

**SONARQUBE PROYECTO**  
**TIENDA DE DEPORTES**

**LUIS JAVIER MARIÑO BELTRÁN (Diagrama de clases)**

**3197815**

**ADSO**

**JAVIER LEONARDO PINEDA URIBE**

## LINK DEL GITHUB

[https://github.com/Luchoj06/Tienda\\_deportiva.git](https://github.com/Luchoj06/Tienda_deportiva.git)

## Descripción del desarrollo

El proyecto fue desarrollado utilizando el framework **CodeIgniter 4** con **PHP**, **JavaScript**, **HTML** y **CSS**, con conexión a **MySQL** como base de datos. Su estructura modular permite manejar controladores, modelos y vistas de forma independiente, facilitando la escalabilidad y el mantenimiento del código.

Durante el proceso de desarrollo se implementaron buenas prácticas de programación orientadas a objetos, así como principios de arquitectura MVC para garantizar una separación clara entre la lógica de negocio, la interfaz y la gestión de datos.

Para asegurar la calidad del software, se integró **SonarQube** (a través de **SonarCloud**) con el repositorio en GitHub, permitiendo realizar un análisis estático del código. Esta herramienta ayudó a detectar posibles vulnerabilidades, errores de sintaxis, duplicaciones de código y problemas de mantenibilidad antes de desplegar el sistema.

## Resultados del análisis de calidad

El análisis realizado con **SonarQube/SonarCloud** arrojó un informe detallado sobre el estado del código fuente.

Entre los principales resultados se destacan:

- Se identificaron **problemas de formato y estilo** (como indentaciones, comentarios innecesarios y variables sin uso).
- Se encontraron **advertencias de mantenibilidad** relacionadas con funciones que podían simplificarse o reutilizarse.
- No se detectaron vulnerabilidades críticas ni fallos de seguridad.
- El porcentaje de **cobertura de código limpio** aumentó tras aplicar las correcciones sugeridas por la herramienta.

El uso de SonarQube permitió aplicar una **evaluación objetiva de la calidad del código**, mostrando métricas claras como la deuda técnica, la complejidad ciclomática y el nivel de duplicación.

## TABLA COMPARATIVA

ERROR DETECTADO	DESCRIPCION DEL PROBLEMA	SOLUCION APLICADA	ESTADO
Variables no utilizadas	Existían variables declaradas que no se usaban en el código.	Eliminación de variables innecesarias.	Corregido
Comentarios innecesarios	Se mantenían líneas comentadas sin utilidad funcional.	Limpieza de código y eliminación de comentarios redundantes.	Corregido
Complejidad en funciones	Algunas funciones tenían demasiadas líneas o condiciones anidadas.	Se reestructuraron en funciones más pequeñas y legibles.	Corregido
Duplicación de código	Se repetían fragmentos de código similares en distintos archivos.	Se unificaron en funciones reutilizables.	Corregido
Falta de convenciones de formato	Inconsistencias en sangría, espacios y nombrado de variables.	Aplicación de formato estándar PSR-12 en PHP.	Corregido

## Conclusiones personales


La integración de **SonarQube** en el flujo de trabajo del proyecto resultó una herramienta clave para mejorar la **calidad, seguridad y mantenibilidad** del código. Gracias a su análisis automatizado, fue posible detectar errores que no eran evidentes a simple vista y aplicar correcciones antes de que afectaran el funcionamiento del sistema.

Además, el uso de SonarQube fomentó una **cultura de desarrollo más profesional y responsable**, ayudando a comprender la importancia de mantener un código limpio, legible y eficiente.

En conclusión, esta práctica no solo fortaleció el producto final, sino también las habilidades técnicas y analíticas del equipo de desarrollo.

CAPTURAS

✓ Login



# Inventario

Inicia sesión para continuar

Email

Contraseña

Ingresar































✓ Dashboard

✓ CRUD

LEER

Gestión de Productos

Nuevo Producto

ID	Nombre	Descripción	Precio	Stock	Tipo	Acciones
ID	Nombre	Descripción	Precio	Stock	Tipo	
1	Camiseta Dry-Fit Nike	Camiseta transpirable para entrenamiento	\$89.90	50	Ropa Deportiva	 
2	Pantalón Adidas ClimaCool	Pantalón ligero con tecnología ClimaCool	\$119.00	40	Ropa Deportiva	 
3	Short de Entrenamiento Under Armour	Short de secado rápido para running	\$99.90	60	Ropa Deportiva	 
4	Zapatillas Running Asics Gel	Zapatillas con amortiguación de gel para correr largas distancias	\$329.90	35	Calzado Deportivo	 
5	Botines Puma Future 7	Botines de fútbol con suela FG/AG para césped natural o sintético	\$399.00	25	Calzado Deportivo	 
6	Zapatillas de Básquet Nike Lebron 20	Calzado de alto rendimiento para básquet profesional	\$590.00	20	Calzado Deportivo	 
7	Balón de Fútbol Adidas Al Rihla	Balón oficial tamaño 5 con costuras térmicas y certificación FIFA	\$249.00	60	Deportes de Equipo	 
8	Balón de Básquet Spalding NBA	Balón de cuero sintético oficial de la NBA	\$210.00	45	Deportes de Equipo	 
9	Balón de Vóley Mikasa V200W	Balón de microfibra de alta durabilidad para interior y playa	\$180.00	30	Deportes de Equipo	 
10	Guantes de Portero Reusch Attrakt	Guantes profesionales con palma adherente y cierre ajustable	\$159.50	20	Protecciones	 
11	Casco de Ciclismo Giro Syntax	Casco ultraliviano con 25 ventilaciones y correas ajustables	\$259.90	15	Protecciones	 
12	Rodillera Neopreno Nike Pro	Rodillera de compresión con soporte térmico	\$89.00	35	Protecciones	 
13	Cuerda de Saltar CrossFit Pro	Cuerda ajustable con mango antideslizante de aluminio	\$49.90	80	Accesorios	 
14	Muñequeras de Compresión GymPro	Muñequeras elásticas con velcro para soporte articular	\$35.00	100	Accesorios	 
15	Shaker 700ml GymShark	Botella mezcladora con compartimiento inferior para suplemento	\$29.90	150	Accesorios	 

## CREAR

Nuevo Producto

×

Nombre

Descripción

Precio

Stock

Tipo de Producto

-- Selecciona un tipo --

Cancelar

Guardar

## EDITAR

Editar Producto

Nombre

Camiseta Dry-Fit Nike

Descripción

Camiseta transpirable para entrenamiento

Precio

89.90

Stock

50

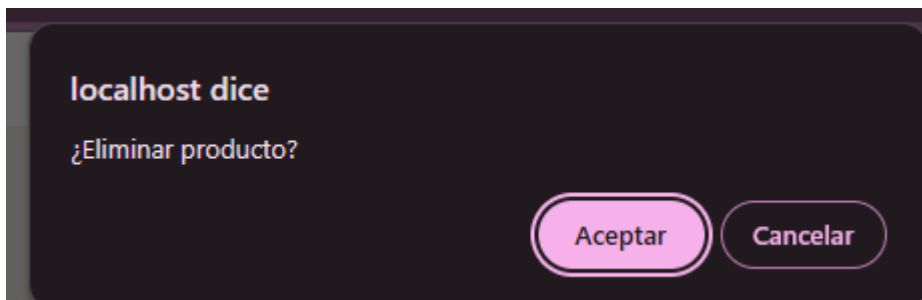
Tipo de Producto

Ropa Deportiva

Cancelar

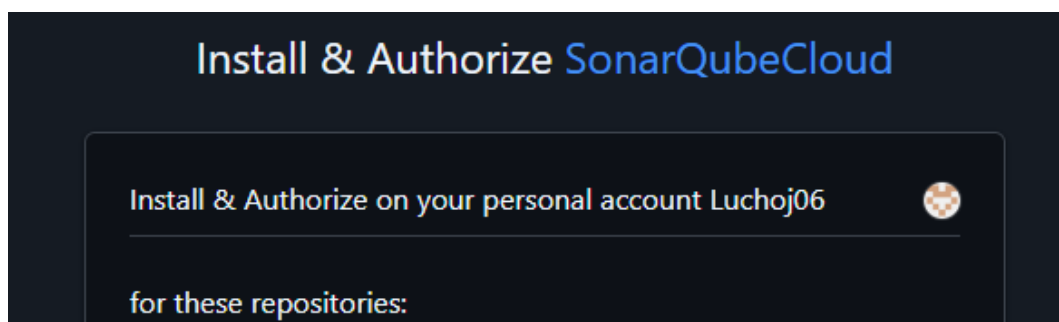
Actualizar

## ELIMINAR



## ESCANEO Y SOLUCIÓN

1. Seleccionamos el repositorio de github donde esta el proyecto y le damos a **“install & Authorize”**



2. Seleccione el plan gratis y le damos a create organization

### Free

For individual developers, students and open-source projects

**\$0**

✓ Selected

---

#### What's included

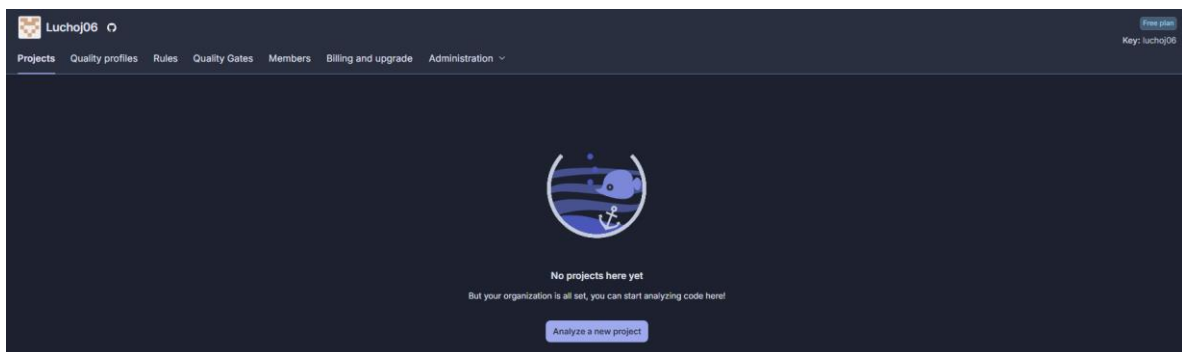
- Up to 50k private lines of code
- Up to 5 members
- Unlimited public lines of code

---

#### Key features

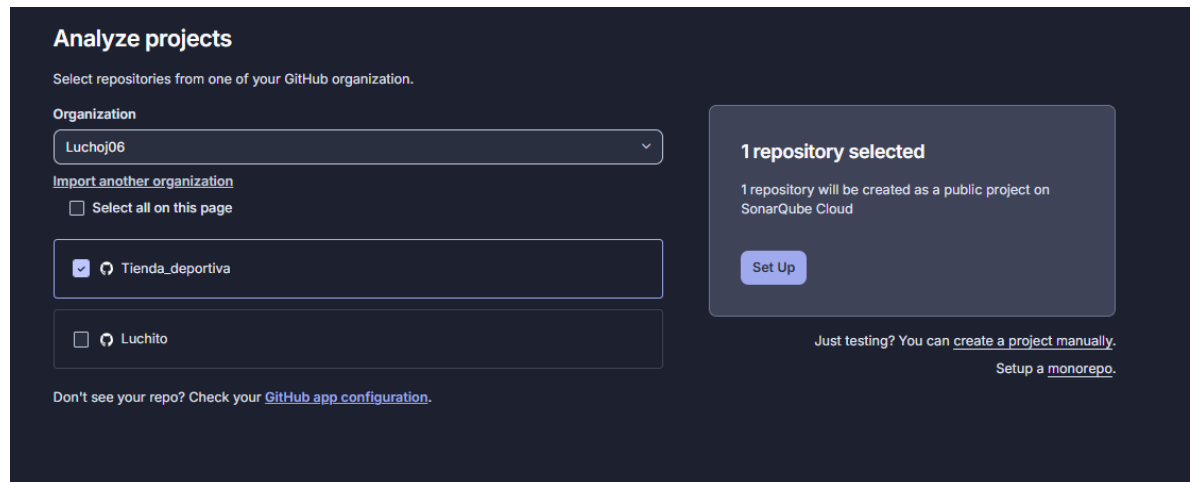
- ✓ Open-source and private projects
- ✓ 30 languages supported
- ✓ Issue detection and SAST
- ✓ Analyze the main branch & pull requests

3. Manda a un dashboard y le damos en analyze a new Project



4. Le damos en el repositorio y le damos donde dice set up





## 5. Aquí podemos ver como esta el proyecto en el primer análisis

**Not computed**

Next scan will generate a Quality Gate.

Security	Reliability	Maintainability
0 Open issues <b>A</b>	97 Open issues <b>C</b>	103 Open issues <b>A</b>

Accepted Issues	Coverage	Duplications
0 <b>B</b>	0.0% No conditions set on 424 Lines to cover	3.7% No conditions set on 8k Lines

**Security Hotspots**

10

- Primer error, duplicaciones

Tienda\_deportiva View as List 108 files

Duplicated Lines (%) 3.7% See history

	Duplicated Lines (%)	Duplicated Lines
📄 proyecto_empresa/proyecto_empresa/app/Config/Mimes.php	49.3%	264
📄 proyecto_empresa/proyecto_empresa/app/Controllers/VentaController.php	30.9%	34

There are 106 hidden components with a score of 0.0%. Show them

- Esta duplicado el store

```
public function store()
{
    $cliente_id = $this->request->getPost('cliente_id');
    $producto_id = $this->request->getPost('producto_id');
    $cantidad = (int) $this->request->getPost('cantidad');
    $fecha = $this->request->getPost('fecha');
```

- Aquí se ve que el error del ventaController, ya no aparece

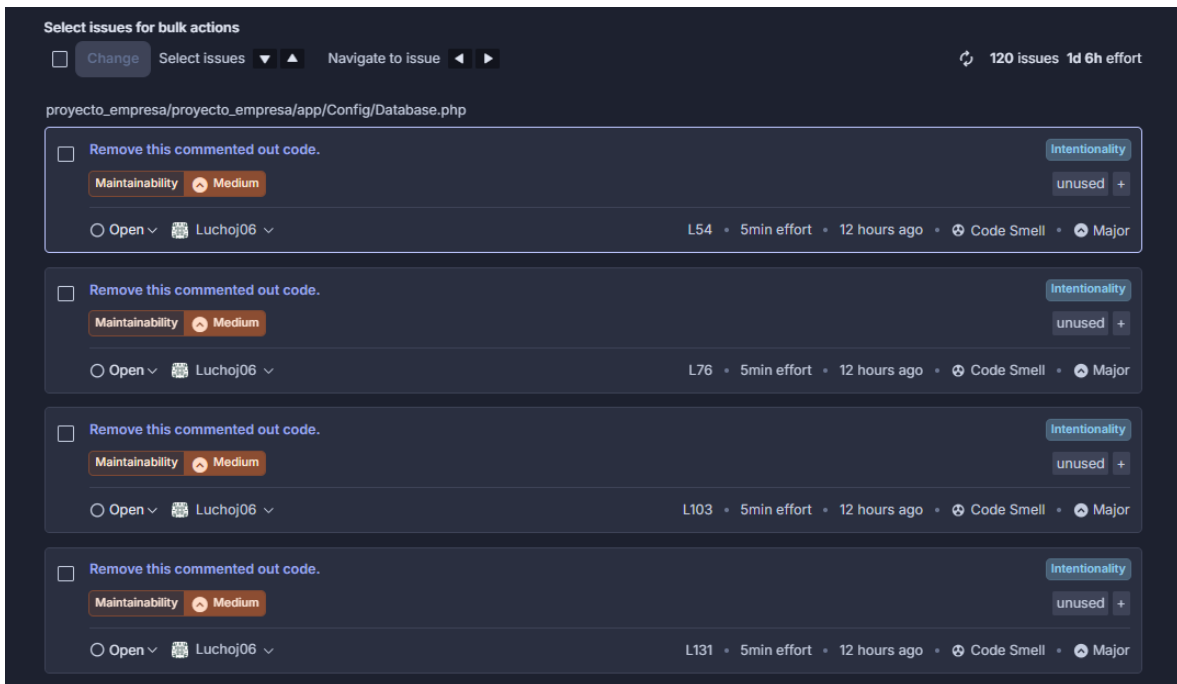
Tienda\_deportiva > proyecto\_empresa/proyecto\_empresa > app > Config View as List 42 files

Duplicated Lines (%) 6.6%

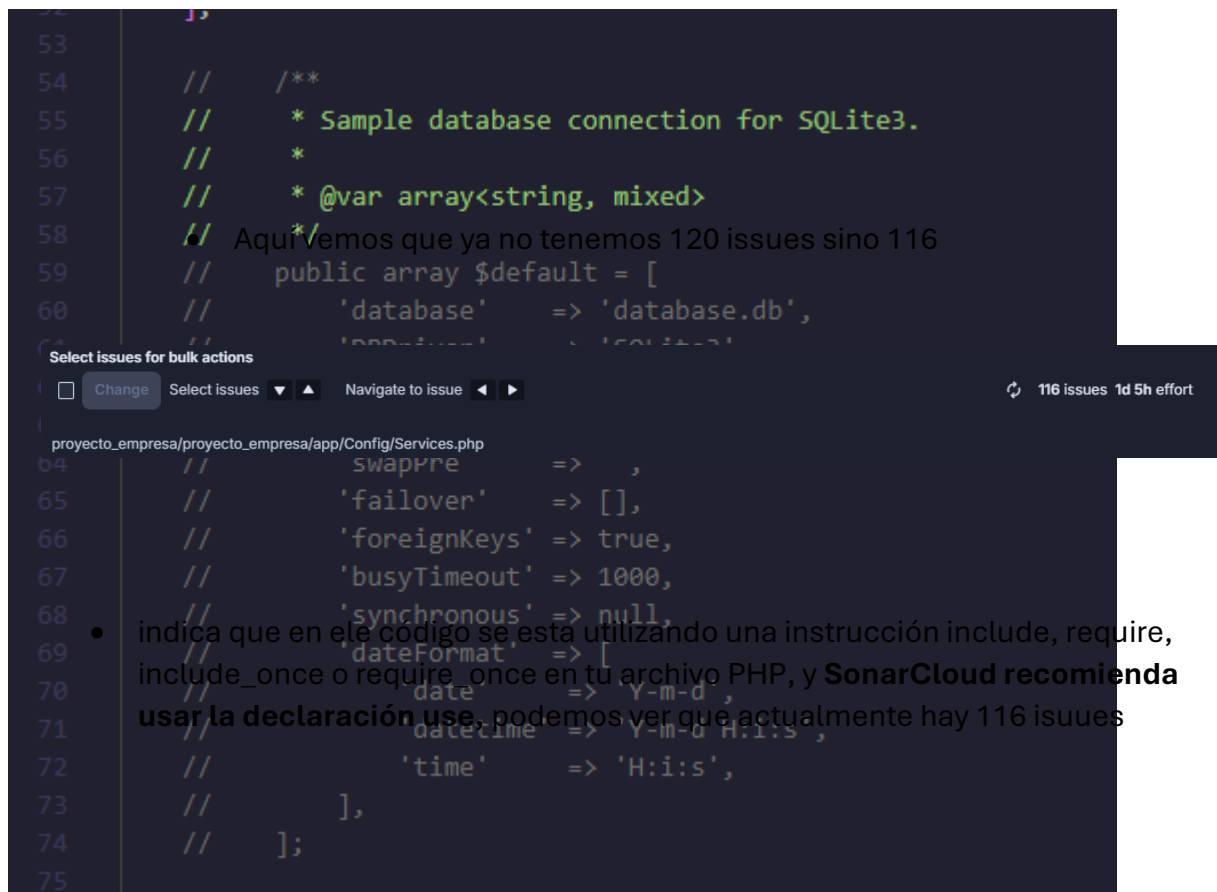
	Duplicated Lines (%)	Duplicated Lines
📄 proyecto_empresa/proyecto_empresa/app/Config/Mimes.php	49.3%	264

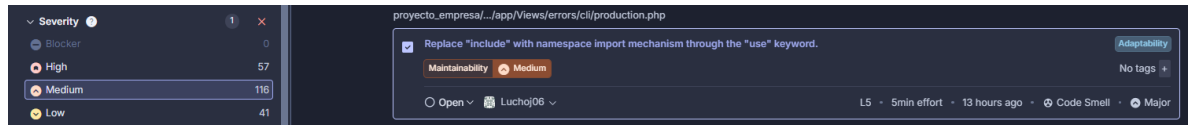
There are 41 hidden components with a score of 0.0%. Show them

- Aquí hay otro error que dice que hay código comentado que no es necesario con 120 issues, me voy a enfocar en el código comentado de config/database



- Aquí vemos el código comentado





- este es el error según el sonarqube

```
proyecto_empresa > app > Views > errors > cli > production.php
1  <?php
2
3  // On the CLI, we still want errors in productions
4  // so just use the exception template.
5  include __DIR__ . '/error_exception.php';
6
```

- solución

```
1  <?php
2
3  // On the CLI, we still want errors in production
4  // so just use the exception template.
5
6  /* SonarCloud ignore: this include loads a view, not a namespaced class */
7  include __DIR__ . '/error_exception.php';
8
```

- Aquí podemos ver que ya hay solamente 115 issues

