

# Documentación proyecto ISO II



|                         |                              |
|-------------------------|------------------------------|
| Javier Rojo Salinas     | Javier.rojo3@alu.uclm.es     |
| Guillermo Juárez García | Guillermo.Juarez@alu.uclm.es |
| David Blanco Bodas      | David.blanco5@alu.uclm.es    |
| Jaime Montalvo Alfageme | Jaime.Montalvo1@alu.uclm.es  |

## Descripción General

Se pretende desarrollar un sistema similar a Airbnb donde personas particulares pueden reservar alojamientos de propietarios particulares. Tanto los inquilinos como propietarios deben registrarse previamente para utilizar el sistema.

- Los propietarios (una vez logeados) pueden dar de alta propiedades (viviendas completas o habitaciones individuales) ofertándolas para ser alquiladas. Los propietarios marcan si desean que los usuarios realicen reservas directas, o guardarse el derecho de confirmar o no las solicitudes de reserva.
- Los usuarios pueden primero buscar alojamiento (destino, fechas, tipo de inmueble, etc.) para examinar las opciones de alquiler. Excepcionalmente también pueden aplicar filtros avanzados para, por ejemplo: ver aquellos

inmuebles con posibilidad de reserva inmediata; seleccionar filtros para ciertas comodidades; o seleccionar política de cancelación de reserva.

- Para realizar la búsqueda de inmuebles no es necesario estar logeado, pero los usuarios registrados pueden agregar algunas propiedades a su lista de deseo.
- Los inquilinos logeados, una vez han seleccionado el inmueble a alquilar, pueden completar la reserva. Si la propiedad permite reserva inmediata se puede completar el pago directamente (vía tarjeta de crédito, débito o paypal).
- Si la propiedad o inmueble no permite reserva inmediata se hace una solicitud de reserva (completando también en este momento el pago). La solicitud de reserva es notificada al propietario quién puede confirmar o no la reserva. Si la confirmación es positiva, esto se le notifica al inquilino. Si la solicitud es denegada, el sistema devuelve el dinero al inquilino y es igualmente notificado al cliente.

## Requisitos

### Funcionales:

#### *1. Gestión de usuarios*

- Registro de usuarios (inquilinos y propietarios).
- Inicio de sesión y cierre de sesión.
- Gestión de perfil (datos personales, métodos de pago, preferencias, etc.).

#### *2. Gestión de propiedades*

- Alta de propiedades por parte de propietarios autenticados.
- Clasificación de propiedades como vivienda completa o habitación individual.
- Configuración de disponibilidad, precios, comodidades y políticas de cancelación.
- Opción para permitir reservas inmediatas o solicitudes de reserva.

#### *3. Búsqueda y filtrado de alojamientos*

- Búsqueda por destino, fechas, tipo de inmueble, etc.
- Filtros avanzados: reserva inmediata, comodidades, política de cancelación, etc.
- Visualización de resultados con detalles de cada propiedad.

#### 4. Lista de deseos

- Usuarios registrados pueden agregar propiedades a su lista de favoritos.

#### 5. Proceso de reserva

- Inquilinos logeados pueden iniciar una reserva.
- Si la propiedad permite reserva inmediata:
- Se realiza el pago directamente (tarjeta de crédito, débito o PayPal).
- Si la propiedad requiere confirmación:
- Se realiza una solicitud de reserva con pago incluido.
- El propietario recibe la solicitud y puede aceptarla o rechazarla.
- Si acepta, se notifica al inquilino.
- Si rechaza, se notifica al inquilino y se devuelve el dinero automáticamente.

#### 6. Notificaciones

- Notificaciones automáticas por correo o en la plataforma para:
- Confirmación o rechazo de reservas.
- Cambios en el estado de propiedades o solicitudes.

### No funcionales:

#### 1. Usabilidad

Interfaz intuitiva y fácil de usar para todos los perfiles de usuario.

Accesibilidad desde dispositivos móviles y de escritorio.

#### 2. Rendimiento

Respuesta rápida en búsquedas y navegación.

Capacidad para manejar múltiples usuarios y transacciones simultáneamente.

#### 3. Seguridad

Autenticación segura de usuarios.

Protección de datos personales y financieros.

Procesamiento seguro de pagos (cumplimiento con PCI-DSS).

#### 4. Disponibilidad

Alta disponibilidad del sistema (24/7).

Mecanismos de recuperación ante fallos.

### 5. Escalabilidad

Capacidad para crecer en número de usuarios, propiedades y transacciones sin degradar el rendimiento.

### 6. Mantenibilidad

Código modular y documentado para facilitar futuras mejoras o correcciones.

Registro de logs para auditoría y diagnóstico de errores.

### 7. Compatibilidad

Compatible con los principales navegadores web.

Integración con pasarelas de pago (Visa, MasterCard, PayPal).

## Objetivo

Desarrollar un sistema similar a Airbnb donde personas particulares pueden reservar alojamientos de propietarios particulares.

## Roles del Sistema

- Propietario: encargado de publicar propiedades, administrar la disponibilidad de sus inmuebles y confirmar las reservas.
- Inquilino: usuario que puede buscar alojamientos, realizar reservas y efectuar pagos dentro de la plataforma.

## Funcionalidades Principales

### └─ Gestión de Usuarios

└─ Registro de propietarios e inquilinos

└─ Autenticación segura de acceso

### └─ Gestión de Propiedades

└─ Alta y edición de inmuebles por parte de propietarios

└─ Administración de disponibilidad y detalles del inmueble

### └─ Búsqueda de Inmuebles

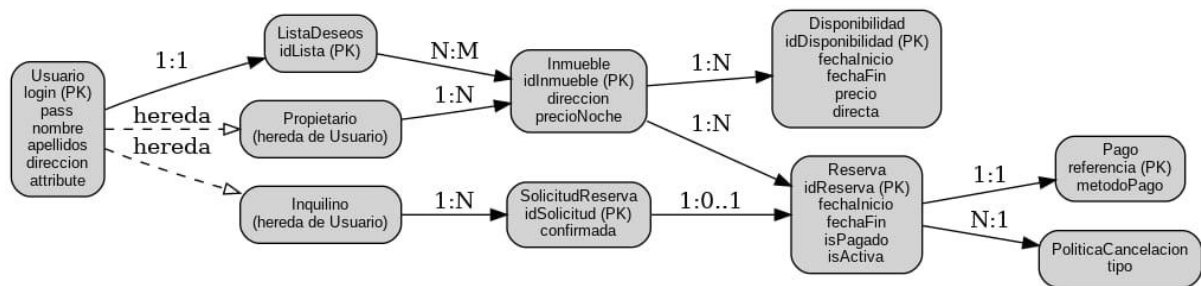
└─ Filtros avanzados:

- |
  - |— Reserva inmediata
  - |— Comodidades específicas
  - |— Políticas de cancelación
- └─ Resultados dinámicos y actualizados
- └─ Reservas y Pagos
  - |— Reserva inmediata o mediante solicitud
  - |— Procesamiento de pagos electrónicos
  - |
    - |— Tarjeta de crédito
    - |— Tarjeta de débito
    - |— PayPal
  - └─ Confirmación y notificación de estado de reserva
- └─ Lista de Deseos (Wishlist)
  - |— Creación y gestión por parte del usuario
  - └─ Almacenamiento de alojamientos favoritos
- └─ Notificaciones Automáticas
  - |— Confirmación de reservas
  - └─ Rechazo o cancelación de solicitudes

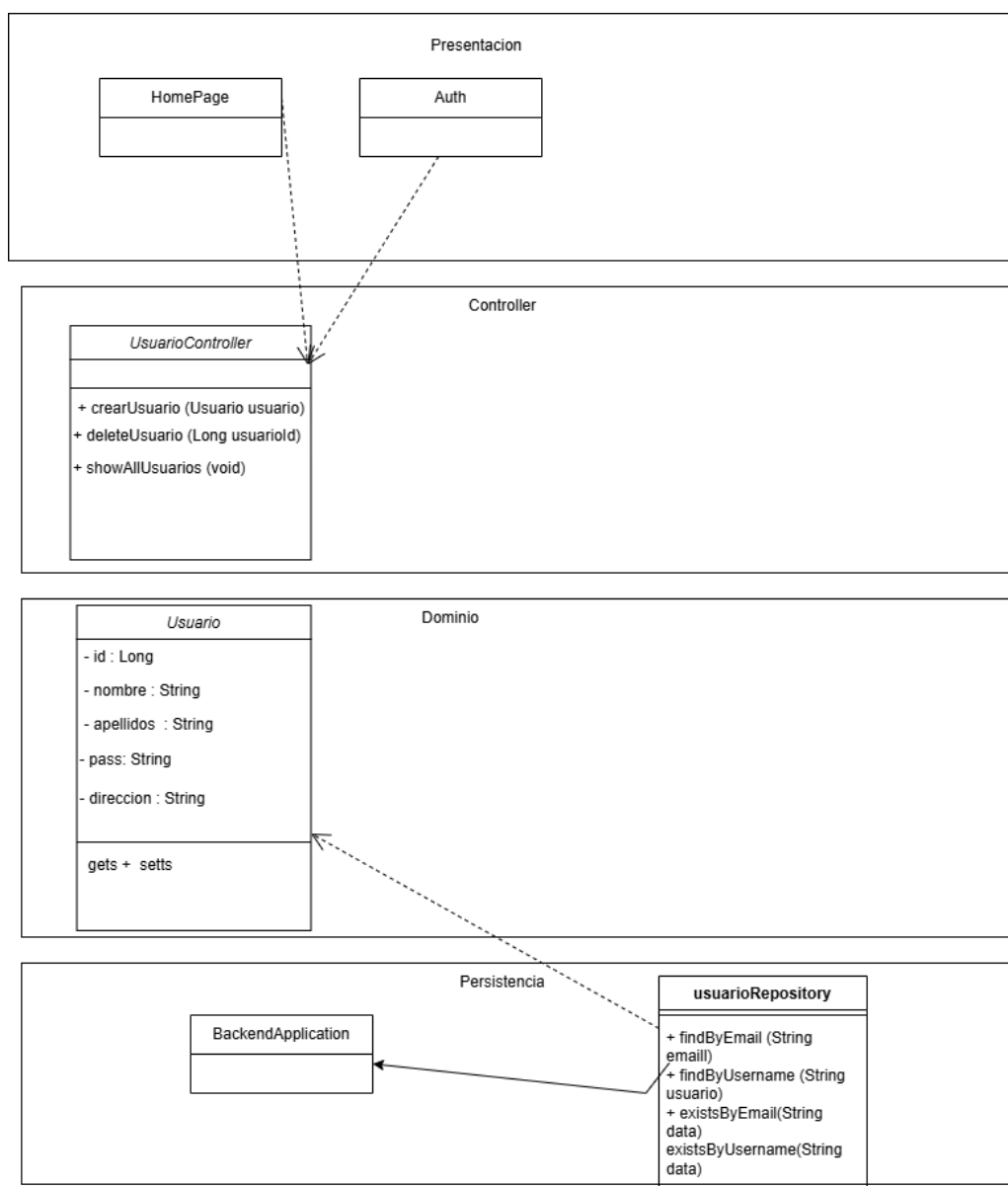
## Tecnología Usada

Para este proyecto estaremos usando H2 como base de datos y java 21 como lenguaje de programación junto con springboot, react y maven.

## Diagrama EER



## Diagrama de clases



## Planificación Sprints

Para el desarrollo del proyecto se opta una combinación de la metodología Scrum con un panel Kanban para el seguimiento de tareas. Cada sprint tiene una duración de una semana.

### *¿Qué es Scrumban?*

Scrumban nace de la fusión entre Scrum (basado en sprints, roles definidos y reuniones periódicas) y Kanban (centrado en la gestión visual del flujo de trabajo y la entrega continua). Su objetivo es ofrecer un marco de trabajo más flexible, visual y adaptable a distintos tipos de proyectos.

### *Características clave*

Sprints opcionales: Aunque se pueden usar ciclos de trabajo como en Scrum, no son obligatorios.

Tablero Kanban: Se utiliza para visualizar tareas, controlar el flujo y limitar el trabajo en curso (WIP).

Sin roles estrictos: No requiere figuras como Product Owner o Scrum Master, aunque pueden mantenerse si el equipo lo desea.

Reuniones flexibles: Se pueden realizar reuniones diarias, retrospectivas o de planificación, pero no son obligatorias.

Adaptabilidad: Ideal para equipos que evolucionan desde Scrum hacia Kanban o que necesitan un enfoque más dinámico.

### *¿Cuándo usar Scrumban?*

Cuando el equipo necesita más flexibilidad que la que ofrece Scrum.

En proyectos con prioridades cambiantes o tareas impredecibles.

Para equipos que quieren visualizar mejor su carga de trabajo y evitar bloqueos.

### *Ventajas*

Combina lo mejor de dos mundos: estructura + fluidez.

Mejora la eficiencia y la transparencia del trabajo.

Reduce la rigidez de Scrum sin perder el enfoque ágil.

## Sprint 0 - Preparación inicial

Objetivo: Establecer la base del proyecto y preparar el entorno de desarrollo.

Tareas realizadas:

Creación del backend y frontend básicos sobre los que se iterará.

Configuración del repositorio y del panel Kanban.

Definición inicial del backlog del producto.

## Sprint 1 - Requisitos y configuración

Objetivo: Definir requisitos funcionales y preparar la infraestructura del proyecto.

Tareas realizadas:

Recopilación de requisitos funcionales y no funcionales.

Creación del panel de seguimiento de tareas.

Generación de documentación en el directorio /docs.

Definición e implementación de la estructura de carpetas en el frontend.

## Sprint 2 - Diseño del backend y autenticación

Objetivo: Inicio del diseño de los formularios de login y registro en el frontend.

Tareas realizadas:

Avance en la implementación de la lógica del backend y conectar con el frontend.

Verificación de errores en el backend.

Modificación y diseño de los formularios en el frontend.

Modificación y extensión de la API REST de usuarios.