client with idcliente 2
  Then I should receive the details of the client

Scenario: Get machinery details for quotation
  Given I have a Cotizacion instance
  And there is a machinery with idmaquinaria 567
  When I fetch the details of the machinery with idmaquinaria 567
  Then I should receive the details of the machinery

Scenario: Get location details for quotation
  Given I have a Cotizacion instance
  And there is a location with idlugar 4
  When I fetch the details of the location with idlugar 4
  Then I should receive the details of the location

Scenario: Get all available machinery for quotation
  Given I have a Cotizacion instance
  When I fetch all available machinery
  Then I should receive a list of all machinery

Scenario: Get all available locations for quotation
  Given I have a Cotizacion instance
  When I fetch all available locations
  Then I should receive a list of all locations
```

## Error mostrado

```
Scenario: Update an existing quotation
  Given I have a Cotizacion instance
  And there is a quotation with idcotizacion 430
  When I update the quotation with idcotizacion 430, setting idcliente to 2, idmaquinaria to 567, idlugar to 4, total to 1500.50, and tiempo to 5
  Then the quotation with idcotizacion 430 should be successfully updated

  Failed to update quotation.
  Failed asserting that null is true.
```



Sin embargo en la base de de datos se visualiza el cambio

DBSOSA.tbcotizacion: 33: >> Siguientes		Mostrar todo		Ordenación		Columnas (6/6)	
idcotizacion	iddiente	idmaquinaria	idlugar	total	tiempo		
423	10	(NULL)	(NULL)	(NULL)	(NULL)		
425	10	(NULL)	(NULL)	(NULL)	(NULL)		
427	10	(NULL)	(NULL)	(NULL)	(NULL)		
429	1	(NULL)	(NULL)	(NULL)	(NULL)		
430	2	567	4	1.500,5	5		
431	2	567	4	1.500,5	5		
432	1	(NULL)	(NULL)	(NULL)	(NULL)		
433	1	(NULL)	(NULL)	(NULL)	(NULL)		
434	1	(NULL)	(NULL)	(NULL)	(NULL)		
435	1	(NULL)	(NULL)	(NULL)	(NULL)		

## **RESULTADO DE LOS TEST PARA LOS ESCENARIOS (FEATURE)**

Mediante el comando `php vendor/bin/behata --format html --out build/html/behata`, explica como funciona y que beneficios nos trae

Se visualiza el porcentaje de los Features realizados, se visualiza 2 con éxito

- 5 features ( 2 success 3 fail )
- 23 escenarios ( 17 success 6 fail )
- 81 steps ( 71 success 3 pending 3 skipped 4 fail )

5 features ( 2 success 3 fail )  
23 escenarios ( 17 success 6 fail )  
81 steps ( 71 success 3 pending 3 skipped 4 fail )  
0m16.10s - 17.97Mb

Suite : default

### Feature: Client Management |+

As a user, I want to manage clients in the system So that I can view, add, edit, delete, and search clients.

Feature has passed

### Feature: Quotation Management |+

As a user, I want to manage quotations in the system So that I can view, add, edit, and delete quotations, as well as retrieve client, machinery, and location details.

Feature has failed Scenarios passed : 85,71%, Scenarios failed : 14,29%

### Feature: Retrieve all places |+

As a user I want to retrieve all places from the database So that I can see a list of available places

Feature has passed

### Feature: Maintenance Management |+

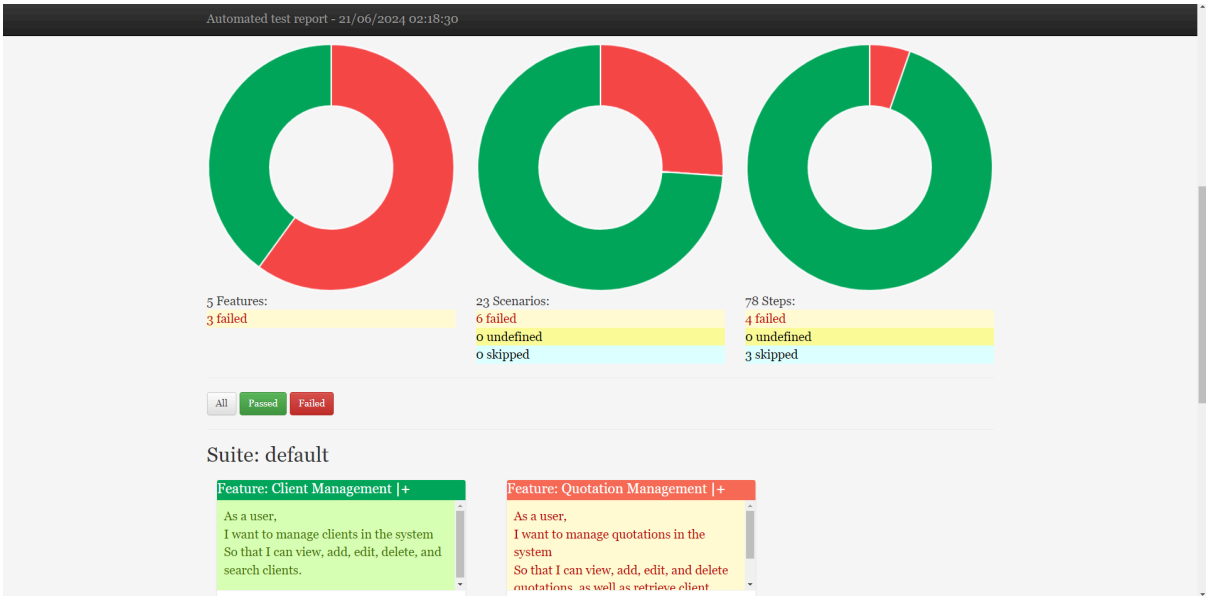
Feature has failed Scenarios passed : 60%, Scenarios failed : 40%

### Feature: Manage machinery |+

As a user I want to manage machinery in the database So that I can see, add, edit, delete, and search machinery

Feature has failed Scenarios passed : 40%, Scenarios failed : 60%

Ahora podemos visualizar de manera gráfica que la mayoría de los Steps(pasos), lograron pasarse.



**LISTA DE TODOS LOS ESCENARIOS REALIZADOS**

cliente.feature	<div>Feature: Client Management</div> <div>As a user, I want to manage clients in the system So that I can view, add, edit, delete, and search clients.</div> <div>Scenario: List clients Given I have a Cliente instance When I request the list of clients Then the list should contain at least one client</div> <div>Scenario: Add a new client Given I have a Cliente instance When I add a new client with nombre "John", apellido "Doe", correo "john.doe@example.com", iddocumento "1", documento "12345678", telefono "987654321" Then the client list should include "John Doe"</div> <div>Scenario: Edit an existing client Given I have a Cliente instance And there is a client with idcliente 240 When I edit the client with idcliente 240, setting nombre to "Jane", apellido to "Smith", correo to "jane.smith@example.com", iddocumento to "1", documento to "87654321", telefono to "654321987" Then the client list should include "Jane Smith"</div>
-----------------	---

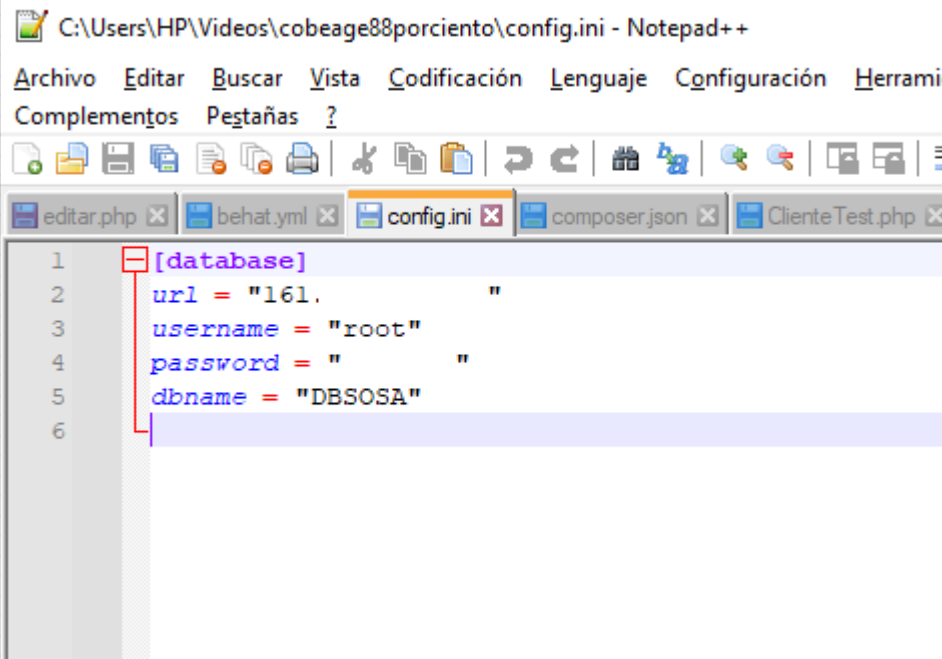
	<p>Scenario: Delete an existing client  Given I have a Cliente instance  When I delete the client with idcliente 239  Then the client list should not include "Jane Smith"</p> <p>Scenario: Search clients by term  Given I have a Cliente instance  When I search for clients with term "John"  Then the search results should include "John Doe"</p>
cotizacion.feature	<p>Feature: Quotation Management  As a user,  I want to manage quotations in the system  So that I can view, add, edit, and delete quotations, as well as retrieve client, machinery, and location details.</p> <p>Scenario: Create a new quotation  Given I have a Cotizacion instance  When I add a new quotation for client with idcliente 1  Then the quotation should be successfully added</p> <p>Scenario: Update an existing quotation  Given I have a Cotizacion instance  And there is a quotation with idcotizacion 431  When I update the quotation with idcotizacion 431, setting idcliente to 2, idmaquinaria to 567, idlugar to 4, total to 1500.50, and tiempo to 5  Then the quotation with idcotizacion 431 should be successfully updated</p> <p>Scenario: Get client details for quotation  Given I have a Cotizacion instance  And there is a client with idcliente 2  When I fetch the details of the client with idcliente 2  Then I should receive the details of the client</p> <p>Scenario: Get machinery details for quotation  Given I have a Cotizacion instance  And there is a machinery with idmaquinaria 567  When I fetch the details of the machinery with idmaquinaria 567  Then I should receive the details of the machinery</p> <p>Scenario: Get location details for quotation  Given I have a Cotizacion instance  And there is a location with idlugar 4  When I fetch the details of the location with idlugar 4  Then I should receive the details of the location</p> <p>Scenario: Get all available machinery for quotation  Given I have a Cotizacion instance  When I fetch all available machinery</p>

	<p>Then I should receive a list of all machinery</p> <p>Scenario: Get all available locations for quotation  Given I have a Cotizacion instance  When I fetch all available locations  Then I should receive a list of all locations</p>
lugar.feature	<p>Feature: Retrieve all places  As a user  I want to retrieve all places from the database  So that I can see a list of available places</p> <p>Scenario: User requests all places  Given there are places in the database  When I request all places  Then I should get a list of places  And each place should have an "idlugar"</p>
mantenimiento.feature	<p>Feature: Maintenance Management</p> <p>Scenario: Add a new maintenance record  Given I have a Mantenimiento instance  When I add a new maintenance record with idmaquinaria 1, fecha "2024-06-30", descripcion "Limpieza y revisión", costopro 200.50, idempleado 1, estado "Pendiente", tipo 1  Then the maintenance list should include "Limpieza y revisión"</p> <p>Scenario: Edit an existing maintenance record  Given I have a Mantenimiento instance  And there is a maintenance record with idmantenimiento 1  When I edit the maintenance record with idmantenimiento 1, setting idmaquinaria to 2, fecha to "2024-07-01", descripcion to "Reparación de motor", costopro to 500.75, idempleado to 2, estado to "Completado", tipo 2  Then the maintenance list should include "Reparación de motor"</p> <p>Scenario: Delete an existing maintenance record  Given I have a Mantenimiento instance  And there is a maintenance record with idmantenimiento 3  When I delete the maintenance record with idmantenimiento 3  Then the maintenance list should not include "Reparación de motor"</p> <p>Scenario: Search maintenance records by term  Given I have a Mantenimiento instance  When I search for maintenance records with term "Limpieza y revisión"</p>

	<p>Then the search results should include "Limpieza y revisión"</p> <p>Scenario: Get maintenance details by ID  Given I have a Mantenimiento instance  And there is a maintenance record with idmantenimiento 2  When I fetch the details of the maintenance record with idmantenimiento 2  Then I should receive the details of the maintenance record</p>
maquinaria.feature	<p>Feature: Manage machinery  As a user  I want to manage machinery in the database  So that I can see, add, edit, delete, and search machinery</p> <p>Scenario: List all machinery  Given there is machinery in the database  When I request all machinery  Then I should get a list of machinery  And each machinery should have an "idmaquinaria"  And each machinery should have a "nombre"</p> <p>Scenario: Add a new machinery  Given I have a Maquinaria instance  When I add a new machinery with numserie "ABC123", nombre "Excavator", marca "CAT", modelo "X1", costoh 100.50, imagenprincipal "excavator.jpg"  Then the machinery list should include "Excavator"</p> <p>Scenario: Edit an existing machinery  Given I have a Maquinaria instance  And there is a machinery with idmaquinaria 1  When I edit the machinery with idmaquinaria 1, setting numserie to "XYZ789", nombre to "Bulldozer", marca to "CAT", modelo to "B2", costoh to 150.75, imagenprincipal to "bulldozer.jpg"  Then the machinery list should include "Bulldozer"</p> <p>Scenario: Delete an existing machinery  Given I have a Maquinaria instance  When I delete the machinery with idmaquinaria 834  Then the machinery list should not include "Excavator"</p> <p>Scenario: Search machinery by term  Given I have a Maquinaria instance  When I search for machinery with term "Excavator"  Then the search results should include "Excavator"</p>

## c) Propuestas de implementacion de Pimera unidad

### Sanitización de la base de datos



```
1 [database]
2 url = "161. "
3 username = "root"
4 password = " "
5 dbname = "DBSOSA"
6
```

### Acciones realizadas:

Revisión del archivo conf.ini, el cual contiene los parámetros de conexión a la base de datos:

url = "161.xxx.xxx.xxx"

username = "root"

password = "\*\*\*\*\*"

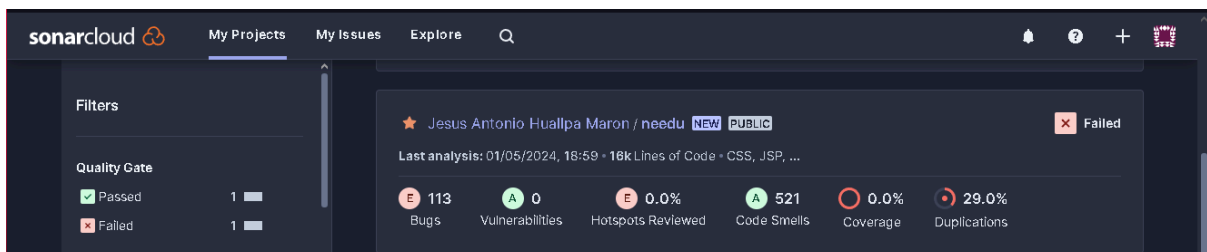
dbname = "DBSOSA"

- Eliminación del almacenamiento del password en texto plano. En su lugar, se implementará un mecanismo seguro de gestión de credenciales.

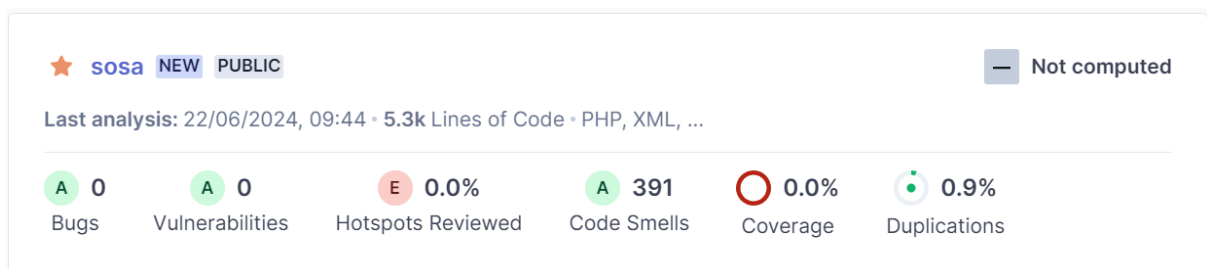
- Restricción del acceso al archivo conf.ini únicamente al personal autorizado, mediante la aplicación de permisos de lectura/escritura adecuados.
- Encriptación de los datos sensibles (url, username, dbname) utilizando técnicas criptográficas robustas.
- Implementación de un proceso de actualización periódica de las credenciales para mantener la seguridad de la conexión.
- Revisión y validación de los cambios por el equipo de seguridad para asegurar la correcta sanitización del archivo.

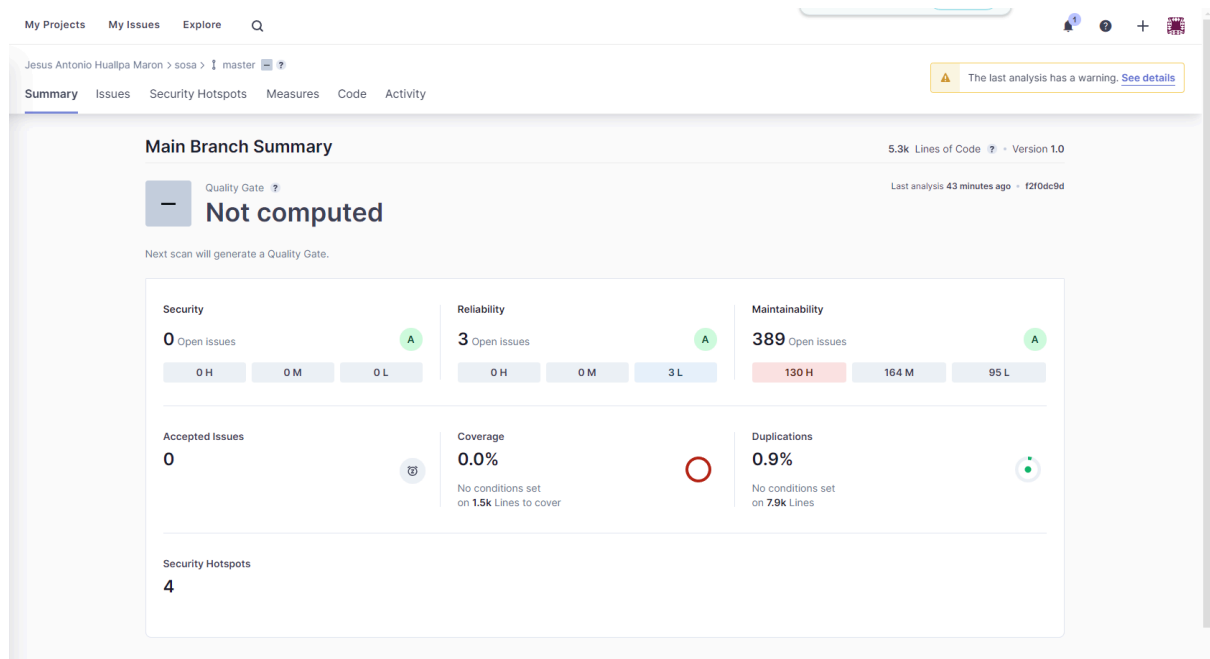
## Código Duplicado

**Version de NeedU (jsp) tiene 113 Bugs y 29% de Duplicado**



**Version de SOSA (php) tiene 10 Bugs y 0.9%, gracias a los incluye en las vistas**





## d)Anexo Behat Instalación

### Paso 1: para configurar Behat en SOSA:

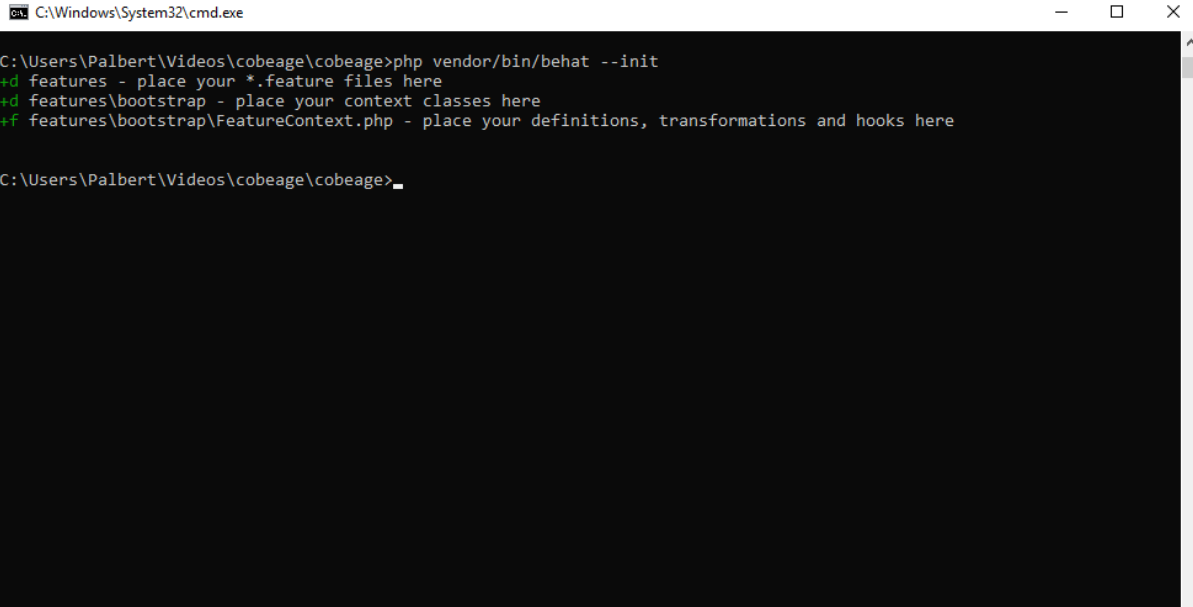
añadir Behat al proyecto con el siguiente comando:  
composer require --dev behat/behat

```
C:\Windows\System32\cmd.exe - composer require --dev behat/behat
- Locking symfony/translation (v7.1.1)
- Locking symfony/translation-contracts (v3.5.0)
- Locking symfony/var-exporter (v7.1.1)
- Locking symfony/yaml (v7.1.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 22 installs, 0 updates, 0 removals
- Downloading symfony/polyfill-ctype (v1.30.0)
- Downloading symfony/yaml (v7.1.1)
- Downloading symfony/translation-contracts (v3.5.0)
- Downloading symfony/polyfill-mbstring (v1.30.0)
- Downloading symfony/translation (v7.1.1)
- Downloading psr/event-dispatcher (1.0.0)
- Downloading symfony/event-dispatcher-contracts (v3.5.0)
- Downloading symfony/event-dispatcher (v7.1.1)
- Downloading symfony/var-exporter (v7.1.1)
- Downloading symfony/deprecation-contracts (v3.5.0)
- Downloading psr/container (2.0.2)
- Downloading symfony/service-contracts (v3.5.0)
- Downloading symfony/dependency-injection (v7.1.1)
- Downloading symfony/polyfill-intl-normalizer (v1.30.0)
- Downloading symfony/polyfill-intl-grapheme (v1.30.0)
- Downloading symfony/string (v7.1.1)
- Downloading symfony/console (v7.1.1)
- Downloading symfony/filesystem (v7.1.1)
- Downloading symfony/config (v7.1.1)
- Downloading behat/transliterator (v1.5.0)
- Downloading behat/gherkin (v4.9.0)
- Downloading behat/behat (v3.14.0)
17/22 [=====] 77%
```

### Ejecutar Behat desde la Ruta Completa



php vendor/bin/behat --init



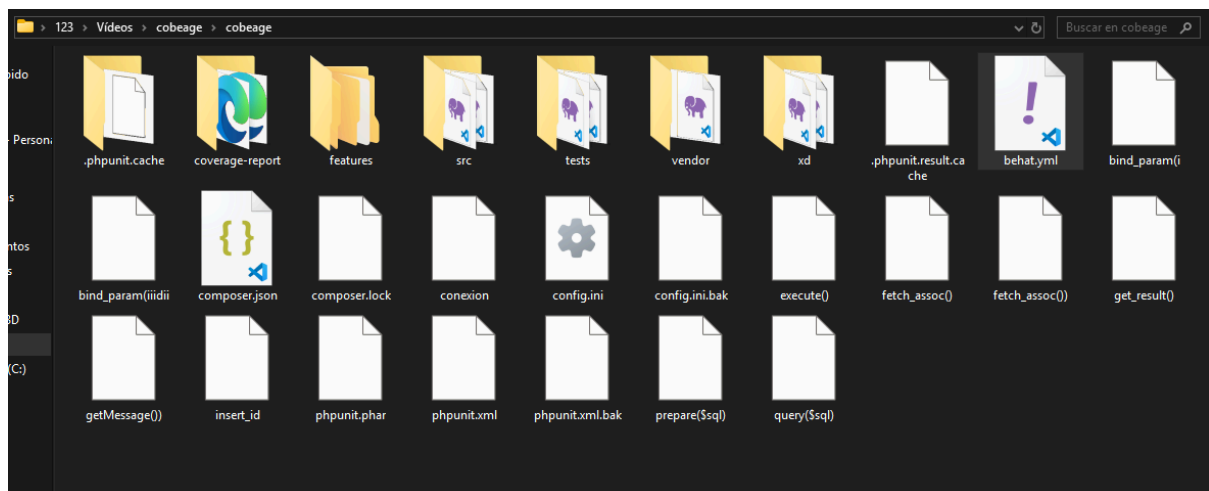
```
C:\Windows\System32\cmd.exe

C:\Users\Palbert\Videos\cobeage\cobeage>php vendor/bin/behat --init
+d features - place your *.feature files here
+d features\bootstrap - place your context classes here
+f features\bootstrap\FeatureContext.php - place your definitions, transformations and hooks here

C:\Users\Palbert\Videos\cobeage\cobeage>
```

composer require --dev behat/behat behat/mink behat/mink-extension  
behat/mink-goutte-driver --with-all-dependencies

**crear behat.yml**



**contenido de behat.yml**

default:

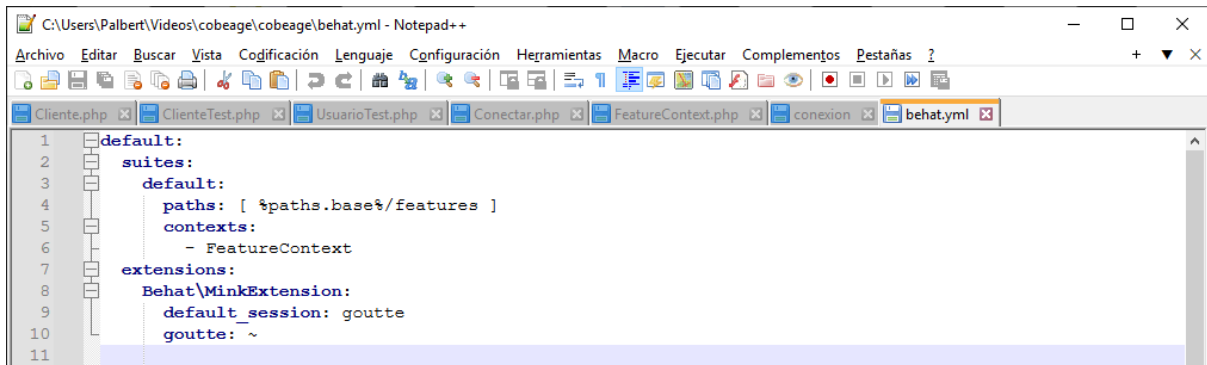
suites:

default:

paths: [ %paths.base%/features ]

contexts:

- FeatureContext  
extensions:  
Behat\MinkExtension:  
  default\_session: goutte  
goutte: ~



```
1 default:
2   suites:
3     default:
4       paths: [ %paths.base%/features ]
5       contexts:
6         - FeatureContext
7   extensions:
8     Behat\MinkExtension:
9       default_session: goutte
10    goutte: ~
11
```

## CREAR UN FEATURE

cliente.feature

Feature: Gestión de Clientes

Scenario: Listar clientes

Given I have a Cliente instance

When I request the list of clients

Then the list should contain at least one client

Scenario: Agregar un nuevo cliente

Given I have a Cliente instance

When I add a new client with nombre "John", apellido "Doe", correo "john.doe@example.com", iddocumento "DNI", documento "12345678", telefono "987654321"

Then the client list should include "John Doe"

Scenario: Editar un cliente existente

Given I have a Cliente instance

And there is a client with idcliente 1

When I edit the client with idcliente 1, setting nombre to "Jane", apellido to "Smith", correo to "jane.smith@example.com", iddocumento to "DNI", documento to "87654321", telefono to "654321987"

Then the client list should include "Jane Smith"

Scenario: Eliminar un cliente existente

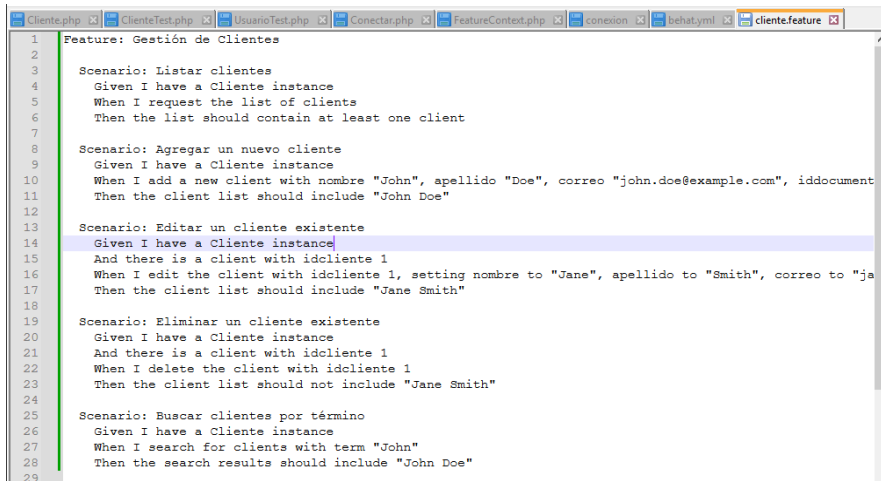
Given I have a Cliente instance

And there is a client with idcliente 1

When I delete the client with idcliente 1

Then the client list should not include "Jane Smith"

Scenario: Buscar clientes por término  
Given I have a Cliente instance  
When I search for clients with term "John"  
Then the search results should include "John Doe"

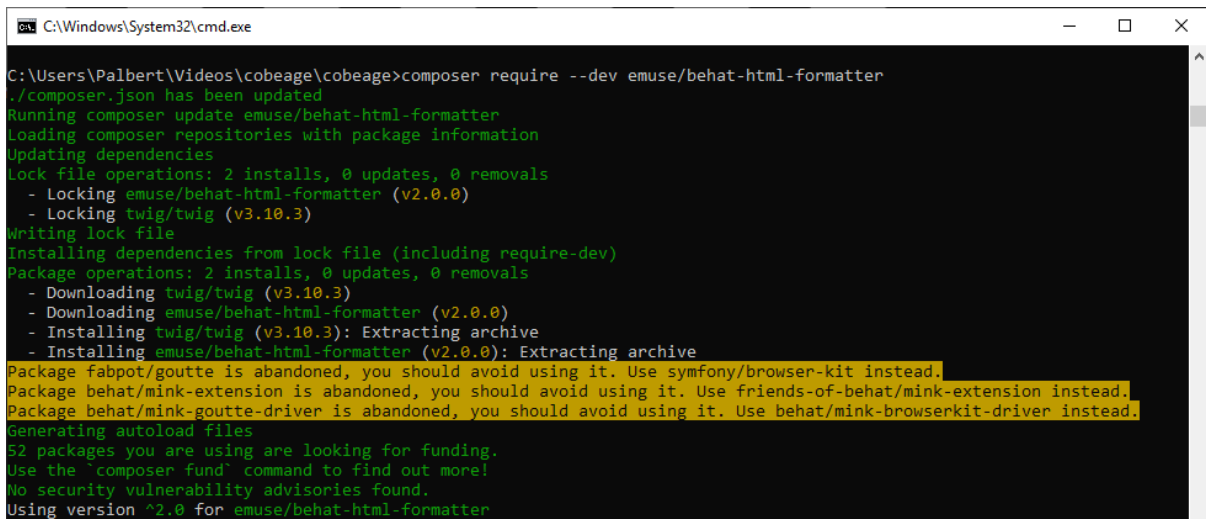


```
1 Feature: Gestión de Clientes
2
3 Scenario: Listar clientes
4   Given I have a Cliente instance
5   When I request the list of clients
6   Then the list should contain at least one client
7
8 Scenario: Agregar un nuevo cliente
9   Given I have a Cliente instance
10  When I add a new client with nombre "John", apellido "Doe", correo "john.doe@example.com", iddocument
11  Then the client list should include "John Doe"
12
13 Scenario: Editar un cliente existente
14   Given I have a Cliente instance
15   And there is a client with idcliente 1
16   When I edit the client with idcliente 1, setting nombre to "Jane", apellido to "Smith", correo to "ja
17   Then the client list should include "Jane Smith"
18
19 Scenario: Eliminar un cliente existente
20   Given I have a Cliente instance
21   And there is a client with idcliente 1
22   When I delete the client with idcliente 1
23   Then the client list should not include "Jane Smith"
24
25 Scenario: Buscar clientes por término
26   Given I have a Cliente instance
27   When I search for clients with term "John"
28   Then the search results should include "John Doe"
29
```

## Paso 2: Para generar un Informe

### GENERAR UN REPORTE

composer require --dev emuse/behat-html-formatter



```
C:\Windows\System32\cmd.exe

C:\Users\Palbert\Videos\cobeage\cobeage>composer require --dev emuse/behat-html-formatter
./composer.json has been updated
Running composer update emuse/behat-html-formatter
Loading composer repositories with package information
Updating dependencies
Lock file operations: 2 installs, 0 updates, 0 removals
  - Locking emuse/behat-html-formatter (v2.0.0)
  - Locking twig/twig (v3.10.3)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Downloading twig/twig (v3.10.3)
  - Downloading emuse/behat-html-formatter (v2.0.0)
  - Installing twig/twig (v3.10.3): Extracting archive
  - Installing emuse/behat-html-formatter (v2.0.0): Extracting archive
Package fabpot/goutte is abandoned, you should avoid using it. Use symfony/browser-kit instead.
Package behat/mink-extension is abandoned, you should avoid using it. Use friends-of-behat/mink-extension instead.
Package behat/mink-goutte-driver is abandoned, you should avoid using it. Use behat/mink-browserkit-driver instead.
Generating autoload files
52 packages you are looking for funding.
Use the `composer fund` command to find out more!
No security vulnerability advisories found.
Using version ^2.0 for emuse/behat-html-formatter
```

### MODIFICAR YML

default:

suites:

default:

name: "Sosa BDD Features" # Nombre personalizado para la suite

paths:

- '%paths.base%/features'

contexts:

- FeatureContext

```
extensions:
  Behat\MinkExtension:
    default_session: goutte
    goutte: ~
formatters:
  custom:
    class: 'Behat\Behat\Formatter\JUnitFormatter'
    parameters:
      output_path: null
extensions:
  - \Behavioral
```

**generar el informe:**

```
php vendor/bin/behat --format html --out build/html/behat
```