Informe de Análisis de Código - SonarQube

# 1. Datos Generales del Proyecto

Nombre del proyecto: obligatorio-devops-iz

URL del repositorio: https://github.com/luciai27/obligatorio-devops-iz

Fecha del análisis: [09/05/2025]

Rama del código analizado: branch

Herramienta utilizada: SonarQube

Versión de SonarQube: Web

Responsable del análisis: Gonzalo Zunino y Lucia Ibarburu

# 2. Resumen General del Análisis

|  |  |
| --- | --- |
| Métrica | Valor |
| Bugs | 4 |
| Vulnerabilidades | 13 |
| Code Smells | 45 |
| Cobertura de Tests | 90% |
| Duplicación de código | 3% |
| Líneas de código | 854 |

# 3. Principales Problemas Detectados

## 🐞 Bugs

* - Add "lang" and/or "xml:lang" attributes to this "<html>" element

result/views/index.html

* - Unexpected missing generic font family

result/views/stylesheets/style.css

* - Unexpected missing generic font family🔐 Vulnerabilidades

vote/static/stylesheets/style.css

* - Add "lang" and/or "xml:lang" attributes to this "<html>" element

vote/templates/index.html

La severidad\* es media para los 4 bugs

\*La gravedad ahora está directamente relacionada con la calidad del software afectada.

Hay cinco niveles de gravedad: bloqueador, alto, medio, bajo e informativo.

## 💨 Code Smells

* **Add the "let", "const" or "var" keyword to this declaration of "data" to make it explicit.**

result/views/app.js

* Unexpected var, use let or const instead.

result/server.js

Este error se repite 6 veces

* Unexpected var, use let or const instead.

result/test/render.js

Este error se repite 3 veces

* Unexpected var, use let or const instead.

result/test/app.js

Este error se repite 10 veces

* Specify a storage request for this container.

k8s-specifications/db-deployment.yaml

* Specify a CPU request for this container.

k8s-specifications/db-deployment.yaml

* Specify a memory request for this container.

k8s-specifications/db-deployment.yaml

* Specify a storage request for this container.

k8s-specifications/redis-deployment.yaml

* Specify a CPU request for this container.

k8s-specifications/redis-deployment.yaml

* Specify a memory request for this container.

k8s-specifications/result-deployment.yaml

* Specify a CPU request for this container.

k8s-specifications/result-deployment.yaml

* Use a specific version tag for the image.

k8s-specifications/result-deployment.yaml

* Specify a memory request for this container.

k8s-specifications/result-deployment.yaml

* Specify a storage request for this container.

k8s-specifications/vote-deployment.yaml

* Specify a CPU request for this container.

k8s-specifications/vote-deployment.yaml

* Use a specific version tag for the image.

k8s-specifications/vote-deployment.yaml

* Specify a memory request for this container.

k8s-specifications/vote-deployment.yaml

* Specify a storage request for this container.

k8s-specifications/worker-deployment.yaml

* Specify a CPU request for this container.

k8s-specifications/worker-deployment.yaml

* Use a specific version tag for the image.

k8s-specifications/worker-deployment.yaml

* Specify a memory request for this container.

k8s-specifications/worker-deployment.yaml

* Remove cache after installing packages or store it in a cache mount.

result/tests/Dockerfile

* Replace this ADD instruction with a COPY instruction.

result/tests/Dockerfile

* Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.

worker/Dockerfile

* Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.

worker/Dockerfile

* Add a 'protected' constructor or the 'static' keyword to the class declaration.

worker/Program.cs

* Change this condition so that it does not always evaluate to 'False'.

worker/Program.cs

## 🔐 Vulnerability

* **Make sure this PostgreSQL database password gets changed and removed from the code.**

result/server.js

* Bind this resource's automounted service account to RBAC or disable automounting.

k8s-specifications/db-deployment.yaml

* Specify a memory limit for this container.

k8s-specifications/db-deployment.yaml

* Specify a storage limit for this container.

k8s-specifications/db-deployment.yaml

* Bind this resource's automounted service account to RBAC or disable automounting.

k8s-specifications/redis-deployment.yaml

* Specify a memory limit for this container.

k8s-specifications/redis-deployment.yaml

* Specify a storage limit for this container.

k8s-specifications/redis-deployment.yaml

* Bind this resource's automounted service account to RBAC or disable automounting.

k8s-specifications/result-deployment.yaml

* Specify a memory limit for this container.

k8s-specifications/result-deployment.yaml

* Bind this resource's automounted service account to RBAC or disable automounting.

k8s-specifications/vote-deployment.yaml

* Specify a memory limit for this container.

k8s-specifications/vote-deployment.yaml

* Bind this resource's automounted service account to RBAC or disable automounting.

k8s-specifications/worker-deployment.yaml

* Specify a memory limit for this container.

k8s-specifications/worker-deployment.yaml

# 4. Cobertura de Tests

Porcentaje general: [Ej: 90%]

# 5. Recomendaciones

* Corregir las vulnerabilidades detectadas, priorizando aquellas de mayor severidad, ya que representan riesgos potenciales para la seguridad del sistema.
* Revisar los bugs funcionales, validando su impacto en la ejecución de la aplicación, especialmente si afectan funcionalidades críticas.
* Refactorizar los 45 code smells, comenzando por los que aparecen en archivos clave o que generan mayor complejidad.
* Mantener y proteger la cobertura de pruebas (90%), incluyendo casos adicionales para nuevas funcionalidades o cambios futuros.
* Monitorear la duplicación de código, y evitar que aumente con futuras versiones mediante prácticas como la reutilización de funciones o módulos comunes.
* Incluir SonarQube en el proceso de integración continua (CI) para garantizar que cada cambio pase por validaciones de calidad antes de integrarse al repositorio principal.

# 6. Capturas de Pantalla

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# 7. Conclusión

El análisis de calidad de código realizado con SonarQube sobre un total de 854 líneas de código muestra un estado general bueno, aunque con áreas que requieren atención.

Se identificaron 4 bugs y 13 vulnerabilidades, lo que representa un nivel de riesgo moderado en términos de seguridad. Las 45 code smells indican posibles problemas de mantenibilidad, aunque no críticos por el momento. La cobertura de pruebas del 90% es un punto destacado, ya que proporciona una alta confianza en el comportamiento del sistema frente a cambios o regresiones. Asimismo, la duplicación de código del 3% se encuentra dentro de un umbral aceptable.

En resumen, el proyecto presenta una base sólida, pero se recomienda priorizar la revisión y corrección de las vulnerabilidades detectadas, con especial atención a su severidad, y realizar mejoras en la limpieza del código para reducir la deuda técnica acumulada.