



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas

**Mejoramiento de la Aplicación
“SISTEMA DE GESTIÓN HOTELERA”**

Curso: Calidad y Pruebas de Software

Docente: Ing. Patrick Cuadros Quiroga

Estudiante:

- **Chávez Linares César Fabián** (2019063854)
- **Nina Vargas, Luigui Augusto** (2019065166)
- **Chambe Torres Edgard Reynaldo** (2019064917)
- **Condori Vargas, Tomas Yoel** (2018000847)
- **Casilla Maquera, Tell Ivan** (2017057888)
- **Alfárez Ponce, Pedro Alberto** (2020066317)

Tacna - Perú

2023

Resumen

Este proyecto se enfoca en el desarrollo de un sistema de gestión hotelera, utilizando el patrón de diseño Modelo-Vista-Controlador (MVC) en lenguaje JavaScript. En este

sistema, un administrador tendrá la capacidad de gestionar las habitaciones y el proceso de reserva a través de un panel de administración web. Este enfoque proporcionará un control eficiente y seguro sobre los recursos y la funcionalidad del sistema, además contribuirá a mejorar la experiencia tanto de los usuarios que empleen el sistema, como del personal encargado y garantice una gestión efectiva en el entorno hotelero e incluirá pruebas de calidad utilizando SonarQube, estas pruebas de test tienen como objetivo garantizar que el código fuente del sistema cumpla con los estándares de calidad y sea eficiente en términos de rendimiento, garantizando que el sistema sea robusto, seguro y de alta calidad antes de su implementación en su entorno de trabajo.

Abstract

This project focuses on the development of a hotel management system using the Model-View-Controller (MVC) design pattern in JavaScript. In this system, an administrator will have the ability to manage rooms and the reservation process through a web-based administration panel. This approach will provide efficient and secure control over system resources and functionality, contributing to improving the experience of both system users and staff while ensuring effective management in the hotel environment. Additionally, it will include quality testing using SonarQube. These tests aim to ensure that the system's source code complies with quality standards and is efficient in terms of performance, guaranteeing that the system is robust, secure, and of high quality before its implementation in the work environment.

1. Introducción

Esta aplicación web tiene como objetivo principal agilizar el proceso de reserva en hoteles, y para lograrlo, se persiguen objetivos secundarios como optimizar los tiempos de reserva, mejorar la facilidad de uso y hacer que el sistema sea intuitivo. En este proceso de mejora, se utilizó SonarQube y Snyk para garantizar la calidad y seguridad del proyecto, lo que contribuye a una experiencia de usuario más eficiente y competitiva para la industria hotelera en general.

2. Título

“Sistema de Gestión Hotelera: Hospedaje Virgen de las Peñas”

3. Autores

- Chávez Linares César Fabián
- Nina Vargas, Luigui Augusto
- Chambe Torres Edgard Reynaldo
- Condori Vargas, Tomas Yoel
- Casilla Maquera, Tell Ivan

4. Planteamiento del problema

4.1. Problema

Cuando una persona desea hacer una reservación en un hotel, se espera que este proceso sea eficiente y sin problemas, el no cumplir con estas necesidades puede provocar que el decida por alojarse en otro lugar. Por lo tanto, para poder satisfacer esta necesidad es necesario que un hotel migre de un proceso de registro manual a un registro online donde el cliente pueda realizar una reserva desde la comodidad del lugar donde se encuentre.

La falta de un sistema para llevar a cabo el proceso de gestión hotelera que va a permitir controlar la información del inventario de las habitaciones, su disponibilidad, tarifas y reservas del hotel.

4.2. Justificación

La finalidad principal de esta aplicación web es reducir el tiempo empleado al realizar una reserva en un hotel de manera satisfactoria. Para conseguir esta meta aparecen como objetivos secundarios un aumento de la ocupación de habitaciones, tiempos de reserva de habitaciones más optimizados, mejoras en la facilidad de uso, y un manejo intuitivo. Otro objetivo es conseguir que el mayor número de usuarios utilice este sistema.

4.3. Alcance

El siguiente proyecto tiene como alcance el “Hospedaje virgen de las peñas”, que brindan el servicio de reservas de habitaciones de manera manual y tienen la necesidad de sistematizar este proceso a través de

un sistema para llevar a cabo el proceso de reserva, la información del inventario de las habitaciones y su disponibilidad, tarifas y reservas realizadas por el cliente.

5. Objetivos

5.1. General

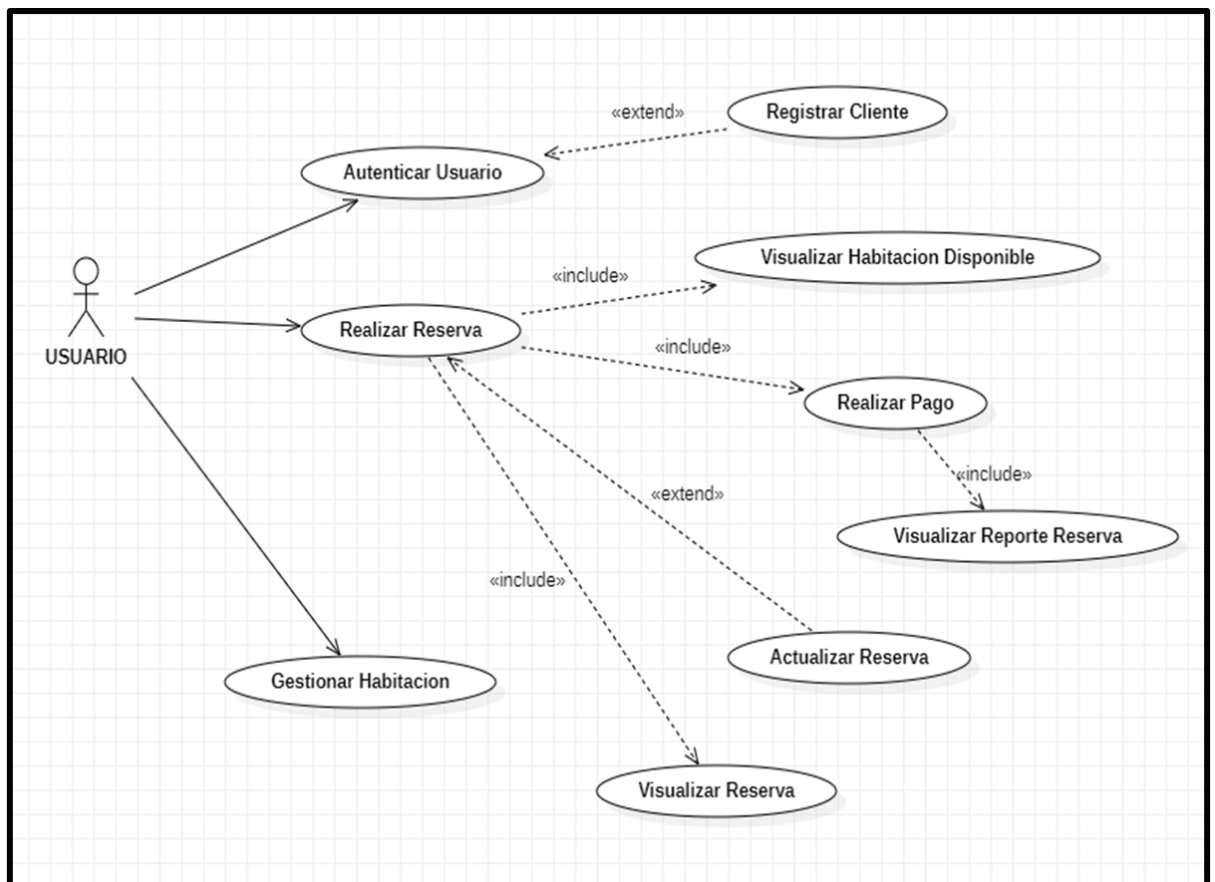
- ❖ Sistematizar el proceso de gestión hotelera y brindar al usuario(cliente,administrador),un diseño simple con el objetivo de facilitar y permitir al usuario el uso de las funcionalidades del sistema para procesar y actualizar la información del inventario de las habitaciones y su disponibilidad, tarifas y reservas del hotel

5.2. Específicos

- Permitir al usuario registrarse y realizar su reserva.
- Dar seguridad al usuario y administrador mediante un login para acceder al sistema.
- Permitir al usuario ver una interfaz para interactuar a través de opciones lo que necesite.
- Mostrar la información de las habitaciones disponibles,tarifas y métodos de pago a los clientes para que puedan realizar su reservación.
- El podrá visualizar la reservación realizada y ver el pago a realizar.
- El administrador podrá gestionar habitaciones y ver el estado de las reservas.

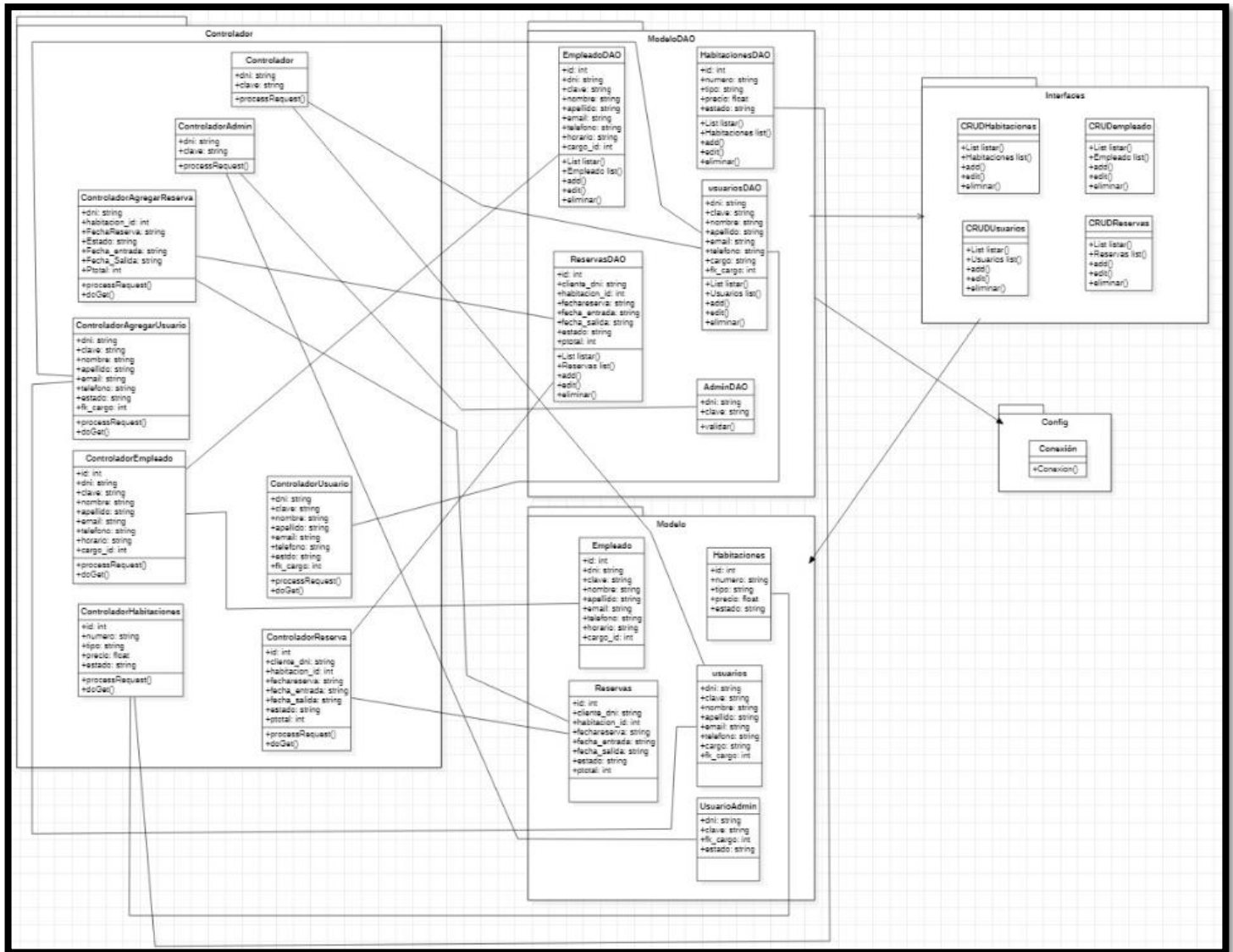
6. Referentes teóricos

6.1 Diagramas de Casos de Uso

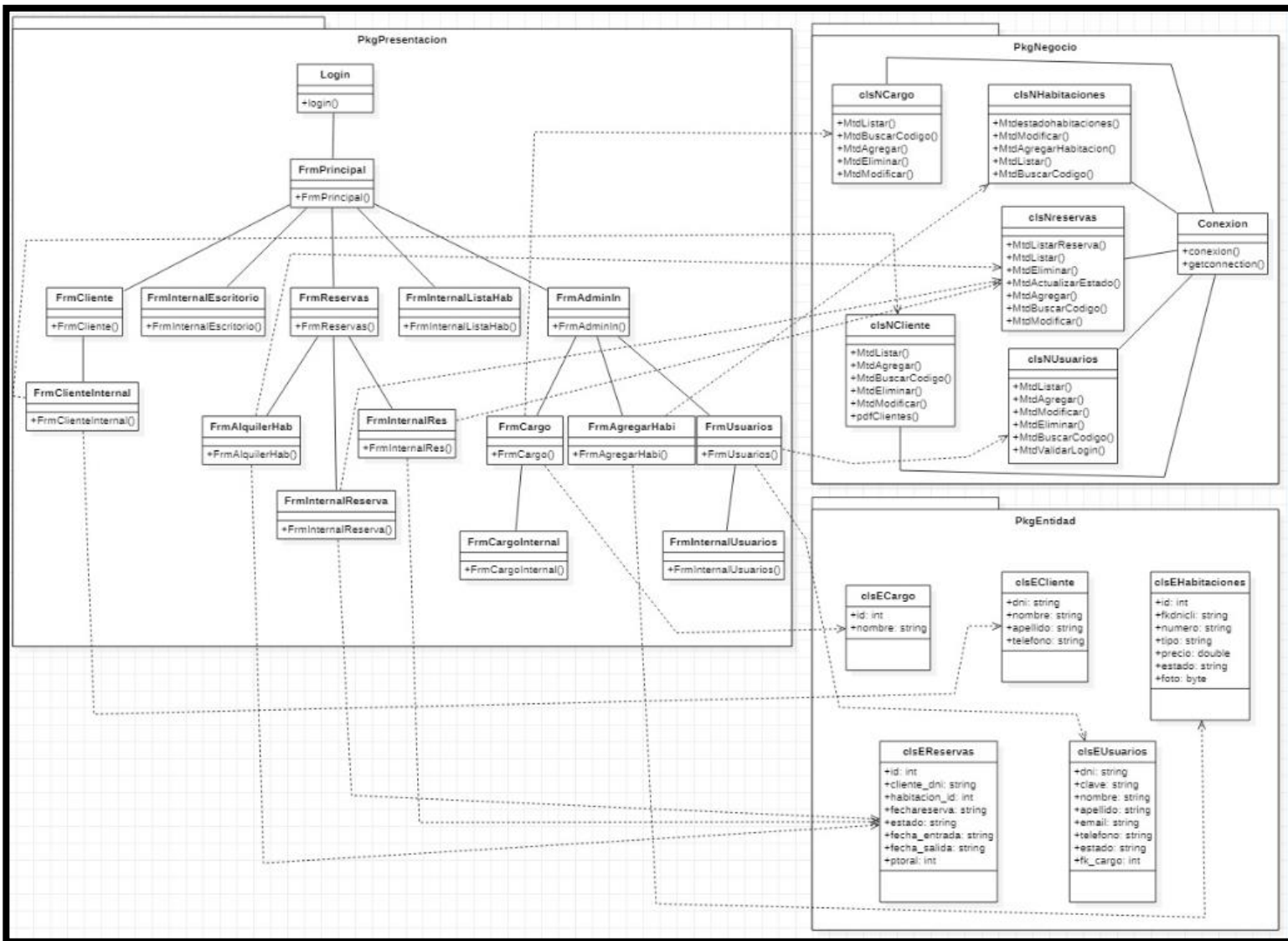


6.2 Diagrama de Clases

Sistema web



Sistema Escritorio



6.3 Diagrama de Componentes y Arquitectura.

DIAGRAMA DE COMPONENTES ESCRITORIO

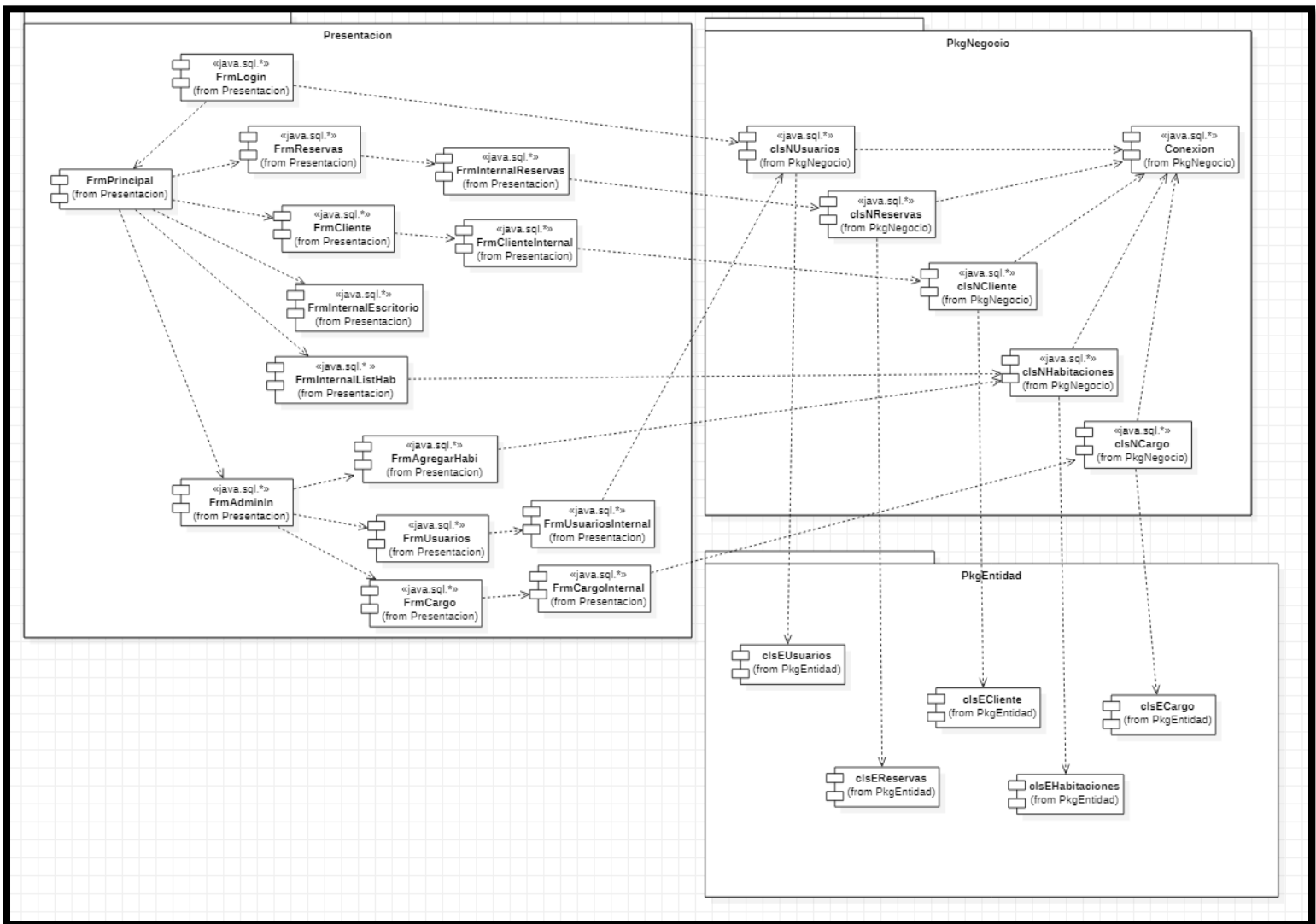


DIAGRAMA DE COMPONENTES SISTEMA WEB

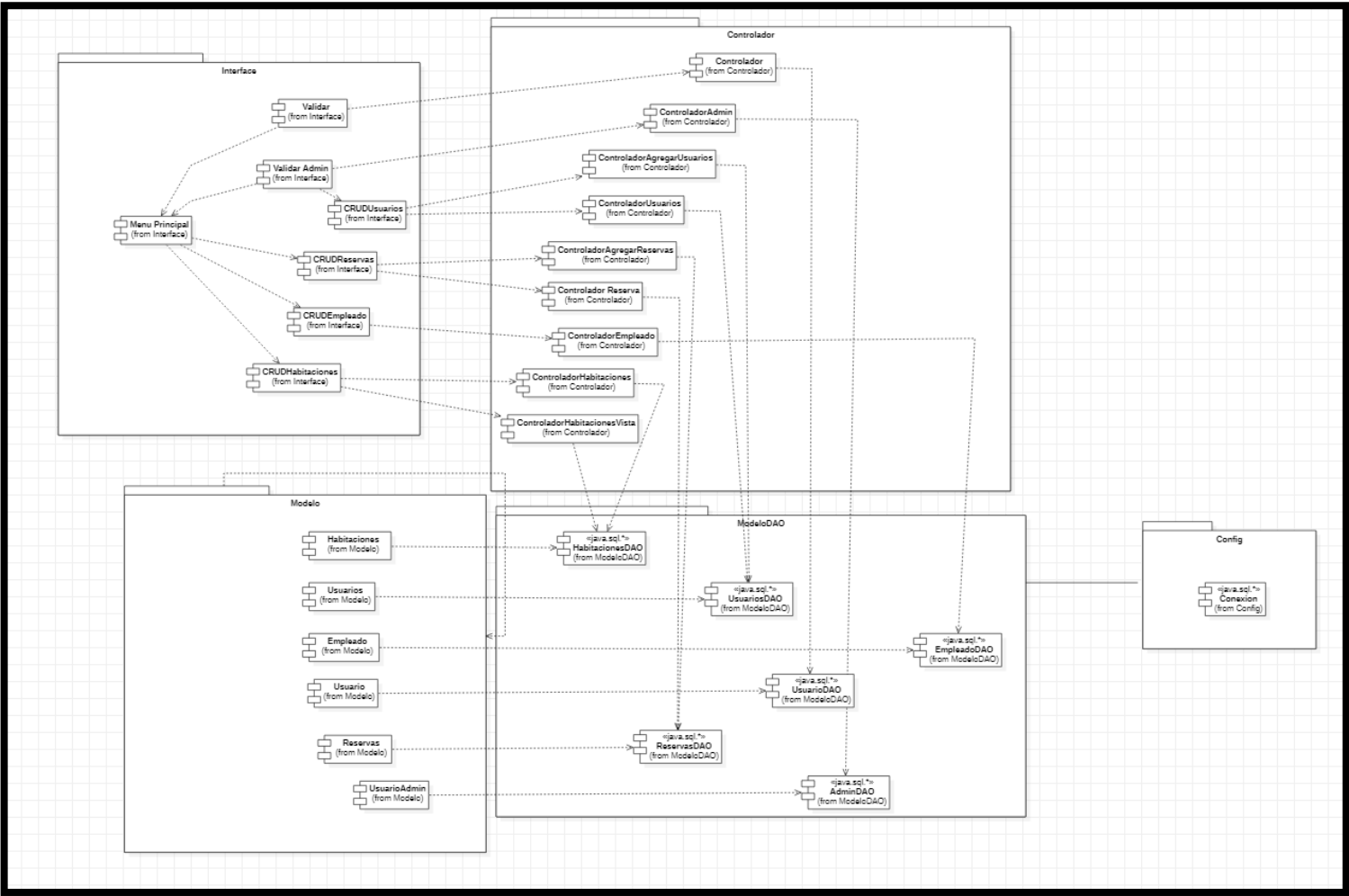
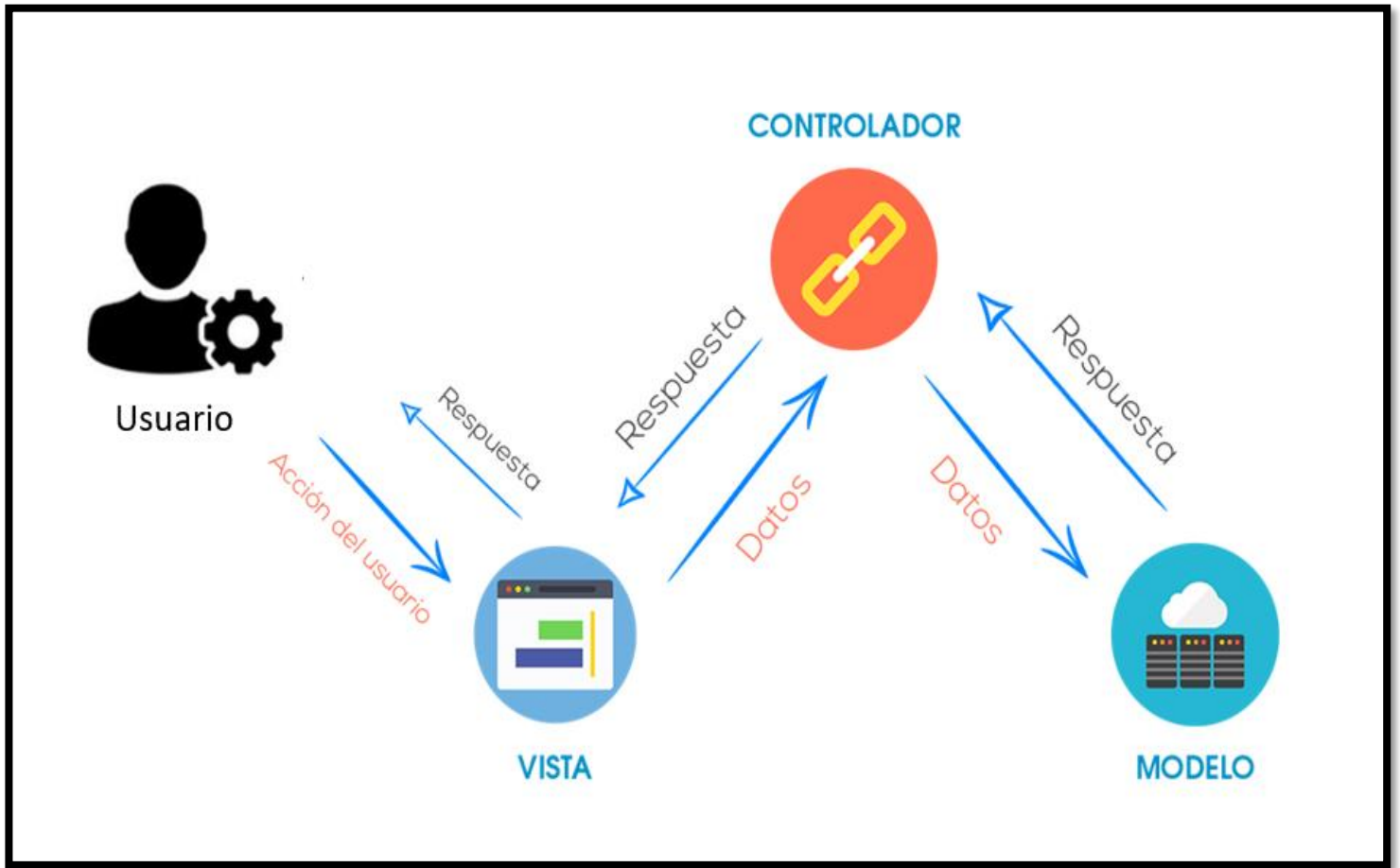


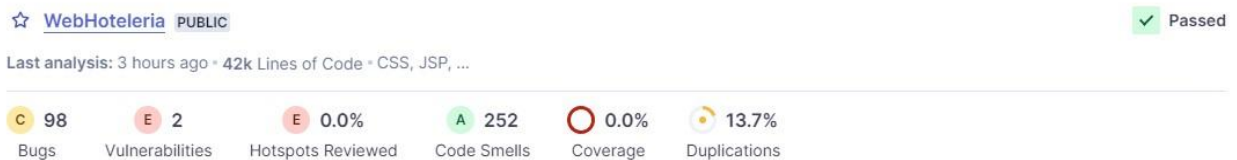
DIAGRAMA DE ARQUITECTURA



7. Desarrollo de la propuesta (Aquí va el análisis de su aplicación con SonarQube y Snyk, para que les muestre todos los aspectos a mejorar de su aplicación)

Análisis de SonarQube

Proyecto: WEB SISTEMA HOTELERO RESERVA



Reliability: se mide mediante análisis estáticos de código fuente para identificar problemas relacionados con la robustez y la estabilidad del software. Esto incluye la detección de defectos de programación, vulnerabilidades de seguridad y otros problemas que pueden afectar negativamente la fiabilidad del software.



Security: se evalúa mediante análisis estáticos de código para identificar posibles vulnerabilidades y debilidades de seguridad en una aplicación

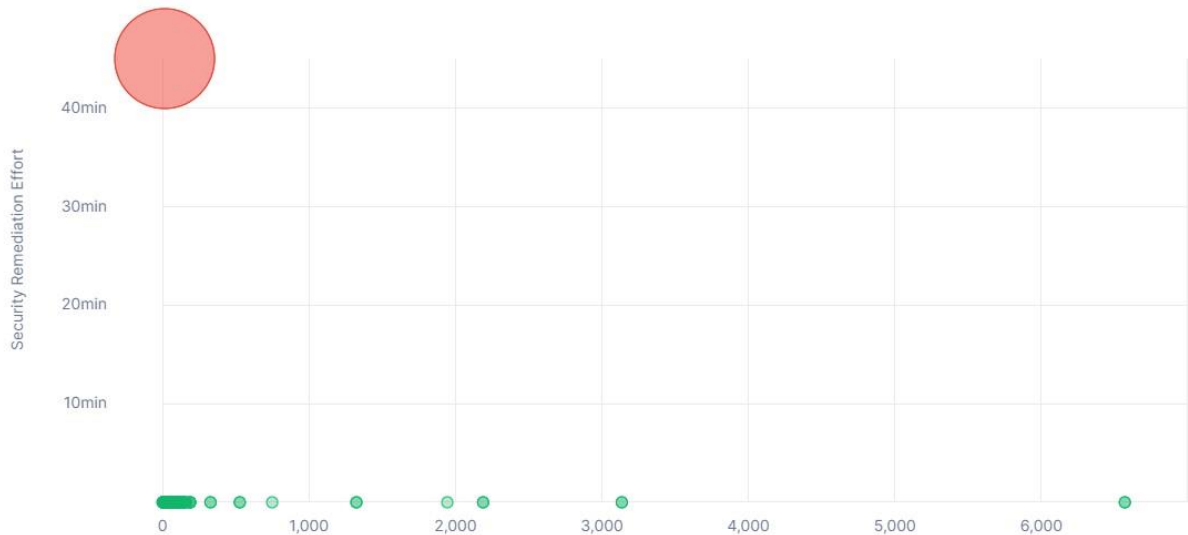
Security Overview ?

[See the data presented on this chart as a list](#)

Color: Security Rating Size: Vulnerabilities

✓ A ✓ B ✓ C ✓ D ✓ E

Zoom: 100%



Maintainability: Esta métrica evalúa la calidad del código fuente y su facilidad de mantenimiento. Una alta puntuación de mantenibilidad en SonarQube indica que el código es más fácil de comprender, modificar y extender, lo que a su vez reduce los costos y riesgos asociados con el mantenimiento del software.

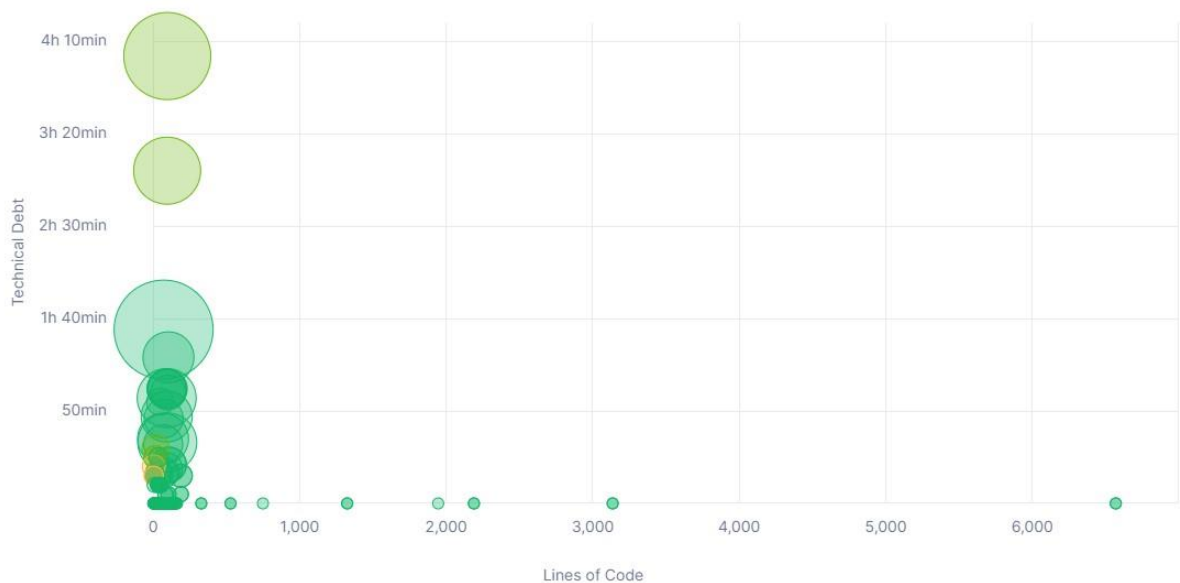
Maintainability Overview ?

[See the data presented on this chart as a list](#)

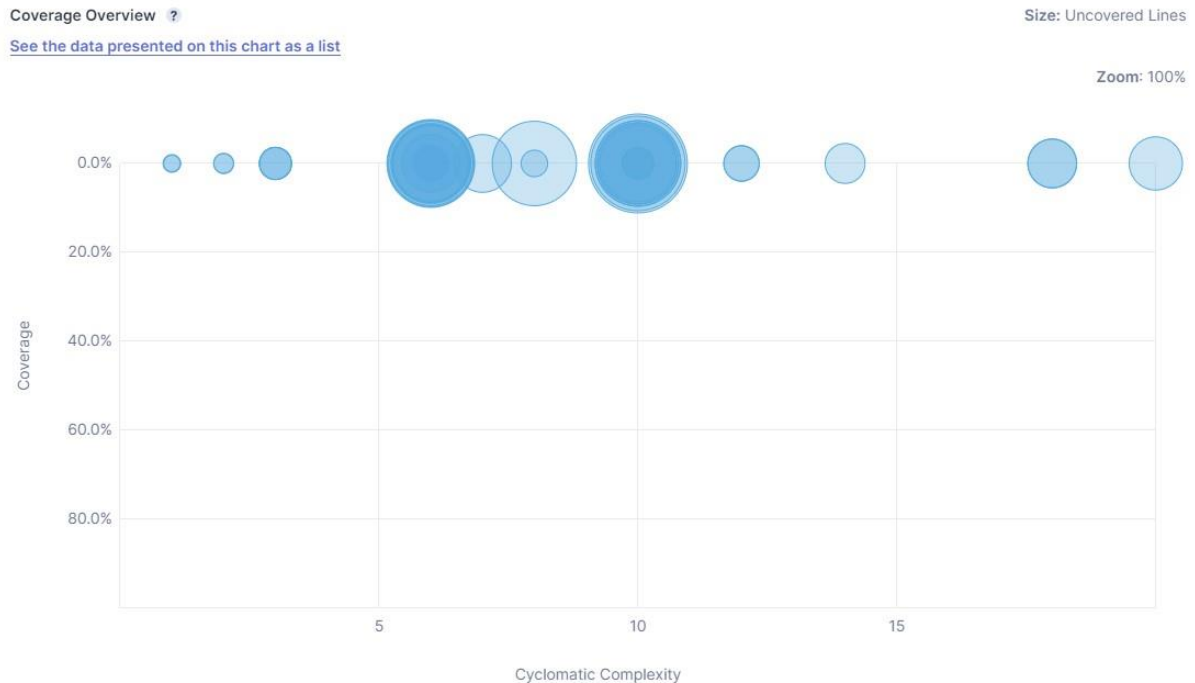
Color: Maintainability Rating Size: Code Smells

✓ A ✓ B ✓ C ✓ D ✓ E

Zoom: 100%



Coverage: La cobertura de código es una métrica importante para evaluar la calidad del código y la efectividad de las pruebas, ya que indica qué parte del código ha sido ejecutada y probada por las pruebas automatizadas en comparación con la totalidad del código fuente.



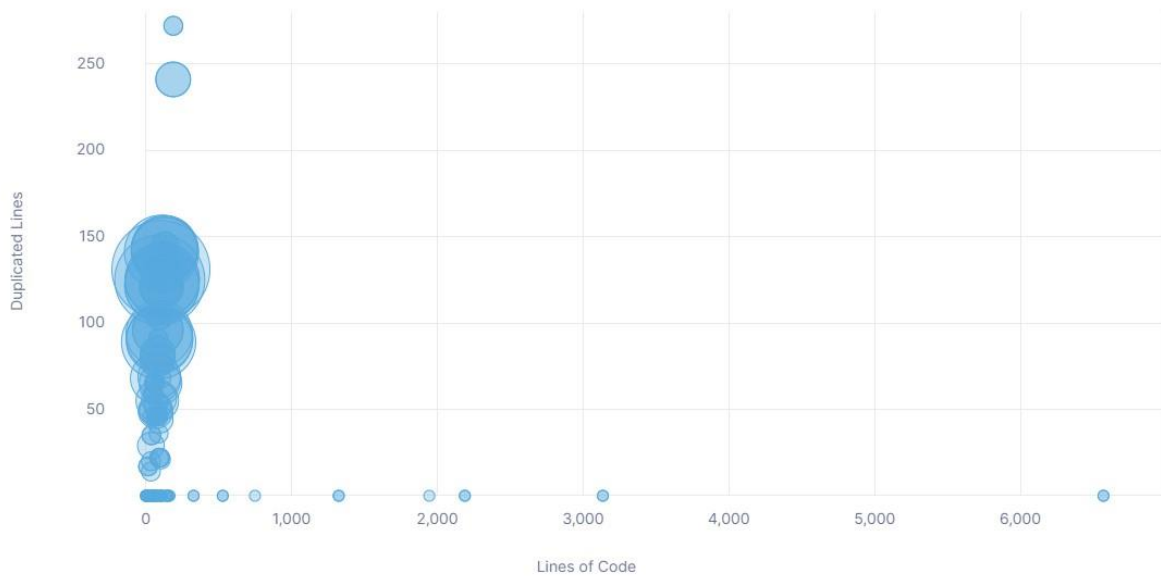
Duplicates: Los duplicados son fragmentos de código que se repiten en múltiples lugares dentro del código fuente. En general, tener un alto número de duplicados en el código no es una buena práctica de programación, ya que puede dificultar la mantenibilidad, aumentar la probabilidad de errores y hacer que el código sea más difícil de entender y mantener.

Duplications Overview ?

Size: Duplicated Blocks

[See the data presented on this chart as a list](#)

Zoom: 100%



Análisis en Synk

Proyecto: WEB SISTEMA HOTELERO RESERVA

SEVERITY

☐ Critical

☐ High

☐ Medium

☐ Low

0020

PRIORITY SCORE

Scored between 0 - 1000

FIXABILITY

☐ Fixable

☐ Partially fixable

☐ No fix available

200

EXPLOIT MATURITY

☐ Mature

☐ Proof of concept

☐ No known exploit

☐ No data

0020

STATUS

☒ Open

☐ Patched

☐ Ignored

200

2 of 2 issues

Sort by highest priority score

M

org.springframework:spring-core - Improper Input Validation

SCORE 429

VULNERABILITY

CWE-20

CVE-2021-22060

CVSS 4.3

MEDIUM

SNYK-JAVA-ORGSRINGFRAMEWORK-2330878

Introduced through

org.springframework:spring-core@5.3.10

Fixed in

org.springframework:spring-core@5.2.19.RELEASE, @5.3.14

Exploit maturity

NO KNOWN EXPLOIT

Show more detail

Learn about this type of vulnerability

Ignore

M

org.springframework:spring-core - Improper Output Neutralization for Logs

SCORE 429

VULNERABILITY

CWE-20

CVE-2021-22096

CVSS 4.3

MEDIUM

SNYK-JAVA-ORGSRINGFRAMEWORK-2329097

Introduced through

org.springframework:spring-core@5.3.10

Fixed in

org.springframework:spring-core@5.3.12, @5.2.18

Exploit maturity

NO KNOWN EXPLOIT

Show more detail

Learn about this type of vulnerability

Ignore

Vulnerabilidades directas

Vulnerabilidades conocidas en el paquete `org.springframework:spring-core`. Esto no incluye vulnerabilidades que pertenecen a las dependencias de este paquete.

VULNERABILITY	VULNERABLE VERSION
M Improper Input Validation	[5.2.19.RELEASE) [5.3.0,5.3.14)
M Improper Output Neutralization for Logs	[5.3.0,5.3.12) [5.2.18)
H Access Restriction Bypass	[5.0.5.RELEASE,5.0.6.RELEASE)
M Multipart Content Pollution	[4.3.0.RELEASE,4.3.14.RELEASE) [5.0.0.RELEASE,5.0.5.RELEASE)
L Denial of Service (DoS)	[3.2.8.RELEASE,3.2.14.RELEASE) [4.1.0.RELEASE,4.1.7.RELEASE)
M Access Restriction Bypass	[3.0.0.RELEASE,3.0.6.RELEASE)
M Directory Traversal	[3.0.0.RELEASE,3.2.9.RELEASE) [4.0.0.RELEASE,4.0.5.RELEASE)
H Expression Language Injection	[2.0.0,2.5.6.SEC03) [3.0.0,3.0.6)
M Arbitrary Code Execution	[2.5.0,2.5.6.SEC01) [3.0.0,3.0.2.RELEASE)
M Regular Expression Denial of Service (ReDoS)	[2.5.6.SEC01)

7.1. Tecnología de información

Para la Aplicación de Escritorio (Java Desktop):

- JDBC (Java Database Connectivity): Para conectar tu aplicación de escritorio a la base de datos MySQL, puedes utilizar JDBC, que es una API estándar de Java para acceder a bases de datos relacionales.
- IDE de Desarrollo: Utiliza un entorno de desarrollo integrado (IDE) como Eclipse, NetBeans para desarrollar y depurar tu aplicación de escritorio en Java.

- Embalaje y Distribución: Considera herramientas como Java Web Start o aplicaciones empaquetadas para distribuir y ejecutar la aplicación en diferentes sistemas operativos.

Para la Aplicación Web (Java Web):

- Servlets y JSP (JavaServer Pages): Estas tecnologías Java son esenciales para manejar solicitudes web y generar páginas dinámicas en el servidor.
- Apache Tomcat: Necesitarás un servidor web o de aplicaciones para desplegar tu aplicación web Java. Apache Tomcat es una opción común y ligera.
- ORM (Object-Relational Mapping): Si deseas simplificar la interacción con la base de datos MySQL, puedes considerar el uso de un ORM.
- HTML, CSS y JavaScript: Además de Java, necesitarás conocimientos en tecnologías web estándar para crear la interfaz de usuario web. HTML y CSS son esenciales, y JavaScript puede ser útil para la interactividad del lado del cliente.
- Pruebas y Depuración: Utiliza herramientas y marcos de pruebas para

garantizar la calidad de tu aplicación web. **7.2. Metodología, técnicas**

usadas

Metodología:

- Metodología Ágil: Puedes optar por una metodología ágil, como Scrum o Kanban, que fomenta la flexibilidad y la colaboración en equipos de desarrollo. Estas metodologías se adaptan bien a proyectos de desarrollo de software, ya que permiten realizar iteraciones y ajustes continuos según las necesidades del cliente.
- Ciclo de Desarrollo: Divide tu proyecto en ciclos o sprints más pequeños, con entregas incrementales de funcionalidad. Esto te permite obtener retroalimentación temprana y realizar mejoras constantes.
- Control de versiones: Emplea un sistema de control de versiones, como Git, para gestionar el código fuente de tu aplicación y facilitar la colaboración en equipo.

Técnicas y Prácticas:

- Desarrollo Guiado por Pruebas (TDD): Escribe pruebas automatizadas antes de desarrollar el código para garantizar que las nuevas funcionalidades sean funcionales y cumplan con los requisitos.
- Revisión de Código: Realiza revisiones de código entre los miembros del equipo para garantizar la calidad del código y el cumplimiento de las mejores prácticas.
- Refactorización: Mantén un proceso constante de refactorización del código para mejorar su legibilidad y mantenibilidad.
- Diseño de Base de Datos: Utiliza técnicas de diseño de bases de datos normalizadas para asegurar la integridad de los datos y un buen rendimiento.
- Seguridad: Implementa prácticas de seguridad, como la validación de entrada de datos y la gestión adecuada de contraseñas, para proteger tu aplicación contra vulnerabilidades.
- Documentación: Mantén una documentación adecuada del código fuente, la arquitectura de la aplicación y los procesos de desarrollo.
- Monitorización: Implementa herramientas de monitorización para supervisar el rendimiento de la aplicación en tiempo real y detectar problemas de forma proactiva.

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.

Para reducir la deuda técnica, abordar vulnerabilidades y minimizar fallos en un proyecto de software con SonarQube, se inició con un análisis que identifica problemas, seguido de una priorización y un plan de acción detallado. Se establecen estándares de codificación y se automatiza la revisión de código con integración continua. Se capacitará al equipo, se realizan revisiones de código regulares y se aplican pruebas de seguridad. El seguimiento constante y la mejora continua son esenciales para mantener la calidad del software a lo largo del tiempo.

Asignado a:	Fecha de inicio	Fecha final	Características	Rating
-------------	-----------------	-------------	-----------------	--------

Cesar Chavez	26/09/23	30/09/23	Reliability	C
Luigui Nina	26/09/23	30/09/23	Security	E
Edgar Chambe	26/09/23	30/09/23	Maintainability	A
Tomar Condori	26/09/23	30/09/23	Security Review	E
Tell Casillas	26/09/23	30/09/23	Duplications	0.0% coverage
Pedro Alf��rez	26/09/23	30/09/23	Coverage	13.7%