



Facultad de
Ciencias Sociales y
Tecnologías de la Información
Talavera de la Reina. UCLM

PROYECTO ISO II

Pablo Gómez García / Álvaro Olmedo Fernández

Pablo Nieto Mayoral / Pablo Sánchez García-Ochoa

ÍNDICE

CONTENIDO

Metodología de desarrollo	3
Product Owner	3
Scrum Master	3
Equipo de Desarrollo	3
Reunión de Planificación del Sprint	3
Gestión de la Configuración del Software	4
Project Board - scrum	5
Issues.....	5
Diagrama de gantt.....	8
Requisitos.....	8
Funcionales.....	8
No funcionales.....	9
DIAGRAMAS UML	10
Diagrama de BD.....	11
Diagramas de comunicación	11
Buscar alojamiento.....	11
Reservar alojamiento.....	11
Dar de alta propiedad.....	12
Diagrama de secuencia	12
Prototipo del sitio web.....	13
Página principal	13
Inicio de sesión y crear cuenta	14
Estructura de la aplicación	15

METODOLOGÍA DE DESARROLLO

En nuestro proyecto utilizaremos Scrum como metodología de gestión de proyectos junto con Kanban como metodología de desarrollo de software.

PRODUCT OWNER

Álvaro Olmedo será el mayor representante del producto.

Define las características del producto y establece los requisitos de este. Prioriza las características del proyecto respecto a su valor de mercado, a su vez establece la prioridad de los requisitos en cada iteración.

SCRUM MASTER

En nuestro caso es Pablo Gómez. Es el responsable de mantener el repositorio GitHub y se asegura de mantener activa la comunicación entre roles. A su vez, Pablo G. se asegura que el grupo permanece funcional.

EQUIPO DE DESARROLLO

Pablo Nieto y Pablo Sánchez se encargarán de desarrollar cualquier tipo de funcionalidad.

Serán ayudados en las tareas por Álvaro O. y Pablo G. en estas tareas.

REUNIÓN DE PLANIFICACIÓN DEL SPRINT

Las reuniones se harán diariamente variando su duración en cuanto a la disponibilidad de los integrantes, con el objetivo de realizar un sprint semanal.

GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE

Con el uso de GitHub utilizaremos el desarrollo por ramas. Emplearemos 4 ramas principales:

- **Rama Developer:**

Esta rama principal permite la revisión en desarrollo o el mantenimiento del código. Permite también corregir los problemas detectados antes del lanzamiento del producto.

- **Ramas features:**

Estas ramas creadas desde developer sirven para implementar nuevas características o funcionalidades.

Cuando se ha completado el desarrollo de la nueva característica se une con la rama developer para liberarla en la siguiente versión.

- **Rama de release:**

Una vez se va a lanzar una nueva versión de la aplicación, se añade a esta rama para hacer pruebas y detectar posibles errores para poder corregirlos sin que afecte al usuario.

- **Rama main:**

Esta rama es la versión final en la que se lanza tras pasar por developer y release, contiene los cambios realizados pulidos de developer. Se genera una versión mediante un etiquetado y se lanza la aplicación.

PROJECT BOARD - SCRUM

Hemos usado un Project Board con estas columnas principales:

- ➔ **Backlog:** inventario maestro con todas las tareas.
- ➔ **Sprint / To Do:** ítems comprometidos para el sprint actual.
- ➔ **In Progress:** tareas que alguien está desarrollando.
- ➔ **QA / In Testing:** pruebas manuales / verificación.
- ➔ **Done:** tareas completadas según la Definition of Done.

ISSUES

Crear repositorio y configurar Maven + Spring Boot y Spring JPA	
Descripción	Nuestro propósito aquí es inicializar el proyecto con una estructura básica y las dependencias, para que todos puedan clonar y ejecutar
Responsables	La persona responsable es el Developer
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 0

Planificar los Sprints	
Descripción	Nuestro propósito aquí es coordinar al equipo y planificar todos los Sprints
Responsables	Las personas responsables son el Scrum Master y el Product Owner
Estimación	Estimamos una duración entre 0.5-1 día
Sprint	Sprint 0

Recogida de requisitos del proyecto	
Descripción	Nuestro propósito aquí es documentar que debe hacer el producto identificando los requisitos funcionales y no funcionales
Responsables	La persona responsable es el Product Owner
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 0

Análisis de requisitos y creación del diagrama de la BD	
Descripción	Nuestro propósito aquí es traducir requisitos a entidades y relaciones para la implementación
Responsables	Las personas responsables son el Product Owner y Developer
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 0

Diseño de la arquitectura básica	
Descripción	Nuestro propósito aquí es documentar las decisiones técnicas
Responsables	La persona responsable es el Developer
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 1

Prototipo del diseño de la interfaz	
Descripción	Nuestro propósito aquí es elaborar bocetos de pantallas clave
Responsables	Las personas responsables son el Product Owner y Developer
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 1

Crear estructura inicial del proyecto	
Descripción	Nuestro propósito aquí es crear las carpetas necesarias y añadir README inicial
Responsables	La persona responsable es el Developer
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 1

Implementar la funcionalidad principal	
Descripción	Nuestro propósito aquí es desarrollar la feature core que constituye el MVP
Responsables	La persona responsable es el Developer
Estimación	Estimamos una duración entre 3-6 días
Sprint	Sprint 2

Hacer el sistema de login	
Descripción	Nuestro propósito aquí es permitir acceso de usuarios mediante un formulario
Responsables	La persona responsable es el Developer
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 2

Guardar y mostrar datos	
Descripción	Nuestro propósito aquí es implementar operaciones básicas en los datos
Responsables	La persona responsable es el Developer
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 3

Probar funciones del sistema	
Descripción	Nuestro propósito aquí es ejecutar pruebas funcionales manuales sobre los flujos principales
Responsables	La persona responsable es el Developer
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 3

Comprobar errores comunes	
Descripción	Nuestro propósito aquí es capturar y corregir validaciones o errores de entrada más frecuentes
Responsables	La persona responsable es el Developer
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 4

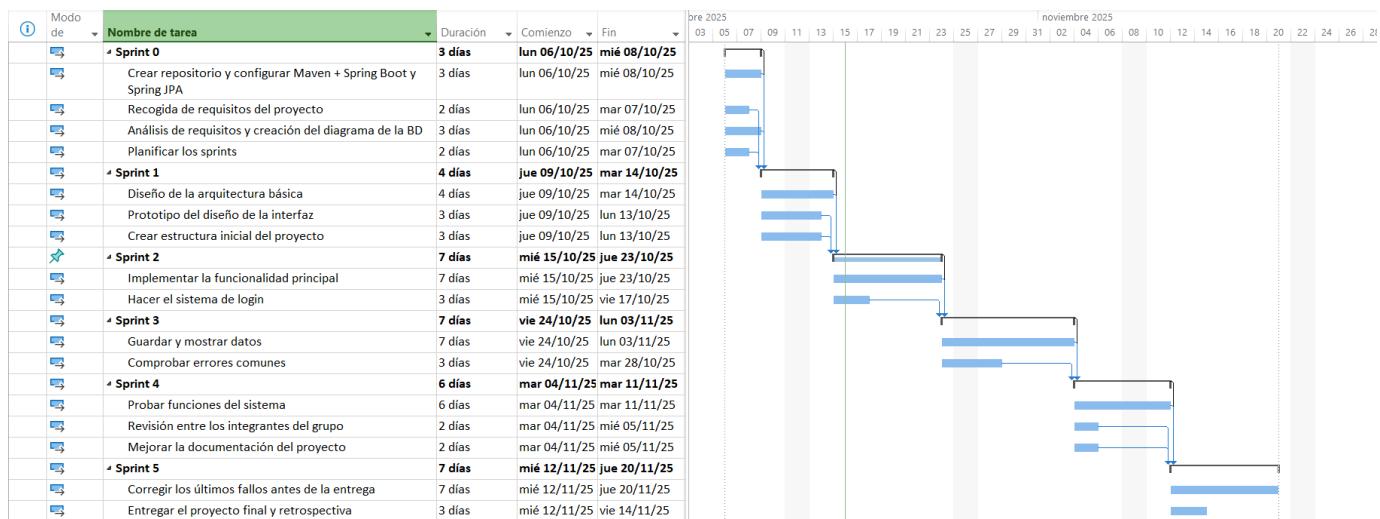
Revisión entre los integrantes del grupo	
Descripción	Nuestro propósito aquí es que cada miembro revise secciones de código de otros compañeros
Responsables	La persona responsable es el Developer (todo el grupo participa).pñ
Estimación	Estimamos una duración entre 0.5-1 día
Sprint	Sprint 4

Mejorar la documentación del proyecto	
Descripción	Nuestro propósito aquí es completar el README con guía de uso
Responsables	Las personas responsables son el Developer y el Product Owner
Estimación	Estimamos una duración entre 1-3 días
Sprint	Sprint 4

Corregir los últimos fallos antes de la entrega	
Descripción	Nuestro propósito aquí es resolver bugs prioritarios detectados en pruebas
Responsables	La persona responsable es el Developer
Estimación	Y estimamos una duración entre 1-3 días
Sprint	Sprint 5

Entrega del proyecto final y retrospectiva	
Descripción	Nuestro propósito aquí es generar el release final e identificar lecciones aprendidas
Responsables	Las personas responsables son el Scrum Master y el Product Owner
Estimación	Estimamos una duración entre 0.5-1 día
Sprint	Sprint 5

DIAGRAMA DE GANTT



El diagrama de Gantt representa un cronograma del proyecto donde cada tarea aparece con una barra en una línea de tiempo, donde la longitud de la barra indica su duración. Por otra parte, las tareas están agrupadas del Sprint 0 al Sprint 5. Además, las flechas conectan a las barras que representan la dependencia Finish->Start que obliga que un Sprint no empiece hasta que no haya terminado el anterior. En resumen, este diagrama nos permite ver de forma rápida qué se debe entregar en cada sprint y qué tareas dependen de otras.

REQUISITOS

FUNCIONALES

RF01 - Los propietarios, después de dar de alta una cuenta personal, pueden asociar propiedades a la misma.

RF02 - Los propietarios, de manera excepcional, pueden aceptar a los usuarios para su servicio.

RF03 - Los usuarios pueden buscar alojamiento filtrando por: destino, fechas, tipo de inmueble... para examinar las opciones de alquiler.

RF04 - Además, los usuarios, de forma excepcional, pueden aplicar filtros avanzados. Ejemplo: ver aquellos inmuebles con posibilidad de reserva inmediata; seleccionar filtros para ciertas comodidades; o seleccionar política de cancelación de reserva.

RF05 - Los inmuebles poseerán una descripción en la que se detallan las características de esta y sus servicios.

RF06 - No es necesario tener una cuenta de cliente para poder ver los alojamientos, pero solo pueden almacenar propiedades en una lista de deseos personal.

RF07 - Es necesario tener una cuenta de usuario para poder aplicar a la reserva del inmueble deseado, no es posible hacerlo sin estar registrado en la plataforma.

RF08 - Los inquilinos una vez logeados, al seleccionar el inmueble a alquilar, pueden aplicar a la reserva.

RF09 - Las propiedades con reserva inmediata se puede completar el pago con tarjeta de crédito, débito o PayPal desde la plataforma.

RF10 - La aplicación no deberá permitir más de una única reserva por inmueble en el mismo periodo de tiempo.

RF11 - La aplicación no deberá permitir más de una reserva para una misma persona en el mismo periodo de tiempo.

RF12 - Los usuarios están asociados a la persona titular de la cuenta, la cual debe ser participante del uso del inmueble.

RF13 - Los propietarios podrán elegir las condiciones para la cancelación de las reservas, las cuales serán comunicadas al inquilino antes del pago. Se pedirá una confirmación de lectura al mismo para saber que es consciente de las mismas.

RF14 - Al terminar una estancia, se podrá puntuar a los usuarios por el estado de los inmuebles y a los inquilinos por el servicio prestado. A su vez, pueden añadir un breve texto argumentando el porqué de su valoración, la cual puede ser de ayuda a otros usuarios de la plataforma.

RF15 - La interfaz poseerá un menú desplegable para que los usuarios puedan acceder a las principales funcionalidades de la plataforma.

- Futuras especificaciones: Permitir el cambio de unidad monetaria a elección del usuario.

NO FUNCIONALES

RNF01 - El sistema debe permitir un uso fluido para los usuarios sin incluir tiempos de carga excesivos.

RNF02 - El sistema debe soportar grandes cantidades de usuarios en temporadas de alta demanda.

RNF03 - La interfaz debe ser intuitiva, sencilla, accesible desde el navegador y fácil de entender para los usuarios.

RNF04 - Planificar escalonadamente los cambios a realizar para no modificar código sin tener claro la idea que se busca.

RNF05 - Presentar código ordenado y documentado para ser entendible de modificar en actualizaciones futuras, fallos detectados o implementar nuevas funcionalidades.

RNF06 - Consistencia en los procesos a tiempo real, si un usuario reserva en una fecha el cambio debe reflejarse inmediatamente.

RNF07 - Generar diferentes ramas del proyecto para generar estabilidad entre versiones.

RNF08 - Antes de la publicación o actualización del código probar previamente las funcionalidades de este y buscar posibles errores del código modificado.

RNF09 - Escalabilidad que permita un mejor soporte para los usuarios en actualizaciones futuras.

DIAGRAMAS UML

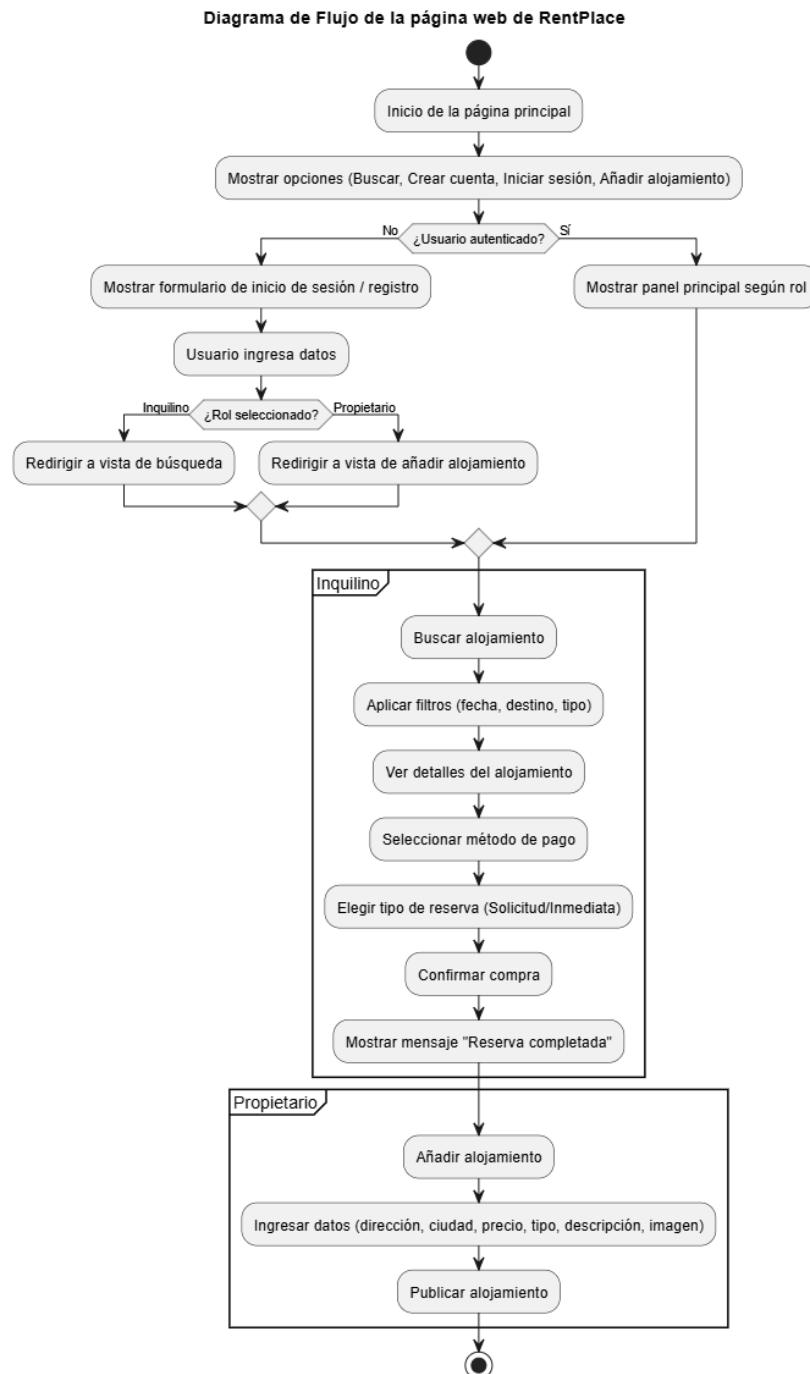
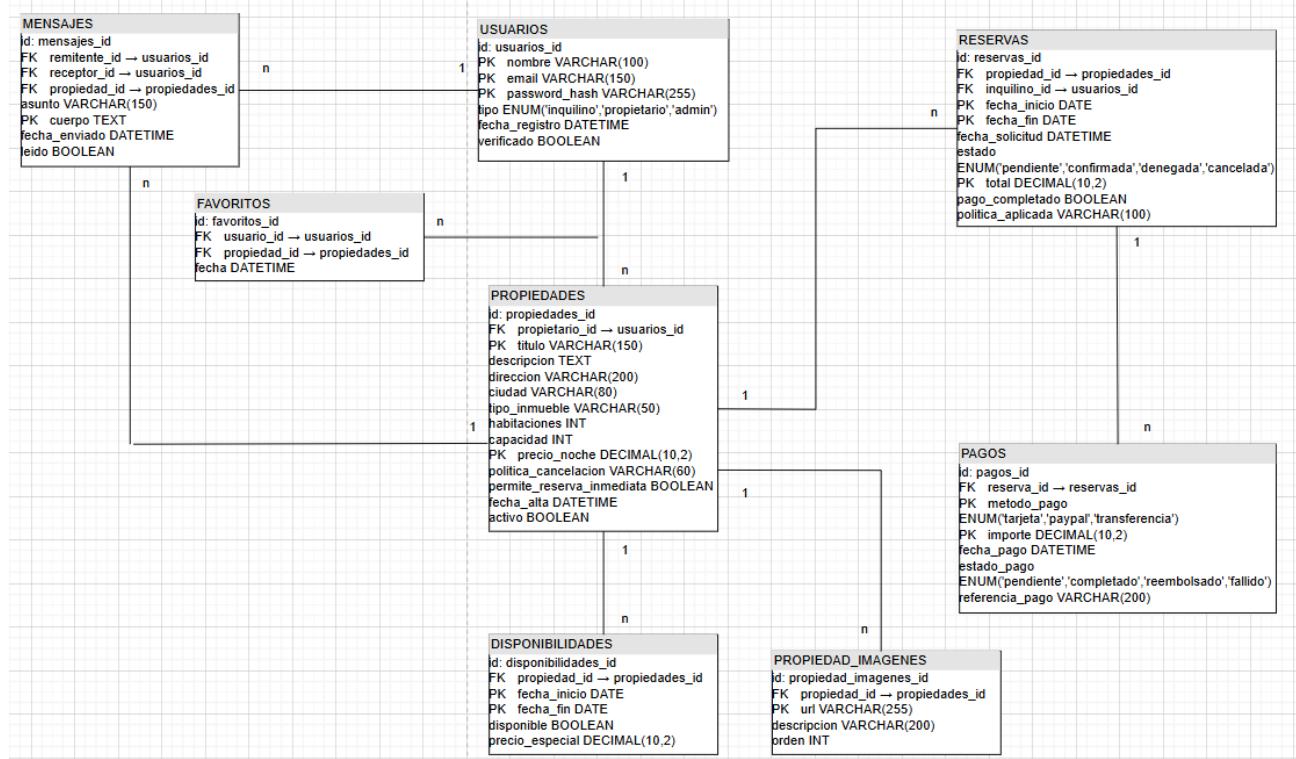


DIAGRAMA DE BD

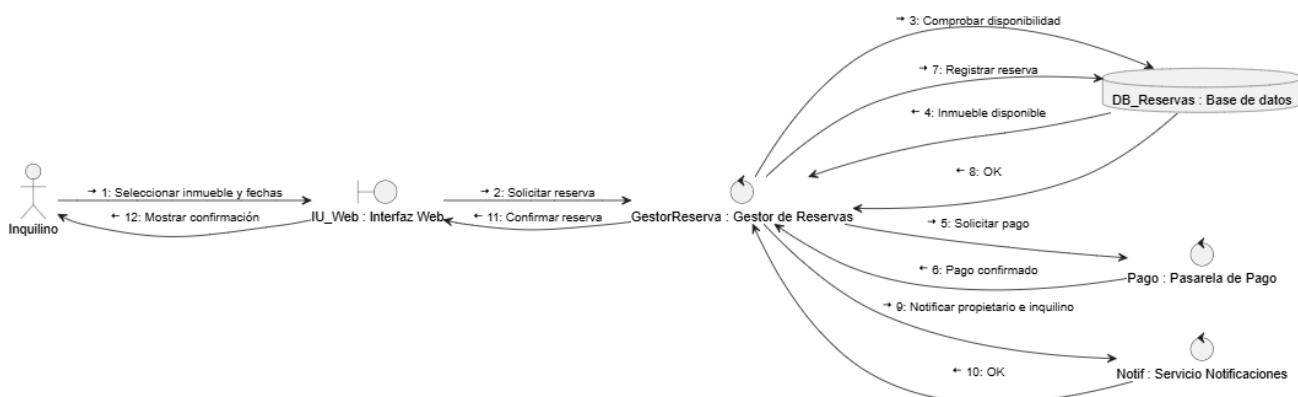


DIAGRAMAS DE COMUNICACIÓN

BUSCAR ALOJAMIENTO



RESERVAR ALOJAMIENTO



DAR DE ALTA PROPIEDAD

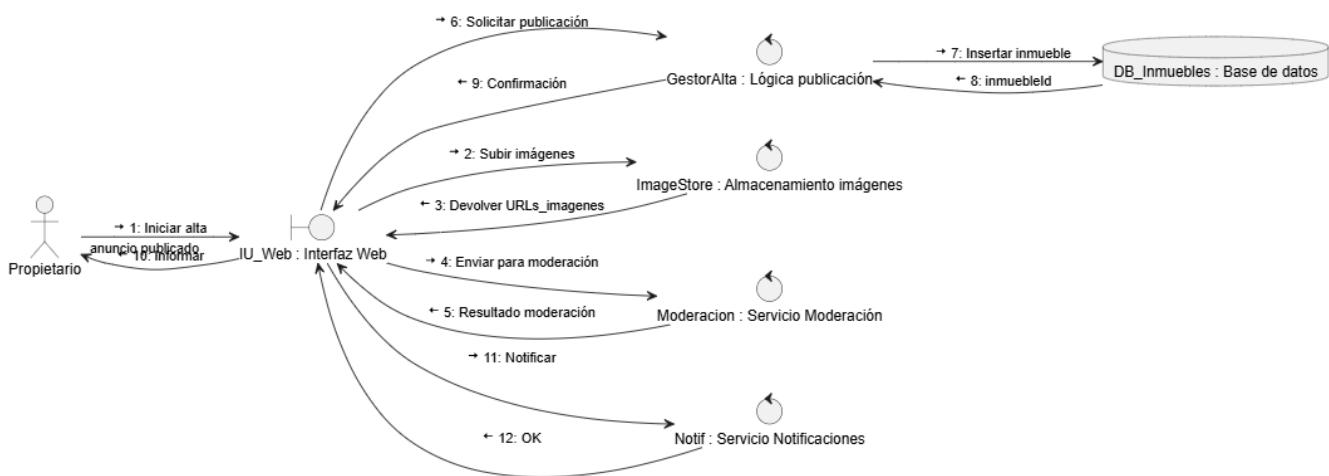
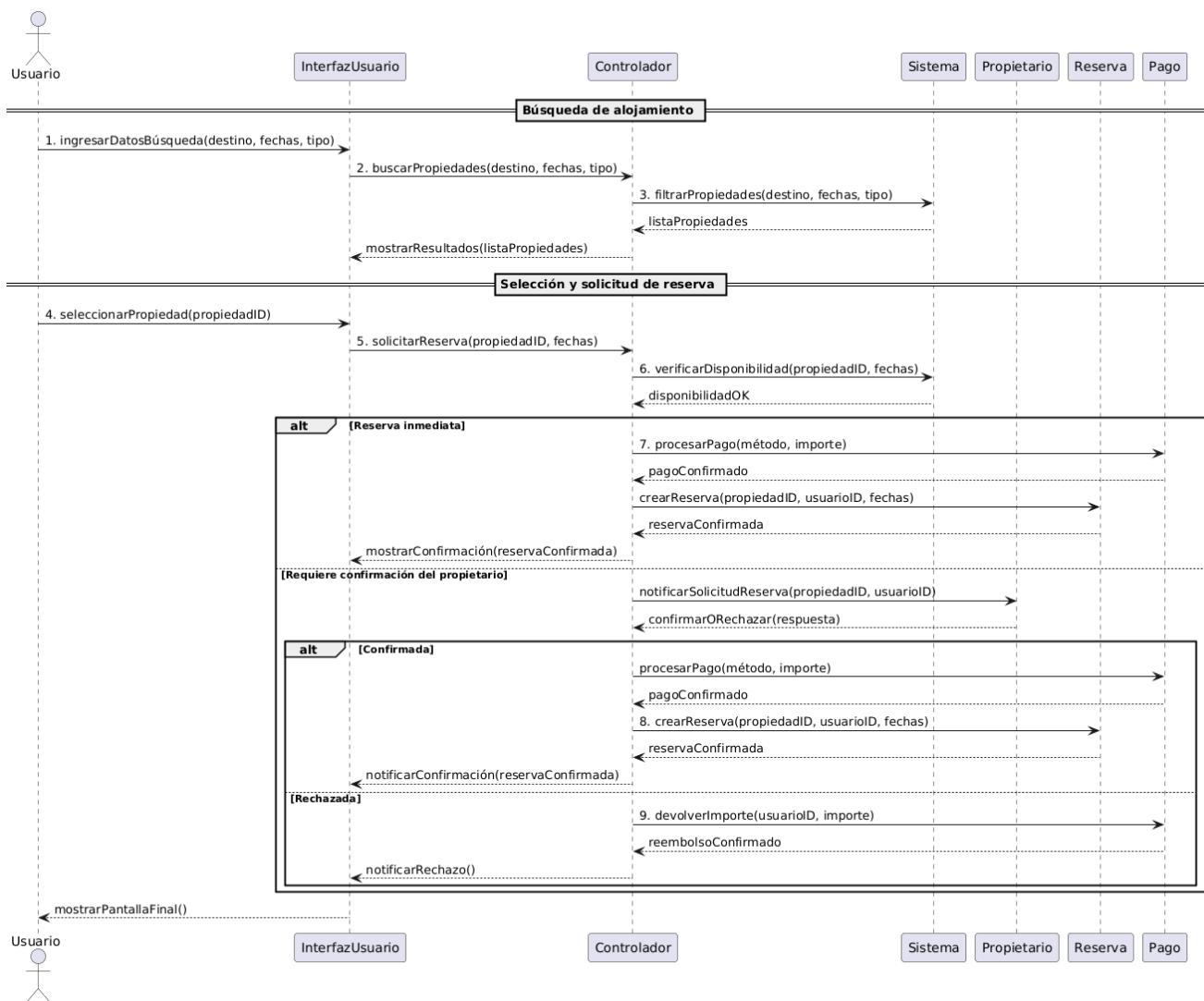
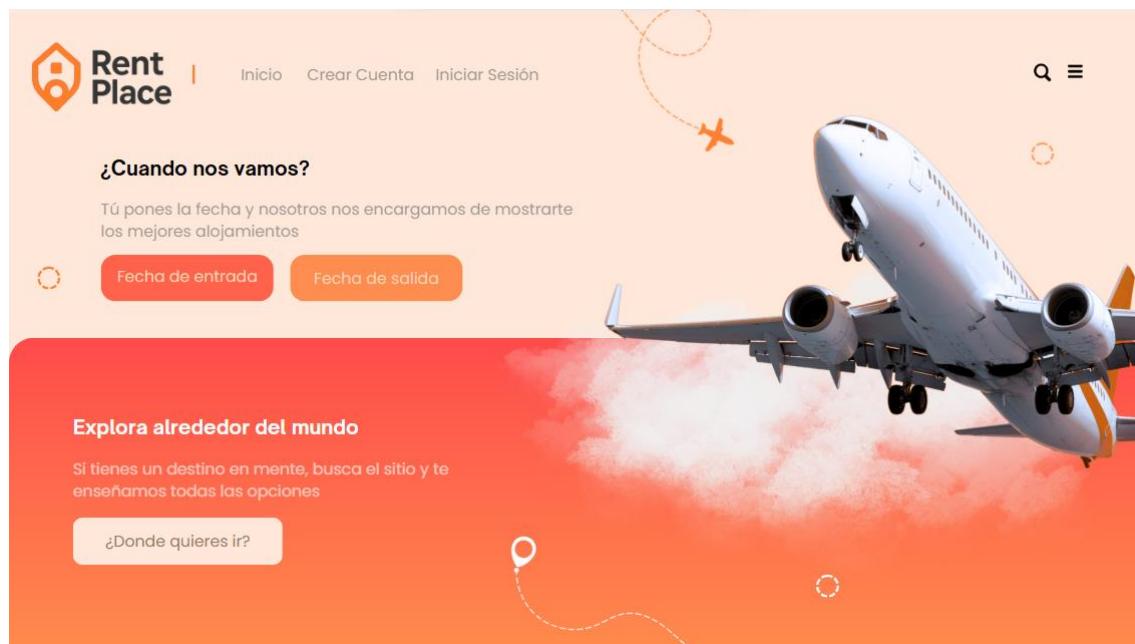


DIAGRAMA DE SECUENCIA

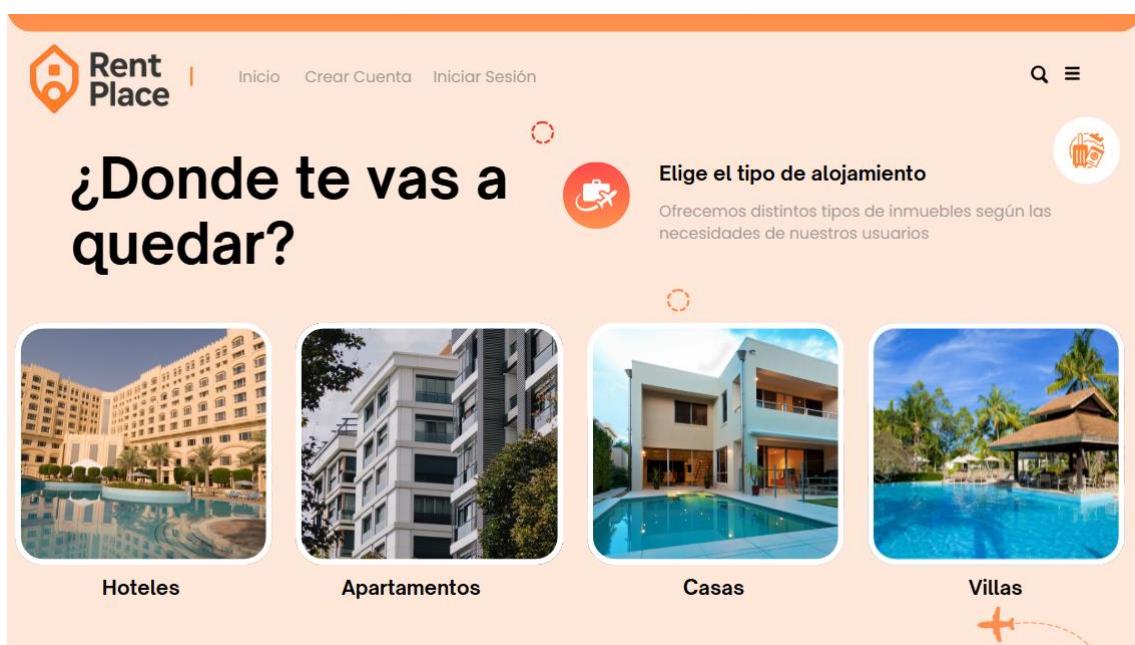


PROTOTIPO DEL SITIO WEB

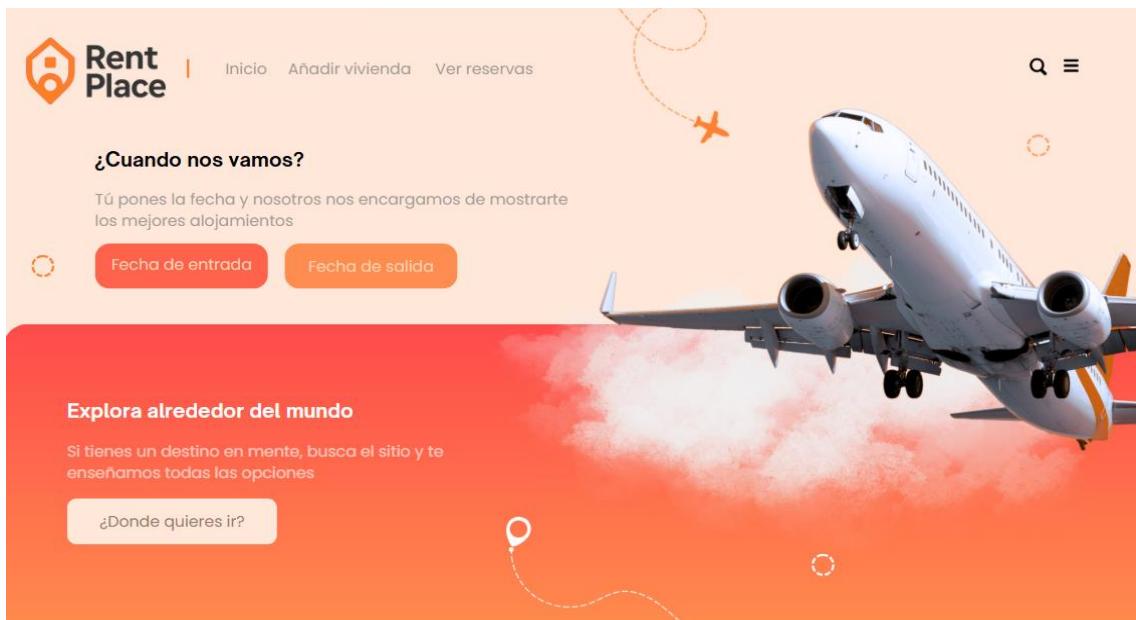
PÁGINA PRINCIPAL



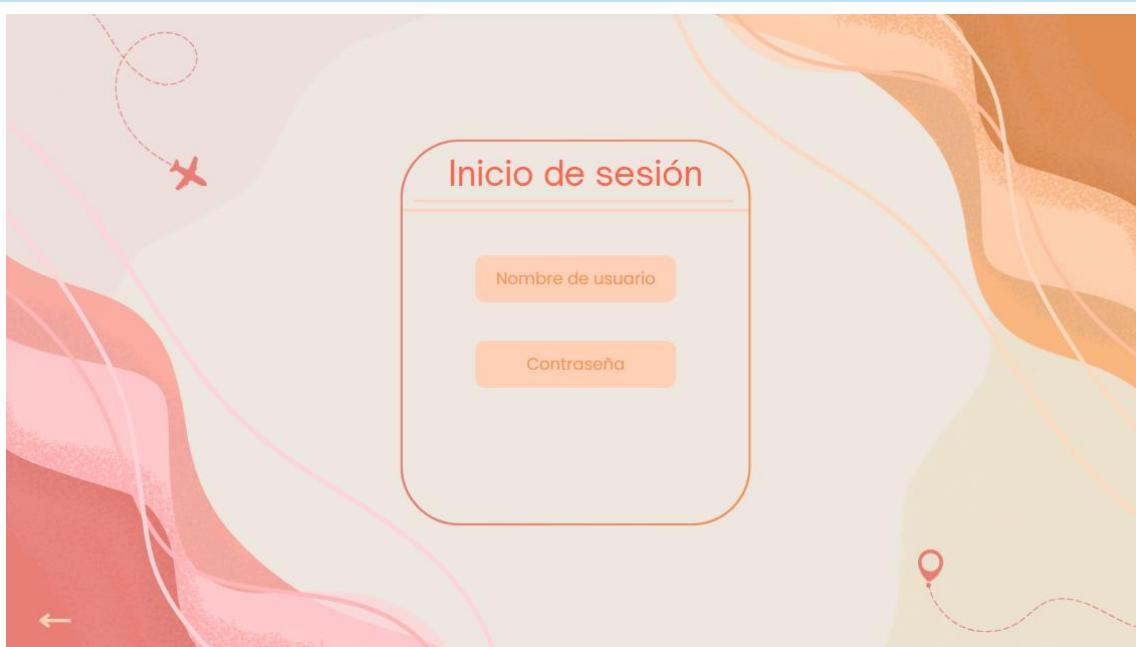
Prototipo de la página principal de RentPlace. La barra superior es azul con el título "PROTOTIPO DEL SITIO WEB". Abajo de ella, una barra azul contiene el subtítulo "PÁGINA PRINCIPAL". El contenido principal tiene un fondo naranja. En la parte superior izquierda, hay el logo "Rent Place" y en la parte superior derecha, iconos para búsqueda y menú. A la izquierda, se muestra un formulario para elegir fechas de entrada y salida. A la derecha, una imagen de un avión en vuelo. En el centro, se muestra un formulario para buscar destinos y un mapa con un marcador.

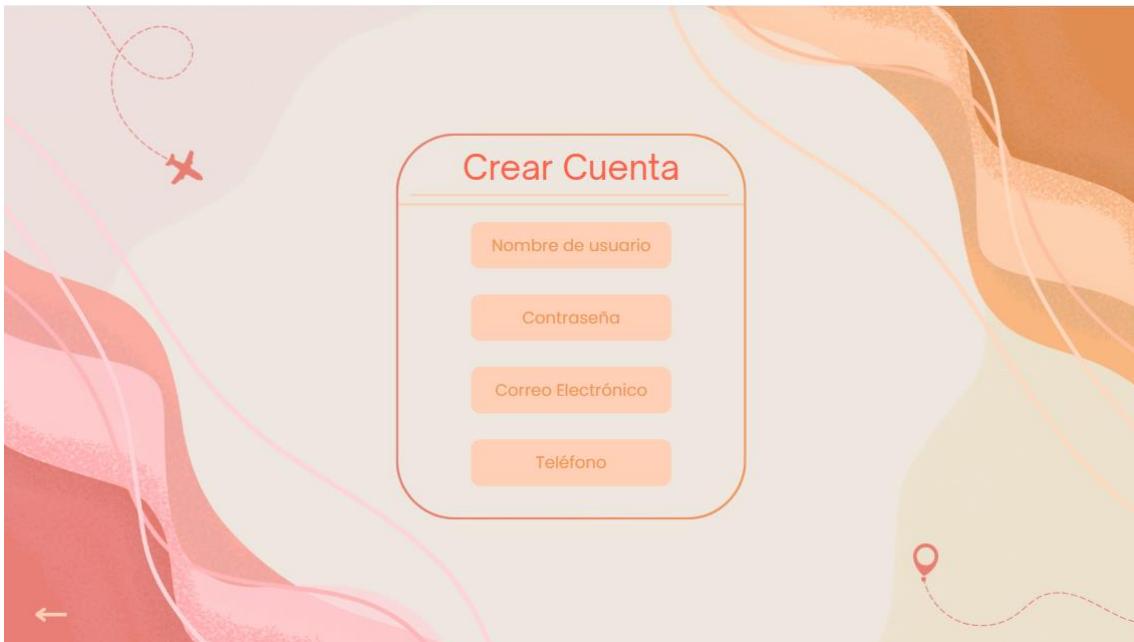


Prototipo de la página de selección de tipo de alojamiento. La barra superior es naranja con el logo "Rent Place" y en la parte superior derecha, iconos para búsqueda y menú. El centro de la página tiene un encabezado "¿Donde te vas a quedar?" con un icono de maleta. A la derecha, se muestra un formulario para elegir el tipo de alojamiento, que incluye imágenes y descripciones para Hoteles, Apartamentos, Casas y Villas.



INICIO DE SESIÓN Y CREAR CUENTA





ESTRUCTURA DE LA APLICACIÓN

- Para empezar, usaremos Spring Boot como marco de desarrollo y JPA para estructurar los datos de la base de datos Derby.
- En primer lugar, tenemos la carpeta **src/main/java** que contiene dos paquetes, el primer paquete contiene el arranque del Spring Boot y el servlet container. Luego en el segundo paquete, tiene los controladores que devuelven las vistas de Home, Login y Registrer. Por último, hay otros tres paquetes para gestionar las entidades (definición de la entidad, operaciones y capa intermedia entre los controladores y la base de datos).
- En segundo lugar, tenemos la carpeta **src/main/resources** que contiene a su vez dos carpetas más (static y templates). Dentro de la carpeta static tenemos hojas de estilo para cada página, fuentes locales y recursos estáticos como: CSS, imágenes.... Por último, en la carpeta templates contiene plantillas HTML que devuelven los controladores para mostrar las páginas al usuario.
- En tercer lugar, tenemos la carpeta **src/test/java** donde contiene el código de pruebas del proyecto para verificar que la aplicación arranca sin errores y servir como punto de partida para añadir test unitarios o de integración.
- Además, tenemos la carpeta **rentplaceDB** donde se almacenan los archivos físicos de la base de datos.
- Por último, tenemos el archivo **pom.xml** que es la configuración principal de Maven donde definimos el nombre del proyecto, sus dependencias y versión de java.