



Uni Roomie: Una aplicación para la búsqueda y gestión de alojamiento estudiantil

Desarrollo Basado en Plataformas (CS2031)

Laboratorio 1.02

INTEGRANTES

| Nombres y Apellidos | Código del Alumno | Participacion |
|-------------------------------------|--------------------------|----------------------|
| Miguel Fernando Ucañani Tintaya | 202310383 | 45% |
| Gabriel Nicolás Frisancho Gálvez | 202310080 | 5% |
| Hector Sebastián Nieto Paz | 202310259 | 100% |
| Milton Joel Cordova Navarro | 202310314 | 70% |

Link del repositorio GitHub:

<https://github.com/CS2031-DBP/proyecto-backend-grupo-4>

DOCENTE: Mateo Noel Rabines

2024-01

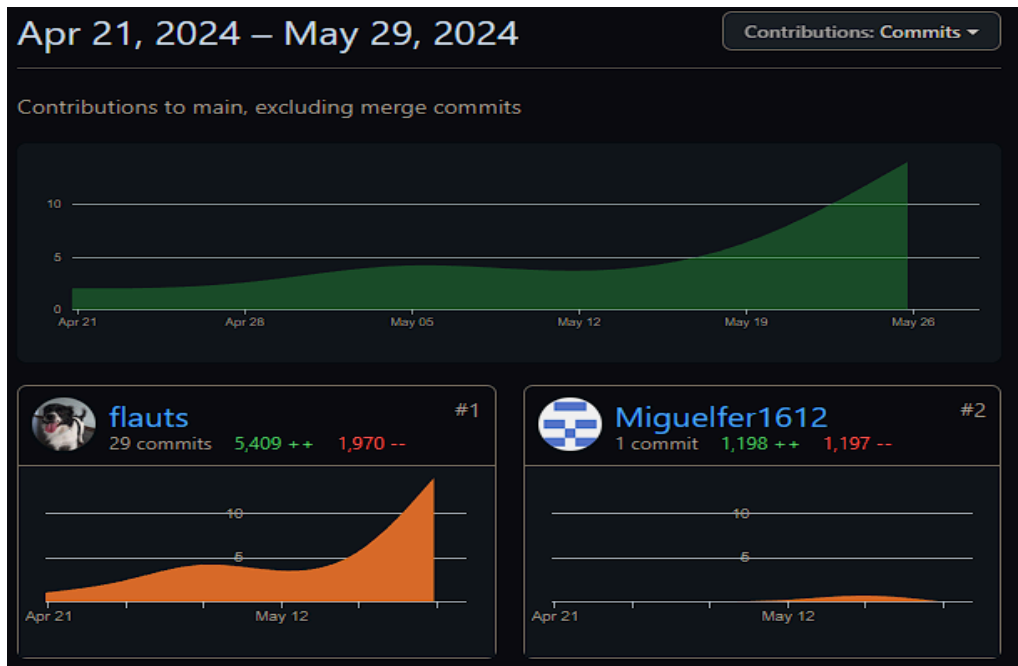
29/05/2024

Observaciones:

- Hubo muchas complicaciones a la hora de trabajar el proyecto. Al inicio del proyecto, 75% de los integrantes no tenían los conocimientos básicos para trabajar en Github, y menos los enseñados en clase. Por lo que solo el 25% de los integrantes tuvo que realizar la mayor parte del proyecto sólo, durante casi todo el tiempo dado para el proyecto.
- **Milton Joel Cordova Navarro** fue el integrante que más ayudó en términos de código del proyecto, aunque fuera en los últimos días de la entrega del proyecto. Se encargó de los eventos sincrónicos con respecto al envío de correos, e incluso realizó una plantilla HTML personalizada para este último. También implementó algo de testing, aunque este último no pudo ser terminado por ninguno de los dos.
- **Miguel Fernando Ucañani Tintayay** **Gabriel Eduardo Romero Diez** principalmente se encargaron de la redacción del proyecto de solo los conceptos generales. No pudieron hacer mucho más ya que no tenían conocimiento completo del código.
- **Gabriel Eduardo Romero Diez** dio la idea para la realización de este proyecto.
- Aquí se adjuntan las contribuciones del proyecto en Github.
- **Milton Joel Cordova Navarro** tiene pocos commits ya que la mayoría de veces iba a su rama para ver los cambios que realizaba y los agregaba a mis commits para evitar complicaciones. **Milton Joel Cordova Navarro** fue esencial en los días posteriores para la finalización del proyecto. El commit perteneciente a **Miguel Fernando Ucañani Tintaya**, que se encuentra en la imagen, consiste en renombrar las carpetas del proyecto, agregar el `AbstractContainerTest`, desarrollo de algunos de los test y el Controlador de Anfitrión, aunque este último fue reescrito después.

- Debido a la falta de integrantes activos, este informe puede que no sea la mejor representación del trabajo hecho en este equipo. Por favor ver el código realizado.

“



Resumen Ejecutivo

- **Propósito del Proyecto:**

El propósito del proyecto UniRoomie es facilitar a los estudiantes universitarios de Utec la búsqueda y gestión de alojamiento estudiantil, proporcionando una plataforma intuitiva y segura que les permita encontrar compañeros de cuarto, alquilar departamentos cercanos a su universidad, y simplificar el proceso de reserva y comunicación entre anfitriones y futuros inquilinos.

- **Solución Propuesta:**

La solución implementada, UniRoomie, es una plataforma digital diseñada para facilitar a los estudiantes universitarios la búsqueda y gestión de alojamiento estudiantil. Ofrece una variedad de características, incluyendo perfiles de usuarios, funcionalidades de búsqueda, sistema de comunicación integrado, gestión de reservas y solicitudes, seguridad y confianza mediante sistemas de puntajes y reseñas, y una interfaz intuitiva para una experiencia de usuario fluida. UniRoomie se centra en proporcionar una solución integral que simplifica y agiliza el proceso de encontrar alojamiento adecuado para estudiantes de Utec.

- **Beneficios Clave:**

UniRoomie ofrece una solución integral que simplifica y agiliza el proceso de búsqueda y gestión de alojamiento estudiantil, brindando una serie de beneficios clave a sus usuarios:

Perfiles de Usuarios:

Los estudiantes que buscan alojamiento pueden crear perfiles detallados donde podrán describir sus hobbies e intereses.

Los estudiantes que ofrecen alojamiento y los propietarios de edificios pueden publicar perfiles con información relevante sobre sus alojamientos disponibles.

Funcionalidades de Búsqueda:

Los inquilinos pueden realizar búsquedas avanzadas de alojamiento por el número de habitación, baño, y precio, facilitando la búsqueda de opciones que se ajusten a sus necesidades.

La clasificación de alojamientos por precios permite a los estudiantes encontrar opciones que se adapten a sus preferencias y presupuestos.

Gestión de Departamentos:

UniRoomie proporciona información acerca de los departamentos que está rentando, como lo es el piso donde se encuentra y reseñas hechas con puntajes y reservas, lo que facilita la presentación completa y precisa de la misma.

Las descripciones detalladas de los alojamientos, junto con fotos y características, ayudan a los usuarios a tomar decisiones informadas sobre dónde vivir.

Reservas y Gestión de Solicitudes:

UniRoomie ofrece un sistema de reserva donde los inquilinos pueden enviar solicitudes de reserva a los anfitriones, simplificando el proceso de asegurar alojamiento.

Los anfitriones pueden aceptar o rechazar solicitudes según el límite de capacidad del alojamiento, lo que garantiza una gestión eficiente de las reservas.

Seguridad y Confianza:

La plataforma cuenta con un sistema de puntajes y reseñas que permite a los usuarios evaluar y compartir sus experiencias, promoviendo la transparencia y la confianza en la comunidad.

Se realiza una verificación de la identidad de los usuarios, especialmente para estudiantes de la UTEC, para garantizar la seguridad de todos los involucrados.

Facilidad de Uso:

UniRoomie ofrece una interfaz intuitiva y amigable para los usuarios, lo que facilita la navegación y el uso de la plataforma.

La navegación sencilla entre perfiles, alojamientos y mensajes garantiza una experiencia de usuario fluida y sin complicaciones.

Introducción

- **Contexto:**

Muchas veces vemos que la cantidad de estudiantes varían con tendencia a aumentar, son muchos los que necesitan una estadía mientras están en su etapa universitaria sin embargo las oportunidades y las opciones son insuficientes por el hecho de que no están en sincronía con su entorno (Barranco). Se ven afectados psicológicamente o se sienten abrumados por no encontrar una necesidad básica por eso con esta aplicación se les facilitará la búsqueda de nuevas viviendas desocupadas o fechas próximas a encontrar la mejor estadía.

- **Objetivos del Proyecto:**

Mejorar la Calidad de Vida de los Estudiantes:

Reducir los casos de discomfort entre los estudiantes al facilitarles la búsqueda de alojamiento adecuado, permitiendo que puedan concentrarse en sus estudios y rendir de la mejor manera posible.

Proximidad a la Universidad:

Brindar a los estudiantes la posibilidad de residir cerca de la universidad, minimizando los tiempos de desplazamiento y asegurando un entorno cómodo y conveniente para su vida estudiantil.

Facilitar la Comunicación y Coordinación:

Fomentar una convivencia armoniosa entre inquilinos y compañeros de cuarto al promover una comunicación abierta y efectiva, resolviendo rápidamente cualquier situación que se presente.

Proporcionar una Plataforma Segura y Confiable:

Ofrecer una plataforma intuitiva y segura que garantice la verificación de la identidad de los usuarios, promoviendo la confianza y transparencia dentro de la comunidad de UniRoomie.

Optimizar la Gestión de Reservas y Solicitudes:

Simplificar el proceso de reserva y gestión de solicitudes para inquilinos y anfitriones, garantizando una experiencia eficiente y libre de complicaciones.

Promover la Convivencia Estudiantil:

Crear un entorno que facilite la interacción y el conocimiento mutuo entre estudiantes, fortaleciendo la comunidad universitaria y promoviendo el apoyo y la colaboración entre sus miembros.

Impulsar la Disponibilidad de Alojamiento:

Ampliar las oportunidades y opciones de alojamiento para los estudiantes al sincronizar la oferta de viviendas disponibles con las necesidades y preferencias de los usuarios, mejorando así la accesibilidad a residencias adecuadas.

Garantizar la Seguridad y la Confianza:

Implementar sistemas de puntajes y reseñas que permitan evaluar y compartir experiencias, fomentando un entorno seguro y de confianza para todos los usuarios de la plataforma.

Identificación del Problema o Necesidad

- **Descripción del Problema:**

El problema central radica en la ubicación geográfica de los estudiantes con respecto a la universidad en la que estudian, esto puede verse agravado por ser nuevos en la ciudad. Como consecuencia, genera un ineficiente uso de su tiempo en cuanto al desplazamiento, lo que impacta negativamente en su rendimiento académico.

- **Justificación:**

- Al reducir el tiempo de movilización, los estudiantes pueden dedicar más tiempo a sus estudios y actividades académicas, mejorando así sus resultados.
- Contribuye al bienestar emocional y social, proveyendo un ambiente de integración y apoyo en el entorno educativo y fomentando la colaboración y el aprendizaje.
- Mejora la situación económica para cada involucrado gracias a que podrán compartir gastos como el alquiler y servicios públicos.

Descripción de la Solución

- **Funcionalidades Implementadas:**

- **Creacion de Usuarios UTEC:** Permite solo a estudiantes UTEC crear y gestionar sus cuentas.
- **Búsqueda Avanzada:** Funcionalidad de búsqueda avanzada por ubicación, precio y puntaje.
- **Sistema de Reservas:** Facilita el proceso de enviar y gestionar solicitudes de reserva.
- **Sistemas de Puntajes y Reseñas:** Permite evaluar y compartir experiencias de alojamiento.
- **Verificación de Identidad:** Garantiza la seguridad mediante la verificación de identidad de los usuarios.
- **Interfaz Intuitiva:** Ofrece una navegación fluida y amigable.

- **Tecnologías Utilizadas:**

- **Lenguajes de Programación:** Java

- **Entorno de Desarrollo:** IntelliJ IDEA
- **Frameworks:** Spring Boot para el backend
- **Seguridad:** JWT para autenticación y autorización
- **Otras Herramientas:** Maven para la gestión de dependencias, JUnit para testing

Proceso de Desarrollo

- **Desarrollo Histórico:**

El desarrollo de este proyecto tuvo 2 etapas significativas:

“Antes del domingo 26 de Mayo”, y “Después del domingo 26 de Mayo”

Durante la etapa de desarrollo, “Antes del domingo 26 de Mayo”, el proyecto fue principalmente realizado principalmente por una persona. El principal reto consistió en la creación del Modelo Entidad-Relación apropiado para nuestro proyecto y el uso apropiado de las anotaciones para relaciones entre entidades.

Estos retos fueron superados a través de los materiales enseñados en clase que mostraban una implementación correcta de las relaciones entre entidades y que entidades debía haber para nuestro proyecto, que resultaban tener similitudes.

Al finalizar esta etapa, se completaron la mayoría de los endpoints, servicios, modelos y dtos de cada entidad con sus respectivas relaciones.

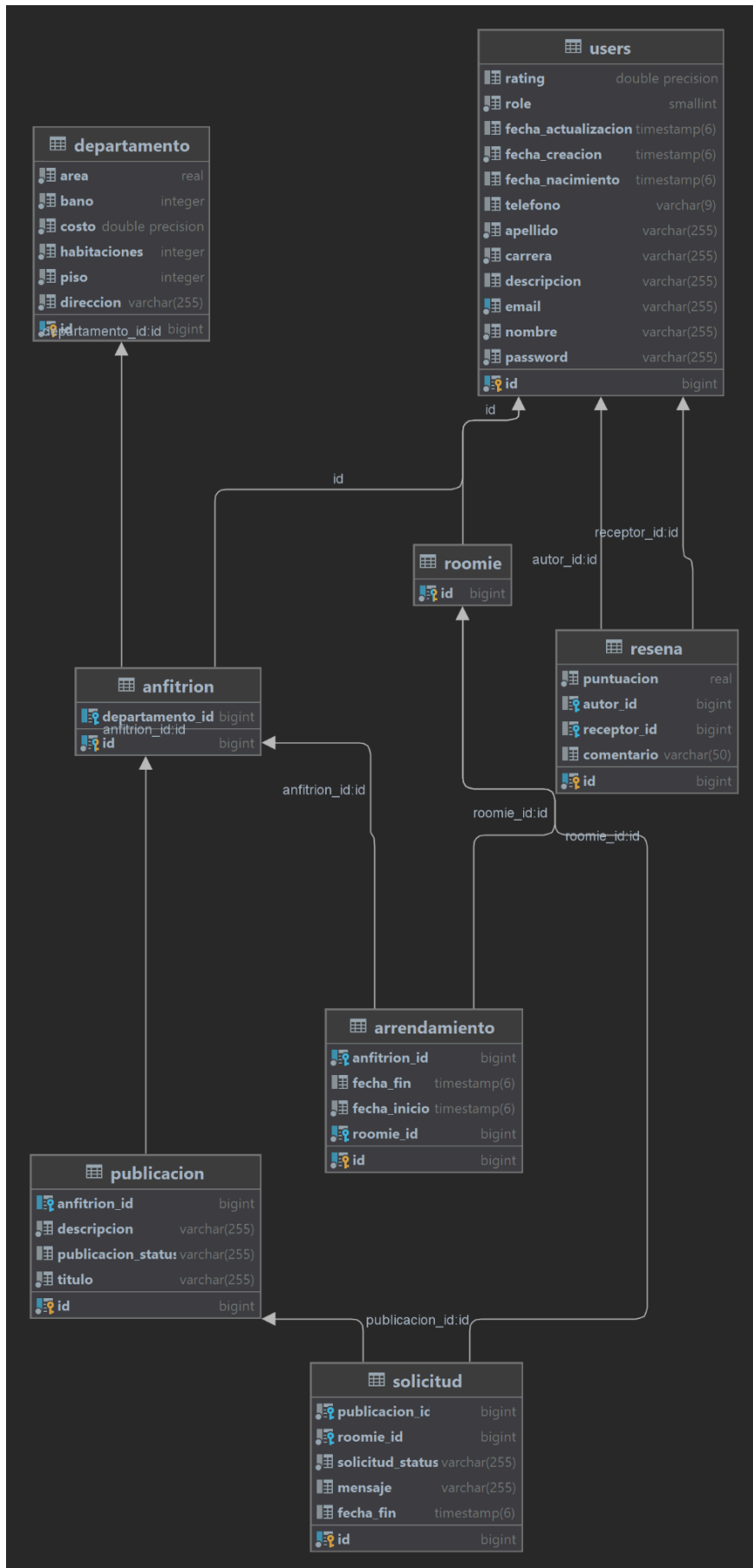
“Después del domingo 26 de Mayo”, hubo más retos como la implementación de seguridad, eventos asincrónicos, la autorización en cada endpoint y testing. Sin embargo, durante esta etapa hubo un integrante más en la producción del proyecto.

Estos retos fueron superados a través del trabajo en equipo.

Al finalizar esta etapa, se logró implementar la autorización a los endpoints, los eventos asincrónicos con envío de correos con plantilla HTML personalizada, y la autorización por rol en cada endpoint, y la implementación de algunos tests.

Modelo Entidad-Relación

- Diagrama ER:



- Descripción de Entidades:

1. User

- Atributos:

id: Identificador único del estudiante (PK).
nombre: Nombre del estudiante.
apellido: Apellido del estudiante.
carrera: Carrera que estudia el estudiante.
descripción: Descripción personal del estudiante.
email: Dirección de correo electrónico del estudiante.
fecha_actualizacion: Fecha de la última actualización del perfil.
fecha_nacimiento: Fecha de nacimiento del estudiante.
fecha_de_creacion: Fecha de creación del perfil.
password: Contraseña del estudiante.
rating: Calificación del estudiante dentro de la aplicación.
teléfono: Número de teléfono del estudiante.
Role: El rol que tiene un estudiante dentro del contexto de la aplicación(ANFITRION o ROOMIE).
fecha_actualizacion: Fecha de la última actualización de los datos del estudiante.

- **Relaciones:**

- Forma parte de una **relación unidireccional Muchos a Uno** proveniente de Reseña.

2. Roomie

- **Atributos:**

id: Identificador único del roomie (PK)(FK).

- **Relaciones:**

- Es una subclase de la superclase User.

- Es parte de una **relación unidireccional Muchos a Uno**, proveniente de la entidad Solicitud. También es parte de una relación unidireccional Uno a Uno, proveniente de la entidad Arrendamiento.

3. Anfitrión

- **Atributos:**

id: Identificador único del anfitrión (PK).

departamento_id: Identificador unico del departamento (FK)

- **Relaciones:**

- Tiene una relación de unidireccional Uno a Uno con la entidad Departamento.

- Es parte de una **relación unidireccional Uno a Uno** proveniente de la entidad Arrendamiento. También es parte de la relación unidireccional Uno a Uno proveniente de la entidad Publicación.

4. Resena

- **Atributos:**

id: Identificador único de la reseña (PK).

comentario: Comentario de la reseña.

puntuacion: Puntuación dada en la reseña.

autor_id: Identificador del autor de la reseña (FK).

receptor_id: Identificador del receptor de la reseña (FK).

- **Relaciones:**

- **2 relaciones Unidireccional Muchos a Uno** con User (ya que puede ser realizada tanto por un Anfitrión, que por un Roomie,). Las 2 relaciones representar al Autor y al receptor, respectivamente.

5. Departamento

- **Atributos:**

id: Identificador único del departamento (PK).

direccion: Direccion del departamento.

area: Área del departamento.

bano: Número de baños en el departamento.

costo: Costo de arrendamiento del departamento.

habitaciones: Número de habitaciones del departamento.

piso: Piso donde se encuentra el departamento.

- **Relaciones:**

- **Unidireccional Uno a Uno** desde anfitrión (cada anfitrión tiene un departamento).

6. Arrendamiento

- **Atributos:**

id: Identificador único del arrendamiento (PK).

fechaFin: Fecha de fin del arrendamiento.

fechaInicio: Fecha de inicio del arrendamiento.

anfitrión_id: Identificador del estudiante anfitrión (FK).

roomie_id: Identificador del estudiante arrendatario (FK).

- **Relaciones:**

- **Unidireccional Uno a Uno** con Anfitrión (Un arrendamiento solo tiene un Anfitrión).

- **Unidireccional Uno a Uno** con Roomie (Un arrendamiento solo tiene un Roomie).

7. Publicación

- **Atributos:**

id: Identificador único de la publicación (PK).

publicacionStatus: Estado de la publicación.

descripción: Descripción de la publicación.

titulo: Título de la publicación.

anfitrión_id: Identificador del departamento (FK).

- **Relaciones:**

- **Uno a Uno** con Anfitrión (Una publicación solo puede tener un anfitrión)

8. Solicitud

- **Atributos:**

id: Identificador único de la solicitud (PK).

solicitudStatus: Estado de la solicitud.

fecha_inicio: Fecha de inicio sugerida por el Roomie que manda la solicitud.

fecha_fin: Fecha de fin sugerida por el Roomie que manda la solicitud.
Puede tomar valor de indefinido.

publicacion_id: Identificador de la publicación (FK).

roomie_id: Identificador del roomie que realiza la solicitud (FK).

- **Relaciones:**

- **Muchos a uno** con publicación (varias solicitudes pueden estar asociadas a una publicación).

- **Muchos a uno** con roomie (varias solicitudes pueden ser realizadas por un roomie).

Testing

- **Niveles de Testing Realizados:**

Para asegurar la calidad del software en el proyecto, se han implementado diferentes niveles de pruebas. A continuación, se describen los niveles de prueba realizados:

1. **Pruebas de Integración**

Clases involucradas:

- PublicacionRepositoryTest
- AbstractContainerBaseTest

Descripción:

Las pruebas de integración se centran en verificar que los diferentes componentes de la aplicación funcionan correctamente juntos. En este caso, se ha utilizado la anotación `@DataJpaTest` en combinación con un entorno de base de datos real proporcionado por Testcontainers para realizar pruebas de integración.

Detalles:

PublicacionRepositoryTest: Esta clase prueba la integración entre el repositorio de publicaciones (PublicacionRepository) y la base de datos.

- Configuración de entidades como Departamento, Anfitrión y Publicacion.
- Persistencia de estas entidades en la base de datos.
- Recuperación de publicaciones basadas en el anfitrión para asegurar que la interacción con la base de datos funciona correctamente.
- Uso de TestEntityManager para gestionar el estado de persistencia de las entidades durante las pruebas.

AbstractContainerBaseTest: Proporciona un entorno de prueba aislado y reproducible utilizando un contenedor PostgreSQL.

- Configuración dinámica de propiedades de la fuente de datos para usar el contenedor PostgreSQL.
- Asegura que las pruebas de integración se ejecuten en un entorno que simula una base de datos real sin depender de la infraestructura local.

2. Pruebas de Sistema

Clase involucrada:

- UtecRoomieApplicationTests

Descripción:

Las pruebas de sistema se realizan para verificar que el sistema completo, como una unidad integrada, cumple con los requisitos especificados. Estas pruebas aseguran que todos los componentes de la aplicación funcionan juntos correctamente.

Detalles:

UtecRoomieApplicationTests: Esta clase contiene una prueba básica para verificar que el contexto de la aplicación Spring Boot se carga correctamente.

- La anotación `@SpringBootTest` se utiliza para arrancar toda la aplicación y comprobar la configuración general.
- La prueba `contextLoads()` asegura que no hay errores en la configuración del contexto y que todos los beans necesarios se crean correctamente.

3. Pruebas Unitarias

Descripción:

Las pruebas unitarias verifican la funcionalidad de componentes individuales, como funciones, métodos y clases, de manera aislada.

Herramientas Utilizadas: JUnit.

Ejemplos de Casos de Prueba:

- Verificación de la creación de usuarios.
- Verificación del cálculo de calificaciones.
- Pruebas de validación de formularios.

4. Pruebas de Seguridad

Descripción:

Estas pruebas identifican vulnerabilidades de seguridad en la aplicación y verifican la robustez de los mecanismos de seguridad implementados.

Herramientas Utilizadas: OWASP ZAP, análisis manual de código.

Ejemplos de Casos de Prueba:

- Prueba de autenticación y autorización mediante JWT.
- Verificación de la seguridad en la transmisión de datos sensibles.
- Prueba de protección contra ataques comunes como SQL injection y XSS.

- **Resultados:**

Las pruebas revelaron cierto tipo de errores, los cuales tratamos de solucionar la mayoría para garantizar un sistema más estable y funcional.

Medidas de Seguridad Implementadas

- **Seguridad de Datos:**

Cifrado: Uso de HTTPS y cifrado de datos sensibles.

Autenticación y Autorización: Implementación de JWT.

Gestión de Permisos: Control de acceso basado en roles.

- **Prevención de Vulnerabilidades:**

Inyección SQL: JpaRepository, implementada en todos los repositorios de las entidades, ya implementa medidas de protección contra inyección SQL.

XSS: En SecurityConfig, se implementó un filtro de seguridad para ataques XSS brindado por una página web. (Baeldung)

Conclusión

- **Logros del Proyecto:**

- Se implementó la creación de perfiles detallados tanto para estudiantes que buscan alojamiento como para aquellos que lo ofrecen, permitiendo descripciones precisas de intereses y características de los alojamientos.
- La plataforma ofrece búsquedas detalladas basadas en múltiples criterios como número de habitaciones, baños, y rango de precios, facilitando la personalización de la búsqueda.
- Información completa y detallada de los departamentos disponibles, incluyendo reseñas y puntuaciones, lo que proporciona una visión clara y confiable de las opciones disponibles.
- Se desarrolló un sistema eficiente para enviar y gestionar solicitudes de reserva, permitiendo a los anfitriones aceptar o rechazar solicitudes según la disponibilidad.
- Implementación de un sistema de evaluación que permite a los usuarios compartir sus experiencias, promoviendo la transparencia y la confianza en la comunidad.
- Aseguramiento de la identidad de los usuarios mediante verificaciones, especialmente para estudiantes de UTEC, garantizando la seguridad en las transacciones y comunicaciones.
- Los estudiantes ahora tienen más opciones para residir cerca de la universidad, reduciendo los tiempos de desplazamiento y mejorando la conveniencia.
- Uso de tecnologías robustas como Java, Spring Boot, JWT para autenticación, y herramientas de testing como JUnit, garantizando la calidad y seguridad del software.
- Implementación de cifrado, control de acceso basado en roles, y prevención de vulnerabilidades como inyección SQL y XSS, asegurando un entorno seguro para todos los usuarios.

- **Aprendizajes Clave:**

El desarrollo del proyecto nos hizo dar cuenta de la importancia de la seguridad y la usabilidad en plataformas de este tipo, así como la necesidad de pruebas exhaustivas para asegurar la calidad.

- **Trabajo Futuro:**

Algunas posibles mejoras o extensiones serían implementar funciones adicionales como notificaciones push, integración con servicios de pago en línea, y una app móvil para mejorar la accesibilidad.

Apéndices

- **Licencia:**

“Apache License 2.0”, ya que permite comercializar y ofrece buena protección.

- **Referencias:**

- Ppts del curso.
- Recursos de los E2E pasados.
- <https://devdocs.io/javascript/>
- Materiales de los repasos y asesorías del curso.
- Postman API Fundamentals Student Expert
- <https://www.baeldung.com/jpa-cascade-types>
- Herramientas como ChatGPT o Github Copilot.
- <https://docs.spring.io/spring-data/jpa/reference/jpa/query-methods.html>
- <https://learning.postman.com/docs/introduction/overview/>
- <https://documenter.getpostman.com/view/15567703/UVyxRtnq>
- [Prevent Cross-Site Scripting \(XSS\) in a Spring Application | Baeldung](#)