Desarrollo de software seguro

PROYECTO DE BIBLIOTECA ESPE

Integrantes:

Tiamba Alucho Henry Anthony

Andrés Valarezo

Jeremy Cadena

Ing. Ángel Cudco

Sangolquí, Febrero 2024

Universidad de las Fuerzas Armadas ESPE

NRC: 15390

1. Introducción

El desarrollo de software seguro aborda estos desafíos integrando prácticas y técnicas de seguridad en todas las etapas del ciclo de vida del desarrollo de software. Esto incluye la implementación de controles de seguridad durante la fase de diseño y arquitectura, la adopción de buenas prácticas de codificación segura, la realización de pruebas exhaustivas de seguridad y la aplicación de parches y actualizaciones de forma regular para abordar nuevas vulnerabilidades. Sistema de prestación de libros, una aplicación web multiplataforma diseñada para gestionar y administrar el proceso de prestación de libros para los usuarios de la en la Universidad de las Fuerzas Armadas ESPE. El sistema tiene como objetivo facilitar el control del inventario en la prestación de libros, permitiendo a los usuarios registrarse y realizar el debido proceso de reserva o pedir prestado un libro de la biblioteca. Para lograr un adecuado manejo de la información y enfrentar el desafío de la seguridad se va a desarrollar implementando las mejores prácticas del desarrollo del software seguro a nivel de requisitos como : el modelado de amenazas , caso de abuso, caso de usos, modelado de ataques , ingeniería de requisitos y para el diseño el patrón DDD a nivel de Backend , patrones de diseño, análisis de riesgo y patrones de diseño UI , la implementación de estas buenas prácticas ayudan a tener un sistema con un nivel alto de seguridad garantizando así la confianza por parte del usuario. La implementación de esta combinación de buenas prácticas del desarrollo del software seguro es esencial para asegurar una gestión eficaz de la CID, Con esta solución, el sistema de prestación de libros de la biblioteca estará preparado para enfrentar los retos de un crecimiento de usuarios y garantizar una experiencia confiable y segura tanto para los usuarios como para los administradores involucrados en el sistema de la biblioteca de la ESPE.

2. Objetivos

- Objetivo general
    - Desarrollar e implementar un sistema de prestación de libros seguro y confiable para la Universidad de las Fuerzas Armadas ESPE, que integre prácticas y técnicas de seguridad en todas las etapas del ciclo de vida del desarrollo de software.
- Objetivos específicos
    - Integrar prácticas de desarrollo seguro, como el modelado de amenazas, casos de abuso y casos de uso, en la fase de requisitos del sistema de prestación de libros.
    - Aplicar el patrón de diseño DDD a nivel de Backend y otros patrones de diseño pertinentes para garantizar la seguridad y la eficacia del sistema.

3. Desarrollo

El desarrollo del sistema de prestación de libros para la Universidad de las Fuerzas Armadas ESPE se llevará a cabo utilizando una arquitectura de microservicios. Esta arquitectura dividirá la aplicación en varios servicios independientes, cada uno encargado de una funcionalidad específica, como la gestión de usuarios, la administración del inventario de libros y el proceso de préstamo. Cada microservicio será desarrollado y desplegado de manera independiente, lo que facilitará la escalabilidad y la mantenibilidad del sistema. Además, permitirá una mayor modularidad y flexibilidad, lo que facilitará la incorporación de nuevas funcionalidades y la adaptación a los cambios en los requisitos del usuario. La comunicación entre los microservicios se realizará de manera segura, utilizando protocolos de comunicación cifrados y autenticados, garantizando así la integridad y la confidencialidad de los datos. Esta arquitectura orientada a microservicios ayudará a garantizar un sistema robusto, escalable y seguro para la gestión eficaz de la prestación de libros en la biblioteca de la ESPE.

- Arquitectura de seguridad

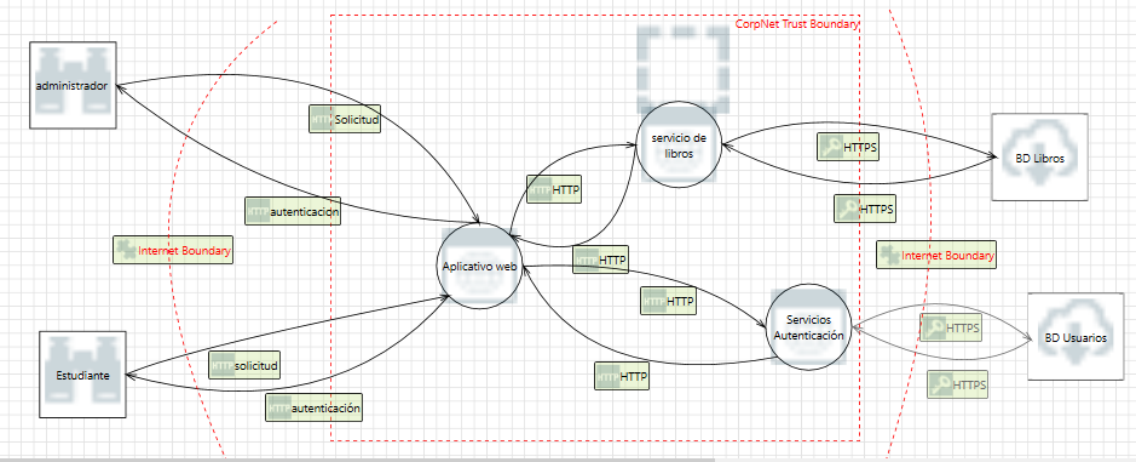

Grafico 1. Arquitectura de modelado de amenazas

Usuarios: Hay dos usuarios identificados, “administrador” y “estudiante”, ambos conectados a un proceso de “solicitud” y “autenticación”.
Aplicativo web: Este actúa como un intermediario, procesando solicitudes HTTP y HTTPS.
Bases de datos: Hay dos bases de datos etiquetadas como “BD Libros” y “BD Usuarios”, que indican bases de datos para libros e información del usuario respectivamente.
Zonas de la red: Los elementos están contenidos dentro de áreas marcadas como “Internet Boundary” y “CorpNet Trust Boundary”, indicando diferentes niveles o zonas dentro de la red. Las líneas rojas discontinuas indican los límites entre estas zonas.

- Tecnologías

Tabla 1. Descripción de tecnologías

Tecnología	Versión	Características
Node.js	v18.12.1	- Soporte para ECMAScript 6 y posteriores.- Mejoras en el rendimiento y la estabilidad del runtime.- Actualizaciones periódicas de seguridad y mantenimiento.
Express.js	v4.17.2	- Enrutamiento flexible y fácil de usar.- Middleware para manejo de solicitudes HTTP.- Amplia comunidad de desarrollo y soporte activo.
MySQL	v8.0.28	- Motor de almacenamiento InnoDB por defecto.- Mejoras en rendimiento y seguridad.- Soporte para JSON y funciones analíticas.
bcrypt	v5.0.1	- Implementación de hash seguro de contraseñas.- Compatible con diferentes versiones de Node.js.- Documentación clara y ejemplos de uso.
JSON Web Tokens	v8.5.1	- Implementación robusta de JWT para autenticación y autorización.- Soporte para algoritmos de firma como HS256, RS256, etc.- Buena integración con frameworks web.
Python	v3.10.0	- Sintaxis clara y legible.- Amplia biblioteca estándar y soporte para paquetes de terceros.- Activo desarrollo y comunidad activa.

- Implementación
  - Microservicio de autenticación con roles y permisos
    - Funcionalidad
      - Registro de usuarios con nombre de usuario y contraseña.
      - Inicio de sesión seguro utilizando JWT (JSON Web Tokens).
      - Asignación de roles a los usuarios.
      - Definición de permisos asociados a roles.
      - Verificación de permisos antes de permitir el acceso a recursos protegidos.

    - Base de datos
      La base utilizada es MySQL donde se encuentra implementada las siguientes tablas.

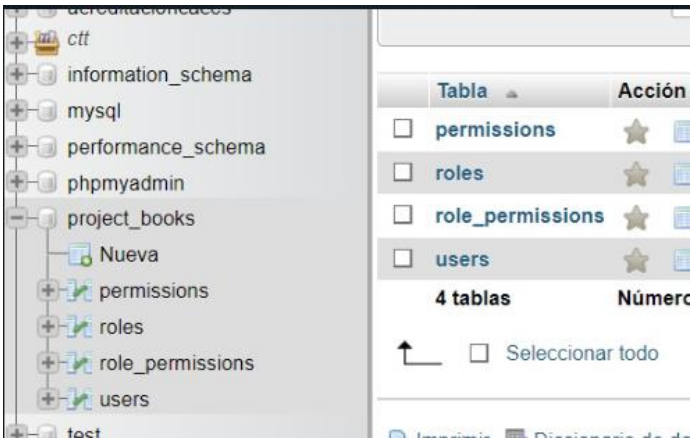

Gráfico 2. Base de datos relacional MySQL

- Rutas
  - Inicio de Sesión
  - Utiliza la ruta /api/login para iniciar sesión. Envía una solicitud POST con el nombre de usuario y la contraseña en el cuerpo de la solicitud.
  - POST /api/login
  - Cuerpo de la solicitud: { "username": "ejemplo", "password": "contraseña" }
  - POST /api/register
  - Cuerpo de la solicitud: { "username": "nuevoUsuario", "password": "contraseña" }

- Operaciones de usuarios

Tabla 2. Operaciones del microservicio de usuarios

Tipo de Operación	Ruta	Descripción
Operaciones de Usuario		
GET	/api/users/:id	Obtener los detalles de un usuario por su ID.
PUT	/api/users/:id	Actualizar los detalles de un usuario por su ID.
DELETE	/api/users/:id	Eliminar un usuario por su ID.
GET	/api/users	Obtener la lista de todos los usuarios.
GET	/api/userName/:username	Obtener los detalles de un usuario por su nombre de usuario.
PUT	/api/userRole/:id	Actualizar el rol de un usuario por su ID.
Operaciones de Rol		
POST	/api/roles	Crear un nuevo rol.
GET	/api/roles/:id	Obtener los detalles de un rol por su ID.

GET	/api/roles	Obtener la lista de todos los roles.
PUT	/api/roles/:id	Actualizar los detalles de un rol por su ID.
DELETE	/api/roles/:id	Eliminar un rol por su ID.
Operaciones de Permiso		
POST	/api/permissions	Crear un nuevo permiso.
GET	/api/permissions/:id	Obtener los detalles de un permiso por su ID.
GET	/api/permissions	Obtener la lista de todos los permisos.
PUT	/api/permissions/:id	Actualizar los detalles de un permiso por su ID.
DELETE	/api/permissions/:id	Eliminar un permiso por su ID.

- Microservicio de Libro
  - Requisitos para el desarrollo
    - Python 3.x instalado en tu sistema.
    - MySQL Server instalado y en funcionamiento
  - Base de datos
    La base de datos utilizada para el microservicio de libros es MySQL
  - Rutas

Tabla 3. Rutas del microservicio de libros

Tipo de Operación	Ruta	Descripción
Crear un Nuevo Libro		
POST	/libros/	Crea un nuevo libro.
Obtener Detalles de un Libro		
GET	/libros/{libro_id}	Obtiene los detalles de un libro específico.
Actualizar Detalles de un Libro		
PUT	/libros/{libro_id}	Actualiza los detalles de un libro específico.
Actualizar Estado de un Libro		
PUT	/libros/{libro_id}/estado	Actualiza el estado de un libro específico.
Eliminar un Libro		
DELETE	/libros/{libro_id}	Elimina un libro específico.
Crear un Nuevo Préstamo		
POST	/prestamos/	Crea un nuevo préstamo.
Obtener Detalles de un Préstamo		

GET	/prestamos/{prestamo_id}	Obtiene los detalles de un préstamo específico.
Eliminar un Préstamo		
DELETE	/prestamos/{prestamo_id}	Elimina un préstamo específico.

- Pruebas con SonarQube

SonarQube es una plataforma de código abierto utilizada para evaluar y analizar la calidad del código fuente. Proporciona un conjunto de herramientas poderosas para identificar automáticamente problemas de código, vulnerabilidades de seguridad, errores de codificación, malas prácticas y duplicaciones de código en proyectos de software. Además de realizar análisis estático de código, SonarQube ofrece métricas de calidad del código, como la cobertura de pruebas, la complejidad ciclomática y la cantidad de código duplicado.

- Desarrollo de prueba en los microservicios

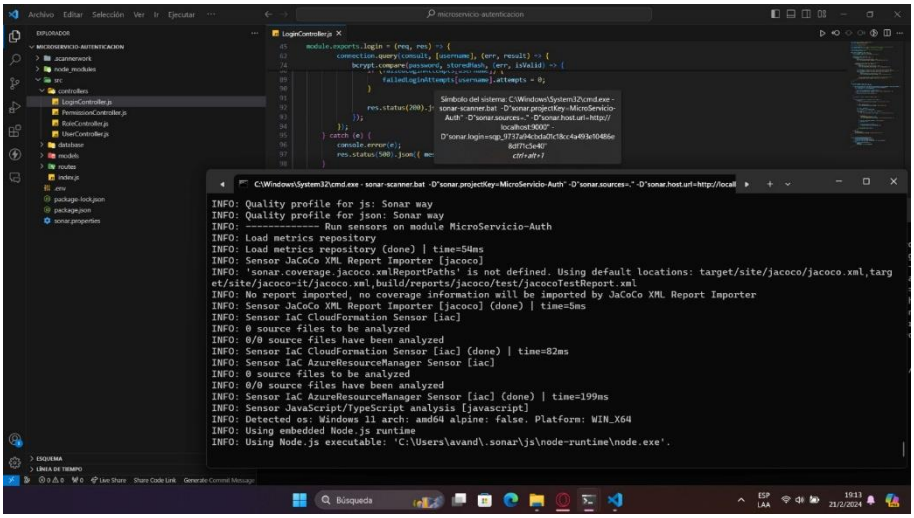

Gráfico 3. Ejecución de SonarQube al código de microservicios

- Análisis de resultados

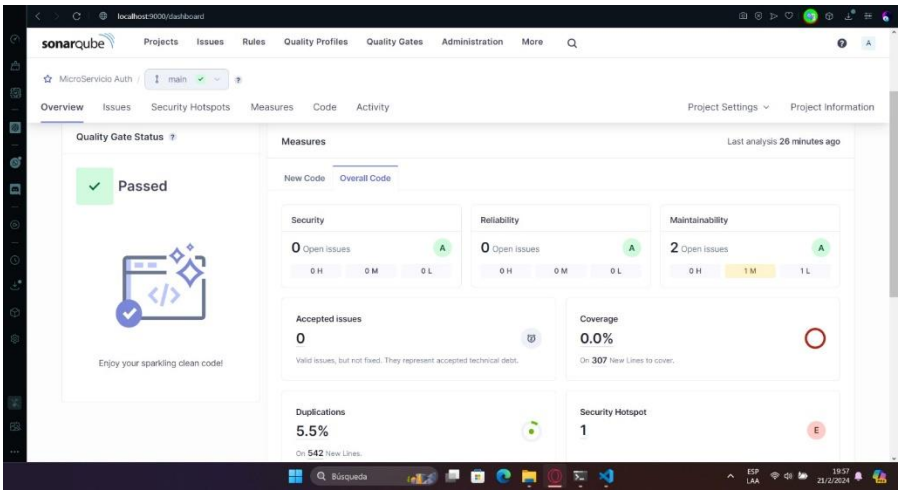

Gráfico 4. Resultados de métricas de SonarQube

Se puede observar el reporte que nos ofrece SonarQube una vez evaluado el código en la cual se obtuvo los siguientes resultados.

Tabla 4. Resultados al aplicar SonarQube

Métrica	Calificación	Descripción
Seguridad	A	No hay problemas abiertos relacionados con la seguridad del código.
Confiabilidad	A	No hay problemas abiertos relacionados con la confiabilidad del código.
Mantenibilidad	A	Hay 2 problemas abiertos relacionados con la mantenibilidad del código.
Cobertura de Código	0.0%	La cobertura del código es del 0.0%, indicando que no hay pruebas o que las pruebas no cubren líneas de código.
Duplicaciones	5.5%	El código tiene un 5.5% de duplicaciones, lo que sugiere la necesidad de refactorizar para eliminar código duplicado.

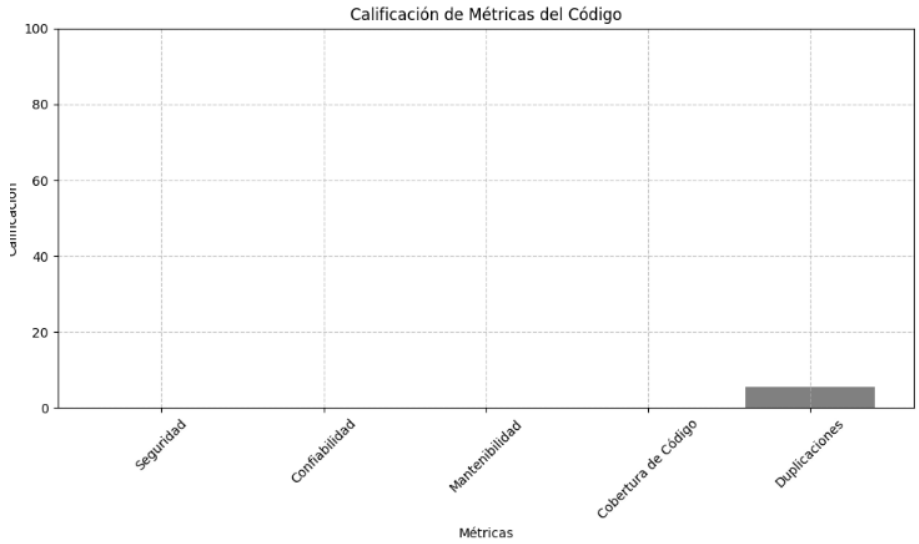

Gráfico 5. Resultados de métricas de SonarQube

4. Conclusiones

La evaluación realizada mediante SonarQube proporciona una visión clara de la calidad del código en los microservicios desarrollados para el sistema de prestación de libros de la Universidad de las Fuerzas Armadas ESPE.

En términos de seguridad y confiabilidad, los resultados son muy positivos, con una calificación de "A" en ambas métricas. Esto indica que no se encontraron problemas abiertos relacionados con la seguridad ni con la confiabilidad del código, lo que sugiere que se han seguido buenas prácticas en el desarrollo para mitigar posibles vulnerabilidades y asegurar la estabilidad del sistema.

Sin embargo, en cuanto a mantenibilidad, se identificaron 2 problemas abiertos. Esto indica que existen áreas en el código que podrían necesitar mejoras en términos de claridad, legibilidad o modularidad para facilitar su mantenimiento a largo plazo. Es crucial abordar estos problemas para garantizar que el sistema pueda evolucionar y adaptarse a futuros cambios de manera eficiente.

La cobertura del código es del 0.0%, lo que sugiere que no se han implementado pruebas automatizadas o que las pruebas existentes no cubren adecuadamente todas las líneas de código. Esto representa un área de mejora significativa, ya que las pruebas automatizadas son esenciales para garantizar la calidad del software y la detección temprana de posibles problemas.

Además, se encontró que el código tiene un 5.5% de duplicaciones, lo que indica la presencia de fragmentos de código repetidos que podrían ser refactorizados para mejorar la mantenibilidad y la legibilidad del código.

En conclusión, si bien el sistema muestra fortalezas en términos de seguridad y confiabilidad, existen áreas de mejora en cuanto a mantenibilidad, cobertura de

código y eliminación de duplicaciones. Es fundamental abordar estos aspectos para garantizar un sistema robusto, escalable y fácil de mantener a largo plazo.

5. Referencia

Diaz Chaparro, L. R. (2014). Desarrollo de software seguro (Bachelor's thesis, Universidad Piloto de Colombia).

Haro, E., Guarda, T., Peñaherrera, A. O. Z., & Quiña, G. N. (2019). Desarrollo backend para aplicaciones web, servicios web restful: Node. js vs spring boot. Revista Ibérica de Sistemas e Tecnologias de Informação, (E17), 309-321

López-Rodríguez, S. A., & García-Peña, V. R. (2021). Metodologías de desarrollo de software seguro con propiedades agiles. Polo del Conocimiento, 5(10), 1027-1046.

Lenarduzzi, V., Lomio, F., Huttunen, H., & Taibi, D. (2020, February). Are sonarqube rules inducing bugs?. In 2020 IEEE 27th international conference on software analysis, evolution and reengineering (SANER) (pp. 501-511). IEEE.

Ferenc, R., Langó, L., Siket, I., Gyimóthy, T., & Bakota, T. (2014, September). Source meter sonar qube plug-in. In 2014 IEEE 14th International Working Conference on Source Code Analysis and Manipulation (pp. 77-82). IEEE.