

Requisito 54

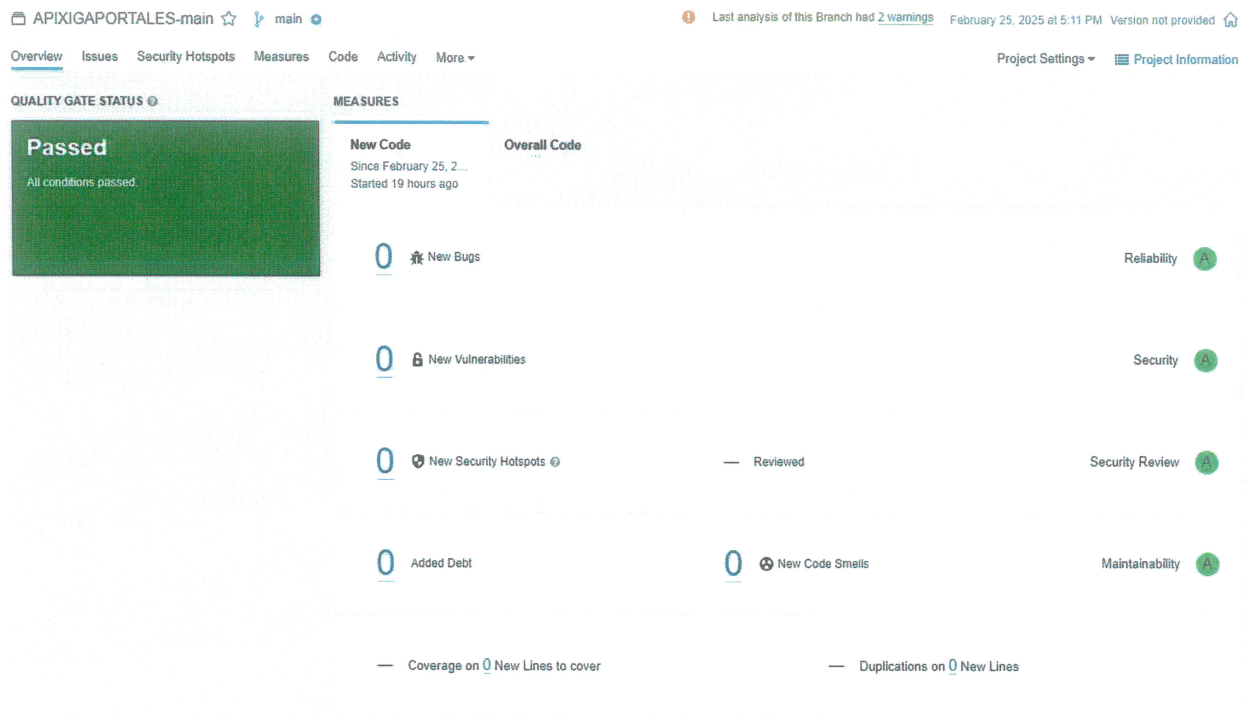
Evidencia de Pruebas de Código Seguro mediante el escaneo con herramienta SonarQube al Portal XIGA Clientes

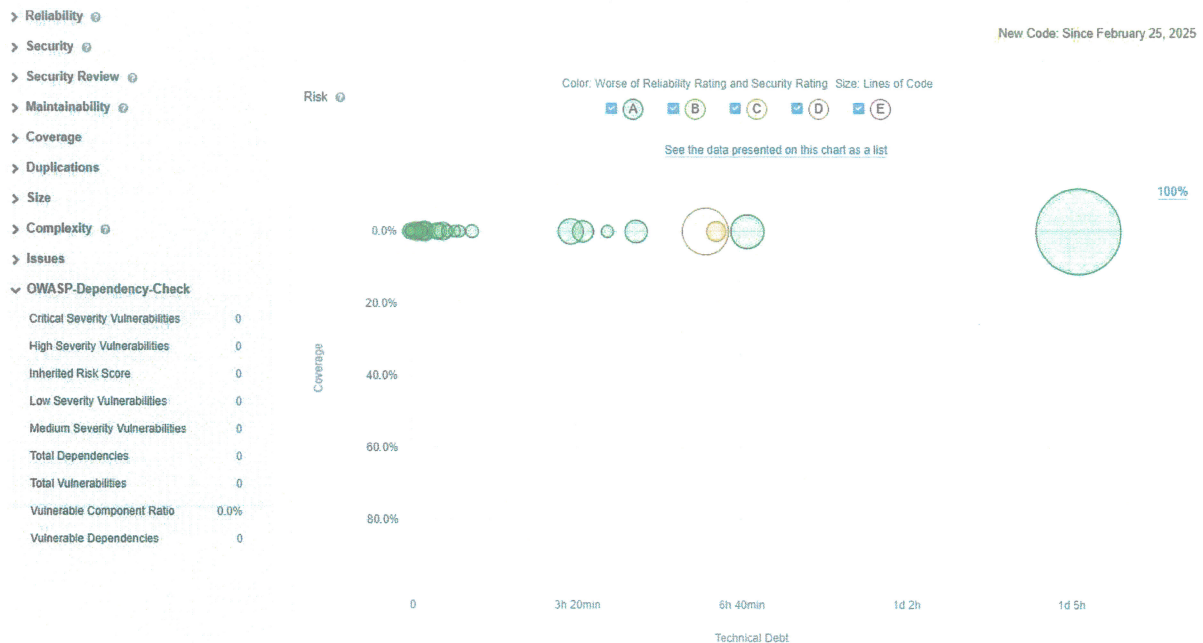
SonarQube es una plataforma de código abierto diseñada para evaluar y mejorar la calidad del código de proyectos de software. Funciona como una herramienta de análisis estático que examina el código fuente en busca de problemas de seguridad, errores, vulnerabilidades, y malas prácticas de codificación. Además de proporcionar una evaluación detallada de la calidad del código SonarQube, ofrece métricas y estadísticas útiles para medir la salud general del proyecto.

En el siguiente resumen muestra los principales **indicadores de calidad del código** después del análisis con la herramienta SonarQube. Se identificaron:

- ✓ 0 bugs,
- ✓ 0 vulnerabilidades y
- ✓ 0 code smells,

Aprobando con clasificación “A” en todos los rubros.





En el rubro de las pruebas en apego a OWASP:

OWASP-Dependency-Check

Critical Severity Vulnerabilities	0
High Severity Vulnerabilities	0
Inherited Risk Score	0
Low Severity Vulnerabilities	0
Medium Severity Vulnerabilities	0
Total Dependencies	0
Total Vulnerabilities	0
Vulnerable Component Ratio	0.0%
Vulnerable Dependencies	0

SonarQube y OWASP

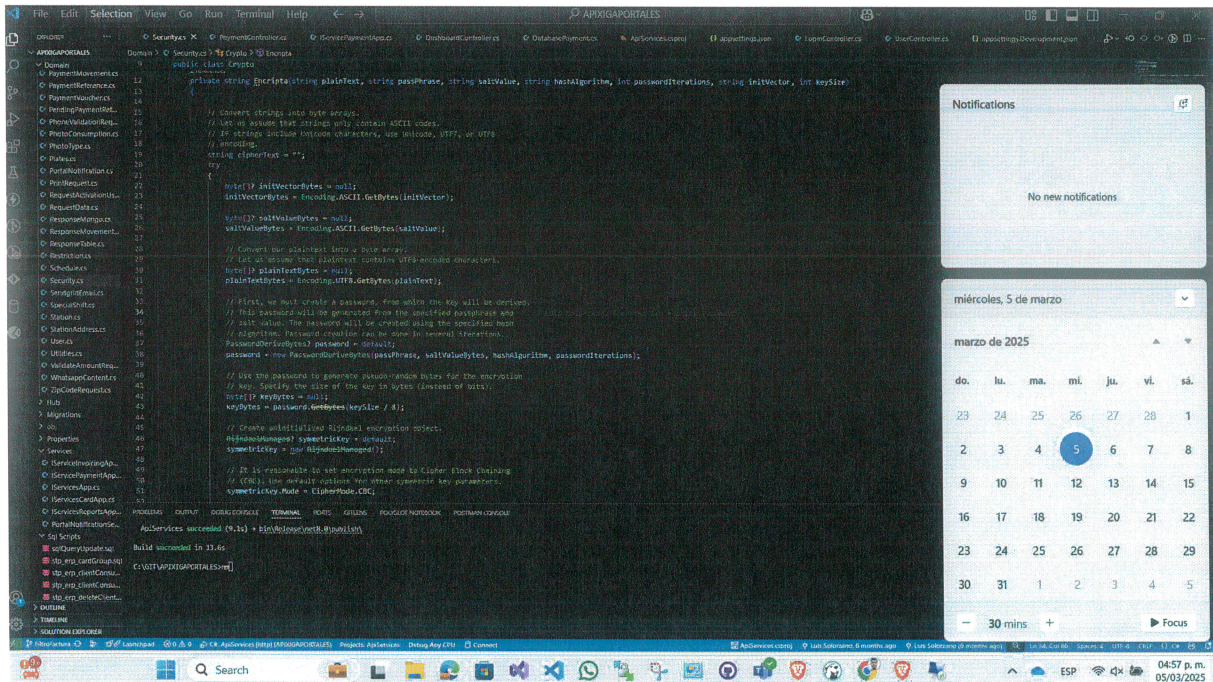
SonarQube es plataforma de análisis estático de código que también puede implementar la revisión con las pautas y recomendaciones de seguridad de OWASP. Esto permite identificar y abordar posibles vulnerabilidades y problemas de seguridad en el código fuente de las aplicaciones web.

La integración con OWASP ayuda a garantizar que las aplicaciones están desarrolladas siguiendo buenas prácticas de seguridad, reduciendo así el riesgo de posibles ataques y vulnerabilidades de software.

Las categorías aprobadas aplicantes al Proyecto:

Cryptographic Failures

La exposición de datos confidenciales se asegura al guardar el password del usuario administrador y/ usuarios secundarios, se encripta antes de almacenarse en la base de datos, mediante la librería Crypto de C#



La contraseña encriptada en base de datos

The screenshot displays the DBeaver SQL client interface. The main window shows a query execution window with the following SQL query:

```
select * from gmc_client go where number=15
```

The query results are displayed in a table with the following columns and data:

	base_type	is_check	status	last_purchase_date	last_payment_date	password	Valor
1	B	A		NULL	NULL	f3a0755d493d4abc7cdc97cp3739177989d482	16

On the right side of the interface, there are three floating windows:

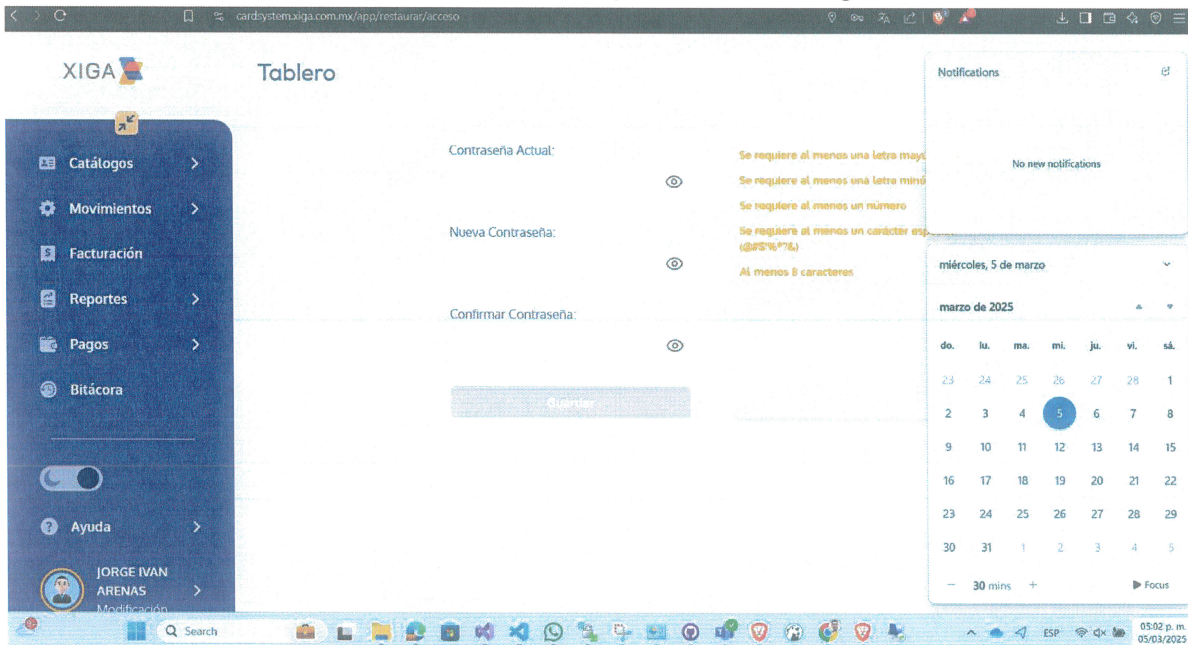
- Notificaciones** (Notifications): A window showing a notification about a screenshot being captured and saved to the clipboard.
- Herramienta Recortes** (Screenshot Tool): A window showing a screenshot of the current screen.
- Calendar**: A calendar for March 2025, showing the date Wednesday, March 5th.

En el código muestra líneas donde valida vigencia de contraseña

The image shows a Windows desktop environment with Visual Studio Code open. The editor displays a C# file named 'Program.cs' within a project structure for 'ServiciosApp'. The code implements a login endpoint that checks client credentials against a database. The right sidebar contains two panels: 'Notifications', which shows 'No new notifications', and a calendar for 'miércoles, 5 de marzo'. The bottom status bar indicates the file is 'Saved' and the project is running on 'Any CPU'.

Broken Authentication

Protección contra ataques donde se le concede a un usuario no autenticado privilegios de acceso una o más cuentas. Vencimiento de contraseña, cuando va a cambiar contraseña, pide que cumpla con las especificaciones de seguridad mandatorias para contraseña segura



Injection

Protección contra ataques de inyección de código en consultas con manejadores de base de datos. Se parametriza la inserción de datos con `I()` y su método de revisión del parámetro, para garantizar que los datos proporcionados por el usuario corresponden únicamente a los datos a capturar en el repositorio de base de datos.

