



UNIVERSIDAD  
**COMPLUTENSE**  
MADRID

# CompluHome

## Acceso al git

## Indice

<u>1. Resumen ejecutivo</u> .....	3
• Breve descripción del proyecto	
• Objetivos principales	
• Alcance general	
• Beneficios esperados	
• Cronograma	
<u>2. Antecedentes y justificación</u> .....	5
• Contexto del problema o necesidad	
• Situación actual	
• Impacto esperado	
<u>3. Objetivos del proyecto</u> .....	7
<u>4. Alcance del proyecto</u> .....	8
<u>5. Requerimientos del sistema</u> .....	9
• Requerimientos funcionales	
• Requerimientos no funcionales	
• Requerimientos de usuario	

<u>6. Arquitectura y diseño del sistema</u>	13
• Subsistemas	
• Interfaces	
• Casos de uso	
<u>7. Plan de trabajo</u>	46
• Fases del proyecto	
• Cronograma	
• Dependencias e hitos clave	
<u>8. Recursos del proyecto</u>	54
• Costes de Personal	
• Costes Tecnológicos	
• Software y herramientas	
• Otros costes	
• Resumen total	
<u>9. Gestión del proyecto</u>	57
• Metodología	
• Gestión de riesgos	
• Control de calidad	
• Gestión de cambios	
• Comunicación	
<u>10. Plan de pruebas</u>	62
• Tipos de Pruebas	
• Estrategia de Validación	
• Casos de Uso y Pruebas Asociadas	
• Herramientas de Testing	

# 1. Resumen ejecutivo

## 1.1 Breve descripción del proyecto:

CompluHome es una plataforma de gestión y oferta de alquiler de habitaciones orientada a los estudiantes de la Universidad Complutense de Madrid. Su objetivo principal es proporcionar un modelo accesible y estructurado que permita responder a la creciente demanda de vivienda universitaria vivida en los últimos años. El proyecto se centra en la colaboración con grandes propietarios de inmuebles por medio de contratos que garanticen una base sólida de clientes a cambio de una regulación en los precios, favoreciendo la reducción de estos a favor del estudiantado.

Además, CompluHome propone un esquema económico y tecnológico que podrá servir de referencia para otras instituciones académicas, de manera que su estructura pueda ser replicada internacionalmente y adaptada a las particularidades del entorno de cada universidad.

## 1.2 Objetivos principales:

Las metas a alcanzar con este proyecto abarcan mejoras tanto económicas como sociales. Por un lado buscamos reducir el precio medio de las habitaciones destinadas a estudiantes de la comunidad universitaria a niveles más asequibles y compatibles con su situación económica, mitigando de esta manera la barrera económica de acceso a los estudios presenciales. Por otro lado, y en términos sociales, pretendemos reducir el inconveniente que resulta encontrar alojamiento en Madrid para aquellos estudiantes tanto nacionales como extranjeros que no poseen apoyo familiar en la capital. Así mismo, se contempla la inclusión de jóvenes investigadores y personal académico en los anteriores grupos, ampliando el impacto de nuestra plataforma más allá del alumnado tradicional.

## 1.3 Alcance general:

Inicialmente el alcance del proyecto se centrará en la comunidad complutense, lo que incluye a estudiantes de grado, posgrado, investigadores predoctorales y postdoctorales, así como otros perfiles asociados a la actividad universitaria.

Sin embargo, del diseño de nuestro modelo se puede percibir una clara escalabilidad: CompluHome podrá ser replicado y adaptado a las condiciones socioeconómicas de otras provincias españolas o en universidades que presenten problemáticas similares en materia de vivienda estudiantil.

## 1.4 Beneficios esperados:

Los beneficios previstos en este caso suponen un mayor beneficio social más que económico de acuerdo con nuestro objetivo. Hay que añadir a ello el carácter sostenible del proyecto, orientado a no depender de ayudas económicas externas en fases avanzadas y generar capital suficiente para mantener la plataforma junto a sus empleados e infraestructura sin perseguir márgenes de beneficio elevados propios de productos puramente comerciales.

## 1.5 Cronograma:

El desarrollo detallado del cronograma asociado junto a demás términos del plan de trabajo se especifican más adelante.

## 2. Antecedentes y Justificación

### 2.1 Contexto del problema o necesidad:

Nos situamos en Madrid, ciudad en la que la regla es la aglomeración y la falta de vivienda, que en adición con la especulación de precios han dejado al estudiante de medios/bajos recursos (procedente de localidades distantes) apartado de la posibilidad de estudiar en la capital sin tener que atravesar un notorio esfuerzo económico para costearse una habitación.

A esto se le suma el predominio de aplicaciones de alquiler generalistas que no contemplan las particularidades y necesidades del alumno. Estas herramientas, al no estar orientadas específicamente al ámbito académico, no ofrecen garantías adaptadas al perfil del estudiante ni suficiente transparencia en la gestión de anuncios, precios y contratos. En algunos casos, estas aplicaciones llegan a aprovecharse de la incertidumbre y vulnerabilidad del alumnado, dando lugar a precios inflados, condiciones poco claras o prácticas que contribuyen a una experiencia precaria.

Así mismo, la idea de un sistema institucional propio que permita centralizar la oferta de alojamiento, ofrecer protección frente abusos o coordinar acuerdos con propietarios de forma regulada no ha sido aún explorada en la mayoría de universidades.

Es ante esta situación de desigualdad que surge la idea de nuestra plataforma.

### 2.2 Situación actual:

En el contexto actual, las soluciones disponibles resultan insuficientes para garantizar un acceso justo y estable a la vivienda universitaria. Los pisos adaptados para estudiantes son escasos, y la mayoría están gestionados por intermediarios privados que priorizan beneficios sin ofrecer respaldo institucional ni estándares claros de calidad o seguridad.

CompluHome nace con la intención de transformar esta situación atacando a los dos principales pilares del problema: La falta de vivienda adecuada para estudiantes y los precios inaccesibles que caracterizan el mercado madrileño. Además de ello, pretendemos mejorar la relación propietario-estudiante por

medio de diversas funcionalidades que se detallarán más adelante, buscando proyectar una imagen de seguridad a ambas partes así como desarrollar un ecosistema basado en la transparencia y la comunicación.

## 2.3 Impacto esperado:

De la implantación de nuestra plataforma esperamos un crecimiento positivo en diversos aspectos tanto de la comunidad universitaria como de movilización de capital: la mayor accesibilidad a los estudios presenciales de los que gozará el alumnado de la UCM se traducirá en un aumento en el número y diversidad de estudiantes en las diferentes facultades, a la vez que se estimula el sector mobiliario.

Por otro lado, y aunque pueda sonar menor, una residencia más cercana a las facultades propiciará hábitos sostenibles y saludables al reforzar la idea de la movilidad activa (caminar o ir en bicicleta), reduciendo la dependencia de transporte privado o público y alineándose con los objetivos de ciudades más sostenibles y descarbonizadas.

# 3. Objetivos del Proyecto

## 3. Objetivos del Proyecto:

### Objetivos generales:

El propósito de este Proyecto es realizar una aplicación web con la que cualquier estudiante de la Universidad Complutense de Madrid (en su defecto se podrá extender para cualquier estudiante de Madrid si se desea) pueda encontrar un alojamiento a un precio razonable, intentando así cuadrar los precios para el presupuesto de cualquier estudiante.

### Objetivos específico:

- ❖ Diseño, creación e implementación en la aplicación web tanto de una base de datos para registrar habitaciones como de otra para registrar estudiantes en busca de habitación.
- ❖ Implementar módulo de autentificación para los usuarios.
- ❖ Creación de módulo de filtrado de viviendas por distancia.
- ❖ Creación de barra de búsqueda.
- ❖ Crear pruebas específicas para los módulos del Proyecto.
- ❖ Creación de posibilidad de mantener un chat privado entre el arrendador y el arrendado.

# 4. Alcance del proyecto

## 4. Alcance del Proyecto

Funcionalidades incluidas:

- ❖ Registro y login de usuarios
- ❖ Filtrado de viviendas
- ❖ Función de chat privado
- ❖ Publicar anuncio
- ❖ Reservar habitación
- ❖ Buscar en maps

Límites del Sistema:

- ❖ Se implementará una aplicación nativa para el móvil, no existirá una versión web.
- ❖ No se incluirá una versión traducida a varios idiomas, solo existirá una versión en castellano.

Entregables principales:

- ❖ Modelo de casos de uso y diagramas UML de análisis/diseño.
- ❖ Diseño de base de datos y script SQL de creación.
- ❖ Código fuente de la aplicación.
- ❖ Plan de pruebas y resultados de pruebas.
- ❖ Documento de especificación de requisitos.

Criterios de aceptación:

- ❖ El sistema permite registrar, autenticar y cerrar sesión de usuarios correctamente.
- ❖ El 100% de los requisitos funcionales marcados como se cumplen.
- ❖ No existen errores críticos en las pruebas de aceptación del profesor.

# 5. Requerimientos del Sistema

## 5.1 Requerimientos funcionales:

- El sistema debe permitir que los caseros ofrezcan sus viviendas de forma segura y que los estudiantes o personal vinculado a la Universidad Complutense de Madrid puedan buscar, contactar y alquilar las viviendas.
- Los caseros podrán registrarse en la plataforma mediante un formulario que les pida la información básica, incluyendo datos de contacto y documentación que acredite su identidad y titularidad o representación legal de la vivienda. Una vez registrados, podrán crear anuncios (en los cuales aparecerá la descripción, fotos, precio, tipo de contrato, servicios incluidos y fechas de disponibilidad). Previamente a su publicación, estos anuncios deberán ser revisados y aprobados por parte de la UCM para garantizar la calidad y veracidad de la oferta.
- Por otro lado, los estudiantes y el personal universitario solo podrán acceder al sistema tras verificar su pertenencia a la comunidad universitaria, ya sea mediante su correo institucional (@ucm.es) o a través de una página de autenticación de la universidad. Una vez dentro, podrán buscar viviendas utilizando filtros tales como: ubicación (preferiblemente cercana a sus facultades), rango de precios, número de habitaciones, presencia de servicios básicos, disponibilidad inmediata, etc. Los resultados se mostrarán de forma clara y ordenada, permitiendo guardar anuncios en una lista de favoritos para su revisión posterior.
- Cuando un usuario esté interesado en una vivienda, podrá enviar una solicitud de contacto al casero. Este recibirá una notificación y, si decide aceptar, se establecerá un canal seguro de comunicación donde ambos podrán intercambiar información adicional y acordar visitas o condiciones. Solo en ese momento se harán visibles los datos de contacto reales. Además, tras la finalización de un alquiler, ambas partes tendrán la opción de dejar una valoración y un comentario, fomentando así la confianza dentro de la comunidad.

- Los administradores de la UCM tendrán funciones de moderación: podrán aprobar o rechazar anuncios, gestionar reportes de usuarios, revisar conversaciones en caso de conflicto y mantener un registro de todas las acciones relevantes realizadas en la plataforma. Finalmente, el sistema enviará notificaciones automáticas por correo electrónico en momentos clave: validación de cuenta, publicación de anuncio, aceptación de contacto, nuevas ofertas según preferencias guardadas, entre otros.

## 5.2 Requerimientos no funcionales:

Más allá de las funcionalidades, el sistema debe cumplir con una serie de criterios de calidad que garanticen su fiabilidad, seguridad y facilidad de uso.

- En cuanto a seguridad, el sistema exigirá autenticación y protegerá todos los datos sensibles (como números de teléfono o direcciones exactas) mediante cifrado, mostrándolos únicamente tras aceptación mutua entre casero e interesado.
- La usabilidad es un aspecto fundamental. La interfaz debe ser intuitiva tanto para jóvenes familiarizados con la tecnología como para caseros que pueden tener menos experiencia digital. El proceso de publicación de una vivienda no debe superar los cinco pasos, y la búsqueda de alojamiento debe ser rápida y visual, con resultados comprensibles al primer vistazo. Además, la plataforma será compatible en ordenadores, tabletas y teléfonos móviles.
- En términos de rendimiento, el sistema debe responder con eficacia, sin que se quede la pantalla congelada, incluso en períodos de alta demanda como el inicio del curso académico. Se diseñará para soportar varios cientos de usuarios concurrentes sin degradación notable del servicio.
- La disponibilidad será especialmente crítica durante los meses previos a septiembre–octubre y enero–febrero, coincidiendo con los principales períodos de búsqueda de alojamiento. Durante esos meses, se buscará una buena disponibilidad. Habrá copias de seguridad diarias para poder recuperar datos que se pierdan si hay fallos.

- El sistema respetará la normativa de protección de datos. Los usuarios podrán acceder, corregir o solicitar la eliminación de sus datos en cualquier momento. No se compartirá información con terceros, ni se incluirá publicidad externa, reforzando la idea de que se trata de un servicio universitario de apoyo, no comercial.
- El sistema no deberá descuidar el aspecto de ciberseguridad, requiriendo de un buen control, seguridad y disciplina para defenderse de ciberataques, malware o phishing.
- Finalmente, el código se diseñará pensando en la mantenibilidad futura, de modo que la UCM pueda actualizar o ampliar funcionalidades sin necesidad de reescribir el sistema completo.

## 5.3 Requerimientos de usuario:

Los usuarios finales del sistema pertenecen a dos colectivos con necesidades distintas

- Los caseros son propietarios que tienen una vivienda libre y desean alquilarla a estudiantes, pero temen caer en estafas o malos tratos. Lo que más valoran es la garantía de identidad del inquilino: saber que quien contacta es realmente un estudiante o empleado de la UCM. Además, buscan un proceso sencillo, y esperan que la universidad actúe como árbitro de confianza en caso de conflicto.
- Por otro lado, los estudiantes, especialmente aquellos que llegan de otras ciudades o países, necesitan encontrar alojamiento rápido, seguro y asequible. Muchos carecen de referencias en Madrid y confían en la UCM para acceder a ofertas verificadas. Queremos que haya transparencia total: precios reales, fotos auténticas, condiciones claras y sin sorpresas en el último momento. También valoran poder leer las experiencias de otros estudiantes antes de tomar una decisión.
- El personal docente e investigador (aunque en menor número) busca alojamientos más estables, a menudo para estancias prolongadas, y aprecia la misma seriedad y validación institucional. En general, todos los usuarios

esperan que la plataforma sea rápida, fiable y neutral. No quieren sentir que están en una web comercial (como otras plataformas ya creadas en la actualidad), porque si no, es como una más. Tiene que dar impresión como servicio apoyo social de la universidad. Por eso, la ausencia de publicidad, la regulación y moderación y la verificación de perfiles son elementos clave para que los usuarios se sientan tranquilos.

# 6. Arquitectura y Diseño del Sistema

## 6.1 Subsistemas:

### **Control Usuario/Registro**

Para acceder a CompluHome se necesita una cuenta verificada por la UCM. En el caso de los alumnos sería con su correo de la universidad y en el caso de los propietarios, con una cuenta entregada por la propia UCM.

### **Casos de Uso:**

Registrarse, Iniciar/cerrar sesión, Cambiar ajustes app.

### **Búsqueda**

El usuario alumno busca ofertas (con la posibilidad de aplicar ciertos filtros) dentro de las disponibles en el sistema, donde puede ir visualizándolas o abrir alguna en específico para ver su información detallada y decidir si está interesado o no.

### **Casos de Uso:**

Buscar oferta, Filtrar búsqueda, Visualizar detalles oferta, Contactar, Ver en maps, Activar alerta.

### **Notificaciones/Mensajería**

El usuario alumno tiene un apartado de notificaciones en el que le llegarán alertas de las habitaciones que ha marcado o en las que está en trámite. Por otro lado, el usuario propietario tiene un apartado de mensajes en el que gestiona todas sus solicitudes.

### **Casos de Uso:**

Ver notificaciones, Gestionar solicitudes, Aceptar solicitud, Rechazar solicitud.

### **Gestión de ofertas (Base de datos)**

El usuario propietario puede gestionar todas sus ofertas que estarán almacenadas en la base de datos. Tiene la opción de añadir nuevas ofertas, eliminarlas o

modificarlas según su necesidad. Además, puede ver una lista con la información de todas sus habitaciones ya ocupadas, y los detalles de los alumnos que las ocupan.

### Casos de Uso:

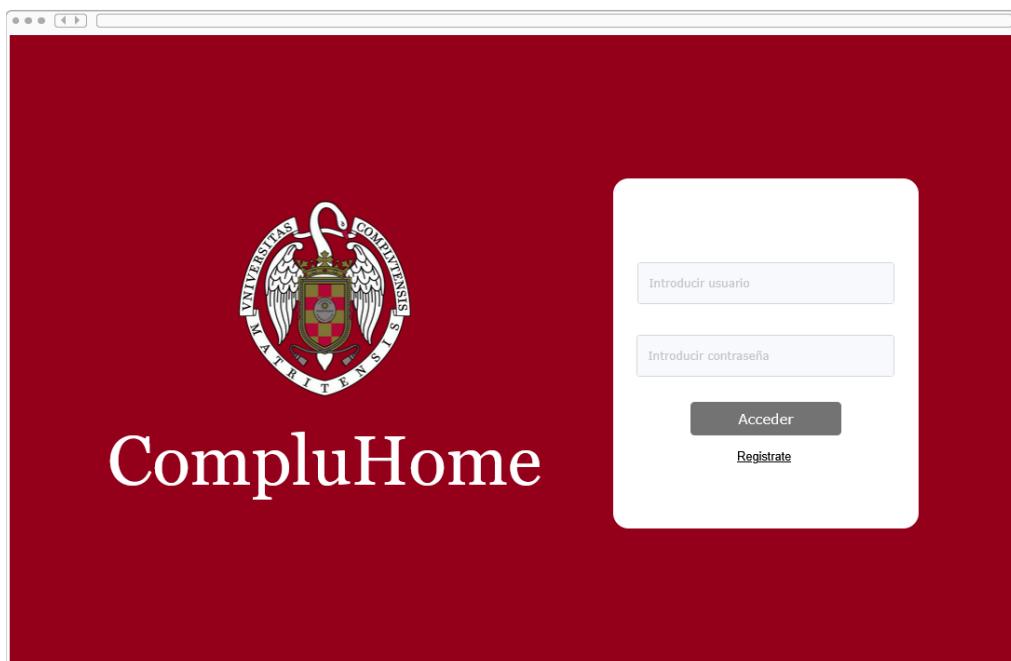
Publicar oferta, Eliminar oferta, Modificar oferta, Ver información de sus habitaciones.

## 6.2 Interfaces:

Nuestra aplicación cuenta con una serie de interfaces que facilitan la experiencia de usuario (UX). Estas interfaces están interconectadas entre sí, y dependiendo del caso de uso, será necesario navegar entre ellas. Cada una de estas interfaces está diseñada para ofrecer la mejor experiencia posible según el rol del usuario, ya sea estudiante o propietario. A continuación, se describen las principales interfaces que forman parte de la aplicación.

### 6.2.1 Interfaz de Inicio:

Esta es la interfaz inicial que ve todo usuario al abrir nuestra aplicación. Su función principal es verificar los datos de registro del usuario y, en caso de ser un usuario registrado, permitirle el acceso.



## **Descripción**

La interfaz cuenta con un sencillo inicio de sesión, pidiendo al usuario que introduzca su correo y contraseña. Se pueden distinguir dos tipos de usuario al iniciar sesión:

- **Estudiantes:** Cuentan con un correo universitario verificado.
- **Propietarios:** Necesitan haber creado una cuenta previamente, habiendo llegado a un acuerdo con la universidad en cuestión. En el caso de que un estudiante no tenga una cuenta registrada, deberá introducir su correo y seleccionar la opción de *registrarse*, lo que enviará un correo a su dirección académica para verificar su registro. Una vez introducidos los datos correctos, dependiendo de si se trata de un correo universitario o de propietario, el usuario será redirigido a la interfaz que le corresponda.
- Los estudiantes serán dirigidos a la interfaz de Búsqueda de piso.
- Los propietarios serán dirigidos a la interfaz de Propietarios.

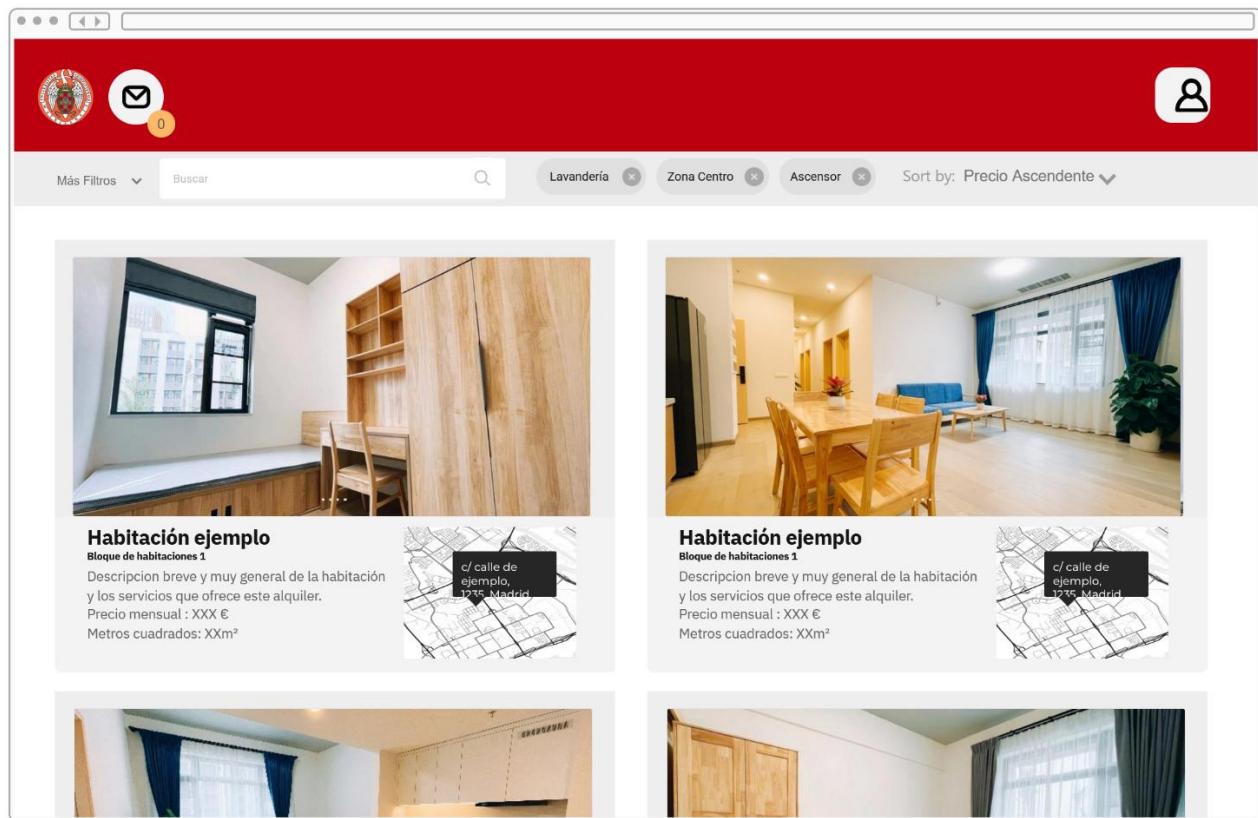
## **Casos de uso**

Esta es una lista de los casos de uso desarrollados en esta interfaz:

- Registro de Usuario.
- Inicio de Sesión.
- Verificación de los datos mediante la UC

## 6.2.2 Interfaz Principal:

La interfaz principal solo puede ser accedida por los estudiantes registrados en la aplicación. Esta interfaz muestra la oferta de habitaciones disponibles.



### Descripción

La interfaz cuenta con una barra superior donde encontramos un ícono de notificaciones a la izquierda, y un ícono de usuario a la derecha. El ícono de notificaciones muestra una lista de las notificaciones entrantes para el usuario. Estas pueden ser la aceptación o denegación de una habitación o la aparición de nuevas habitaciones en la aplicación. El ícono de usuario muestra un menú donde se pueden seleccionar distintas opciones de accesibilidad o cerrar sesión.

Bajo la barra superior, encontramos la sección de búsqueda. Esta cuenta con una barra de búsqueda por nombre y un menú seleccionador de filtros, entre los que encontramos:

- Filtros por campus o zona de la habitación.

- Precios máximos y mínimos por habitación.
- Opciones de accesibilidad como ascensores o transporte público cercano.
- Filtro por carreras, seleccionando habitaciones en edificios donde se hospedan estudiantes de la carrera del usuario. Los filtros seleccionados se muestran a la derecha, en píldoras, con un botón para eliminar el filtro.

El resto de la interfaz contiene las ofertas resultantes tras aplicar los filtros seleccionados. Estas habitaciones se muestran con fotos, una breve descripción de la oferta y un "iframe" con la ubicación de la habitación en un mapa. Al hacer clic en una de estas ofertas, se abrirá una vista más detallada de la respectiva habitación, la Interfaz detallada.

## Casos de uso

Esta es una lista de los casos de uso desarrollados en esta interfaz:

- Buscar oferta.
- Filtrar ofertas mediante los filtros mencionados.
- Ajustes de cuenta
- Cerrar sesión
- Visualizar notificaciones

### 6.2.3 Interfaz Detallada:

La interfaz detallada ofrece toda la información relativa a una habitación ofertada. Es desde esta interfaz desde la que un estudiante, en caso de interesarse por una habitación, puede ponerse en contacto con el propietario.

The screenshot shows a web-based room detail interface. At the top is a red header bar with the text "Habitación 1, bloque x". Below the header is a small thumbnail image of a room interior. To the right of the image is a horizontal ellipsis (...). The main content area begins with a bold heading "Bloque de habitaciones 1 - Pisos disponibles 17/50". Underneath this, there is a descriptive paragraph about the room's features, followed by another paragraph about external services like laundry and gyms. Further down are details about monthly cost, additional costs, surface area, and other relevant statistics. To the right of the text is a map of a city street with a callout box indicating the address "c/ calle de ejemplo, 1235, Madrid". At the bottom of the interface is a dark button labeled "Mandar mensaje".

#### Descripción

Esta interfaz se puede dividir en tres apartados principales:

1. **Carrusel de imágenes:** El primer apartado es un carrusel con fotografías de la habitación, el edificio donde se encuentra y sus instalaciones.
2. **Descripción detallada:** La segunda sección contiene una serie de párrafos donde se describe en detalle la oferta, incluyendo tanto las características físicas de la habitación y los utensilios o electrodomésticos con los que está equipada, como los servicios externos disponibles. Entre estos servicios se encuentran la lavandería o un servicio de cantina para estudiantes entre

semana. También se incluye un mapa para ver la ubicación precisa del edificio y su cercanía al campus.

3. **Precio y contacto:** Por último, en la parte inferior se encuentra la sección con el precio del alquiler, que ha sido previamente verificado y ajustado por la universidad, y un método de contacto con el propietario de la habitación. Este último puede recibir un mensaje con nuestro interés al hacer clic en el botón en la parte inferior de la página.

## Casos de Uso

Esta es una lista de los casos de uso desarrollados en esta interfaz:

- Visualizar detalles de una oferta.
- Contactar con el propietario de la habitación.
- Suscribirse a una alerta para la habitación

### 6.2.4 Interfaz Propietario:

La interfaz de propietario es la que acceden los propietarios una vez han sido verificados al iniciar sesión. Su objetivo principal es la gestión y visualización de las habitaciones que el propietario tiene ofertadas en la aplicación.

The screenshot shows a web-based application interface for property owners. At the top, there is a red header bar with a logo, a search bar, and a button to add a new property. Below the header is a table titled "Mis Habitaciones" (My Apartments) with the following columns: Ocupancia (Occupancy), Estudiante (Student), and Tiemp restante alquiler (Remaining lease time). The table lists several apartments with their status (e.g., Ocupado, Libre, En trámite), student email, and lease duration. The rows alternate in color between white and light pink.

Mis Habitaciones	Ocupancia	Estudiante	Tiemp restante alquiler
Habitacion 441, bloque ejemplo xx	■ Ocupado	ejemplo@ucm.es	14 meses y 27 días
Habitacion 445, bloque ejemplo xx	■ Ocupado	ejemplo@ucm.es	14 meses y 27 días
Habitacion 452, bloque ejemplo xx	■ En trámite	ejemplo@ucm.es	-
Habitacion 23, bloque ejemplo otro x	■ Libre	-	-
Habitacion 15, bloque ejemplo otro x	■ Libre	-	-
Habitacion 18, bloque ejemplo otro x	■ Ocupado	ejemplo@ucm.es	14 meses y 27 días

## Detalles

Esta interfaz cuenta con el mismo menú superior que la interfaz principal de los estudiantes, lo que permite acceder a las notificaciones entrantes, cambiar los ajustes de la cuenta y cerrar sesión. La parte principal de la interfaz tiene un formato de *Dashboard*, que muestra una tabla con todas las habitaciones que el propietario tiene ofertadas en la aplicación. También se incluyen opciones de filtro para seleccionar qué información se muestra en cada fila de la tabla, así como la posibilidad de ordenar las habitaciones según diferentes criterios. Para cada habitación se muestra su nombre y su estado de ocupación. Las habitaciones pueden estar:

- **Libres:** Si nadie está alquilándola actualmente.
- **Ocupadas:** Si un estudiante está residiendo en ella.
- **En trámite:** Si la habitación está en proceso de ser alquilada. Si el propietario selecciona cualquiera de las filas, se mostrará una vista detallada. Los propietarios también pueden eliminar, modificar o añadir nuevas ofertas desde esta interfaz.

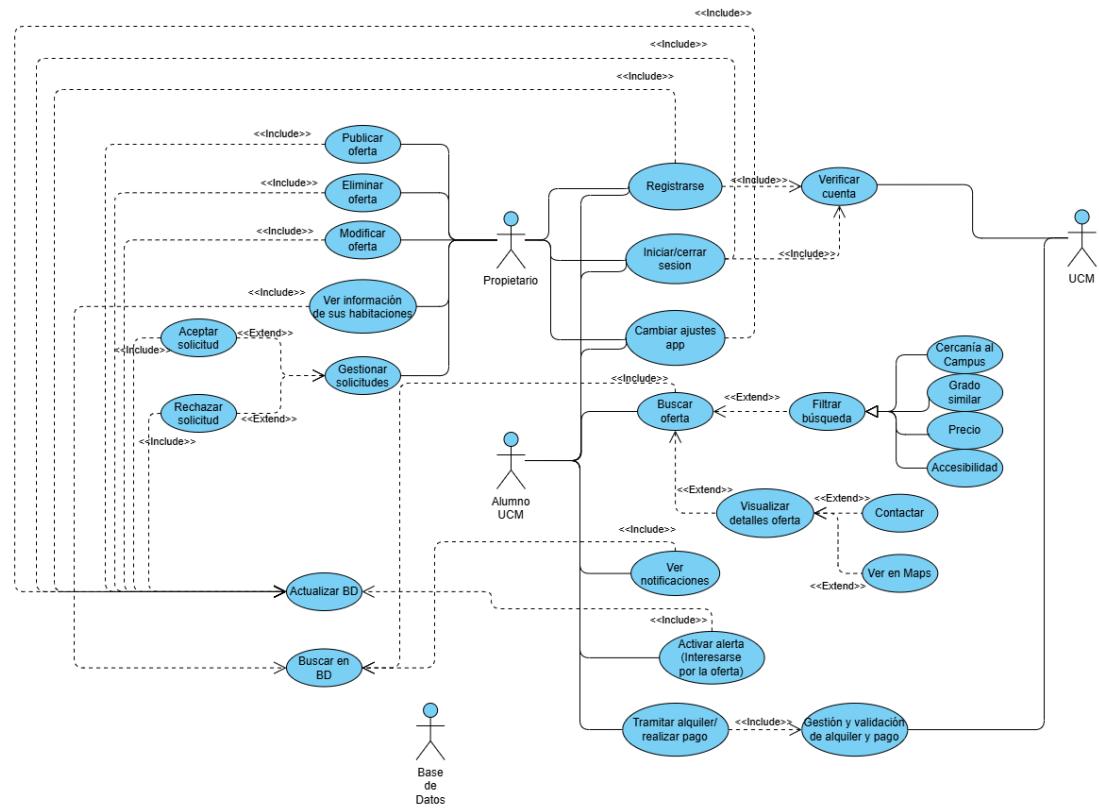
## Casos de Uso

Esta es una lista de los casos de uso desarrollados en esta interfaz

- Ver información de las habitaciones y los residentes.
- Añadir o modificar una oferta.
- Gestionar solicitudes mediante las notificaciones.

## 6.3 Casos de uso:

Para visualizar correctamente todas las funcionalidades de recoge nuestro proyecto, hemos modelado el siguiente diagrama de casos de uso. En él agrupamos los actores que forman parte de la plataforma y las relaciones entre ellos, así como sus dependencias. Más abajo se puede ver cada caso de uso detallado y especificado junto al diagrama de actividad correspondiente.



### 6.3.1 Especificación del Caso de Uso : Buscar y solicitar oferta:

Identificador: CU-01

Nombre: Buscar y Solicitar Oferta de Alojamiento

Actores:

- Actor Principal: Alumno UCM.
- Actor Secundario: Sistema (Base de Datos).

**Descripción:** El alumno busca alojamiento utilizando filtros específicos, consulta los

detalles de una vivienda y contacta con el ofertante para mostrar interés.

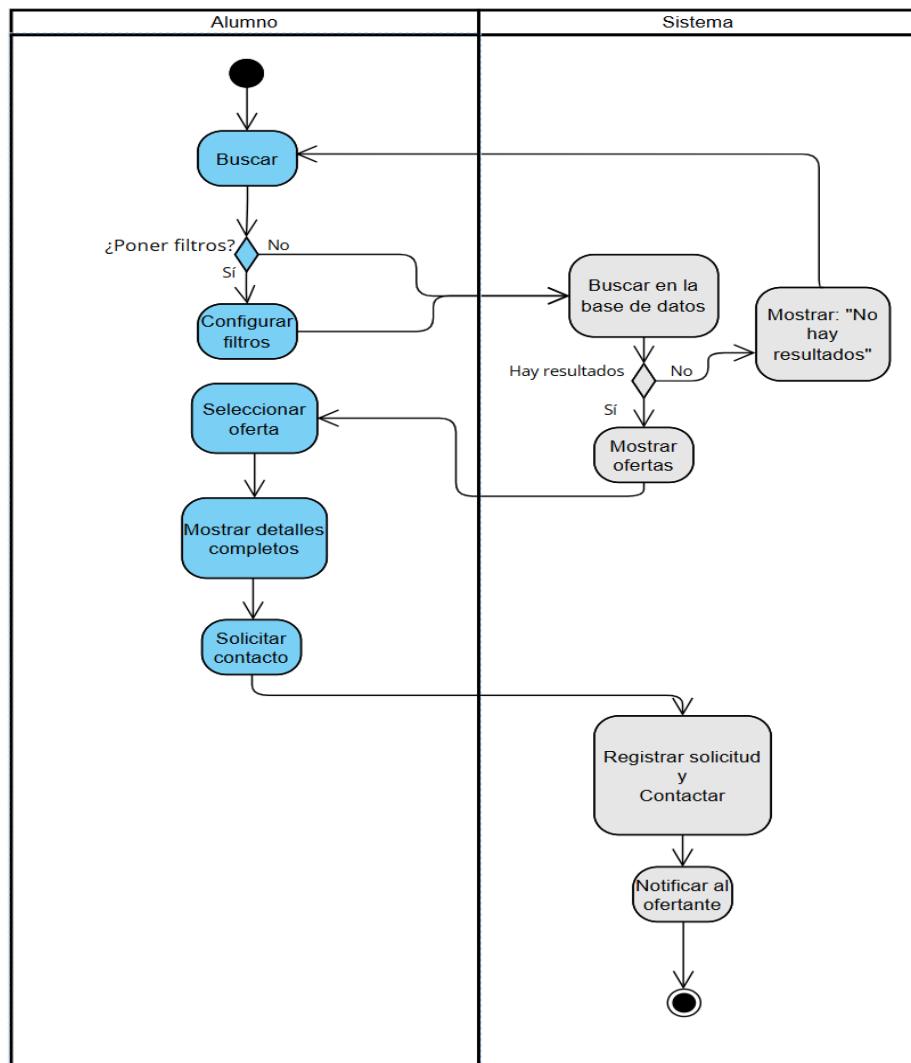
**Precondiciones:**

El usuario necesita haber iniciado sesión.

**Postcondiciones:**

La oferta queda marcada como "Con solicitud pendiente" para ese usuario.

El Ofertante tiene una nueva notificación en su dashboard.



## 6.3.2 Especificación del Caso de Uso : Filtrar búsqueda:

Identificador: CU-02

Nombre: Configurar filtros de búsqueda

Actores:

- Actor Principal: Alumno UCM.
- Actor Secundario: Sistema (Base de Datos).

Descripción: El alumno configura las opciones de filtrado en la búsqueda de habitaciones, pudiendo filtrar por cercanía a un campus seleccionado, por similitud del grado universitario de los residentes al grado seleccionado, por precio y por opciones de accesibilidad.

Precondiciones:

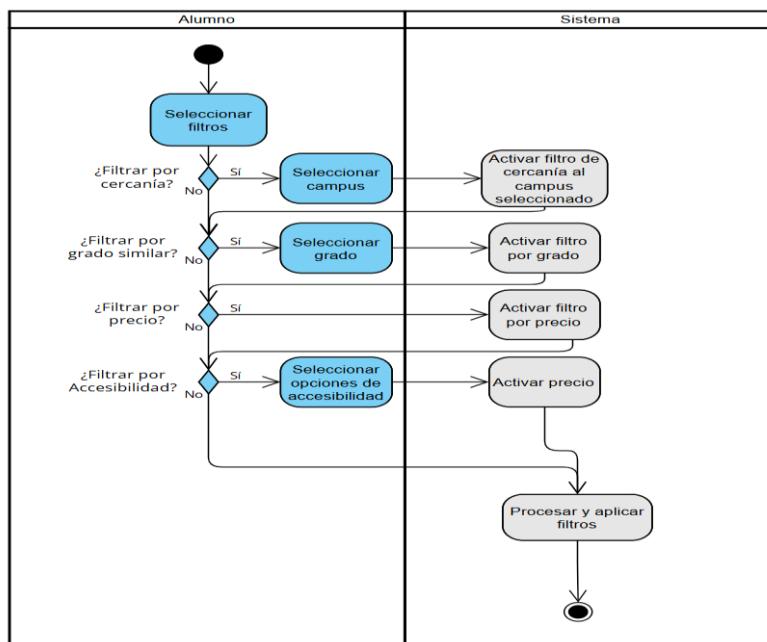
El usuario necesita haber iniciado sesión.

El usuario debe haber comenzado la búsqueda de habitaciones.

Postcondiciones:

Los filtros configurados deben estar procesados y aplicados correctamente.

Se debe proceder con la ejecución de la búsqueda.



### 6.3.3 Especificación del Caso de Uso : Añadir oferta de Habitación

Identificador: CU-03

Nombre: Añadir nueva oferta de Habitación.

Actores:

- Actor Principal: Propietario de la Habitación.
- Actor Secundario: Sistema (Base de Datos).

Descripción: El propietario tiene una habitación que quiere subir a la plataforma y añade una oferta para esta misma a la aplicación. La nueva habitación se mostrará para todos los estudiantes en la aplicación.

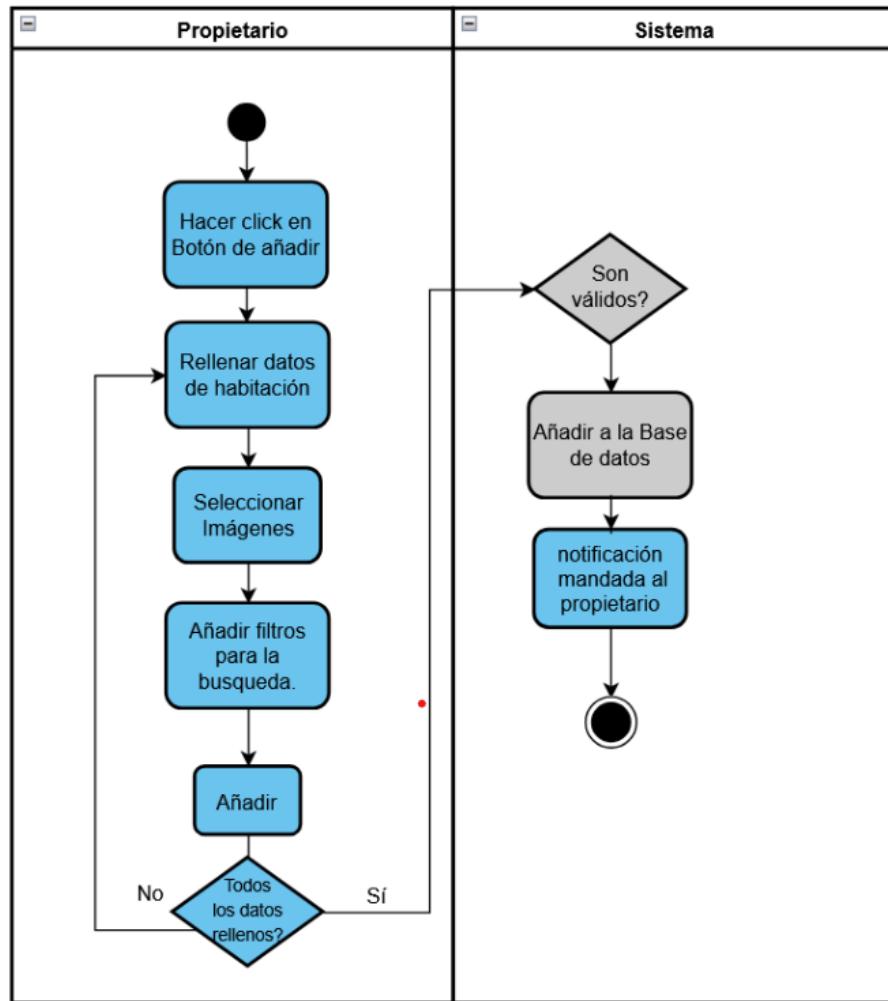
Precondiciones:

El usuario necesita haber iniciado sesión.

Postcondiciones:

El propietario tiene una nueva habitación ofertada bajo su perfil.

El propietario recibe una notificación indicando si se ha añadido la habitación a la BD con éxito.



### 6.3.4 Especificación del Caso de Uso : Eliminar habitación

Identificador: CU-04

Nombre: Añadir nueva oferta de Habitación.

Actores:

- Actor Principal: Propietario de la Habitación.
- Actor Secundario: Sistema (Base de Datos).

Descripción: El propietario selecciona una de sus habitaciones subidas a la aplicación y la selecciona para eliminarla de la Base de Datos. La oferta de la habitación deja de estar disponible para los estudiantes de la plataforma.

Precondiciones:

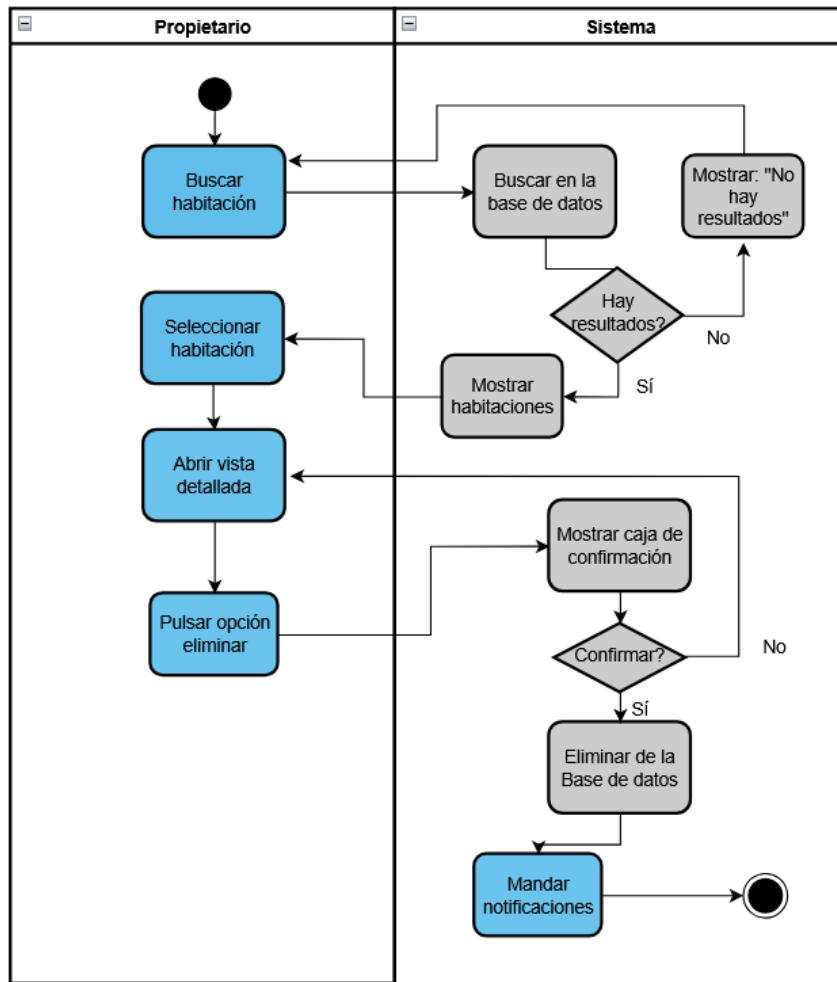
El usuario necesita haber iniciado sesión.

Postcondiciones:

El propietario elimina la habitación seleccionada de su perfil.

El propietario recibe una notificación indicando que se ha eliminado con éxito la habitación

Los alumnos interesados en la habitación reciben una notificación indicando que ya no está disponible.



## 6.3.5 Especificación del Caso de Uso : Ver Información de las Habitaciones.

Identificador: CU-05

Nombre: Ver información de las habitaciones (Propietario).

Actores:

- Actor Principal: Propietario de la Habitación.
- Actor Secundario: Sistema (Base de Datos).

Descripción: El Propietario accede a un menú específico y conectado a la Base de Datos en el que se muestran sus propiedades (Habitaciones). Para cada una de ellas se despliegan los datos pertinentes.

Precondiciones:

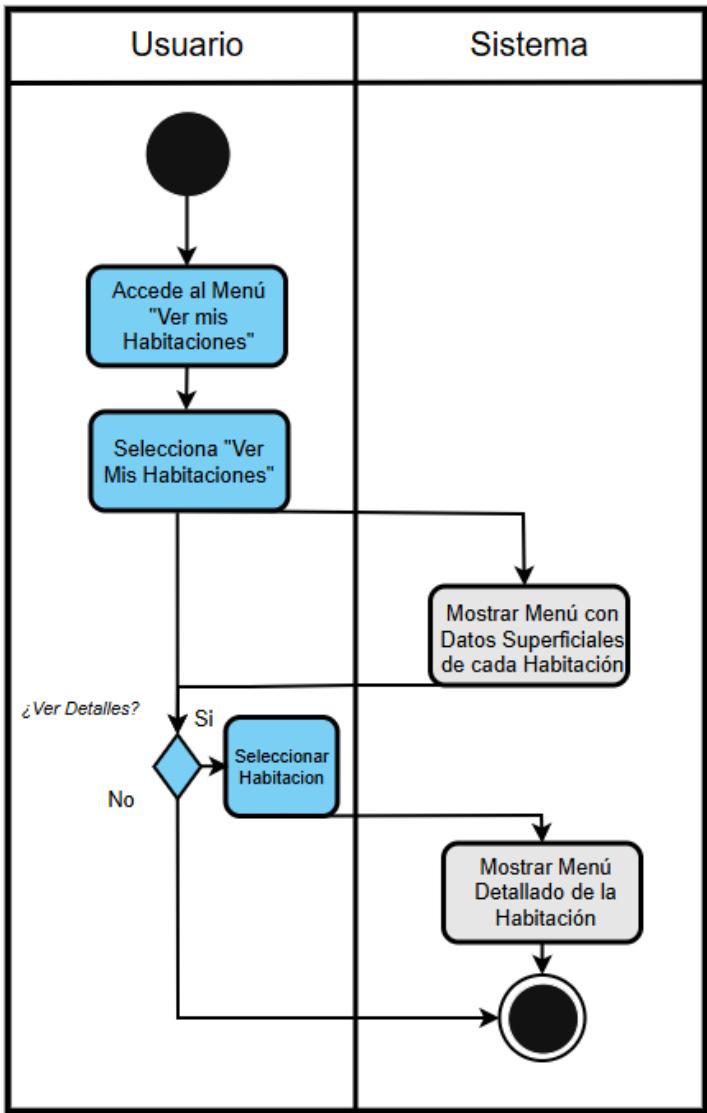
El Propietario debe haber iniciado sesión.

Las Habitaciones deben estar registradas en la base de datos.

Postcondiciones:

Las Habitaciones quedan cargadas y mostrando datos importantes (disponibilidad, estudiante/s ocupándola, link a la descarga del contrato... )

Diagrama actividad:



### 6.3.6 Especificación del Caso de Uso : Inicio de Sesión.

Identificador: CU-06

Nombre: Iniciar Sesión.

Actores:

- Actor Principal: Usuario del Sistema (Administrador, Estudiante o Arrendador).
- Actor Secundario: Sistema (Base de Datos).

**Descripción:** Un Usuario accede a la página principal de inicio de sesión. Se contrasta los datos con la Base de Datos y en caso de éxito se otorga acceso a la plataforma.

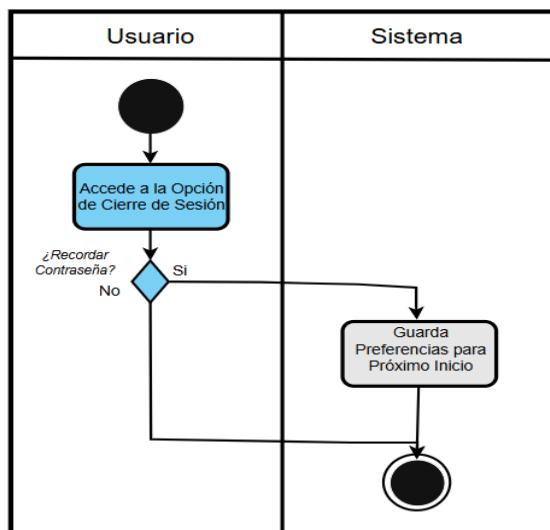
**Precondiciones:**

El Usuario debe tener una cuenta.

**Postcondiciones:**

La página queda accesible en caso de que haya iniciado sesión.

**Diagrama actividad:**



### 6.3.7 Especificación del Caso de Uso : Cierre de Sesión.

Identificador: CU-07

Nombre: Cerrar sesión

Actores:

- Actor Principal: Usuario del Sistema (Administrador, Estudiante o Arrendador).
- Actor Secundario: Sistema (Base de Datos).

**Descripción:** Un Usuario accede a la opción de cierre de sesión. Se pregunta si quiere recordar la contraseña para un inicio de sesión futuro. En todo caso se sale de la página.

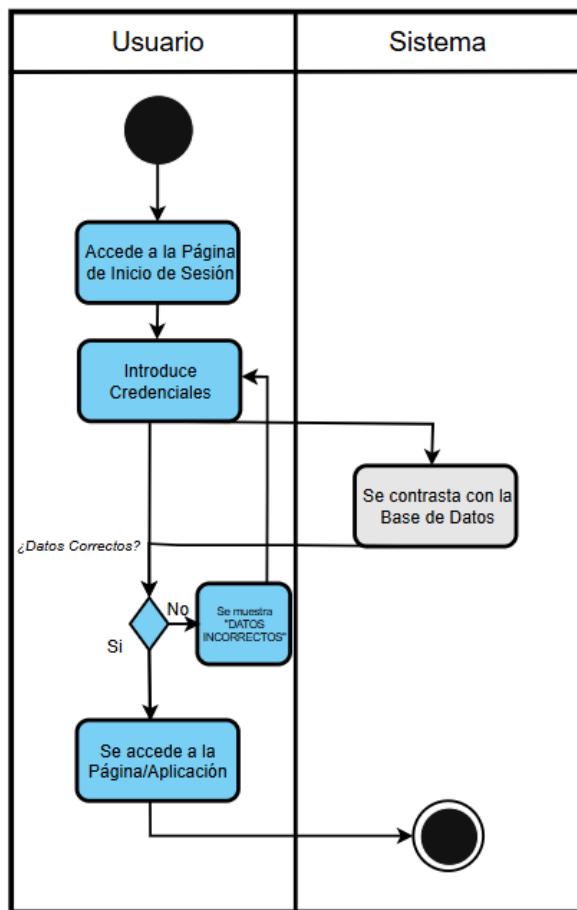
**Precondiciones:**

El Usuario debe haber iniciado sesión.

**Postcondiciones:**

El Usuario sale de la página y la sesión queda correctamente cerrada.

**Diagrama actividad:**



## 6.3.8 Ver mensajes de solicitudes-notificaciones (Alumno)

Identificador: CU-08

Nombre: Ver mensajes de solicitudes-notificaciones (Alumno).

Actores:

- Actor Principal: Alumno UCM.
- Actor Secundario: Sistema (Base de Datos).

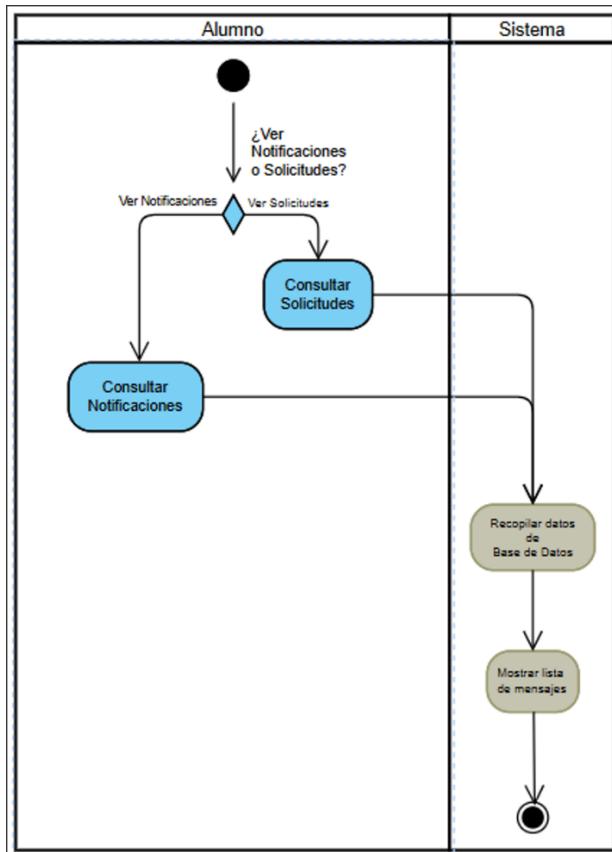
Descripción: El alumno consulta los mensajes de solicitudes propios.

Precondiciones:

El usuario necesita haber iniciado sesión.

Postcondiciones:

Se le muestra la lista de solicitudes al usuario.



### 6.3.9. Modificar oferta vigente (Propietario).

Identificador: CU-09

Nombre: Modificar oferta vigente (Propietario).

Actores:

- Actor Principal: Propietario.
- Actor Secundario: Sistema (Base de Datos).

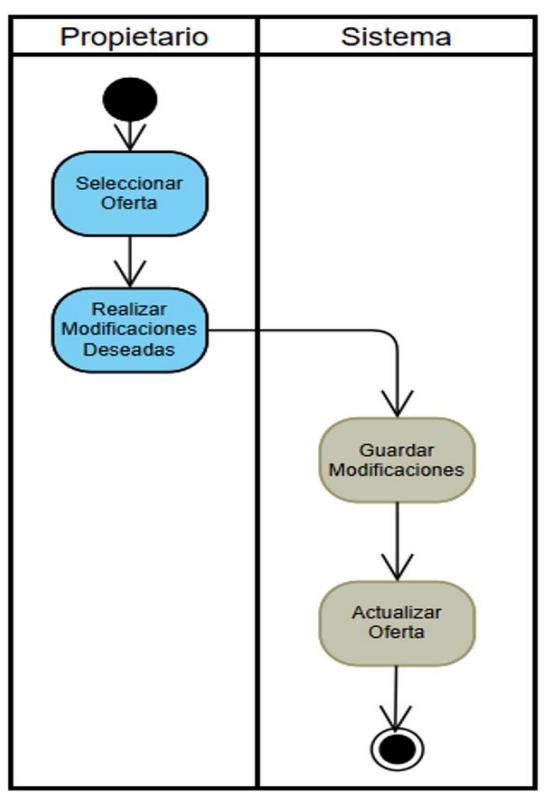
Descripción: El alumno consulta los mensajes de solicitudes propios.

Precondiciones:

- El usuario necesita haber iniciado sesión.
- El usuario debe tener al menos una oferta vigente

Postcondiciones:

- Se modifica la oferta vigente.



### 6.3.10 Aceptar solicitud estudiante y comenzar conversación (Propietario)

Identificador: CU-10

Nombre: Aceptar solicitud estudiante y comenzar conversación (Propietario)

Actores:

- Actor Principal: Propietario.
- Actor Secundario: Sistema (Base de Datos).

Descripción: El alumno consulta los mensajes de solicitudes propios.

Precondiciones:

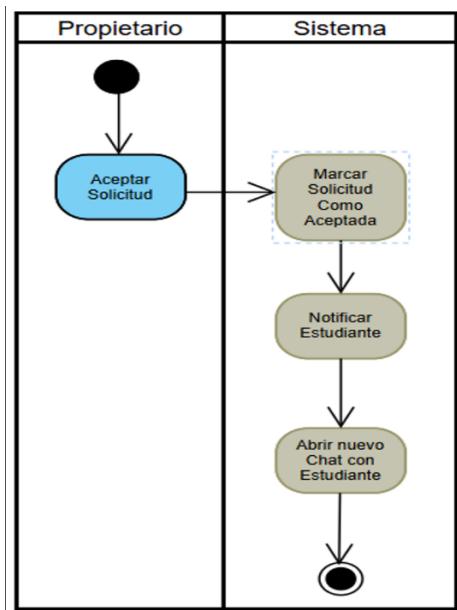
El usuario necesita haber iniciado sesión.

El usuario debe tener una solicitud vigente.

Postcondiciones:

La solicitud queda como aceptada.

El estudiante es informado de la aceptación.



### 6.3.11 Rechazar estudiante (Propietario)

Identificador: CU-11

Nombre: Rechazar estudiante (Propietario).

Actores:

- Actor Principal: Propietario.
- Actor Secundario: Sistema (Base de Datos).

Descripción: El alumno consulta los mensajes de solicitudes propios.

Precondiciones:

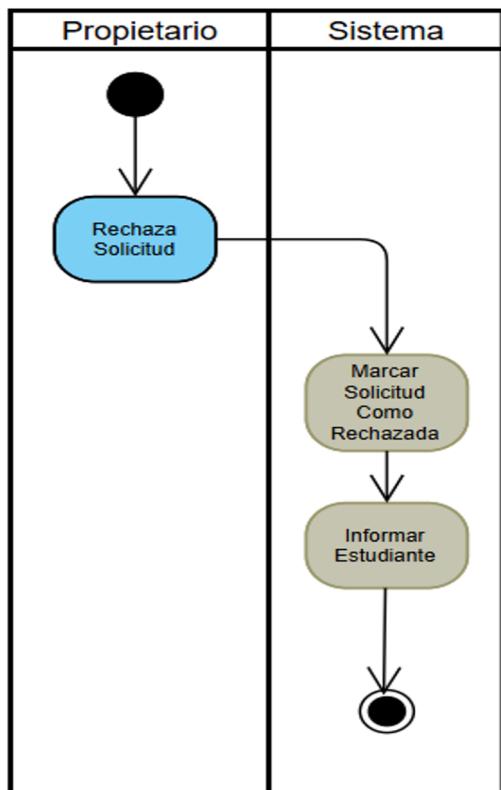
El usuario necesita haber iniciado sesión.

El usuario debe tener una solicitud vigente

Postcondiciones:

La solicitud queda como rechazada.

El estudiante es informado del rechazo.



### 6.3.12. Especificación del Caso de Uso : Registrarse.

Identificador: CU-12

Nombre: Registro de Usuario en la Plataforma

Actores:

- Actor Principal: Alumno UCM.
- Actor Secundario: Sistema (Base de Datos).

Descripción: El alumno accede al formulario de registro y proporciona sus datos personales y credenciales iniciales. El sistema valida la información introducida, comprueba si el correo cumple los requisitos establecidos (correo institucional UCM en el caso de estudiantes) y almacena los datos en la base de datos. Una vez registrado, se envía un correo de verificación al usuario y su cuenta queda pendiente de activación según su rol.

Precondiciones:

El alumno no debe existir previamente en el sistema.

Postcondiciones:

El alumno queda registrado en la base de datos con su rol correspondiente.

Se genera y envía un correo de validación al usuario para completar el proceso.

En caso de ser una inmobiliaria/propietario, su cuenta queda en estado "Pendiente de aprobación".



### 6.3.13 Especificación del Caso de Uso : Interesarse por Oferta.

Identificador: CU-13

Nombre: Interesarse por una Oferta de Alojamiento

Actores:

- Actor Principal: Alumno UCM.
- Actor Secundario: Sistema (Base de Datos).

Descripción: El alumno, mientras visualiza la ficha de una habitación que actualmente no permite enviar una solicitud directa (por encontrarse en trámite o temporalmente no disponible), puede indicar su interés activando una suscripción de alerta. El sistema registra dicho interés en la base de datos y monitoriza el estado de la habitación. Cuando la oferta vuelva a estar disponible, el sistema enviará automáticamente una notificación al alumno para informarle de que ya puede solicitarla.

Precondiciones:

El alumno debe haber iniciado sesión.

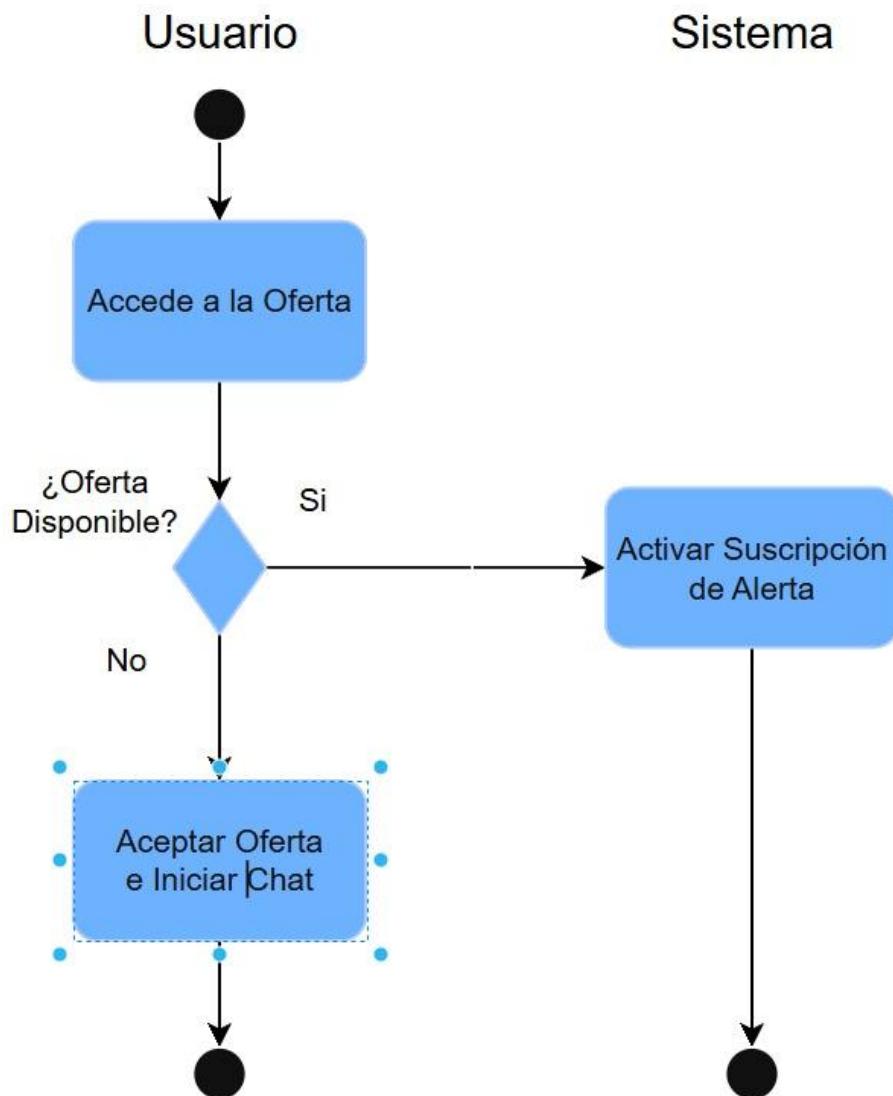
La oferta debe encontrarse no disponible para solicitud, pero visible en el sistema.

Postcondiciones:

La habitación queda asociada al alumno como “Interés activado”.

El sistema registra una suscripción de alerta para esa habitación.

El alumno recibirá una notificación automática cuando la oferta pase a estado disponible.



## 6.3.14 Especificación del Caso de uso: Cambiar Ajustes de la App

Identificador: CU-14

Nombre: Cambiar ajustes de la aplicación

Actores:

Actor Principal: Alumno UCM.

Actor Secundario: Sistema

Descripción:

El alumno accede al menú de ajustes de su perfil dentro de la aplicación y modifica parámetros de configuración relacionados con su experiencia de uso, como preferencias de notificaciones, idioma, accesibilidad o privacidad. El sistema valida los cambios realizados y actualiza la información correspondiente en la base de datos.

Precondiciones:

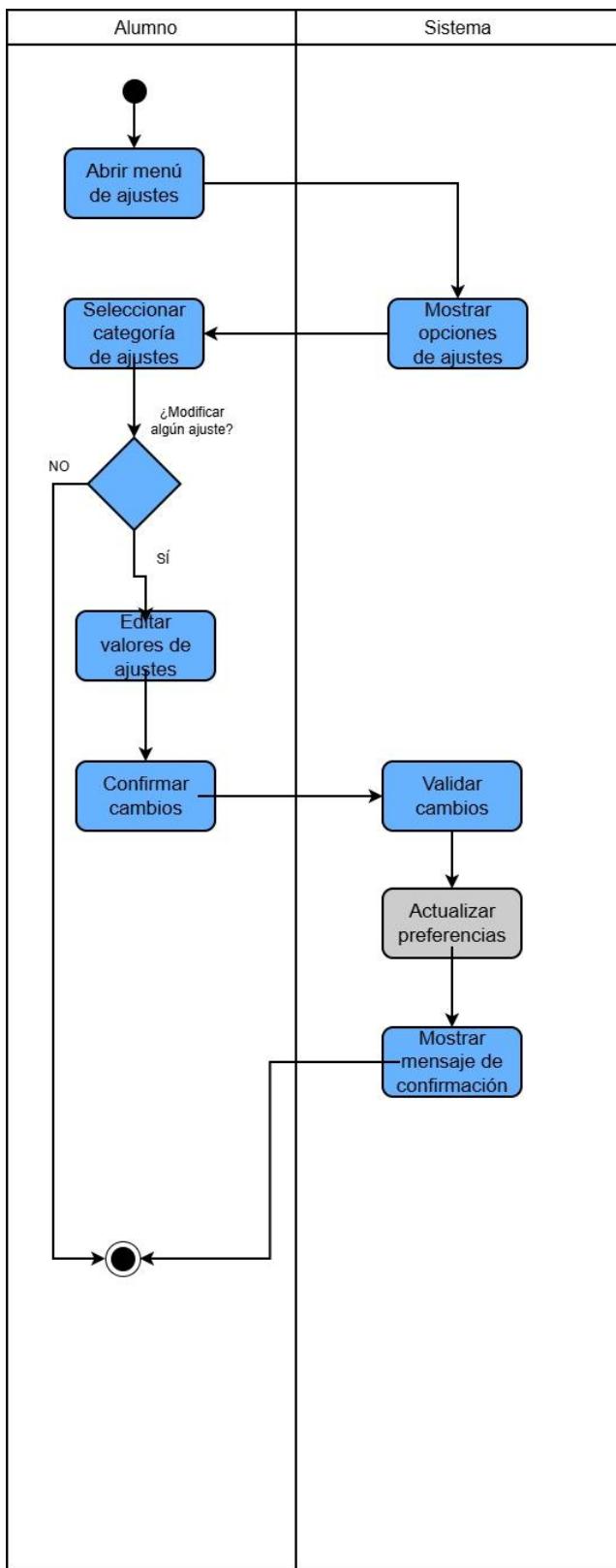
El usuario debe haber iniciado sesión.

El menú de ajustes debe estar disponible en la aplicación.

Postcondiciones:

Las preferencias del usuario quedan actualizadas correctamente en la base de datos.

El sistema aplicará los nuevos ajustes en las próximas interacciones del usuario.



## 6.3.15 Especificación del Caso de uso: Abrir localización de habitación en Maps

Identificador: CU-15

Nombre: Visualizar habitación en Maps

Actores:

Actor Principal: Alumno UCM.

Actor Secundario: Sistema (Servicios externos de mapas: Google Maps, OpenStreetMaps, etc.)

Descripción:

El alumno consulta la ubicación de una habitación seleccionada y solicita abrirla en un visor de mapas externo o integrado mediante iframe. El sistema recupera las coordenadas asociadas a la oferta y carga el mapa correspondiente.

Precondiciones:

El usuario debe haber iniciado sesión.

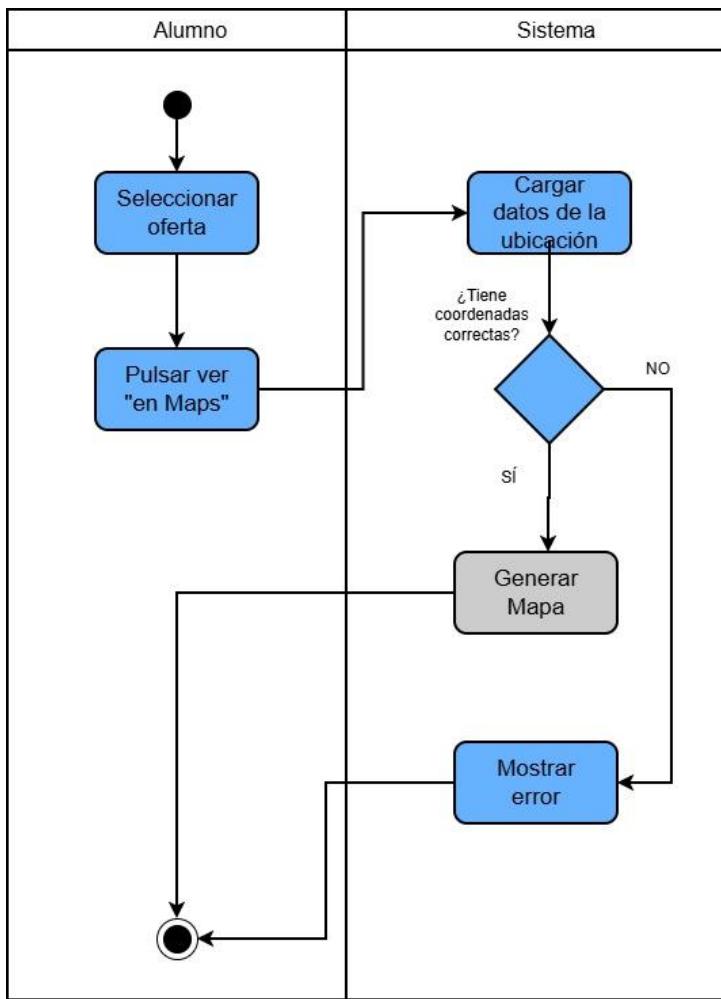
El usuario debe haber seleccionado una oferta concreta.

La oferta debe contener coordenadas válidas o una dirección interpretable.

Postcondiciones:

Se muestra al usuario un mapa con la ubicación exacta de la habitación.

El usuario puede usar el mapa para navegar o calcular rutas externas.



### 6.3.16 Especificación del caso de Uso: Añadir oferta a una lista de favoritos

Identificador: CU-16

Nombre: Añadir oferta a una lista de favoritos

Actores:

- Actor Principal: Alumno UCM.
- Actor Secundario: Sistema (Base de Datos).

**Descripción:** El alumno UCM añade una oferta de alojamiento a su lista personal de favoritos para poder acceder a ella de forma rápida en futuras sesiones

**Precondiciones:**

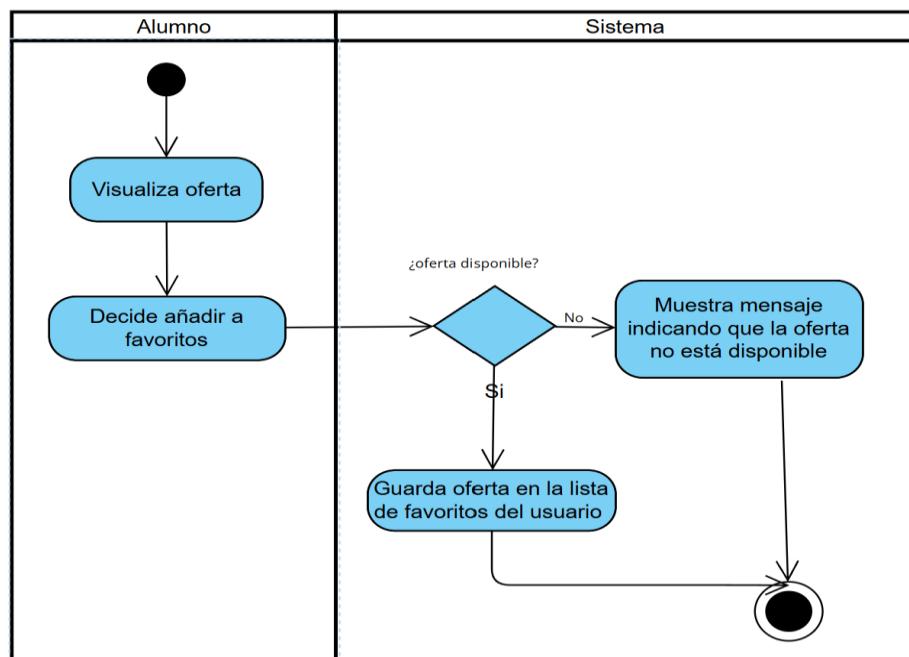
El usuario debe haber iniciado sesión en la app

El usuario debe estar mirando una oferta de alojamiento y que sea válida y disponible.

**Postcondiciones:**

La oferta seleccionada queda registrada en la lista de favoritos del alumno en la base de datos

El sistema actualiza la interfaz para reflejar visualmente que la oferta ha sido marcada como favorita.



### 6.3.17 Especificación del Caso de uso: Solicitar quedada con el propietario para ver la oferta

Identificador: CU-17

Nombre: Solicitar quedada con el propietario para ver la oferta

Actores:

Actor Principal: Alumno UCM.

Actor Secundario: Sistema (Base de Datos).

Descripción:

El alumno UCM envía una solicitud al propietario de una oferta de alojamiento para concertar una cita presencial o virtual y visualizar la vivienda. La solicitud incluye una fecha y hora propuestas, así como un mensaje opcional.

Precondiciones:

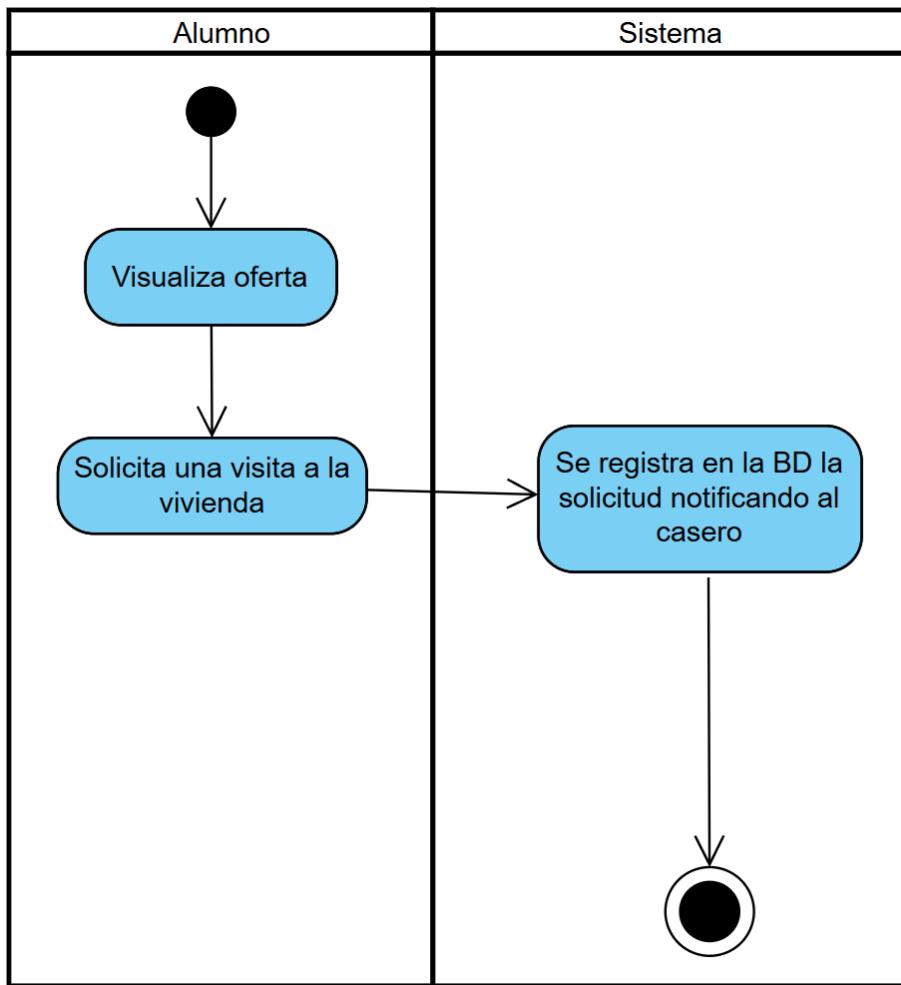
El usuario debe estar visualizando una oferta de alojamiento válida y disponible.

El propietario de la oferta debe tener su perfil activo y accesible.

Postcondiciones:

La solicitud de quedada queda registrada en la base de datos, asociada al alumno, al propietario y a la oferta.

El sistema notifica al propietario de la nueva solicitud (por ejemplo, mediante mensaje interno o alerta en la plataforma).



### 6.3.18 Especificación del Caso de Uso: Reportar Propietario por Incidencia

Identificador: CU-18

Nombre: Reportar propietario por Incidencia

Actores:

Actor Principal: Alumno UCM.

Actores Secundarios: Propietario, Sistema(Base de Datos).

Descripción: El Alumno observa una incidencia en su trato con el propietario. Prácticas abusivas, una oferta que no se asemeja a la realidad de la habitación o

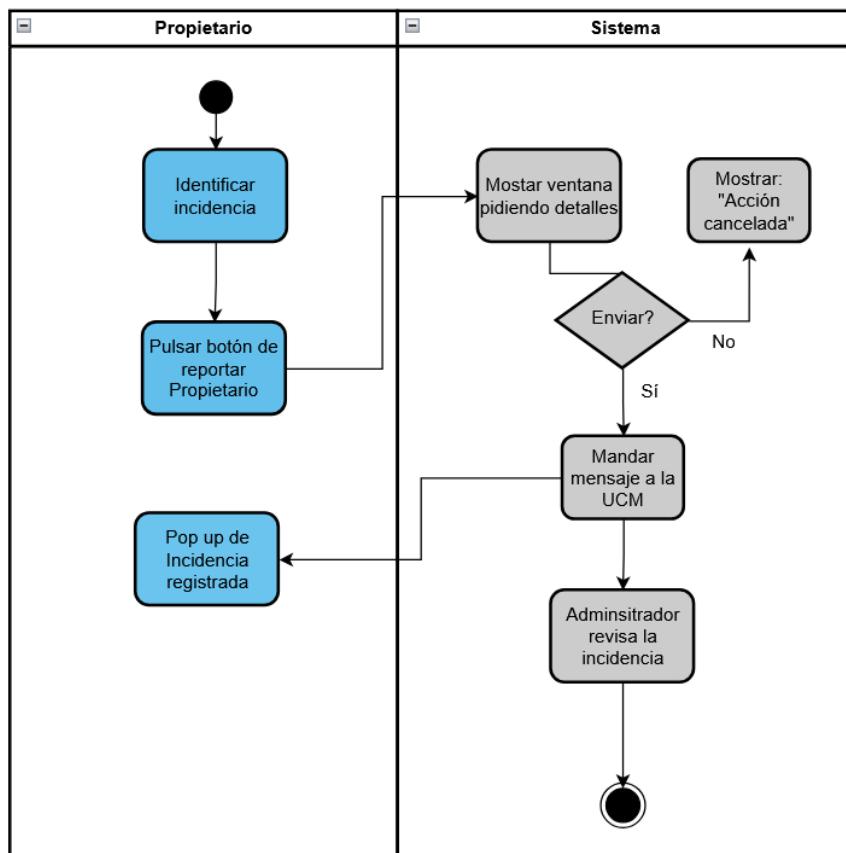
un mal trato personal. El usuario tiene la posibilidad de notificar esta incidencia a la UCM, y estos la tratarán de manera anónima con el Propietario.

Precondiciones:

El usuario necesita haber iniciado sesión Debe haber una incidencia real a reportar.

Postcondiciones:

El propietario será enviado a la UCM para tratar la incidencia. El usuario será notificado de que se ha enviado el mensaje correctamente.



# 7. Plan de Trabajo

En este proyecto, los estudiantes se ponen de acuerdo con el representante de la universidad, en este caso Borja Manero Iglesias, y en un futuro cuando el proyecto esté más desarrollado, se presentará al rectorado de la UCM. Para poder definir el siguiente plan de trabajo y así desarrollar la aplicación, con el fin de solucionar un problema real de los estudiantes: los altos precios de las habitaciones cerca de la universidad.

A continuación, enumeramos todas las fases del proyecto con sus respectivas tareas, una breve descripción, sus duraciones y fechas orientativas, además de las personas que se encargarán de ellas provisionalmente. Todas ellas con el consenso del representante de la universidad: Borja Manero Iglesias.

## 7.1 Fases del proyecto

### 7.1.1 Análisis:

Creación del documento de diseño con todas las siguientes tareas (1 - 8), para poder cumplir con la tarea número 9.

<p><b>1. Gestión de requisitos:</b> Definición, clasificación y documentación de requisitos funcionales, no funcionales, incluyendo su priorización.</p> <p><b>-Encargados:</b> Axel Metodiev Ojeda y Senbo Zhou Pan. <b>-Duración:</b> 5 días <b>-Fechas:</b> 6/11/2025 - 12/11/2025</p>	<p><b>2. Definición del alcance:</b> Establecimiento de los límites del sistema, las funcionalidades incluidas, las excluidas y las restricciones principales de la aplicación y de los usuarios que la usarán.</p> <p><b>-Encargados:</b> Ramiro Alejandro Gallegos Cuji y Diego Razvan Cozma. <b>-Duración:</b> 5 días <b>-Fechas:</b> 6/11/2025 - 12/11/2025</p>
<p><b>3. Búsqueda de competencias y/o proyectos similares:</b> Revisión de sistemas similares, estándares y tecnologías disponibles para identificar</p>	<p><b>4. Modelado del sistema:</b> Representación inicial del funcionamiento del sistema mediante casos de uso, diagramas de actividades y modelos conceptuales que</p>

<p>referencias y oportunidades de inspiración o mejora.</p> <p><b>-Encargado:</b> Ramiro Alejandro Gallegos Cuji  <b>-Duración:</b> 5 días  <b>-Fechas:</b> 6/11/2025 - 12/11/2025</p>	<p>estructuran la aplicación.</p> <p><b>Encargado:</b> Sergio Viñas González.  <b>-Duración:</b> 7 días  <b>-Fechas:</b> 13/11/2025 - 21/11/2025</p>
<p><b>5. Análisis de viabilidad:</b>  Evaluación técnica, económica, operativa y legal para determinar la posibilidad real del proyecto y las condiciones necesarias para su desarrollo.</p> <p><b>-Encargados:</b> Miguel Cruz Gómez y Ramiro Alejandro Gallegos Cuji.  <b>-Duración:</b> 7 días  <b>-Fechas:</b> 13/11/2025 - 21/11/2025</p>	<p><b>6. Identificación y análisis de riesgos:</b>  Detección de riesgos potenciales, estimación de impacto en los estudiantes, y definición inicial de estrategias de mitigación.</p> <p><b>-Encargado:</b> Héctor Manuel Díaz Bernal  <b>-Duración:</b> 7 días  <b>-Fechas:</b> 13/11/2025 - 21/11/2025</p>
<p><b>7. Requisitos de calidad y criterios de aceptación:</b>  Determinación de los atributos de calidad del sistema (rendimiento, seguridad, mantenibilidad, etc.) y criterios para validar su cumplimiento por la UCM.</p> <p><b>-Encargado:</b> Héctor Manuel Díaz Bernal  <b>-Duración:</b> 7 días  <b>-Fechas:</b> 13/11/2025 - 21/11/2025</p>	<p><b>8. Estrategia provisional de desarrollo:</b>  Definición del enfoque general del desarrollo, posibles tecnologías, modelo de proceso, restricciones técnicas y metodología inicial.</p> <p><b>-Encargado:</b> Miguel Cruz Gómez  <b>-Duración:</b> 2 días  <b>-Fechas:</b> 24/11/2025 - 25/11/2025</p>
<p><b>9. Validación y cierre del análisis:</b>  Revisión conjunta con Borja Manero para confirmar que el análisis es correcto, coherente y completo antes de pasar a diseño.</p> <p><b>-Encargados:</b> (todos)  <b>-Duración:</b> 1 día  <b>-Fecha:</b> 11/12/2025</p>	

## 7.1.2 Diseño

Creación del documento de diseño con todas las siguientes tareas (1 - 5), para poder cumplir con la tarea número 6.

<p><b>1. Diseño de módulos y componentes:</b> Definir los módulos principales: gestión de anuncios, gestión de usuarios, sistema de mensajería, favoritos, filtrado/búsqueda, gestión de fotos, etc., junto con sus interfaces internas.</p> <p>-<b>Encargados:</b> Diego Razvan Cozma y Hector Castro Portillo -<b>Duración:</b> 10 días -<b>Fecha:</b> 24/11/2025 - 5/12/2025</p>	<p><b>2. Diseño de datos:</b> Modelar entidades como Usuario, Anuncio, Habitación/Piso, Imagen, Mensaje, Favorito, Reserva/Contacto, incluyendo esquemas de bases de datos y restricciones.</p> <p>-<b>Encargados:</b> Ramiro Alejandro Gallegos Cuji y Hugo Lozano Ruiz -<b>Duración:</b> 10 días -<b>Fecha:</b> 24/11/2025 - 5/12/2025</p>
<p><b>3. Diseño de requisitos no funcionales:</b> Definir medidas de rendimiento, seguridad (protección de datos personales), escalabilidad (almacenamiento de imágenes, carga del feed), disponibilidad y mantenibilidad.</p> <p>-<b>Encargados:</b> Héctor Manuel Díaz Bernal y Axel Metodiev Ojeda -<b>Duración:</b> 10 días -<b>Fecha:</b> 24/11/2025 - 5/12/2025</p>	<p><b>4. Diseño de la arquitectura del sistema:</b> Definir cómo se estructura la aplicación: arquitectura cliente-servidor, servicios, base de datos, almacenamiento de imágenes, sistemas de búsqueda y notificaciones.</p> <p>-<b>Encargados:</b> Sergio Viñas González y Miguel Cruz Gómez -<b>Duración:</b> 5 días -<b>Fecha:</b> 24/11/2025 - 28/11/2025</p>
<p><b>5. Diseño de interfaces y experiencia de usuario (UI/UX):</b> Diseñar las pantallas clave: inicio, búsqueda, detalles del piso, chat, perfil del arrendador, publicar habitación, etc. Definir las APIs necesarias para que la aplicación se comunique con el backend.</p> <p>-<b>Encargados:</b> Sergio Viñas González y Miguel Cruz Gómez -<b>Duración:</b> 8 días -<b>Fecha:</b> 1/12/2025 - 10/12/2025</p>	<p><b>6. Revisión de documentación del diseño:</b> Revisar el documento de diseño con el equipo y con Borja Manero para confirmar coherencia antes de comenzar el desarrollo.</p> <p>-<b>Encargados:</b> (todos) -<b>Duración:</b> 1 día -<b>Fecha:</b> 11/12/2025</p>

### 7.1.3 Desarrollo

Creación de la aplicación siguiendo las tareas (1 - 5), para poder integrarlas en la tarea 6.

<p><b>1. Implementación de la arquitectura y la infraestructura base:</b> Desarrollo de la estructura inicial del backend, la configuración de servicios clave: servidor, base de datos, autenticación.</p> <p>-<b>Encargados:</b> Diego Razvan Cozma y Hector Castro Portillo -<b>Duración:</b> 70 días -<b>Fecha:</b> 19/1/2026 - 24/4/2026</p>	<p><b>2. Implementación de módulos y funcionalidades principales:</b> Desarrollo de los módulos que dan forma a la aplicación como los de: gestión de usuarios (registro, login, verificación), publicación de anuncios, sistema de búsqueda y filtros, sistema de mensajería entre usuarios, etc.</p> <p>-<b>Encargados:</b> Hugo Lozano Ruiz, Sergio Viñas González y Ramiro Alejandro Gallegos Cuji -<b>Duración:</b> 70 días -<b>Fecha:</b> 19/1/2026 - 24/4/2026</p>
<p><b>3. Desarrollo de la interfaz de usuario:</b> Construcción de las pantallas y flujos definidos en diseño: inicio, búsqueda, detalles del piso, chat, perfil del arrendador, publicar habitación, etc.</p> <p>-<b>Encargados:</b> Miguel Cruz Gómez y Senbo Zhou Pan -<b>Duración:</b> 70 días -<b>Fecha:</b> 19/1/2026 - 24/4/2026</p>	<p><b>5. Implementación de requisitos no funcionales:</b> Aplicación de medidas de rendimiento, seguridad, protección de datos personales, control de acceso, optimización de carga de imágenes, y prácticas para asegurar escalabilidad y mantenibilidad.</p> <p>-<b>Encargados:</b> Héctor Manuel Díaz Bernal y Axel Metodiev Ojeda -<b>Duración:</b> 70 días -<b>Fecha:</b> 19/1/2026 - 24/4/2026</p>
<p><b>6. Documentación técnica del código:</b> Generación de documentación interna: estructuras, APIs, dependencias, guías de uso de la aplicación, y comentarios relevantes para facilitar el mantenimiento y las pruebas.</p> <p>-<b>Encargados:</b> (todos) -<b>Duración:</b> 70 días -<b>Fecha:</b> 19/1/2026 - 24/4/2026</p>	<p><b>6. Integración de componentes:</b> Conexión de todas las partes del sistema: infraestructuras bases, módulos principales, interfaces, y validación de que funcionan de forma coherente en conjunto.</p> <p>-<b>Encargados:</b> (todos) -<b>Duración:</b> 10 días -<b>Fecha:</b> 27/4/2026 - 8/5/2026</p>

## 7.1.4 Pruebas

Realización de las siguientes pruebas para posteriormente y al mismo tiempo solucionar y refactorizar la aplicación para su futura publicación.

<p><b>1. Pruebas unitarias:</b> Verificar el funcionamiento correcto de componentes individuales del sistema (funciones, clases o módulos) de forma aislada para detectar errores en la lógica interna.</p> <p>-<b>Encargados:</b> (todos) -<b>Duración:</b> 15 días -<b>Fecha:</b> 1/6/2026 - 19/6/2026</p>	<p><b>2. Pruebas de integración:</b> Comprobar que los distintos módulos del sistema funcionan correctamente entre sí, validando las interacciones y la comunicación entre componentes (APIs, base de datos, servicios externos...).</p> <p>-<b>Encargados:</b> (todos) -<b>Duración:</b> 15 días -<b>Fecha:</b> 1/6/2026 - 19/6/2026</p>
<p><b>3. Pruebas de sistema:</b> Evaluar el sistema completo funcionando como un todo, comprobando que cumple los requisitos especificados en condiciones similares a las reales.</p> <p>-<b>Encargados:</b> (todos) -<b>Duración:</b> 15 días -<b>Fecha:</b> 1/6/2026 - 19/6/2026</p>	<p><b>4. Pruebas de aceptación:</b> Pruebas realizadas junto a Borja Manero y el rectorado para confirmar que el sistema cumple sus expectativas y requisitos, y que está listo para su publicación.</p> <p>-<b>Encargados:</b> Miguel Cruz Gómez y Sergio Viñas González -<b>Duración:</b> 15 días -<b>Fecha:</b> 1/6/2026 - 19/6/2026</p>
<p><b>5. Refactorización y soluciones tras las pruebas:</b> Durante y tras las pruebas continuas, el equipo correspondiente al apartado se encargará de solucionar los problemas surgidos durante las pruebas.</p> <p>-<b>Encargados:</b> (todos) -<b>Duración:</b> 20 días -<b>Fecha:</b> 1/6/2026 - 26/6/2026</p>	

## 7.1.5 Implementación

Una vez todas las pruebas realizadas, las refactorizaciones y soluciones también probadas y todo aprobado por la universidad, la aplicación se publicará el 29 de junio de 2026 (provisionalmente) a través de todas las aplicaciones de descarga de apps tales como: Google Play Store, Apple App Store, etc. Y si es conveniente una posible descarga complementaria a través de la página web oficial de la universidad para mayor seguridad y control.

La descarga de la aplicación será totalmente gratuita, pero una vez dentro solo la podrán usar aquellos estudiantes con un correo @ucm y los arrendador que tengan convenio con la universidad para poder administrar sus anuncios.

## 7.1.6 Mantenimiento

El mantenimiento de la aplicación se realizará una vez se inicie su vida útil, de forma que gracias a la comisión por cada alquiler a través de la plataforma, la universidad se lleva dinero con el que poder pagar a un pequeño **equipo de mantenimiento** que tenga conocimientos para realizar las siguientes tareas:

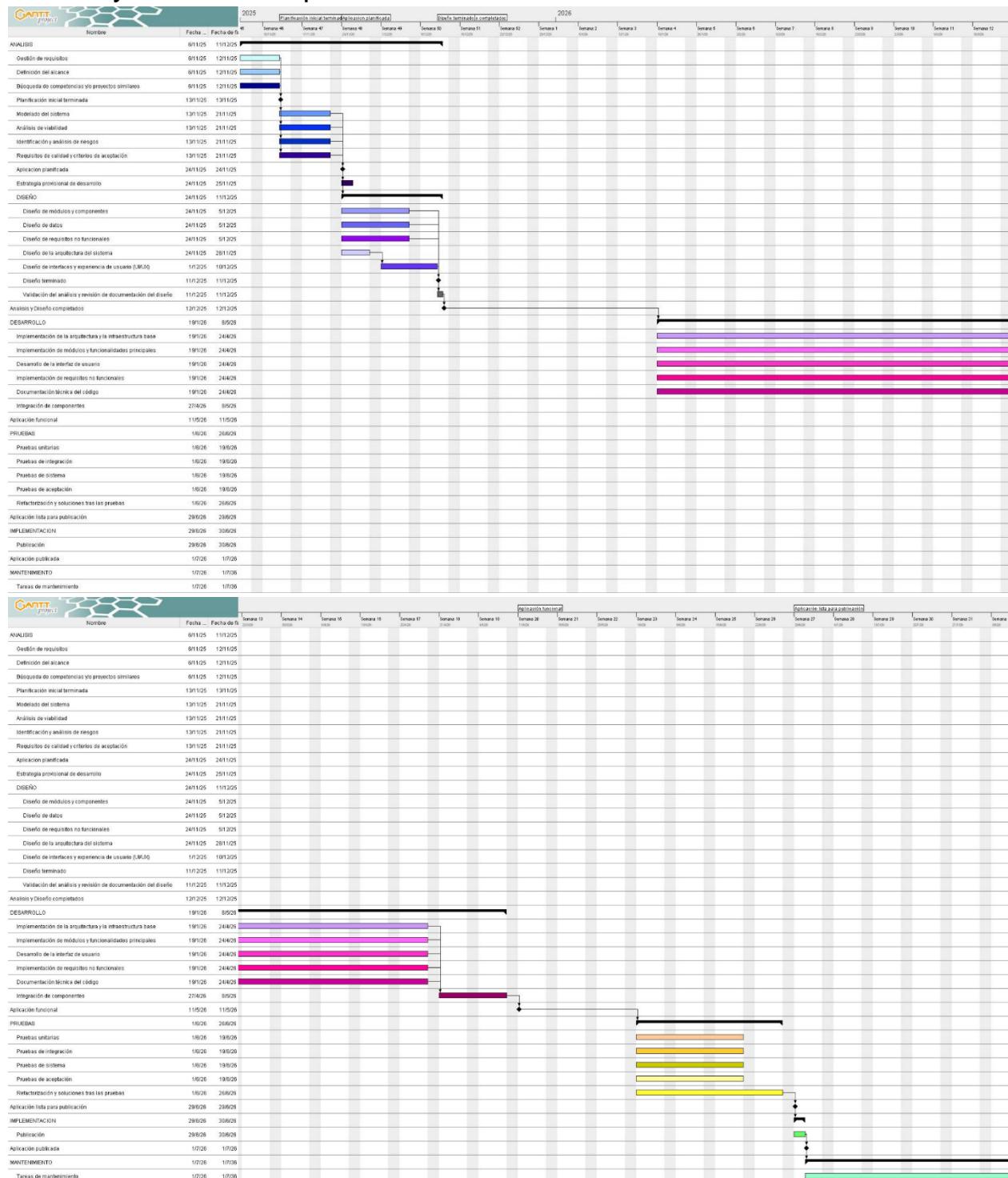
- Arreglos de errores que vayan surgiendo a lo largo del tiempo.
- Pago de servidores para mantener siempre la base de datos.
- Posibles nuevas implementaciones propuestas por la universidad o los usuarios.

Además, será necesario de un equipo de la UCM encargado de crear y administrar las cuentas y sus respectivas contraseñas, para los arrendadores durante toda la vida de la aplicación, mientras se quiera seguir añadiendo más anuncios.

Por supuesto esta fase podrá durar tanto como la universidad pueda permitirse a lo largo de los años.

## 7.2 Cronograma (Gantt)

Diagrama de Gantt teniendo en cuenta que solo se trabajará de lunes a viernes y fuera de temporadas de exámenes:



## 7.3 Dependencias e hitos clave

Las dependencias e hitos se pueden apreciar en el diagrama de Gantt anterior, representadas por las flechas entre tareas para las dependencias y rombos para los hitos.

Sin embargo estos son los hitos importantes del proyecto:

1. -Planificación inicial terminada
2. -Aplicación terminada
3. -Diseño terminado
4. -Análisis y diseño completados
5. -Aplicación funcional
6. -Aplicación lista para publicación
7. -Aplicación publicada

# 8. Recursos del Proyecto

## Presupuesto estimado:

El presupuesto se calcula considerando costes reales pero moderados, asumiendo que el equipo desarrollará internamente el diseño, el código y la documentación, evitando contratar servicios externos.

### 8.1. Costes de Personal

Se estiman 96 horas por integrante para desarrollo, pruebas y documentación.

- $9 \text{ integrantes} \times 6 \text{ h/semana} \times 16 \text{ semanas} \times 12 \text{ €/hora} = 10.368 \text{ €}$

Total personal: 10.368 €

#### 8.1.1 Equipo de trabajo

El equipo de trabajo estará formado por:

1. Miguel Cruz Gómez
2. Axel Metodiev Ojeda
3. Ramiro Alejandro Gallegos Cuji
4. Senbo Zhou Pan
5. Héctor Manuel Díaz Bernal
6. Hugo Lozano Ruiz
7. Diego Razvan Cozma
8. Sergio Viñas González
9. Hector Castro Portillo

Para este proyecto no tenemos unos roles definidos. Sin embargo, en el plan de trabajo se puede apreciar que tareas tiene asignado cada integrante.

## 8.2. Costes Tecnológicos

- Infraestructura básica (1 año)
- Dominio web (.es o .com): 12 €
- Servidor VPS básico (DigitalOcean/OVH): 10 €/mes → 120 €
- Base de datos administrada o alojada en el mismo VPS: 0–50 €
- Almacenamiento de imágenes (S3, Backblaze o similar): 30 €
- Servicio de correo (Mailgun/SendGrid básico): 50 €

**Total infraestructura: entre 212–262 € (ponemos la cifra promedio: 240 €)**

## 8.3. Software y herramientas

- Solo herramientas que realmente podrían requerir coste en un proyecto universitario:
- GitHub Team (opcional): 4 €/mes × 12 meses × 9 usuarios = 432 €
- Herramientas de diseño → se usarán herramientas gratuitas → 0 €

**versión mínima: 0 €**

**Total SW: 432 €**

## 8.4 Otros costes

- Reuniones (desplazamientos, material): 40 €
- Imprevistos técnicos: 100 €

**Total otros costes: 140 €**

## 8.5 Resumen total

- Categoría Coste
  - Personal 10.368 €
  - Infraestructura 240 €
  - Software (opcional) 432 €
  - Otros costes 140 €

**TOTAL ESTIMADO 11.180 €**

# 9. Gestión del Proyecto

## 9.1 Metodología

Para este proyecto se empleará una metodología ágil basada en Scrum, adaptada al entorno académico y a las dinámicas de trabajo en clase. Debido a que el equipo se reúne físicamente durante las sesiones presenciales, las reuniones diarias (“daily meetings”) se realizan al inicio de cada clase, permitiendo sincronizar el progreso, resolver dudas y detectar posibles bloqueos. Las reuniones semanales de planificación y retrospectiva se mantienen según la estructura Scrum habitual.

### **Roles Scrum definidos para el proyecto**

Product Owner (PO): Diego Cozma Responsable de priorizar el backlog, validar requisitos, representar la visión del cliente (profesorado) y aceptar las entregas parciales del equipo.

Scrum Master (SM): Miguel Cruz Encargado de velar por el correcto uso del marco Scrum, organizar las reuniones, facilitar el trabajo del equipo, eliminar impedimentos y coordinar la comunicación.

El resto del equipo actúa como Development Team, participando en el desarrollo de funcionalidades, documentación, pruebas y despliegue.

### **Estructura Scrum utilizada:**

Sprints de 1 o 2 semanas, como se puede apreciar en el diagrama de Gantt.

Reuniones de sincronización semanales para evaluar el progreso.

Backlog del producto mantenido en GitHub Projects.

Revisión y retrospectiva al final de cada sprint.

Esta metodología permite adaptarse a cambios en requisitos, mejorar progresivamente la plataforma y asegurar una comunicación continua dentro del equipo.

## 9.2 Gestión de riesgos

Se identifican y analizan los riesgos más relevantes relacionados con el desarrollo de la plataforma:

Nº	Riesgo	Descripción	Probabilidad	Consecuencia	Nivel (SQAS-SEI)
1	Retraso en entregas	Mala planificación o acumulación de trabajos académicos	Probable	Serio	Alto (H)
2	Fallos de integración frontend-backend	Incompatibilidad entre módulos desarrollados por distintos miembros	Ocasional	Serio	Medio (M)
3	Pérdida de código del proyecto	Fallos en repositorio, borrado accidental o problemas en PCs	Improbable	Catastrófica	Medio (M)
4	Cambios tardíos en requisitos	Solicitudes nuevas del cliente que afecten a módulos ya desarrollados	Ocasional	Crítico	Medio (M)
5	Baja disponibilidad del equipo	Enfermedades, exámenes o carga académica	Frecuente	Menor	Medio (M)

<b>Nº</b>	<b>Riesgo</b>	<b>Descripción</b>	<b>Probabilidad</b>	<b>Consecuencia</b>	<b>Nivel (SQAS-SEI)</b>
6	Publicación de anuncios fraudulentos	Usuarios malintencionados que suben habitaciones falsas	Ocasional	Serio	Medio (M)
7	Problemas de seguridad (ataques y fuga de datos)	Riesgos como inyección SQL o accesos no autorizados a información personal	Remoto	Catastrófico	Alto (H)
8	Sobrecarga del sistema en picos de demanda	Alta afluencia de estudiantes al inicio de cuatrimestres	Probable	Serio	Alto (H)
9	Errores en la base de datos	Habitaciones no disponibles que aparecen como libres o duplicados	Ocasional	Menor	Bajo (L)
10	Fallo en el sistema de verificación de propietarios	No detectar perfiles falsos o malintencionados	Ocasional	Serio	Medio (M)

## Planes generales de mitigación

**Control del retraso de entregas:** Mantener un Gantt actualizado, revisar semanalmente y dividir tareas grandes en tareas pequeñas.

**Pérdida de código:** Uso obligatorio de GitHub y copias en Drive.

**Cambios tardíos en requisitos:** Solicitarlos por escrito y evaluarlos solo al final del sprint.

**Fallos técnicos:** Pruebas continuas de integración (CI), revisión de código entre compañeros.

## 9.3 Control de calidad

Para garantizar la calidad del producto se implementarán los siguientes mecanismos:

**Puntos de control (milestones)** Revisión de requisitos antes de crear prototipos. Validación del diseño de BD antes de programar. Pruebas unitarias tras completar cada módulo. Pruebas de integración al finalizar cada sprint.

**Revisiones internas** Cada entrega será revisada por un miembro del equipo distinto del autor. Cambios pequeños se gestionan entre autor y revisor. Cambios grandes requerirán reunión del equipo.

**Auditorías informales** Reuniones de verificación del progreso previo a entregas formales. Registro de decisiones tomadas en GitHub / Drive

## 9.4 Gestión de cambios

La gestión de cambios sigue el procedimiento:

**Elementos susceptibles a cambio** Documentación: Plan de Proyecto y SRS. Código de frontend y backend. Modelo de datos.

**Proceso** Si NO es línea base → cambio directo acordado entre autor y revisor. Si SÍ es línea base →

1. Se envía solicitud de cambio (plantilla).
2. Se analiza en reunión del equipo.

3. Se aprueba/rechaza por voto o consenso.
4. Se documenta versión nueva.

**Versionado** Formato estándar: V1.0, V1.1, V1.2... Cambios de línea base generan número principal nuevo. Correcciones menores generan un decimal.

## 9.5 Comunicación

Para mantener un flujo de comunicación claro y constante se establecen los siguientes canales:

**Herramientas de comunicación interna** WhatsApp / Telegram para comunicación rápida diaria. Google Meet para reuniones semanales de sprint. Google Drive para almacenamiento colaborativo. GitHub Issues para seguimiento técnico.

**Frecuencia** Reunión formal semanal. Reuniones extraordinarias en caso de cambios o problemas importantes. Notificación diaria del avance mediante mensajes cortos en grupo.

**Comunicación con representante de la Universidad (Borja)** Envío de entregas parciales según está especificado en el diagrama de Gantt, o sea, al final de cada fase. Preparación de demos cuando se requiera.

# 10. Plan de pruebas

## 10.1. Tipos de Pruebas

Tipo	Objetivo	Validación principal
Unitarias	Lógica interna	Reglas, validaciones
Integración	Comunicación entre módulos	BD, estados, servicios
Sistema	Flujo completo	Funcionalidad real
Aceptación	Uso real	Claridad y utilidad

## 10.2 Estrategia de Validación

La estrategia de validación definida para el sistema tiene como objetivo garantizar que todas las funcionalidades implementadas cumplan con los requisitos funcionales, de calidad y de usabilidad establecidos. Para ello, se adopta un enfoque progresivo y estructurado, basado en distintos niveles de prueba que permiten detectar errores en fases tempranas y asegurar la estabilidad del sistema antes de su uso final.

En primer lugar, se realizan pruebas unitarias, cuyo propósito es validar el correcto funcionamiento de los componentes individuales del sistema de forma aislada. Estas pruebas se centran principalmente en la lógica interna de la aplicación, como la validación de datos de entrada, el cumplimiento de reglas de negocio y el control de permisos básicos. De este modo, se asegura que cada módulo responde correctamente ante entradas válidas e inválidas antes de ser integrado con el resto del sistema.

Una vez superadas las pruebas unitarias, se llevan a cabo pruebas de integración. En esta fase se verifica la correcta comunicación entre los distintos módulos y servicios de la aplicación, como la interacción entre la interfaz, la lógica de negocio y la base de datos. Estas pruebas permiten comprobar que las

operaciones más relevantes, como el registro de usuarios, la publicación y modificación de ofertas, el envío de solicitudes o la gestión de notificaciones, se realizan de forma coherente y consistente, manteniendo la integridad de los datos y los estados del sistema.

Posteriormente, se ejecutan pruebas de sistema, que validan el funcionamiento completo de la aplicación desde el punto de vista del usuario final. En este nivel se prueban los flujos completos de los casos de uso, incluyendo la navegación por la interfaz, la ejecución de acciones y la respuesta del sistema ante diferentes escenarios. El objetivo de estas pruebas es confirmar que todas las funcionalidades principales están correctamente implementadas y que el sistema se comporta de acuerdo con lo esperado en un entorno lo más similar posible al real.

Finalmente, se realizan pruebas de aceptación, orientadas a evaluar la calidad percibida del sistema por parte de los usuarios. Estas pruebas se centran en aspectos como la claridad de la interfaz, la facilidad de uso de las funcionalidades y la comprensión general de los flujos de interacción. A través de este tipo de pruebas se valida que la aplicación no solo es funcional, sino también intuitiva y adecuada para su uso en un contexto real.

Un caso se considera válido únicamente cuando supera todas las pruebas aplicables.

### 10.3 Casos de Uso y Pruebas Asociadas

CU-01 — Buscar y solicitar oferta (Alumno)

Tipo	Prueba
Integración	Búsqueda en BD
Sistema	Envío de solicitud
Aceptación	Flujo claro

Criterios:

- Solicitud registrada

#### CU-02 — Filtrar ofertas

Tipo	Prueba
Unitaria	Filtros
Integración	Filtros combinados
Sistema	Resultados filtrados

Criterios:

- Resultados coherentes

#### CU-03 — Publicar oferta

Tipo	Prueba
Unitaria	Permisos
Integración	Guardado de oferta
Sistema	Oferta visible
Aceptación	Facilidad de publicación

Criterios:

- Solo usuarios autorizados

#### CU-04 — Eliminar oferta

Tipo	Prueba
Sistema	Eliminación
Aceptación	Confirmación

Criterios:

- Oferta no accesible

#### CU-05 — Ver info de las habitaciones (Propietario)

Tipo	Prueba
Sistema	Visualización
Aceptación	Información clara

Criterios:

- Información completa

#### CU-06 — Iniciar sesión

Tipo	Prueba
Unitaria	Validación de credenciales
Integración	Inicio de sesión
Sistema	Acceso a área privada
Aceptación	Facilidad de uso

Criterios:

- Acceso solo con credenciales válidas

#### CU-07 — Cerrar sesión

Tipo	Prueba
Sistema	Cierre de sesión
Aceptación	Comprensión del proceso

Criterios:

- No acceso tras cierre de sesión

#### CU-08 — Ver mensajes y notificaciones (Alumno)

Tipo	Prueba
Integración	Recuperación de mensajes
Sistema	Visualización
Aceptación	Claridad de bandeja

Criterios:

- Mensajes actualizados

#### CU-09 — Modificar oferta vigente (Propietario)

Tipo	Prueba
Integración	Actualización
Sistema	Cambios visibles
Aceptación	Flujo claro

Criterios:

- Cambios reflejados

#### CU-10 — Aceptar solicitud y comenzar conversación (Propietario)

Tipo	Prueba
Integración	Cambio de estado
Sistema	Conversación habilitada
Aceptación	Flujo intuitivo

Criterios:

- Estado aceptado

#### CU-11 — Rechazar estudiante (Propietario)

Tipo	Prueba
Integración	Cambio de estado
Sistema	Notificación
Aceptación	Mensaje claro

Criterios:

- Estado rechazado

#### CU-12 — Registrarse

Tipo	Prueba
Unitaria	Validación de email y contraseña
Integración	Creación de usuario en BD
Sistema	Registro completo
Aceptación	Claridad del formulario

Criterios:

- Usuario creado correctamente
- Mensajes de error claros

#### CU-13 — Interesarse por oferta (alerta)

Tipo	Prueba
Integración	Registro de alerta
Sistema	Envío de notificación
Aceptación	Utilidad percibida

Criterios:

- Notificación enviada al cambiar estado

#### CU-14 — Cambiar ajustes de la app

Tipo	Prueba
Integración	Guardado de ajustes
Sistema	Persistencia
Aceptación	Claridad de opciones

Criterios:

- Cambios guardados correctamente

#### CU-15 — Abrir localización en Maps

Tipo	Prueba
Sistema	Carga de mapa
Aceptación	Visualización correcta

Criterios:

- Ubicación correcta

#### CU-16 — Añadir oferta a favoritos (Alumno)

Tipo	Prueba
Unitaria	Evitar duplicados
Integración	Guardado
Sistema	Lista actualizada
Aceptación	Utilidad

Criterios:

- Sin duplicados

CU-17 — Solicitar quedada con propietario (Alumno)

Tipo	Prueba
Integración	Registro de cita
Sistema	Notificación
Aceptación	Flujo claro

Criterios:

- Cita registrada

CU-18 — Reportar Propietario por Incidencia

Tipo	Prueba
Integración	Registro de reporte
Sistema	Reportar
Aceptación	Flujo claro

Criterios:

- Cita registrada

## 10.4 Herramientas de Testing

### Introducción

Para la validación y verificación del sistema se utilizarán únicamente herramientas compatibles con el entorno tecnológico del proyecto, basado en **Java** y **Java Swing**. La selección de estas herramientas permite realizar pruebas coherentes con la arquitectura de la aplicación, facilitando la detección de errores y garantizando la calidad del software desarrollado.

## **JUnit**

JUnit es la principal herramienta utilizada para la realización de **pruebas unitarias** en aplicaciones Java. Permite comprobar el correcto funcionamiento de métodos y clases de forma aislada, validando la lógica interna del sistema.

### **Uso en el proyecto**

JUnit se empleará para:

- Validar reglas de negocio.
- Comprobar validaciones de datos de entrada.
- Verificar el comportamiento de métodos críticos.
- Detectar errores de forma temprana durante el desarrollo.

### **Ventajas**

- Integración nativa con Java.
- Ejecución automática de pruebas.
- Facilita el mantenimiento y la reutilización de tests.
- Amplio soporte y documentación.

## **Mockito**

Mockito es una herramienta complementaria a JUnit que permite simular el comportamiento de objetos y dependencias externas. Es especialmente útil cuando se realizan pruebas unitarias de componentes que dependen de otros módulos del sistema.

### **Uso en el proyecto**

Mockito se utilizará para:

- Simular accesos a la base de datos.
- Aislar componentes durante las pruebas unitarias.
- Comprobar interacciones entre clases sin depender de implementaciones reales.

## Ventajas

- Permite pruebas más controladas y precisas.
- Reduce dependencias entre módulos.
- Facilita la detección de errores lógicos.

## Pruebas de Integración en Java

Las **pruebas de integración** se realizarán utilizando JUnit, ejecutando escenarios en los que varios módulos interactúan entre sí, como la lógica de negocio y el acceso a datos.

### Uso en el proyecto

Estas pruebas permitirán:

- Verificar la correcta comunicación entre capas.
- Comprobar la persistencia y recuperación de datos.
- Validar cambios de estado en el sistema.

## Pruebas de Sistema en Java Swing

Dado que la aplicación utiliza **Java Swing** para la interfaz gráfica, las pruebas de sistema se llevarán a cabo principalmente de forma manual, ejecutando la aplicación y validando los flujos completos de los casos de uso.

### Uso en el proyecto

Se validará:

- Correcta navegación entre pantallas.
- Respuesta de la interfaz a las acciones del usuario.
- Visualización correcta de datos y mensajes.
- Manejo adecuado de errores en la interfaz.

## Pruebas de Aceptación

Las **pruebas de aceptación** se realizarán mediante la ejecución manual de escenarios reales por parte de usuarios o miembros del equipo, utilizando la aplicación Java Swing.

### **Uso en el proyecto**

Estas pruebas permitirán:

- Evaluar la facilidad de uso del sistema.
- Comprobar que los flujos son comprensibles.
- Validar que el sistema cumple los requisitos funcionales