



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Diseño de un portal web para alquiler de habitaciones a estudiantes

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Ávila González, José Luis

**Tutor:** Sáez Barona, Sergio

2016/2017

# Diseño de un portal web para alquiler de habitaciones a estudiantes

# Resumen

---

En este trabajo se desarrolla un portal web de alquiler de habitaciones para estudiantes que permite al usuario del portal realizar distintas acciones relacionadas con el alquiler de habitaciones, como pueden ser el ofertar una habitación para alquiler o alquilar una de estas. En el trabajo se desarrolla tanto la parte frontend como la parte backend.

**Palabras clave:** web, back-end, front-end, portal, Laravel, habitaciones.

# Abstract

---

In this project is developed a web portal for student room rental that allows the user of the web to perform different actions related to the rental of rooms, such as offering a room for rent or rent one of these. The project develops both the frontend and the backend.

**Keywords:** web, back-end, front-end, portal, Laravel, room.

# Tabla de contenidos

---

1	Introducción .....	7
1.1	Objetivos .....	7
1.2	Contexto.....	7
1.3	Estructura del documento .....	8
2	Especificación de requisitos.....	9
2.1	Introducción .....	9
2.1.1	Propósito .....	9
2.1.2	Ámbito .....	9
2.1.3	Definiciones, Acrónimos y Abreviaturas.....	9
2.1.4	Visión general del documento .....	10
2.2	Descripción General.....	10
2.2.1	Perspectiva del producto .....	10
2.2.2	Funciones del producto .....	10
2.2.3	Características del usuario .....	11
2.2.4	Restricciones .....	11
2.2.5	Supuestos y dependencias .....	11
2.2.6	Requisitos futuros .....	12
2.3	Requisitos específicos .....	12
2.3.1	Requisitos de interfaces externos.....	12
2.3.2	Requisitos funcionales.....	12
2.3.3	Atributos del sistema.....	19
2.3.3.1	Mantenimiento .....	19
2.3.3.2	Portabilidad .....	19
2.3.3.3	Seguridad .....	19
3	Análisis.....	20
3.1	Introducción.....	20
3.2	Casos de uso.....	20
3.2.1	Actores .....	20
3.2.2	Caso de uso: Usuario no Registrado .....	21
3.2.3	Caso de uso: Usuario Registrado .....	21
3.2.4	Especificación de los casos de uso .....	22
3.3	Diagramas de actividad.....	31
3.3.1	Diagramas de actividad Búsqueda .....	32

3.3.2	Diagramas de actividad Inicio Sesión .....	32
3.3.3	Diagramas de actividad Publicar Habitación.....	33
3.4	Diagrama de clases .....	33
4	Diseño .....	34
4.1	Introducción .....	34
4.2	Arquitectura física.....	34
4.3	Arquitectura de software .....	35
4.3.1	Capa de persistencia .....	35
4.3.1.1	Modelo.....	36
4.3.1.2	Modelo entidad relación de la BBDD .....	36
4.3.1.3	Descripción detallada de tablas .....	37
4.3.2	Capa de presentación .....	39
4.3.2.1	Vista .....	39
4.3.2.2	Mockups web .....	40
4.3.3	Capa de lógica.....	41
4.3.3.1	Controlador .....	41
4.3.3.2	Enrutamiento .....	41
4.3.3.3	Mapa web.....	42
5	Desarrollo.....	43
5.1	Tecnologías .....	43
5.1.1	FrontEnd .....	43
5.1.1.1	HTML.....	43
5.1.1.2	CSS.....	43
5.1.1.3	JavaScript .....	44
5.1.2	BackEnd.....	44
5.1.2.1	PHP .....	44
5.1.2.2	Laravel .....	44
5.1.2.3	MySQL .....	45
5.2	Entorno de desarrollo .....	45
5.3	Herramientas utilizadas .....	45
5.3.1	PhpStorm.....	45
5.3.2	phpMyAdmin.....	46
5.3.3	Composer.....	46
5.3.4	Github.....	46
5.3.5	Artisan .....	47
5.4	Estructura .....	47
5.4.1	Directorio app.....	47



5.4.2	Directorio config.....	47
5.4.3	Directorio database .....	48
5.4.4	Directorio public.....	48
5.4.5	Directorio resources .....	48
6	Pruebas .....	49
6.1	Visualización en diferentes dispositivos.....	49
6.2	Visualización en diferentes navegadores.....	51
6.3	Análisis estático .....	52
6.4	Pruebas de uso .....	54
6.4.1	Ver Perfil.....	54
6.4.2	Crear Habitación .....	55
6.4.3	Búsqueda Avanzada.....	57
6.4.4	Ver Habitación.....	59
7	Conclusiones .....	60
8	Bibliografía .....	61

# 1 Introducción

---

Un proyecto final de grado constituye una parte importante para todo estudiante porque en este se sintetiza todo lo aprendido dando forma a una idea en un producto, además porque permite no solo que se consoliden los conocimientos adquiridos, sino que también permite adquirir nuevos. Por esto en desarrollo se hace uso de las tecnologías que se usan actualmente para el desarrollo web.

En este proyecto se desarrolla una aplicación web de alquiler de habitaciones a estudiantes que se centra principalmente en facilitar al estudiante el alquiler u oferta de una habitación.

La aplicación contendrá una información muy detallada de las características de las habitaciones, siendo esta información enfocada principalmente hacia las necesidades que puede tener un estudiante.

## 1.1 Objetivos

El objetivo principal de este trabajo de fin de grado es aplicar, ampliar y consolidar la gran cantidad de conocimientos que se han ido adquiriendo durante todo el proceso formativo.

También se tiene como objetivo que la aplicación web facilite al estudiante la búsqueda de habitaciones que se adapten a sus necesidades para el correcto desarrollo de sus estudios, así como facilitar la oferta de habitaciones para compartir con otros estudiantes.

## 1.2 Contexto

En la actualidad existen gran variedad de aplicaciones web que ofertan servicios inmobiliarios, sin embargo, la mayor parte de estas aplicaciones están enfocadas a un mercado muy generalizado.

Por lo tanto, la aplicación web que se piensa desarrollar está orientada a ofrecer un servicio inmobiliario enfocado en los estudiantes universitarios, estos podrán buscar u ofertar habitaciones que cumplan con sus necesidades.

Para lograr alcanzar los objetivos para la aplicación web, se ha decidido la existencia de dos diferentes tipos de usuarios, registrados y no registrados. Los usuarios registrados tienen un acceso a funciones más avanzadas de la aplicación que dependiendo del rol pueden llevar a cabo determinadas funciones, por otra parte, los usuarios no registrados tendrán solo acceso a las funciones básicas de la aplicación.

### 1.3 Estructura del documento

El documento está estructurado siguiendo las etapas de desarrollo de software más comunes, siendo estas etapas las siguientes:

**Especificación de requisitos:**

Se describe detalladamente el software a desarrollar.

**Análisis:**

Se describe el comportamiento esperado del software y su interacción con los usuarios y/u otros sistemas.

**Diseño:**

Se explica la arquitectura que se ha utilizado en la aplicación.

**Desarrollo:**

Se comentan las tecnologías y herramientas utilizadas, así como detalles del desarrollo.

**Pruebas:**

Se describen las pruebas que se han realizado para asegurar la calidad en el desarrollo del proyecto, así como su correcto funcionamiento.

# 2 Especificación de requisitos

---

## 2.1 Introducción

### 2.1.1 Propósito

La Especificación de Requisitos Software (ERS) tiene como propósito el establecer el conjunto de funcionalidades de la aplicación web, la definición y el resumen de cada una de estas.

La ERS pretende dar una visión completa de las funcionalidades y requisitos de la aplicación, siendo el punto de partida del desarrollo de la aplicación web.

La ERS se ha realizado siguiendo el estándar IEEE Std. 830-1998, el cual describe el contenido y las calidades de una buena especificación de requisitos de software.

### 2.1.2 Ámbito

La aplicación web que se va a desarrollar tiene como principal objetivo facilitar al estudiante la búsqueda y oferta de habitaciones para compartir con otros estudiantes.

Para facilitar la búsqueda y oferta la aplicación proporcionara información importante para el estudiante como son la ubicación respecto a las universidades, el transporte público cercano, la disponibilidad de acceso a internet y alguna información más general sobre las características de las habitaciones que se ofertan.

Además, el estudiante tendrá la opción de seleccionar sus habitaciones favoritas pudiendo así realizar posteriormente una comparativa de las habitaciones que puedan resultar de su interés y elegir la habitación que mejor se adapte a sus necesidades.

### 2.1.3 Definiciones, Acrónimos y Abreviaturas

En este apartado se definen los términos, acrónimos y abreviaturas utilizadas en el apartado de especificación de requisitos.

1. ERS: Especificación de Requisitos Software
2. Framework: Estructura conceptual y tecnológica que sirve de base para la organización y el desarrollo de software.
3. Responsive Design: Es una filosofía de diseño y desarrollo que busca adaptar la apariencia de las páginas web al dispositivo utilizado.

#### **2.1.4 Visión general del documento**

A continuación, se realizará una descripción general de la aplicación, describiendo todos los factores que afectan a la aplicación y a sus requisitos.

También se realizará la descripción de requisitos específicos de la aplicación con un alto nivel de detalle los requisitos que se deben satisfacer en la aplicación.

## **2.2 Descripción General**

### **2.2.1 Perspectiva del producto**

La aplicación que se va a desarrollar es totalmente independiente de cualquier otro sistema, es decir no está relacionada ni forma parte de ninguna otra aplicación.

En un futuro podría que se relacionase con otras aplicaciones como podría ser alguna equivalente para dispositivos móviles o integrarse en alguna red social o servicio web de búsqueda de habitaciones.

### **2.2.2 Funciones del producto**

La aplicación ofertara distintas funcionalidades según el tipo de usuario. Existirán dos tipos de usuarios: Usuarios registrados y Usuarios no registrados.

Los usuarios no registrados podrán:

- Buscar habitaciones.
- Ver información básica de la habitación.
- Registrarse en la aplicación.

Los usuarios registrados podrán hacer todo lo que hacen los usuarios no registrados y además podrán:

- Comunicarse con otros usuarios.
- Ofertar habitaciones.
- Gestionar perfil de usuario.
- Gestionar habitaciones ofertas.
- Gestionar habitaciones favoritas.
- Ver información avanzada de la habitación.

### 2.2.3 Características del usuario

La aplicación tendrá como principales usuarios las personas que se van a iniciar sus estudios o ya estén estudiando y se encuentren en la búsqueda de una habitación que se adapte a sus necesidades.

El nivel informático de los usuarios de la aplicación a priori es desconocido, esto se tendrá en cuenta en las distintas fases del desarrollo de la aplicación siendo evaluando todo lo que se diseñe constantemente ya que siempre es más económico reconducir un diseño que rediseñarlo completamente.

### 2.2.4 Restricciones

Para acceder a la aplicación será necesario disponer de un dispositivo con acceso a Internet, principalmente ordenadores ya sean portátiles o de sobremesa, en dispositivos móviles la aplicación web no estará adaptada.

La aplicación tendrá que ser accesible desde los navegadores más importantes y usados actualmente para acceder a la web desde ordenadores como son Mozilla Firefox, Google Chrome y Microsoft Edge.

Las contraseñas de los usuarios que se registran en la aplicación se deben guardar encriptadas en base de datos.

Para poder contactar con los ofertantes de pisos/habitaciones será necesario ser un usuario registrado.

### 2.2.5 Supuestos y dependencias

La aplicación requiere que se utilice un navegador, un servidor web, una base de datos y una conexión a Internet siendo estas las dependencias más importantes de la aplicación.

Los navegadores puede que en un futuro dejen de ser compatibles con la tecnología que se usa, por lo que se debería evaluar si la aplicación debe evolucionar a tecnologías que soporten los navegadores.

En cuanto a un cambio de servidor solo se tendría que realizar las configuraciones necesarias para su funcionamiento como el que se tenía antiguamente.

El acceso a internet es la dependencia más importante ya que si internet no se puede acceder a la aplicación, sin embargo, hoy en día el acceso a internet cada vez es mayor.

### 2.2.6 Requisitos futuros

En un futuro se podrían realizar distintas mejoras en la aplicación. Algunas de las mejoras que se podrían estudiar y realizar serían las siguientes:

Rediseñar la aplicación para que se adapte a todo tipo de dispositivos conservando toda su funcionalidad mediante uso de las últimas técnicas de diseño de interfaces de usuario como puede ser el Responsive Design.

Añadir nuevas funcionalidades que permitan realizar todas las gestiones del alquiler de habitaciones mediante la misma web, como la firma de contratos de forma digital, gestión de incidencias respecto a los servicios ofertados por la habitación, así como la domiciliación y gestión de los pagos.

## 2.3 Requisitos específicos

### 2.3.1 Requisitos de interfaces externos

La interface de la aplicación deberá ser diseñada para un usuario con conocimientos medios de informática, siendo fácil de aprender y entender su uso.

### 2.3.2 Requisitos funcionales

A continuación, se detallarán los requisitos funcionales para los dos tipos de usuarios existentes, usuarios registrados y usuarios no registrados.

Dado que los usuarios registrados pueden actuar como usuarios no registrados los requisitos funcionales de este último tipo de usuarios aplican a los usuarios registrados.

#### ***Usuario no Registrado***

RF-001	Registrarse en la aplicación
Propósito	El usuario puede darse de alta como usuario registrado
Entrada	Datos personales y contraseña.
Proceso	El usuario rellenara el formulario de registro. Se validan los campos que ha rellenado el usuario. Se inicia una sesión.
Respuesta	Se visualiza la página principal en caso de que se cree el registro correctamente en la aplicación, en caso contrario se indicara al usuario mediante un mensaje de error.

RF-002	Iniciar sesión en la aplicación
Propósito	El usuario puede iniciar sesión en la aplicación
Entrada	Correo electrónico y contraseña.
Proceso	El usuario rellena el formulario de inicio de sesión. Se validan los campos que ha rellanado el usuario. Se inicia una sesión.
Respuesta	Se visualiza el perfil del usuario en caso de que se cree correctamente, en caso contrario se indicara al usuario mediante un mensaje de error.

RF-003	Ver listado de habitaciones
Propósito	El usuario puede ver el listado de habitaciones ofertadas.
Entrada	
Proceso	El usuario accede a la página principal de la aplicación.
Respuesta	Se muestran todas las habitaciones disponibles mostrando las más recientes primero.

RF-004	Ver habitación
Propósito	El usuario puede ver los detalles básicos de una habitación de las habitaciones ofertadas.
Entrada	Identificador de habitación.
Proceso	El usuario hace clic sobre el botón “Ver Habitación” de la habitación que desea.
Respuesta	Se muestran los detalles básicos de la habitación seleccionada por el usuario.

RF-005	Buscar habitación
Propósito	El usuario puede realizar una búsqueda en el listado de habitaciones ofertadas.
Entrada	Filtros con los que se desea buscar el usuario en el listado de habitaciones ofertadas.
Proceso	El usuario selecciona los filtros con los que desea buscar en el listado de habitaciones ofertadas y hace clic sobre el botón “Buscar”.
Respuesta	Se muestra el listado de todas las habitaciones que cumplen con los criterios de búsqueda seleccionados.

### **Usuario Registrado**

RF-006	Ver imágenes de la habitación
Propósito	El usuario puede ver las imágenes de una habitación.
Entrada	Identificador de habitación.
Proceso	El usuario hace clic en los botones de la imagen para ir viendo todas las imágenes de la habitación.
Respuesta	Se va mostrando en la página de detalles de la habitación cada una de las imágenes de esta.

RF-007	Crear habitación
Propósito	El usuario puede crear una habitación a ofertar.
Entrada	Datos de la habitación a ofertar.
Proceso	<p>El usuario hace clic en el botón “Publicar Habitación” situado en el perfil del usuario y en el menú de la cabecera de web una vez inicia sesión.</p> <p>Se mostrará el formulario de creación de la habitación.</p> <p>El usuario rellena el formulario y hace clic en el botón “Publicar”.</p>
Respuesta	<p>En caso de error se muestra un mensaje en el formulario indicando los errores.</p> <p>Si se ha creado correctamente muestra los detalles de la habitación.</p>

RF-008	Modificar habitación
Propósito	El usuario puede modificar una habitación ofertada.
Entrada	<p>Identificadora habitación.</p> <p>Datos a modificar de la habitación.</p>
Proceso	<p>El usuario selecciona la acción “Editar” de la habitación deseada en el listado de habitaciones ofertadas de su perfil.</p> <p>Se muestra un formulario donde se pueden modificar algunos de los datos de la habitación.</p> <p>Una vez modificados los datos se hace clic en “Guardar Modificación”</p>
Respuesta	<p>En caso de error se muestra un mensaje en el formulario indicando los errores.</p> <p>Si se ha modificado correctamente muestra los detalles de la habitación modificada.</p>

<b>RF-009</b>	<b>Eliminar habitación</b>
Propósito	El usuario puede eliminar una habitación ofertada.
Entrada	Identificador de habitación.
Proceso	El usuario selecciona la acción “Borrar” de la habitación deseada en el listado de habitaciones ofertadas de su perfil.
Respuesta	Se muestra el listado de las habitaciones ofertadas, donde ya no está la habitación eliminada.

<b>RF-010</b>	<b>Añadir imágenes de la habitación</b>
Propósito	El usuario puede añadir imágenes de la habitación ofertada.
Entrada	Imagen
Proceso	Mediante el formulario de creación o modificación de una habitación ofertada el usuario añadirá imágenes.
Respuesta	Una vez creada o modificada la habitación, se muestran los detalles de la habitación donde se pueden observar las imágenes.

<b>RF-011</b>	<b>Eliminar imágenes de la habitación</b>
Propósito	El usuario puede eliminar imágenes de la habitación ofertada.
Entrada	Identificador de imagen.
Proceso	El usuario eliminara las fotos deseadas haciendo clic en el botón “X” que aparece sobre las imágenes en el formulario de modificación de la habitación seleccionada.
Respuesta	Se recargará el formulario de modificación donde ya no está la imagen eliminada.

<b>RF-012</b>	<b>Buscar habitaciones por universidad cercana</b>
Propósito	El usuario registrado podrá realizar la búsqueda de habitaciones con el criterio de universidad cercana.
Entrada	Universidad
Proceso	Mediante el formulario de búsqueda el usuario registrado podrá seleccionar un nuevo filtro “Universidad” que filtrará las habitaciones que estén cercanas a la Universidad seleccionada.
Respuesta	Se muestra el listado de todas las habitaciones que cumplen con los criterios de búsqueda seleccionados y con los criterios de búsqueda Universidad y la distancia.

RF-013	Ver Distancia Universidades de la habitación
Propósito	El usuario registrado podrá ver la distancia que separa la habitación de las universidades.
Entrada	Identificador de habitación.
Proceso	El usuario registrado hace clic sobre el botón “Ver Habitación” de la habitación que desea.
Respuesta	Se mostrarán además de los detalles básicos de la habitación seleccionada un nuevo apartado donde se indica la distancia de la habitación respecto a distintas universidades.

RF-014	Ver perfil de usuario
Propósito	El usuario podrá ver su perfil.
Entrada	Identificador usuario
Proceso	El usuario hace clic en el botón “Perfil” del menú de la cabecera de la página cuando ha iniciado sesión.
Respuesta	Se muestra el perfil de usuario.

RF-015	Modificar perfil de usuario
Propósito	El usuario podrá modificar sus datos de perfil.
Entrada	Identificador de usuario. Datos a modificar del perfil.
Proceso	Estando el usuario en su perfil hace clic en el botón de edición.  Se muestra un formulario donde se pueden modificar algunos datos de su perfil.  Una vez modificados los datos se hace clic en “Guardar”
Respuesta	En caso de error se muestra un mensaje en el formulario indicando los errores, si se ha modificado correctamente muestra el perfil del usuario.

RF-016	Dar de baja usuario
Propósito	El usuario se podrá dar de baja.
Entrada	Identificador de usuario.
Proceso	Estando el usuario en su perfil hace clic en el botón de darse de baja.  Se muestra un mensaje de confirmación.  El usuario acepta o rechaza darse de baja.
Respuesta	Se muestra un mensaje indicando que la baja ha sido correcta y se redirige al usuario a la página principal.

RF-017	Añadir habitación a favoritas
Propósito	El usuario añadirá una habitación a su listado de favoritas.
Entrada	Identificador de habitación.
Proceso	Estando el usuario visualizando una habitación hará clic en el botón “Añadir a favoritas”.
Respuesta	Se mostrará la marca de una estrella en la zona superior derecha del detalle de la habitación añadida. También se mostrará la habitación añadida en el perfil del usuario.

RF-018	Eliminar habitación de favoritas
Propósito	El usuario eliminara una habitación de su listado de favoritas.
Entrada	Identificador de habitación.
Proceso	El usuario selecciona la acción “Borrar” de la habitación deseada en el listado de habitaciones favoritas de su perfil o también puede borrarla de su lista haciendo clic en la estrella en la zona superior derecha del detalle de la habitación que se desea eliminar de las favoritas.
Respuesta	Se muestra el listado de las habitaciones favoritas, donde ya no está la habitación eliminada.

RF-019	Ver listado de habitaciones ofertadas por el usuario
Propósito	El usuario podrá ver el listado de las habitaciones ofertadas por él.
Entrada	Identificador usuario.
Proceso	El usuario accede a su perfil.
Respuesta	Se muestra el listado de habitaciones ofertadas en el perfil del usuario.

RF-020	Ver listado de habitaciones favoritas
Propósito	El usuario podrá ver el listado de todas sus habitaciones favoritas.
Entrada	Identificador usuario.
Proceso	El usuario accede a su perfil.
Respuesta	Se muestra el listado de habitaciones favoritas en el perfil del usuario.

RF-021	Ver mensajes
Propósito	El usuario podrá ver el listado de mensajes.
Entrada	Identificador usuario
Proceso	El usuario accede a su perfil y hace clic sobre el botón “Mensajes” Se va a una pantalla con el listado de mensajes, tanto los enviados como los recibidos y se hace clic en el botón “leer” del mensaje deseado.
Respuesta	Se muestran los detalles del mensaje.

RF-022	Enviar mensajes.
Propósito	El usuario podrá enviar o responder mensajes a otros usuarios en relación a una habitación.
Entrada	Mensaje
Proceso	El usuario envía mensajes a otros usuarios haciendo clic al botón “Enviar Mensaje” que aparece en los detalles de una habitación, el listado de mensajes y el listado de habitaciones favoritas.
Respuesta	Se muestra un mensaje informando si el mensaje ha sido enviado correctamente o ha sido erróneo.

RF-023	Desactivar habitación.
Propósito	El usuario podrá desactivar una habitación para que no se muestre en el listado de habitaciones y no pueda ser buscada.
Entrada	Identificadora habitación.
Proceso	Estando en su perfil el usuario hace clic sobre la acción “Desactivar” de una de las habitaciones activas de su listado de habitaciones ofertadas.
Respuesta	Se marca la habitación como desactivada en el listado de habitaciones ofertadas. Desaparece la acción “Desactivar” y aparece la acción “Activar”. La habitación ya no se muestra en el listado de habitaciones

RF-024	Activar habitación.
Propósito	El usuario podrá activar una habitación para que se muestre en el listado de habitaciones y pueda ser buscada.
Entrada	Identificadora habitación.
Proceso	Estando en su perfil el usuario hace clic sobre la acción “Activar” de una de las habitaciones desactivadas de su listado de habitaciones ofertadas.
Respuesta	Se marca la habitación como activa en el listado de habitaciones ofertadas. Desaparece la acción “Activar” y aparece la acción “Desactivar”.

### **2.3.3 Atributos del sistema**

#### **2.3.3.1 *Mantenimiento***

La aplicación deberá ser diseñada para que sea fácil realizar su mantenimiento, por lo que se debe utilizar una arquitectura en la aplicación que ayude a ello, además de cumplir con los principios de diseño.

#### **2.3.3.2 *Portabilidad***

La aplicación deberá ser portable permitiendo que se pueda hacer uso de esta en diferentes sistemas operativos, dependiendo únicamente de que el usuario tenga un navegador web.

Por otra parte, la aplicación podrá ser desplegada fácilmente en cualquier servidor web, sin realizar modificaciones. En cuanto a el servidor de base de datos se debe poder cambiar a diferentes motores de bases de datos relacionales, sin cambiar el código de la aplicación.

#### **2.3.3.3 *Seguridad***

La aplicación tiene que tener un sistema de autenticación para que se pueda acceder a las funcionalidades avanzadas de la aplicación. Conjuntamente, se hará uso de un mecanismo para autorizar las acciones de los usuarios sobre cualquier recurso del sistema, siendo permitidas solo las acciones para el propietario de este.

## 3 Análisis

---

En este capítulo se detallará el análisis inicial realizado sobre la aplicación a desarrollar describiendo la funcionalidad y estructura que debe tener la aplicación mediante el uso de diagramas.

### 3.1 Introducción

Este análisis describirá las funcionalidades de la aplicación mediante diagramas UML, que es el lenguaje de modelado de sistemas de software más conocido y esta estandarizado bajo la normativa ISO/IEC 19505-1 e ISO/IEC 19505-2.

UML se utiliza en este proyecto para visualizar los aspectos conceptuales tales como procesos y funciones del sistema de forma gráfica, teniendo así un modelo a la hora de desarrollar la aplicación.

Para la descripción del proyecto se va hacer uso de diagramas UML como:

- Diagramas de casos de uso
- Diagramas de actividad
- Diagrama de clases

### 3.2 Casos de uso

#### 3.2.1 Actores

La aplicación tiene dos actores que son: Usuario no Registrado y Usuario Registrado.



Imagen 3.1 Actores Aplicación

El usuario no registrado podrá realizar tan solo acciones de consulta de información muy básica en la aplicación, así como acciones de registro o inicio de sesión.

El usuario registrado podrá realizar todas las acciones de las que dispone la aplicación, incluyendo las que puede realizar el usuario no registrado.

### 3.2.2 Caso de uso: Usuario no Registrado



Imagen 3.2 Caso de Uso Usuario No Registrado

El usuario no registrado es el más básico de la aplicación, el usuario no registrado tan solo tendrá acceso a contenido principalmente informativo.

Las acciones que puede llevar a cabo este usuario son la búsqueda básica de habitación, ver el listado de habitaciones que se ofertan, ver el detalle básico de una de las habitaciones que se ofertan, así como acciones de inicio de sesión y de registro en la aplicación.

### 3.2.3 Caso de uso: Usuario Registrado

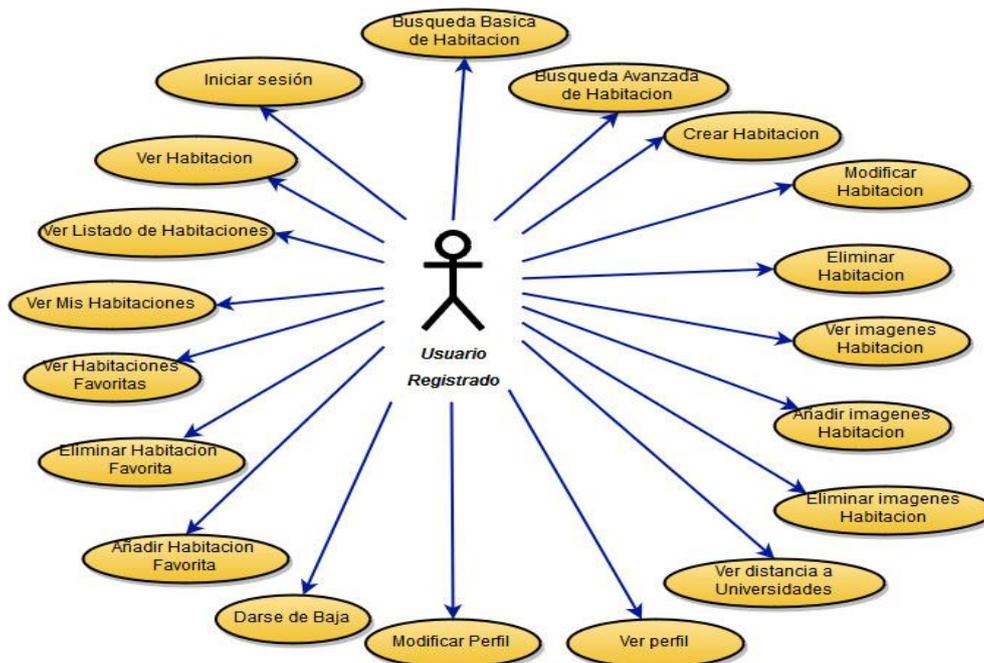


Imagen 3.3 Caso de Uso Usuario Registrado

En usuario registrado podrá realizar acciones adicionales respecto al usuario no registrado que le permitirán obtener información más detallada, así como acceder a funcionalidades más avanzadas de la aplicación.

### 3.2.4 Especificación de los casos de uso

A continuación, se detallan los casos de uso de la aplicación describiendo los pasos que llevan a cabo los usuarios para realizar una operación en el sistema y el comportamiento de este mediante su interacción con los usuarios.

Caso de Uso	Registrarse
Descripción	Permite al usuario registrarse en la aplicación web accediendo así a todas las ventajas que tiene ser un usuario registrado.
Actores	Usuario No Registrado.
Precondiciones	Ninguna.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Registrarse” en el menú de la barra superior.</li> <li>2. El usuario introduce con sus datos el formulario de registro y pulsa “Registrarse”.</li> <li>3. Se validan si los datos son correctos.</li> </ol>
Postcondiciones	Un nuevo usuario debe ser creado en la base de datos y este es redirigido a la pantalla principal con la sesión ya iniciada.
Excepciones	<ol style="list-style-type: none"> <li>4. Si el usuario ha introducido datos incorrectos, el sistema le informará y permitirá volver a internar el registro.</li> </ol>

Caso de Uso	Iniciar Sesión
Descripción	Permite al usuario identificarse para iniciar sesión en la aplicación web.
Actores	Usuario No Registrado.
Precondiciones	Registrarse.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Iniciar Sesión” en el menú de la barra superior.</li> <li>2. El usuario introduce el correo electrónico y la contraseña con las que se ha registrado y pulsa “Iniciar Sesión”.</li> <li>3. Se validan si los datos son correctos.</li> </ol>
Postcondiciones	Se inicia la sesión de usuario y redirige a la pantalla principal.
Excepciones	<ol style="list-style-type: none"> <li>4. Si el usuario ha introducido datos incorrectos o no existentes en el sistema se le informará y permitirá volver a intentar el inicio de sesión.</li> </ol>

Caso de Uso	Ver Listado de Habitaciones
Descripción	El usuario podrá ver el listado de todas las habitaciones que se estén ofertando.
Actores	Usuario No Registrado.
Precondiciones	Ninguna.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a la página principal de la aplicación.</li> <li>2. Se muestran las habitaciones ofertadas.</li> <li>3. El usuario puede navegar por el listado pulsando los enlaces de paginación.</li> </ol>
Postcondiciones	Ninguna.
Excepciones	Ninguna.

Caso de Uso	Búsqueda Básica de Habitación
Descripción	El usuario puede realizar una búsqueda básica en el listado de habitaciones ofertadas.
Actores	Usuario No Registrado.
Precondiciones	Ninguna.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a la página principal de la aplicación.</li> <li>2. El usuario seleccionara los filtros que desea aplicar a la búsqueda de la habitación y pulsara “Buscar”.</li> <li>3. Se muestran los resultados de la búsqueda básica de habitación.</li> </ol>
Postcondiciones	Ninguna.
Excepciones	<ol style="list-style-type: none"> <li>4. Si no se encuentran habitaciones con los filtros seleccionados por el usuario, el sistema informa al usuario y permite volver a realizar una nueva búsqueda.</li> </ol>

Caso de Uso	Ver Habitación
Descripción	Permite al usuario ver el detalle de una de las habitaciones que se ofertan.
Actores	Usuario No Registrado / Usuario Registrado.
Precondiciones	La habitación exista en el sistema.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario pulsa “Ver Habitación” en la habitación que se desea ver del listado de habitaciones de la pantalla principal o de su perfil para los usuarios registrados.</li> </ol>

## Diseño de un portal web para alquiler de habitaciones a estudiantes

	2. Se muestra una nueva página con los detalles de la habitación seleccionada.
Postcondiciones	Ninguna.
Excepciones	3. Si la habitación que se desea ver no existe en el sistema, se redirige al usuario a la pantalla principal y se le informa al usuario.

Caso de Uso	Búsqueda Avanzada de Habitación
Descripción	El usuario puede realizar una búsqueda avanzada en el listado de habitaciones ofertadas.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a la página principal de la aplicación.</li> <li>2. El usuario seleccionara los filtros básicos y avanzados que desea aplicar a la búsqueda de la habitación y pulsara "Buscar".</li> <li>3. Se muestran los resultados de la búsqueda avanzada de habitación.</li> </ol>
Postcondiciones	Ninguna.
Excepciones	4. Si no se encuentran habitaciones con los filtros seleccionados por el usuario, el sistema informa al usuario y permite volver a realizar una nueva búsqueda.

Caso de Uso	Crear Habitación
Descripción	El usuario creara una habitación para ofertarla en la aplicación, indicando todas las características de estas.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario pulsa "Publicar habitación" ya sea en su perfil o en el menú de la barra superior.</li> <li>2. El usuario introduce los datos en el formulario para la creación de habitaciones y pulsa "Dar de Alta".</li> <li>3. Se valida que los datos sean correctos.</li> </ol>
Postcondiciones	Una nueva habitación se debe crear en la base de datos y se redirige a la habitación creada.
Excepciones	4. Si el usuario introduce datos incorrectos se le informa y permite intentar el alta de la habitación nuevamente.

Caso de Uso	Modificar Habitación
Descripción	El usuario podrá modificar cualquiera de sus habitaciones ofertadas.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión. Ser el creador de la habitación que se desea modificar.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>Desde el perfil el usuario selecciona la acción “Modificar” de la habitación que desea modificar.</li> <li>El usuario modifica todos los datos del formulario que desea cambiar y pulsa “Guardar Modificación”.</li> <li>Se validan que todos datos modificados sean correctos.</li> </ol>
Postcondiciones	Se modifican los datos de la habitación en la base de datos y se redirige a la página de la habitación.
Excepciones	<ol style="list-style-type: none"> <li>Si el usuario introduce datos incorrectos se le informa y permite intentar realizar la modificación nuevamente.</li> </ol>

Caso de Uso	Eliminar Habitación
Descripción	El usuario podrá eliminar cualquiera de sus habitaciones ofertadas.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión. Ser el creador de la habitación que se desea modificar.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>Desde el perfil de usuario selección la acción “Eliminar” de la habitación que desea eliminar</li> </ol>
Postcondiciones	Se elimina la habitación de la base de datos y se redirige a el perfil del usuario.
Excepciones	<ol style="list-style-type: none"> <li>Si ocurre un error al eliminar la habitación se informa al usuario.</li> </ol>

Caso de Uso	Ver Imágenes Habitación
Descripción	El usuario podrá ver todas las imágenes de una habitación.
Actores	Usuario Registrado / Usuario No registrado.
Precondiciones	Ninguna.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>El usuario accederá a la pantalla de detalle de la habitación.</li> </ol>

	2. El usuario visualizara las imágenes de la habitación pulsando las flechas laterales del espacio de imágenes.
Postcondiciones	Ninguna.
Excepciones	3. Si no se dispone de imágenes para la habitación, se muestra un gráfico que lo indica.

Caso de Uso	Añadir Imágenes Habitación
Descripción	El usuario puede añadir nuevas imágenes en sus habitaciones ofertadas.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión. Ser el creador de la habitación a la que se desea añadir una imagen.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accederá a la pantalla de modificación de la habitación a la que desea añadir imágenes.</li> <li>2. En el campo imagen pulsa “Elegir archivos”.</li> <li>3. Selecciona todas las imágenes que desea añadir a la habitación.</li> <li>4. Se verifica que las imágenes.</li> </ol>
Postcondiciones	Se guardan las imágenes en la base de datos. Se redirige al usuario a la pantalla de la habitación.
Excepciones	5. Si ocurre un error al añadir las imágenes se informa al usuario y se permite que intente añadir las imágenes otra vez.

Caso de Uso	Eliminar Imágenes Habitación
Descripción	El usuario eliminara cualquiera de las imágenes de una de sus habitaciones ofertadas.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión. Ser el creador de la habitación a la que se desea eliminar una imagen.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accederá a la pantalla de modificación de la habitación a la que desea eliminar alguna de las imágenes.</li> <li>2. Pulsando sobre la “x” que se muestra en cada imagen se eliminará esta.</li> </ol>
Postcondiciones	Se eliminan las imágenes de la base de datos.
Excepciones	3. Si se produce un error al eliminar la imagen se informa al usuario y se permite que intente eliminar las imágenes nuevamente.

Caso de Uso	Ver distancia a Universidades
Descripción	En el detalle de una de las habitaciones ofertadas el usuario podrá ver la distancia a la que se encuentran las universidades respecto a esta.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión. Debe existir datos sobre las distancias a las universidades en la base de datos.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accederá a la pantalla de detalle de la habitación.</li> <li>2. Visualizará la distancia que existe a las universidades en los detalles de la habitación.</li> </ol>
Postcondiciones	Ninguna.
Excepciones	<ol style="list-style-type: none"> <li>3. Si no existen datos sobre distancias se indica en los detalles de las habitaciones.</li> </ol>

Caso de Uso	Ver perfil
Descripción	El usuario accederá a su perfil donde podrá ver toda su información.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario pulsa "Perfil" en el menú de la barra superior.</li> <li>2. Visualizará toda la información de su perfil.</li> </ol>
Postcondiciones	Ninguna.
Excepciones	<ol style="list-style-type: none"> <li>3. Si se produce algún error se le indica al usuario.</li> </ol>

Caso de Uso	Modificar perfil
Descripción	El usuario podrá modificar la información de su perfil.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario pulsa "Perfil" en el menú de la barra superior.</li> <li>2. En la visualización de perfil modifica la información.</li> <li>3. Pulsa "Guardar Datos".</li> <li>4. Se verifica que los datos sean correctos.</li> </ol>
Postcondiciones	Se deben modificar los datos del usuario en la base de datos.

Excepciones	5. Si se produce algún error al modificar el perfil se informa al usuario, pudiendo este volver a intentar modificarlo.
-------------	---

Caso de Uso	Darse de Baja
Descripción	Un usuario puede darse de baja en la aplicación cuando así lo desee.
Actores	Usuario Registrado.
Precondiciones	Inicias Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario pulsa “Perfil” en el menú de la barra superior.</li> <li>2. Pulsa “Dar de Baja”.</li> </ol>
Postcondiciones	Se elimina de la base de datos toda la información de ese usuario.
Excepciones	3. Si se produce un error en el proceso de baja se le indica al usuario, pudiendo este volver a intentarlo.

Caso de Uso	Añadir Habitación favorita
Descripción	Cuando el usuario encuentre una habitación de las ofertadas que resulte de su interés podrá añadirla a su listado de habitaciones favoritas.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a la página de detalle de una habitación.</li> <li>2. Pulsa “Añadir a Favoritas”.</li> <li>3. Se marca la habitación como favorita.</li> </ol>
Postcondiciones	Se debe añadir en base de datos la habitación favorita del usuario y debe mostrar la opción para eliminarla de las favoritas.
Excepciones	4. En caso de error no se añade la habitación a favoritas, dando la opción de volver a intentar añadirla al usuario.

Caso de Uso	Eliminar Habitación favorita
Descripción	El usuario podrá eliminar una habitación de su listado de habitaciones favoritas.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.

Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a la página de detalle de una habitación.</li> <li>2. Pulsa “Eliminar de favoritas”.</li> <li>3. Se marca la habitación como no favorita.</li> </ol>
Flujo Alternativo	<ol style="list-style-type: none"> <li>4. El usuario accede a su perfil.</li> <li>5. En el listado de habitaciones favoritas pulsa sobre la acción “Eliminar”</li> <li>6. Se elimina la habitación del listado de habitaciones favoritas.</li> </ol>
Postcondiciones	Se debe eliminar en base de datos la habitación favorita del usuario y se debe mostrar la opción para añadirla a favoritos.
Excepciones	<ol style="list-style-type: none"> <li>7. En caso de error en el flujo normal o alternativo no se elimina la habitación de favoritas, dando la opción de volver a intentar eliminarla.</li> </ol>

Caso de Uso	Ver Habitaciones favoritas
Descripción	En el perfil el usuario podrá ver el listado de habitaciones favoritas que ha ido añadiendo.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a su perfil.</li> <li>2. Visualiza el listado de habitaciones en la sección habilitada para ello.</li> </ol>
Postcondiciones	Ninguna.
Excepciones	<ol style="list-style-type: none"> <li>5. En caso de que no exista ninguna habitación favorita no se muestra el listado vacío.</li> </ol>

Caso de Uso	Ver mis Habitaciones
Descripción	En el perfil el usuario podrá ver el listado de habitaciones que está ofertando en la aplicación.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a su perfil.</li> <li>2. Visualizara el listado de habitaciones que oferta en la sección habilitada para ella.</li> </ol>
Postcondiciones	Ninguna.
Excepciones	<ol style="list-style-type: none"> <li>3. En caso de que no se estén ofertando habitaciones se muestra el listado vacío.</li> </ol>

Caso de Uso	Ver mensajes
Descripción	El usuario podrá ver los mensajes que le envían otros usuarios solicitando información sobre alguna de las habitaciones que está ofertando.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a su perfil.</li> <li>2. Pulsa “Ver Mensajes”.</li> <li>3. Se dirige al usuario a una página donde se muestran los mensajes enviados y recibidos.</li> <li>4. Pulsa “Leer” sobre el mensaje que desea.</li> </ol>
Postcondiciones	Ninguna.
Excepciones	<ol style="list-style-type: none"> <li>5. Si no existe ningún mensaje se informa al usuario.</li> </ol>

Caso de Uso	Enviar Mensajes
Descripción	El usuario podrá enviar mensajes a otro usuario solicitando información sobre una habitación o respondiendo a un mensaje recibido.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de detalles de una habitación.</li> <li>2. Rellena el formulario de Contacto.</li> <li>3. Pulsa “Enviar Mensaje”</li> <li>4. Se informa que el mensaje ha sido enviado.</li> </ol>
Flujo Alternativo	<ol style="list-style-type: none"> <li>5. El usuario accede a su perfil.</li> <li>6. Pulsa “Ver Mensajes”.</li> <li>7. Pulsa sobre la pestaña “Recibidos”.</li> <li>8. En el mensaje deseado pulsa “Responder”.</li> <li>9. Rellena el formulario y pulsa “Enviar Mensaje”.</li> </ol>
Postcondiciones	Se envía el mensaje al otro usuario almacenándolo en base de datos.
Excepciones	<ol style="list-style-type: none"> <li>10. Si se produce un error al enviar el mensaje se informa al usuario dándole la oportunidad de volver a intentar enviarlo.</li> </ol>

Caso de Uso	Desactivar Habitación.
Descripción	El usuario podrá desactivar la habitación para que no se muestre a los demás usuarios en el listado de habitaciones.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a su perfil.</li> <li>2. En el listado de habitaciones ofertadas pulsa sobre "Desactivar".</li> </ol>
Postcondiciones	Se marca la habitación para que no se muestre a los demás usuarios en el listado.
Excepciones	<ol style="list-style-type: none"> <li>3. Si se produce un error la habitación no se marca pudiendo el usuario intentarlo nuevamente.</li> </ol>

Caso de Uso	Activar Habitación.
Descripción	El usuario podrá activar la habitación para que se muestre a los demás usuarios en el listado de habitaciones.
Actores	Usuario Registrado.
Precondiciones	Iniciar Sesión.
Requisitos no funcionales	Ninguno.
Flujo Normal	<ol style="list-style-type: none"> <li>1. El usuario accede a su perfil.</li> <li>2. En el listado de habitaciones ofertadas pulsa sobre "Activar".</li> </ol>
Postcondiciones	Se marca la habitación para que se muestre a los demás usuarios en el listado.
Excepciones	<ol style="list-style-type: none"> <li>3. Si se produce un error la habitación no se marca pudiendo el usuario intentarlo nuevamente.</li> </ol>

### 3.3 Diagramas de actividad

El diagrama de actividad nos permite representar de forma gráfica los procesos de la aplicación, siendo este útil ya que permite identificar problemas y oportunidades de mejora del proceso, además favorece la comprensión del proceso. En los diagramas que se muestran a continuación podemos ver tres procesos de la aplicación web que se ha desarrollado.

### 3.3.1 Diagramas de actividad Búsqueda

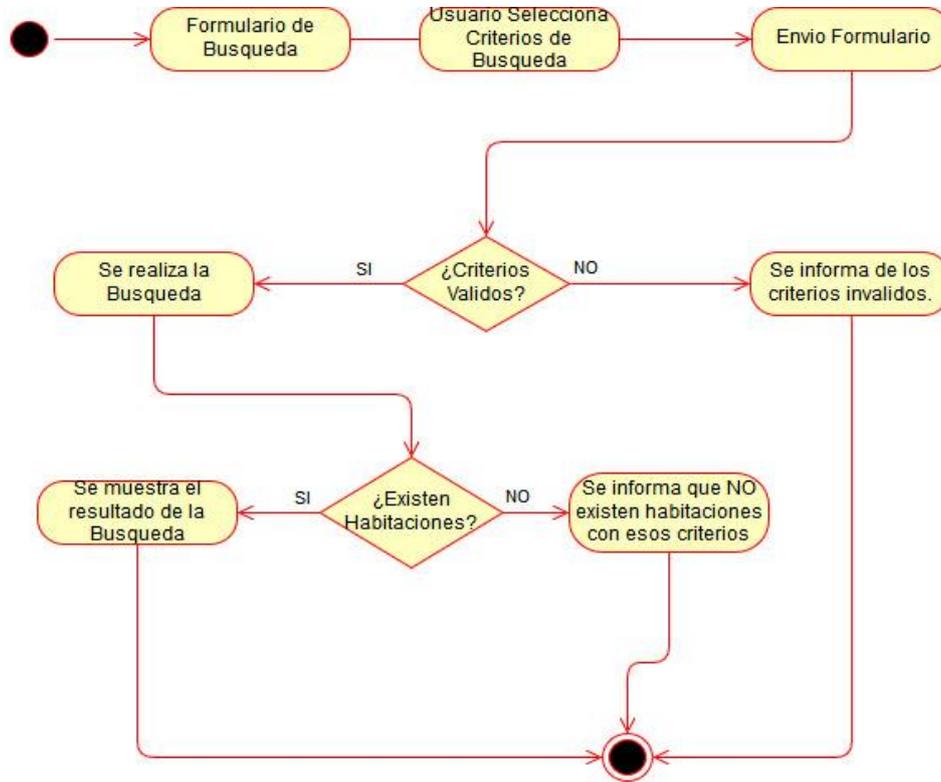


Imagen 3.4 Búsqueda

### 3.3.2 Diagramas de actividad Inicio Sesión

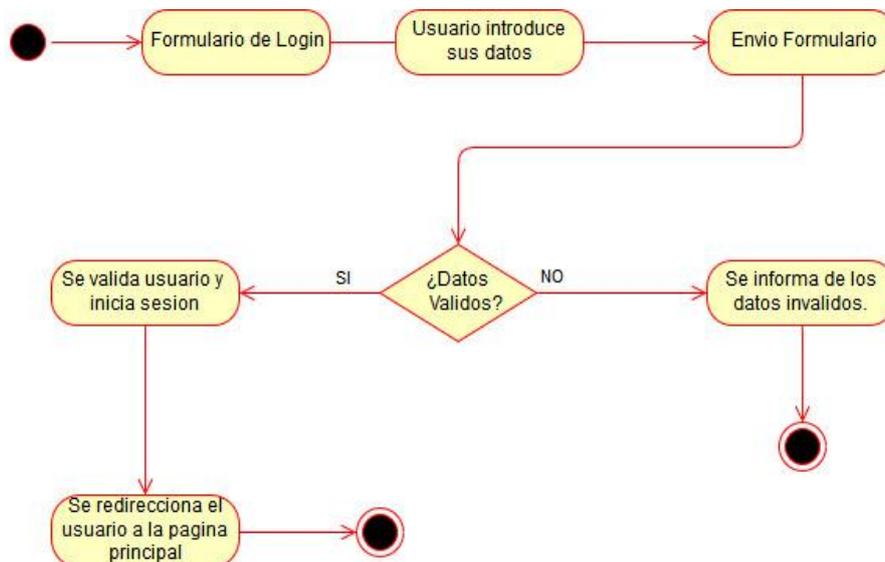


Imagen 3.5 Inicio Sesión

### 3.3.3 Diagramas de actividad Publicar Habitación

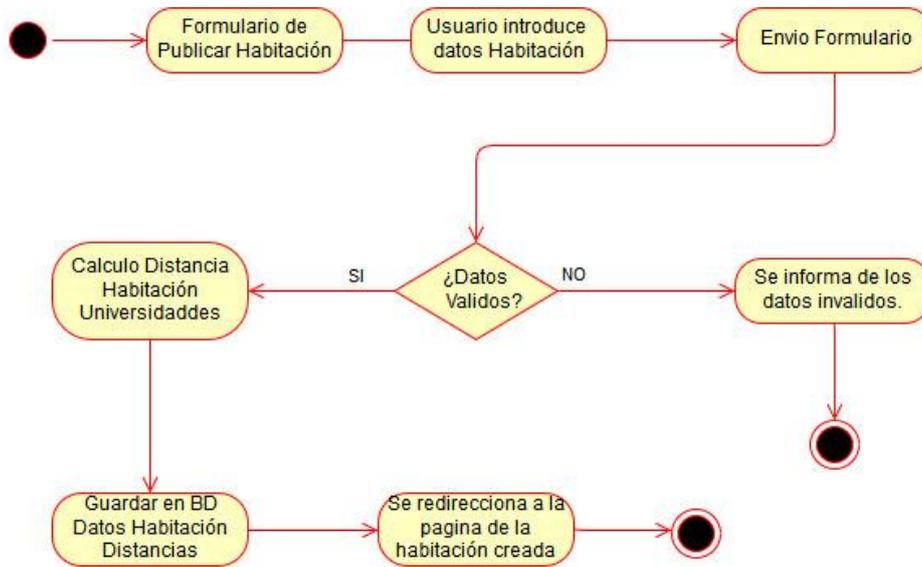


Imagen 3.6 Publicar Habitación

### 3.4 Diagrama de clases

El diagrama de clases representa las principales entidades que intervienen en la aplicación, mostrando sus clases y sus relaciones, para mayor claridad no se han descrito los métodos, ni los atributos.

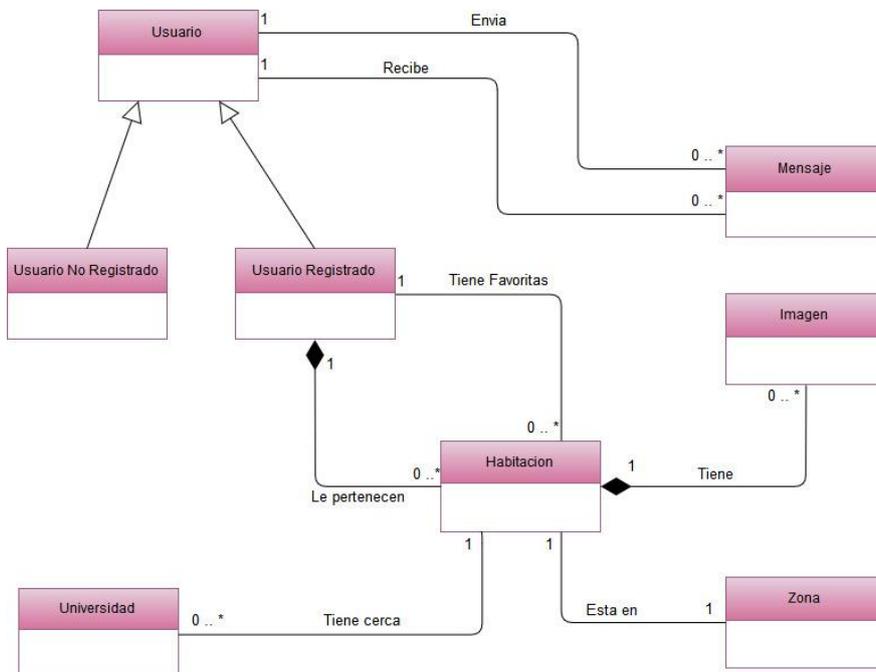


Imagen 3.7 Diagrama de Clases

## 4 Diseño

---

En este capítulo de diseño se plantea como se va a llevar a cabo la implementación de la aplicación en base al análisis que se ha hecho previamente. En este capítulo se presentará el diseño separándolo en dos partes muy importantes la arquitectura física y la arquitectura de software.

### 4.1 Introducción

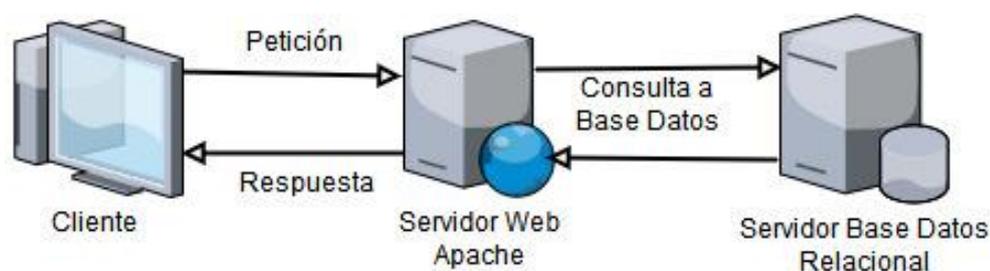
Para el diseño de la aplicación se tomara como base el patrón MVC que consta de tres componentes que son el modelo, la vista y el controlador.

Este patrón de diseño de software se fundamenta en la separación del código de los tres componentes mencionados anteriormente permitiendo así que se puedan separar la lógica de negocio, los datos de la aplicación y la interfaz de usuario.

El patrón de diseño se basa en la idea de permitir que el código de la aplicación sea reutilizable y exista una separación de conceptos buscando así facilitar el desarrollo y mantenimiento de la aplicación.

### 4.2 Arquitectura física

A continuación, se muestra los distintos componentes que forma parte de la arquitectura física de la aplicación.



*Imagen 0.1 Arquitectura física de la aplicación.*

Esta arquitectura es dinámica tanto en la parte del cliente como en la del servidor debido a las tecnologías que se utilizan actualmente en el desarrollo de aplicaciones web y han sido utilizadas en el proyecto como son JavaScript para el dinamismo en el cliente, así como PHP para obtenerlo en la parte del Servidor Web.

Esta arquitectura nos permite modelar la aplicación mediante el modelo de tres capas, persistencia, lógica y presentación.

## 4.3 Arquitectura de software

La aplicación toma como base el patrón MVC añadiendo un componente adicional que es un módulo de enrutamiento que permite procesar las peticiones HTTP que realiza el usuario de la aplicación, direccionando estas al controlador correspondiente. Este módulo aporta más claridad en el código, al separar la responsabilidad de enrutamiento, aportando así una mejora en la mantenibilidad del código.

Este diseño es muy útil en el desarrollo de una aplicación web ya que separa responsabilidades, permite reutilizar código y facilita el trabajo para equipos multidisciplinares pudiendo trabajar independientemente la parte Frontend y Backend de la aplicación sin sufrir afectaciones.

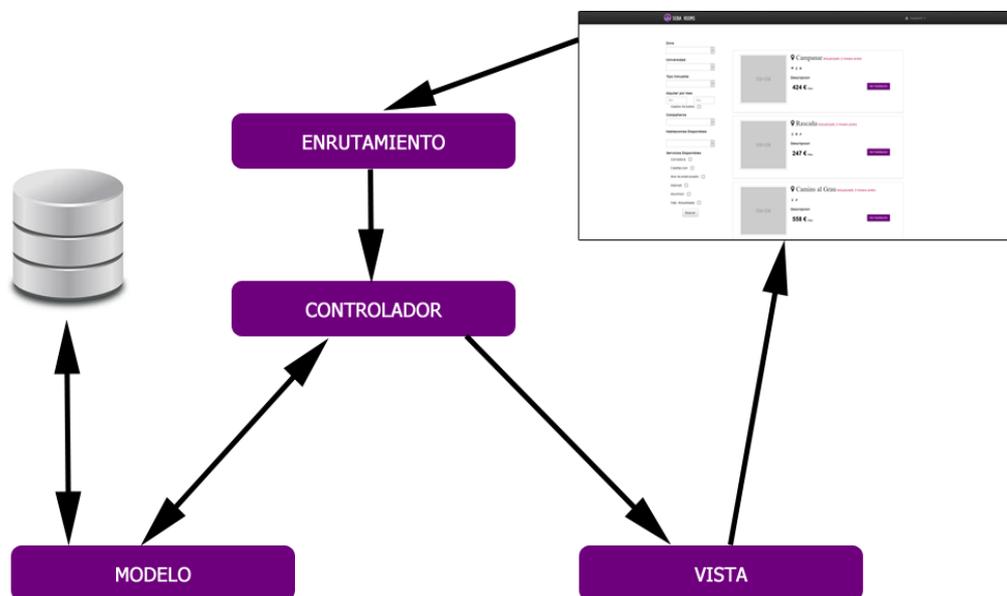


Imagen 0.2 Patrón de diseño Aplicación

Usando la arquitectura física mencionada anteriormente y el patrón MVC podemos modelar fácilmente la aplicación con una arquitectura software de tres capas.

### 4.3.1 Capa de persistencia

En esta capa están representados y almacenados los datos de la aplicación, permitiendo realizar operaciones de consulta, modificación y mantenimiento, todas estas operaciones son solicitadas por la capa de lógica de la aplicación estableciendo así la comunicación entre el servidor web y el servidor de base de datos.

#### 4.3.1.1 Modelo

En la aplicación se representa los datos de la aplicación bajo el modelo que nos ofrece el patrón MVC de Laravel.

El modelo es el componente que representa la información en la base de datos mediante un conjunto de clases para ser gestionada por la aplicación.

Para la gestión de la información de la base de datos y la representación de esta mediante clases se utiliza un ORM que se encarga de hacer un mapeo de datos relacional.

Laravel nos ofrece el ORM Eloquent que permite la creación de los modelos, definir sus relaciones y realizar operaciones CRUD sobre la base de datos mediante objetos PHP sin hacer uso de código SQL. Además, nos permite tener independencia en relación al sistema gestor de base de datos sin realizar ningún cambio en el código.

#### 4.3.1.2 Modelo entidad relación de la BBDD

Para el almacenamiento de los datos de la aplicación se utiliza en este caso una base de datos relacional. Esta base de datos se transforma al modelo mediante ORM.

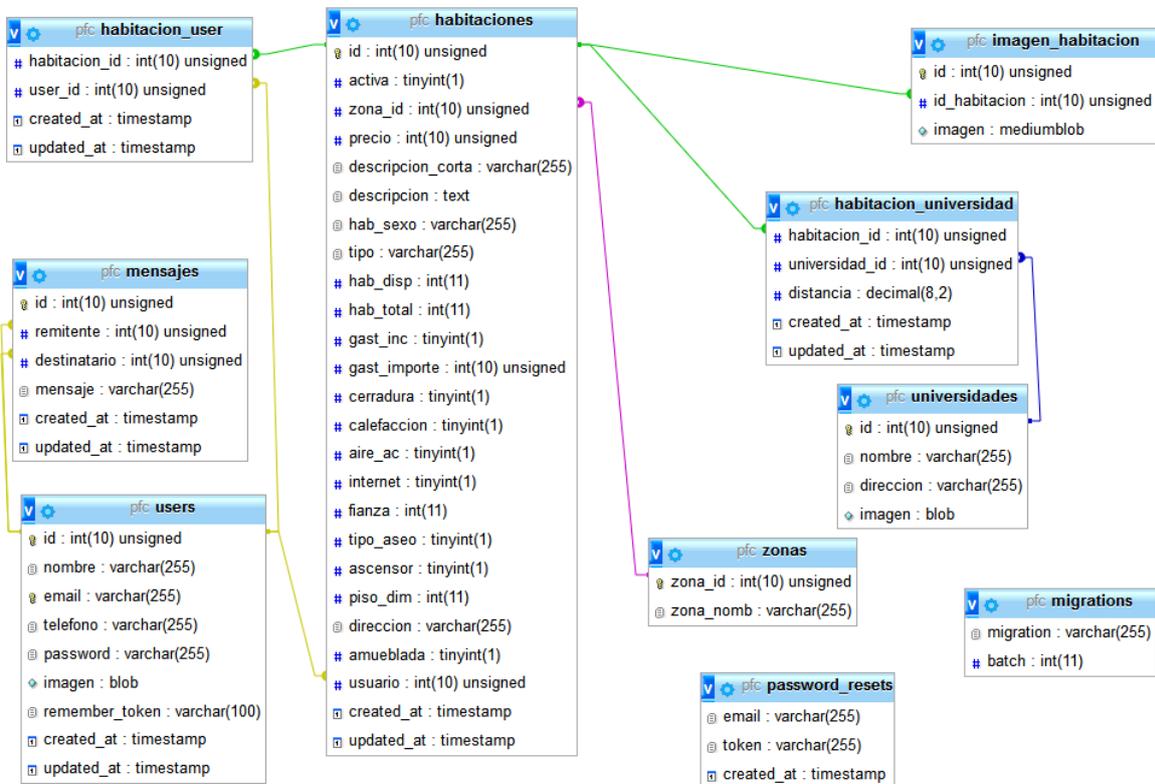


Imagen 0.3 Base de Datos

### 4.3.1.3 Descripción detallada de tablas

**Tabla habitaciones**

Nombre de Columna	Tipo	Descripción
id	int	Identificador de la habitación.
activa	int	Indica si se muestra en el listado.
zona_id	int	Identificador de Zona.
precio	int	Precio mensual alquiler.
descripción_corta	varchar	Descripción breve de la habitación.
descripción	text	Descripción larga de la habitación.
hab_sexo	varchar	Preferencia de chicos, chicas o ambos.
tipo	varchar	Tipo de propiedad de habitación.
hab_disp	int	Numero Habitaciones disponibles en la propiedad.
gast_inc	int	Indica si se incluyen gastos en el precio.
gast_importe	int	Gastos medios en caso de no venir incluidos
cerradura	int	Indica si la habitación tiene cerradura.
calefaccion	int	Indica si la habitación dispone de calefacción.
aire_ac	int	Indica si la habitación tiene aire acondicionado.
internet	int	Indica si hay internet.
fianza	int	Indica la fianza.
tipo_aseo	int	Indica si el aseo es compartido o privado.
ascensor	int	Indica si la propiedad dispone de ascensor.
piso_dim	int	Indica las dimensiones de la propiedad.
direccion	varchar	Dirección de la propiedad donde está la habitación.
amueblada	int	Indica si la habitación esta amueblada.
usuario	int	Identificador usuario creador habitación.
created_at	date	Fecha creación de habitación.
update_at	date	Fecha actualización habitación.

**Tabla users**

Nombre de Columna	Tipo	Descripción
id	int	Identificador de Usuario
nombre	varchar	Nombre del Usuario.
email	varchar	Correo electrónico Usuario.
telefono	varchar	Telefono Usuario
password	varchar	Contraseña encriptada Usuario.
imagen	blob	Imagen de perfil Usuario.
remember_token	varchar	Preferencia de chicos, chicas o ambos.
created_at	date	Fecha creación Usuario.
update_at	date	Fecha actualización Usuario.



### Tabla universidades

Nombre de Columna	Tipo	Descripción
id	int	Identificador de Universidad
nombre	varchar	Nombre de Universidad.
direccion	varchar	Dirección de Universidad.
imagen	blob	Imagen de Universidad.

### Tabla zonas

Nombre de Columna	Tipo	Descripción
zona_id	int	Identificador de Zona
Zona_nomb	varchar	Nombre de Zona.

### Tabla mensajes

Nombre de Columna	Tipo	Descripción
id	int	Identificador de Universidad
remitente	int	Identificador Usuario remitente.
destinatario	int	Identificador Usuario destinatario.
mensaje	varchar	Mensaje.
asunto	varchar	Asunto del mensaje.
created_at	date	Fecha envió mensaje.
updated_at	date	Fecha actualización mensaje.

### Tabla habitacion\_user

Nombre de Columna	Tipo	Descripción
habitacion_id	int	Identificador de Habitación Favorita.
user_id	int	Identificador Usuario.
created_at	date	Fecha creación.
updated_at	date	Fecha actualización.

### Tabla habitacion\_universidad

Nombre de Columna	Tipo	Descripción
habitacion_id	int	Identificador de Habitación.
universidad_id	int	Identificador Universidad
distancia	decimal	Distancia de Habitación a Universidad.
created_at	date	Fecha creación.
updated_at	date	Fecha actualización.

### Tabla imagen\_habitacion

Nombre de Columna	Tipo	Descripción
id	int	Identificador de Imagen.
habitacion_id	int	Identificador Habitación.
imagen	mediumblob	Imagen habitación.

### Tabla password\_resets

Nombre de Columna	Tipo	Descripción
email	varchar	Correo electrónico Usuario.
token	varchar	Token aleatorio para reiniciar password.
created_at	date	Fecha creación.

### Tabla migrations

Nombre de Columna	Tipo	Descripción
migration	varchar	Nombre migration.
batch	int	Numero de ejecución.

## 4.3.2 Capa de presentación

Esta capa es la que permite al usuario interactuar con el sistema, se encarga de recoger la información del usuario y comunicarse con la capa de lógica. Una vez la capa de lógica ha procesado la información esta capa se encarga de generar la presentación mediante vistas.

### 4.3.2.1 Vista

La vista se encarga presentar toda la información del modelo que nos proporciona el controlador. Estas contienen el código HTML con el que se va presentar la información al usuario.

Laravel nos ofrece el motor de plantillas Blade para generar las diferentes salidas que puede tener la aplicación, el motor aporta a la aplicación un gran modularidad gracias a la existencia de la herencia de plantillas.

Otra de las ventajas que nos otorga el motor es que gracias al uso de condicionales y bucles permite tener una menor cantidad de código repetido, controlar en qué casos se muestra el contenido y tener un código más limpio.



### 4.3.2.2 Mockups web

En este apartado se mostrará mediante mockups una representación del diseño de las principales páginas de la aplicación. Estos mockups enseñan cómo se visualizará el contenido, demuestran las funcionalidades básicas de manera estática y permiten revisar la parte visual de la aplicación fácilmente antes de realizar el diseño definitivo.

#### Vista listado habitaciones

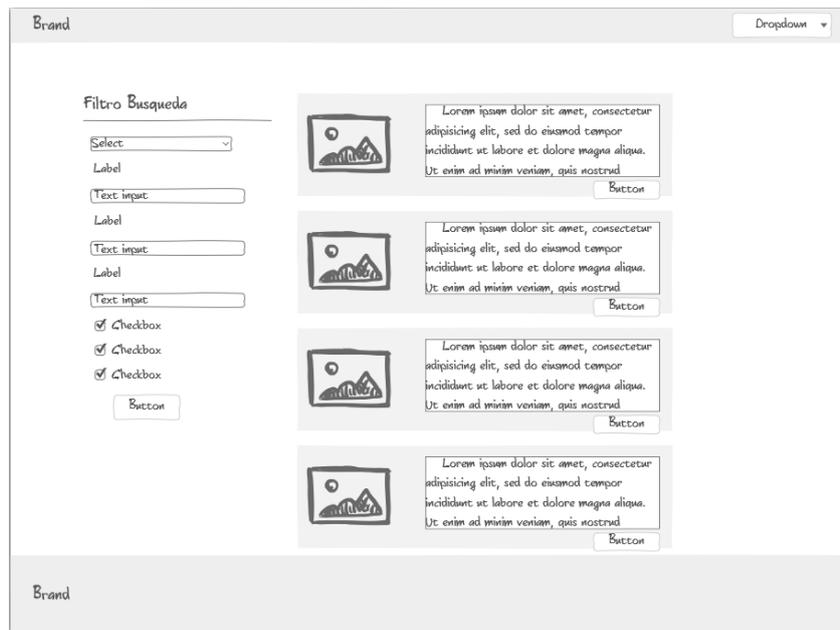


Imagen 0.4 Mockup Vista Listado Habitaciones

#### Vista detalle habitación

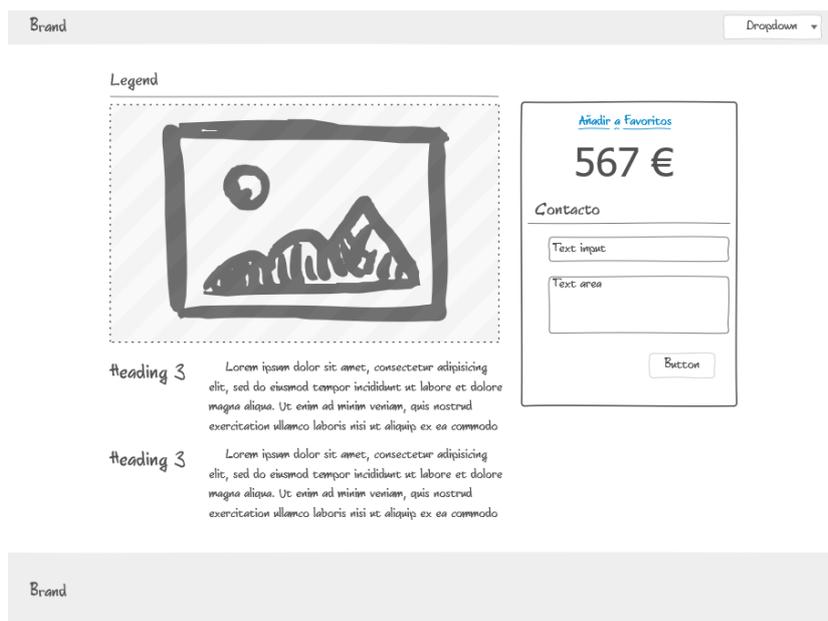


Imagen 0.5 Mockup Vista Detalle Habitación

## Vista perfil

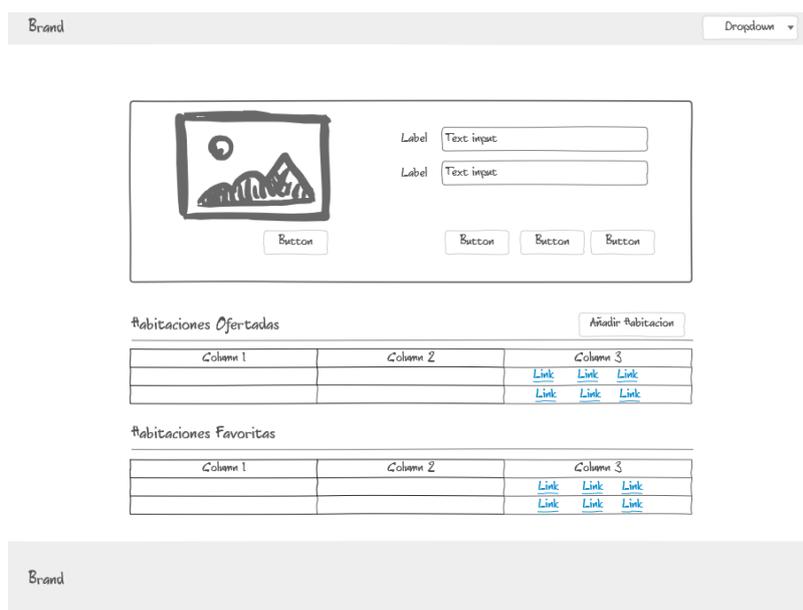


Imagen 0.6 Mockup Vista Perfil

### 4.3.3 Capa de lógica

Esta capa se encarga de procesar las interacciones del usuario con la aplicación y realizar los cambios que sean necesarios en el modelo o en la vista. En nuestra aplicación esta capa está representada por el enrutamiento y controlador.

#### 4.3.3.1 Controlador

El controlador es el que se encarga de responder a las acciones que se solicitan en la aplicación, gestionando toda la lógica de negocio.

Es además el encargado de invocar peticiones al modelo cuando se requiere y enlazar estos con las vistas, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación, como puede ser visualizar un elemento, realizar una búsqueda, etc.

#### 4.3.3.2 Enrutamiento

El enrutamiento que nos ofrece Laravel permite determinar cuál es la acción que se ejecuta ante una petición HTTP que realiza el usuario de la aplicación y también que controlador la debe ejecutar.

Todas las rutas de la aplicación están en un mismo fichero facilitando así su mantenimiento, además permiten crear grupos de rutas aplicándole algunas reglas para

permitir su acceso para por ejemplo solo usuarios autorizados, así como validar el formato de los parámetros que se puedan recibir.

#### 4.3.3.3 Mapa web

A continuación, se muestra una imagen del mapa web de la aplicación donde podremos observar una vista general del contenido. Esto permite al usuario conocer lo que ofrece la aplicación de forma fácil y la mejor forma de navegar por esta.

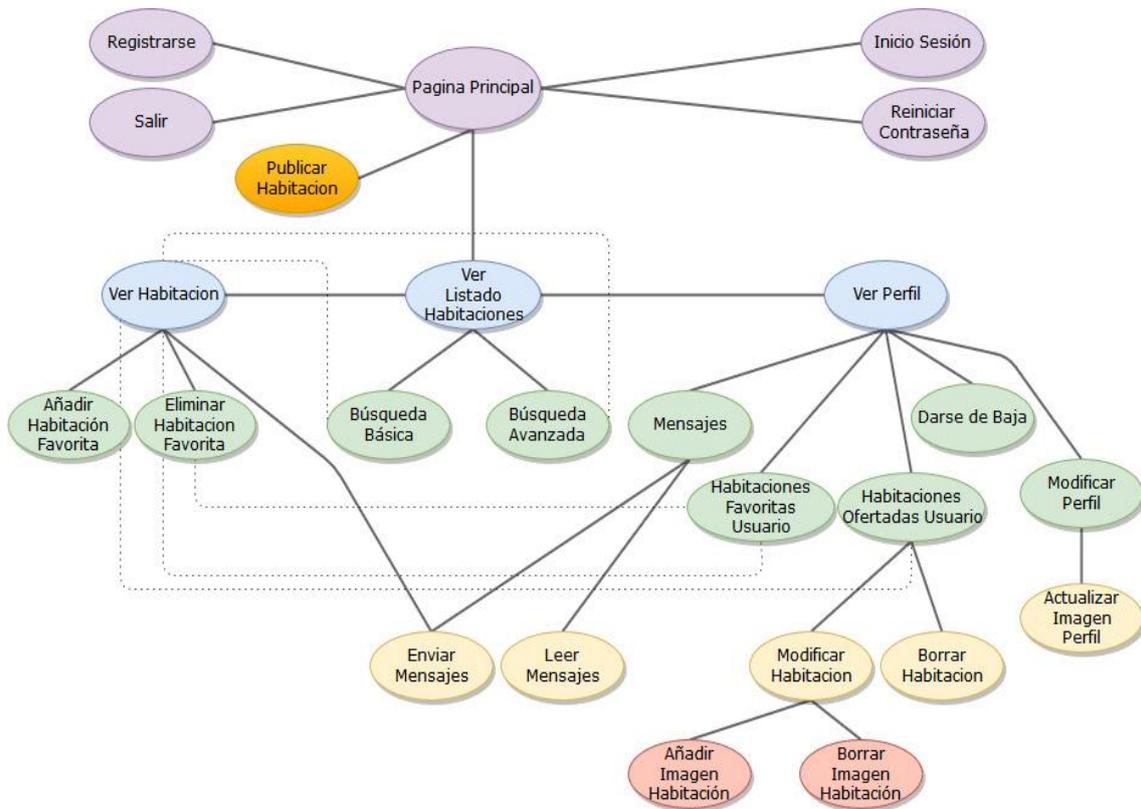


Imagen 0.7 Mapa web

# 5 Desarrollo

---

En este capítulo se pretende mostrar las tecnologías con las cuales se ha construido la aplicación, así como el entorno y todas las herramientas que se han usado para la elaboración de la web.

## 5.1 Tecnologías

Actualmente existen gran número de tecnologías para el desarrollo de aplicaciones web, para llevar a cabo el desarrollo de la aplicación se ha optado por hacer uso de tecnologías que más se suelen utilizar en la industria informática.

A continuación, se detallarán las tecnologías que se van a usar diferenciando entre la parte frontend y la parte backend.

### 5.1.1 FrontEnd

En la parte frontend están todas las tecnologías que se encargan de interactuar con el usuario y que el usuario puede ver. Estas tecnologías son ejecutadas del lado del cliente por el navegador web.

#### 5.1.1.1 HTML

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), este lenguaje de marcado es utilizado para el desarrollo de páginas web, para crear y representar visualmente la página. Determina el contenido de la página web, pero no su funcionalidad.

El lenguaje HTML es un estándar a cargo de la World Wide Web Consortium (W3C), organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web.

#### 5.1.1.2 CSS

Las hojas de estilo en cascada o CSS, Siglas en inglés de Cascading StyleSheets es un lenguaje creado para controlar el aspecto o presentación de documentos estructurados escritos en un lenguaje de marcado como HTML.

El CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes.



En la aplicación se ha hecho uso de las hojas de estilo para definir la presentación de los elementos de los documentos HTML.

### **5.1.1.3 JavaScript**

JavaScript es un lenguaje de programación interpretado, se utiliza principalmente en su forma del lado del cliente implementando como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

El uso más común de JavaScript y por tanto el que se ha aplicado en el desarrollo de la aplicación es la interacción con documentos HTML, mediante la manipulación del árbol DOM (Document Object Model), así como el manejo de eventos.

### **5.1.2 BackEnd**

En la parte backend se encuentran todas las tecnologías que se ejecutan del lado del servidor. Estas tecnologías son las encargadas de procesar las peticiones que los usuarios efectúan desde el frontend.

#### **5.1.2.1 PHP**

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

Fue creado originalmente por Rasmus Lerdorf, es un lenguaje libre publicado bajo la licencia PHP, siendo la implementación principal la producida por The PHP Group.

El gran parecido que posee con lenguajes de programación tan comunes como C, C++ o Java, su gran popularidad, la cantidad de documentación existente y la posibilidad de ejecutarse en la mayoría de servidores web son algunas de las ventajas más importantes del lenguaje PHP.

#### **5.1.2.2 Laravel**

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP. Su filosofía es desarrollar código de forma elegante y simple. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP.

Las principales características que nos ofrece este framework son: sistema de enrutamiento, un motor de plantillas, sistema ORM, soporte para manejo de dependencias, así como el uso del patrón MVC.

En el proyecto se ha hecho uso de este framework y del lenguaje PHP para el desarrollo de toda la parte del lado del servidor.

### 5.1.2.3 MySQL

Para la persistencia de datos de la aplicación se ha optado por SQL y el sistema gestor de base de datos MySQL.

MySQL es uno de los sistemas de gestión de bases de datos relacionales open source más populares, donde se almacenarán los datos de la aplicación, comunicándose así con los modelos de esta para obtener y actualizar datos de la aplicación.

## 5.2 Entorno de desarrollo



*Imagen 0.1 Laragon entorno WAMP*

En el desarrollo de la aplicación se ha hecho uso de una suite de desarrollo para PHP que funciona sobre Windows y que ha sido especialmente diseñada para trabajar con el framework Laravel.

La suite que se usa es Laragon la cual proporciona un entorno de desarrollo rápido, potente y aislado.

Este entorno es un WAMP (Windows, Apache, MySQL, PHP), es fácil de instalar, permite crear la estructura de un proyecto con Laravel de forma sencilla y además evita todo el proceso de configuración del entorno de desarrollo permitiendo que se pueda ir directamente al desarrollo de la aplicación.

## 5.3 Herramientas utilizadas

### 5.3.1 PhpStorm

El Entorno de desarrollo Integrado (IDE) que se ha utilizado para el desarrollo es PhpStorm, este es un IDE comercial por lo cual se ha utilizado una licencia educativa que nos permite acceder a todas sus funcionalidades.

PhpStrom muy completo ya que ofrece la gran variedad de características que nos ofrecen todos los IDE, además de todo lo necesario para trabajar con el lenguaje PHP y soporte para el framework Laravel.

### 5.3.2 phpMyAdmin

Para la administración de MySQL se usa la herramienta phpMyAdmin que funciona a través de una interface web.

Mediante esta herramienta se pueden crear y eliminar Bases de Datos, crear, eliminar y modificar tablas, borrar, editar y añadir campos, administrar claves, privilegios y ejecutar cualquier sentencia SQL.

### 5.3.3 Composer

Composer es un manejador de dependencias de proyectos desarrollados en PHP, es decir nos permite gestionar de manera ágil los paquetes de software en los que se basa un nuestro proyecto.

Esta herramienta nos permite declarar las librerías que deseamos utilizar en nuestro proyecto y nos ayuda a resumir las tareas de descarga y mantenimiento de las librerías.

La suite de desarrollo Laragon nos ofrece ya esta herramienta evitando así preocuparnos de su instalación.

### 5.3.4 Github

Se ha hecho uso en el proyecto del sistema de control de versiones Git junto con la plataforma de desarrollo colaborativo GitHub.

El uso de Git nos ayuda a mantener todo el histórico de cambios que se han producido en el proyecto, pudiendo volver a algún punto de estos, también nos permite crear ramas de desarrollo paralelas de distintas funcionalidades.

Por su parte GitHub nos permite tener nuestro código fuente almacenado en la nube, pudiéndolo gestionar haciendo uso de Git. Además de ofertarnos características como pueden ser la creación de una wiki para el proyecto, información sobre el trabajo de los desarrolladores en el proyecto o bifurcaciones que hagan de este, la posibilidad de gestionar el proyecto mediante Kanban o funcionalidades propias de una red social como puede ser tener seguidores del proyecto.

### 5.3.5 Artisan

La herramienta Artisan es una de las herramientas que nos proporciona el framework Laravel para el desarrollo de aplicaciones.

Esta herramienta es una interfaz de línea de comandos donde el desarrollador puede dar instrucciones en forma de línea de comandos para realizar diferentes tareas durante el desarrollo e incluso cuando la aplicación se encuentra en producción.

## 5.4 Estructura

La estructura de directorios de la aplicación que se ha usado es la recomendada por Laravel.

### 5.4.1 Directorio app

Es el directorio más importante de nuestro proyecto ya que es el que contiene la mayor parte de la aplicación como son las clases que ofrecen funcionalidad a la aplicación, archivos de configuración y demás ficheros importantes para la aplicación.

El directorio app tiene varios subdirectorios, pero el más importante y de los más utilizados es el directorio Http en el cuál ubicaremos nuestros Controllers, Middlewares y Requests en sus carpetas correspondientes, además dentro del subdirectorio Http encontremos también el archivo routes.php donde escribiremos las rutas de la aplicación.

A nivel de la raíz del directorio app se ubicarán los modelos, aunque igual es posible estructurarlos de la forma que queramos.

### 5.4.2 Directorio config

La configuración tanto para el framework como para la aplicación se mantiene en este directorio. En la configuración de Laravel existe un conjunto de archivos PHP que contienen matrices clave-valor. Entre los archivos más usados del directorio config se encuentran:

- app.php: En este archivo nos puede interesar configurar el lenguaje de nuestra aplicación, la zona horaria, los providers y alias de las clases más comunes.
- database.php: En este archivo podemos configurar principalmente el motor de base de datos al cuál deseamos conectarnos.



### 5.4.3 Directorio database

Aquí se encuentran los archivos relacionados con el manejo de la base de datos. Dentro de este directorio se encuentran:

- **factories:** Aquí escribiremos nuestras fábricas de modelos, que son muy útiles para la carga de datos de prueba y almacenar estos en la base de datos.
- **migrations:** Sistema que nos permite la creación de la base de datos mediante programación orientada a objetos.
- **seeds:** Almacena la clase que se encarga de rellenar con datos de prueba la base de datos haciendo uso para ello de los factories.

### 5.4.4 Directorio public

Dentro de este directorio están todos los recursos estáticos de nuestra aplicación, es decir, archivos css, js, imágenes y fuentes.

Es recomendable crear una carpeta por cada tipo de recurso.

### 5.4.5 Directorio resources

Dentro de este directorio se encuentran los subdirectorios:

- **assets:** Aquí se ubican todos los archivos less de nuestra aplicación.
- **lang:** Aquí se encuentran todos los archivos de internacionalización, es decir, los archivos para poder pasar nuestro proyecto de un idioma a otro. Normalmente habrá una carpeta por cada idioma.
- **views:** Aquí se ubican las plantillas php.blade que son llamadas por los controladores pasándoles los datos para que estas generen la respuesta final que se le muestra al usuario.

# 6 Pruebas

En este capítulo se tiene como objetivo presentar información sobre la calidad de la aplicación que se ha desarrollado. A continuación, se describen las pruebas que se han llevado a cabo para comprobar y asegurar el correcto funcionamiento de la aplicación

## 6.1 Visualización en diferentes dispositivos

Se prueba la aplicación en diferentes dispositivos para comprobar que se visualiza de forma correcta. Los dispositivos en lo que se ha probado son un ordenador de sobremesa con resolución 1920 x 1080, un portátil con resolución 1366 x 768 y un dispositivo móvil iPhone 6 Plus.

Aunque no se ha especificado que la aplicación deba funcionar en dispositivos móviles mediante alguna APP se realiza la prueba en móviles ya que estos podrán acceder a la aplicación a través de un navegador web.

### Visualización en ordenador de sobremesa, resolución 1920 x 1080

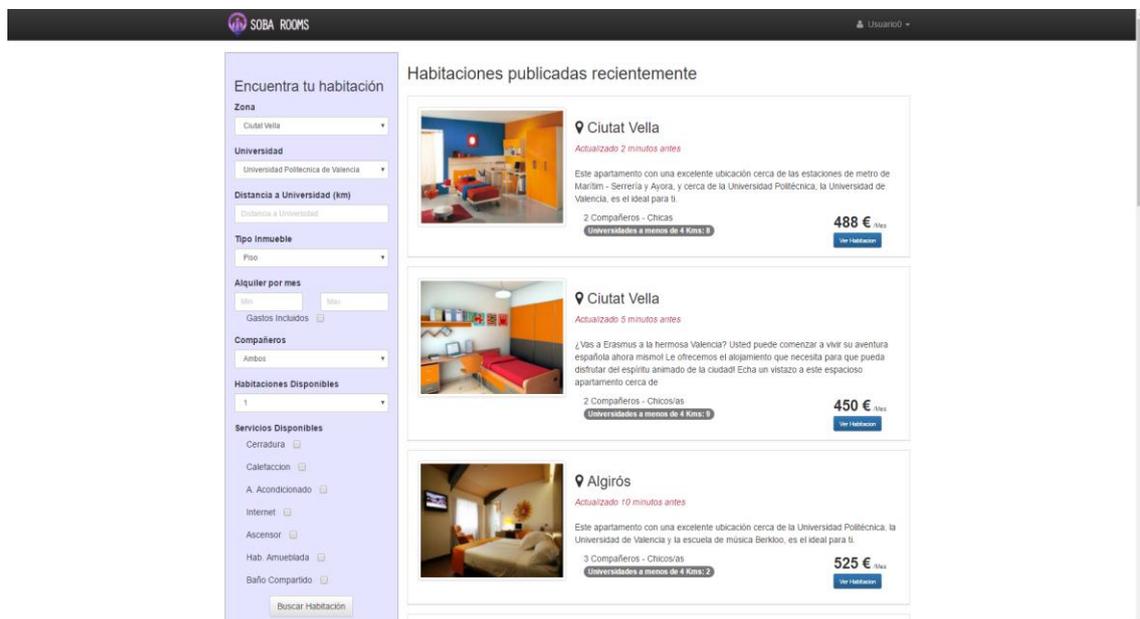


Imagen 6.1 Visualización en sobremesa

# Diseño de un portal web para alquiler de habitaciones a estudiantes

## Visualización en portátil, resolución 1366 x 768

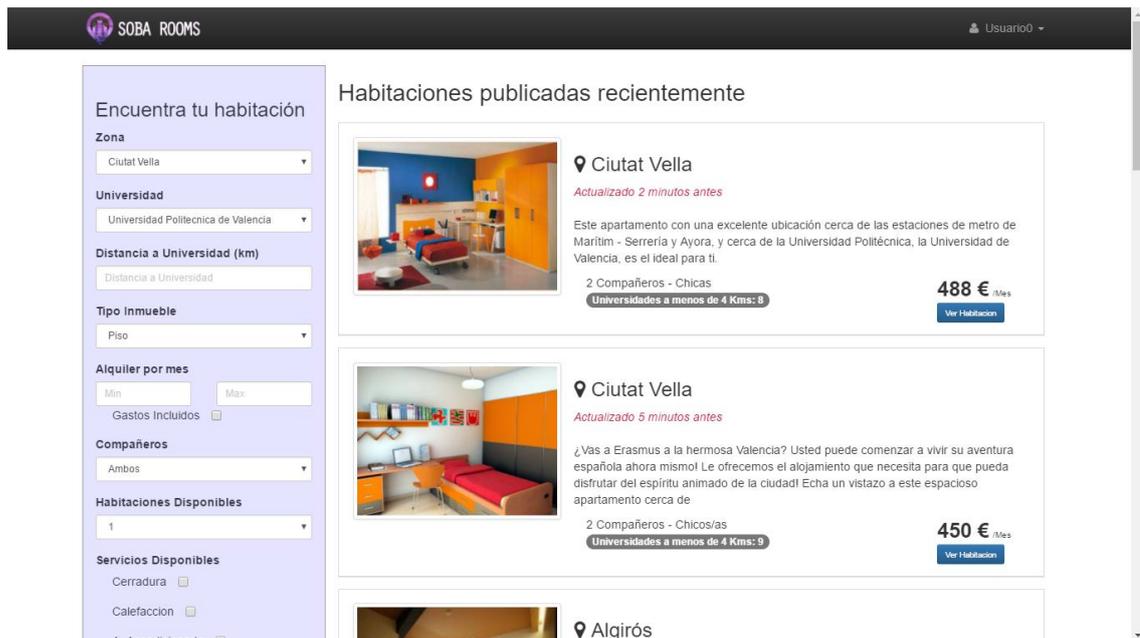


Imagen 6.2 Visualización portátil

## Visualización en móvil, iPhone 6 Plus

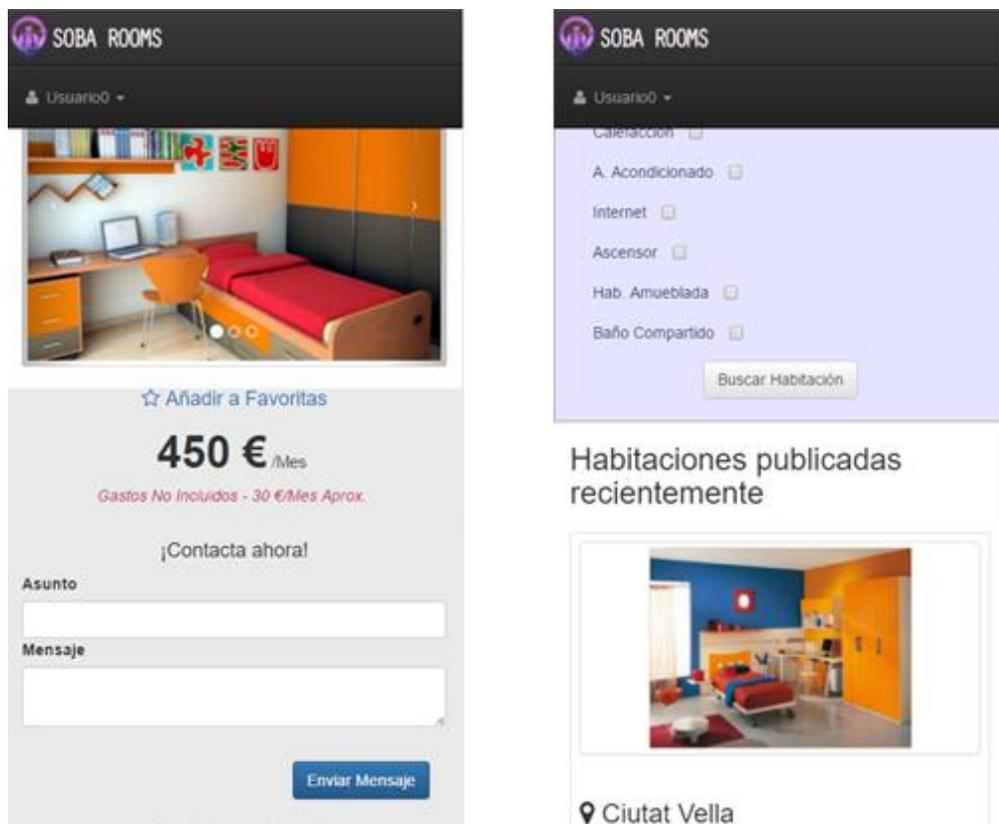


Imagen 6.3 Visualización móvil

## 6.2 Visualización en diferentes navegadores

En esta prueba se verificará que la aplicación web se visualiza correctamente y no presenta ningún tipo de error en todos los navegadores para los cuales ha sido desarrollada.

### Visualización en Microsoft Edge

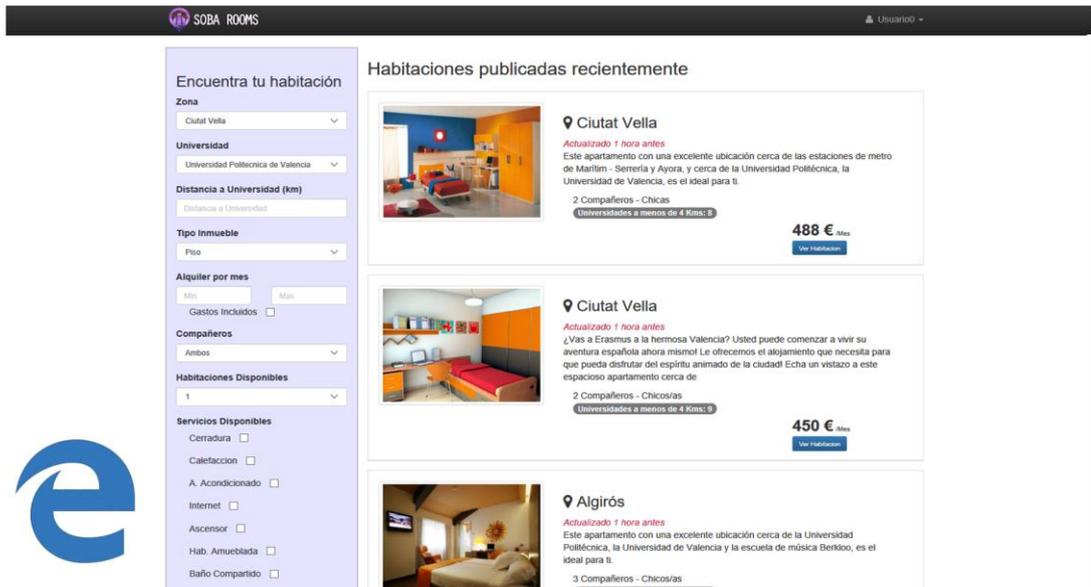


Imagen 6.4 Visualización en navegador Edge

### Visualización en Mozilla Firefox

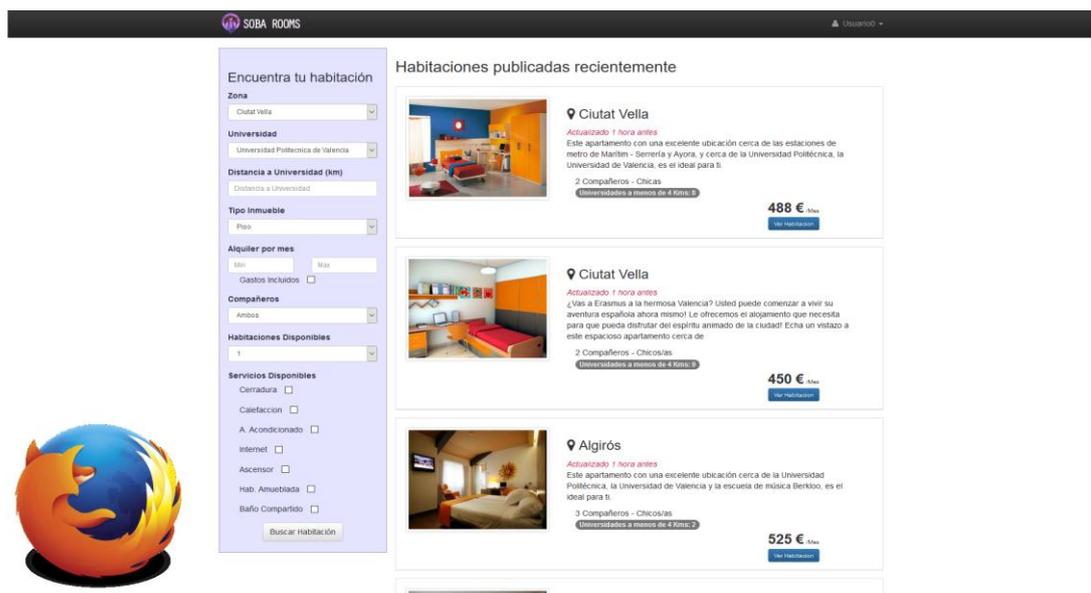


Imagen 6.5 Visualización en navegador Firefox

## Visualización en Google Chrome

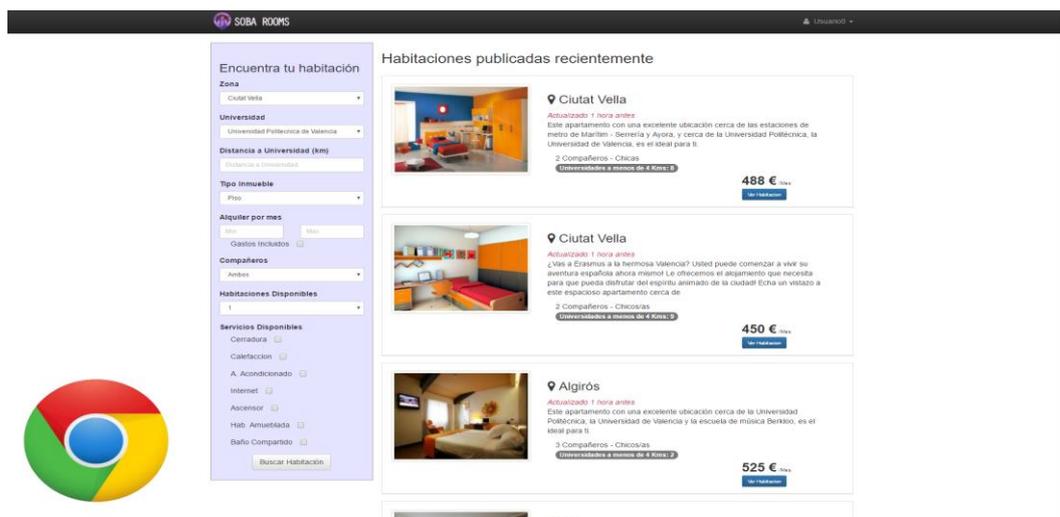


Imagen 6.6 Visualización en navegador Chrome

## 6.3 Análisis estático

Mediante el análisis estático de la aplicación web podremos medir la calidad del código mediante diferentes métricas, este análisis se realiza sobre el código fuente de la aplicación sin necesidad de ejecutarla.

Para la realización de este análisis se ha usado la plataforma para evaluar código fuente Sonarqube, esta plataforma utiliza diversas herramientas de análisis estático de código fuente para obtener métricas que pueden ayudar a mejorar la calidad del código.

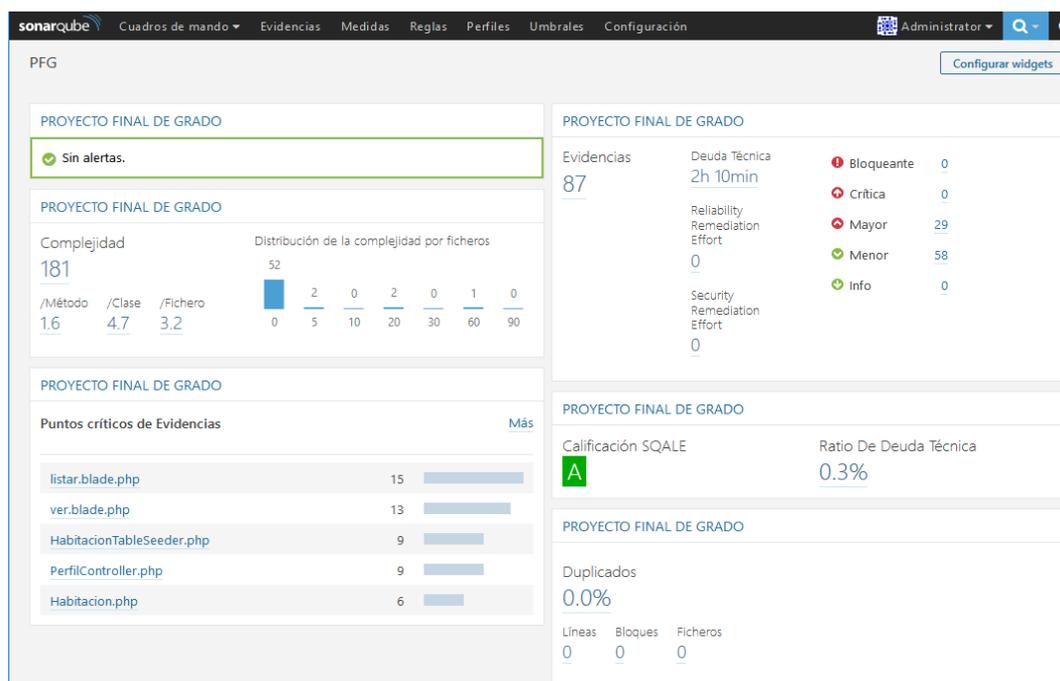


Imagen 6.7 Cuadro de mandos SonarQube

Haciendo un análisis obtenemos unos buenos resultados como podemos ver en la imagen del cuadro de mandos de Sonarqube, donde nos indica la existencia de 87 evidencias siendo mayormente de tipo Menor, también nos indica una buena calificación SQALE, así como un 0.0% en duplicados.

En cuanto a la complejidad ciclométrica total del proyecto podemos ver que esta es de 181, sin embargo, este valor no nos aporta demasiado a la hora de determinar qué acciones llevar a cabo para mejorar la aplicación. Por tanto, nos centramos en el valor de la Complejidad/Método y Complejidad/Clase, donde podemos observar que el proyecto no tiene mucha complejidad ciclométrica.

Por otra parte, en los puntos críticos de evidencias podemos ver como las plantillas en Blade son uno de nuestros mayores puntos críticos, accedemos al detalle de listar.blade.php donde podemos ver el detalle de las evidencias encontradas en este fichero.



Imagen 6.8 Evidencias en Fichero del proyecto

En la imagen podemos observar el detalle de evidencias para el fichero listar.blade.php, donde se observa que gran parte de las evidencias hacen referencia a líneas muy largas lo cual dificulta la lectura del código al tener que hacer scroll horizontal.



## 6.4 Pruebas de uso

Mediante esta prueba verificaremos que la aplicación se comporta como se ha descrito en la especificación de los casos de uso, se mostraran a continuación algunos de los casos de uso más representativos de la aplicación que se han probado.

### 6.4.1 Ver Perfil

Una vez se ha iniciado sesión, como redirigidos a la página principal de la aplicación, siendo esta una precondition de nuestro caso de uso procedemos realizar la prueba en sus distintos flujos.

#### Flujo Normal

En la barra de navegación superior se selecciona la opción “Perfil”.



Imagen 6.9 Ir a Perfil

Se muestra entonces toda la información del perfil de usuario como se puede ver en la siguiente imagen.

Jose Luis Avila

Usuario desde: 11/06/2017

Email: correo0@gmail.com

Telefono: 961938380

Ver Mensajes Guardar Datos Dar de Baja

Imagen Guardar Imagen

Habitaciones Ofertadas [+ Publicar Habitación](#)

Activar	Zona	Dirección	Hab. Disp.	Actualizado	Precio	Acciones
<input checked="" type="checkbox"/>	Extramurs	Calle san juan bosco 83 46019	4	13/06/2017	233	<a href="#">o</a> <a href="#">e</a> <a href="#">x</a>
<input checked="" type="checkbox"/>	Algirós	Camino de Vera s/n, Valencia	2	25/06/2017	525	<a href="#">o</a> <a href="#">e</a> <a href="#">x</a>
<input checked="" type="checkbox"/>	Ciutat Vella	Carrer Del Marí Albesa 8, Valencia	2	25/06/2017	450	<a href="#">o</a> <a href="#">e</a> <a href="#">x</a>
<input checked="" type="checkbox"/>	Ciutat Vella	Carrer De José Aguilar 5, Valencia	3	25/06/2017	488	<a href="#">o</a> <a href="#">e</a> <a href="#">x</a>

Habitaciones Favoritas

Zona	Dirección	Hab. Disp.	Actualizado	Precio	Acciones
El Pla del Real	Calle san juan bosco 83 46019	2	11/06/2017	233	<a href="#">o</a> <a href="#">x</a>

Imagen 6.10 Ver Perfil

## 6.4.2 Crear Habitación

Una vez se ha iniciado sesión, como redirigidos a la página principal de la aplicación, siendo esta una precondition de nuestro caso de uso procedemos realizar la prueba en sus distintos flujos.

### Flujo Normal

Estando en el perfil de usuario se pulsa sobre el botón “Publicar Habitación”.

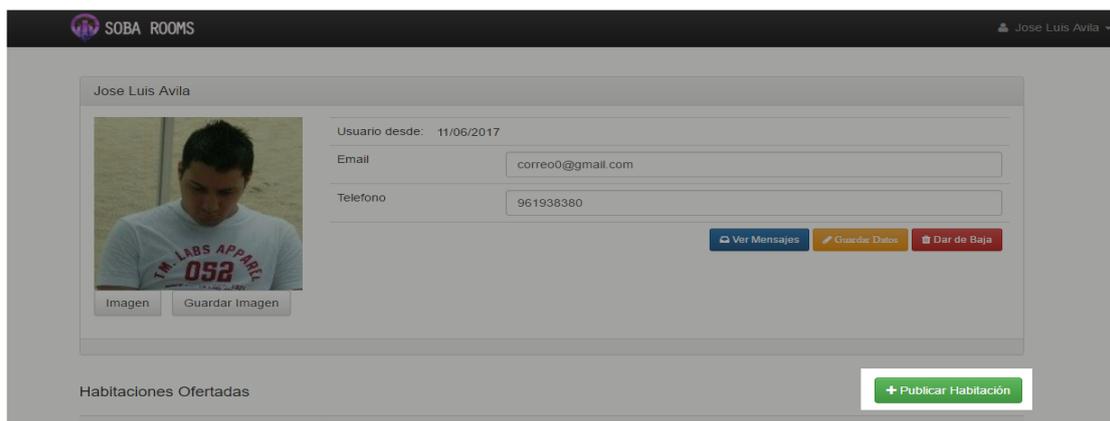


Imagen 6.11 Ir Publicar Habitación

Se nos dirige al formulario de creación de la habitación donde se deben rellenar los campos para su creación y pulsar en el botón “Dar de Alta”.

**Crear Habitación**

**Zona**  
Benimaclet

**Dirección**  
Avenida Alfahuir 55

**Alquiler por mes**  
300

**Imagen**  
Elegir archivos 2 archivos

**Facturas Incluidas**

**Importe facturas por mes**

**Tipo Inmueble**

**Nº Habitaciones Disponibles**  
3

**Nº Habitaciones Totales**  
3

**Compañeros**

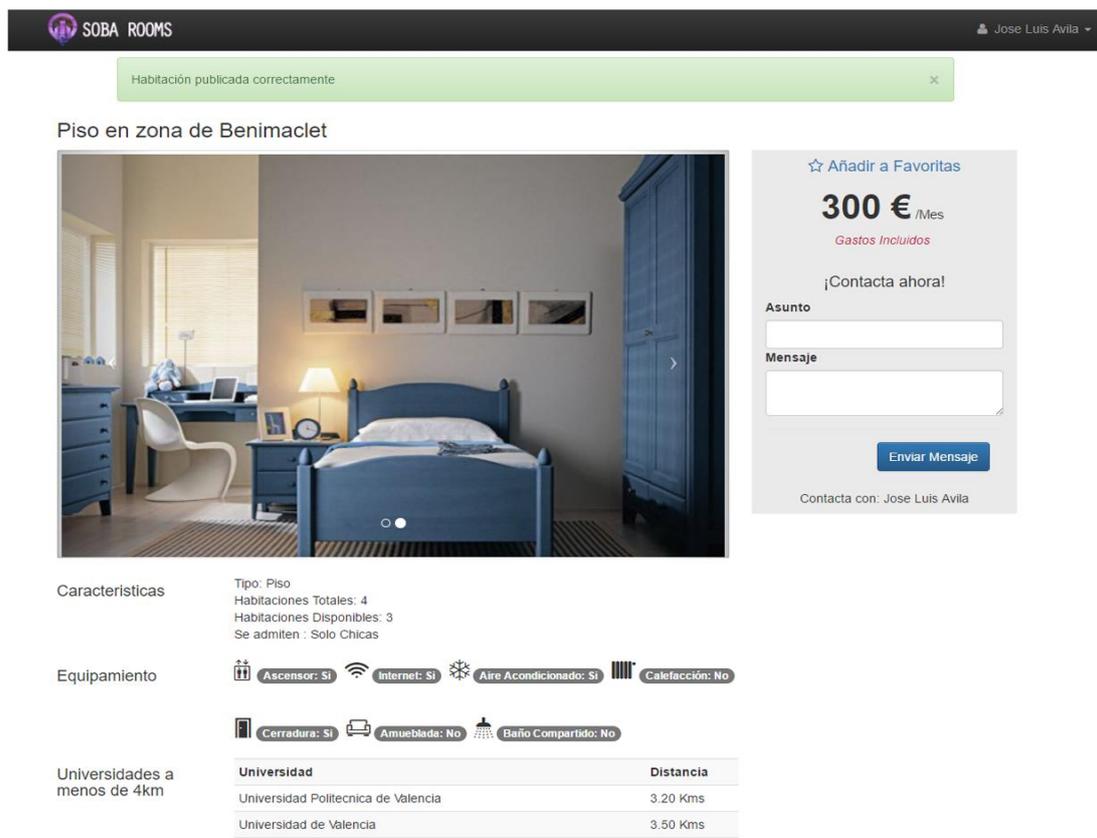
**Servicios Disponibles**  
 Cerradura  Calefacción  Aire Acondicionado  Internet  Ascensor  
 Hab. Amueblada  Baño Compartido

**Descripción Corta**  
Se busca compañera para una habitación individual que se queda libre a de finales de junio. El alquiler son 3000€ gastos incluidos. Buscamos alguien responsable, limpio y que no fume. Nosotras somos tres chicas trabajadoras. A poder ser que se quede hasta finales de agosto.

Imagen 6.12 Formulario de creación Habitación

## Diseño de un portal web para alquiler de habitaciones a estudiantes

Una vez se han validado todos los datos del formulario de creación de la habitación se nos redirige correctamente a la habitación creada.



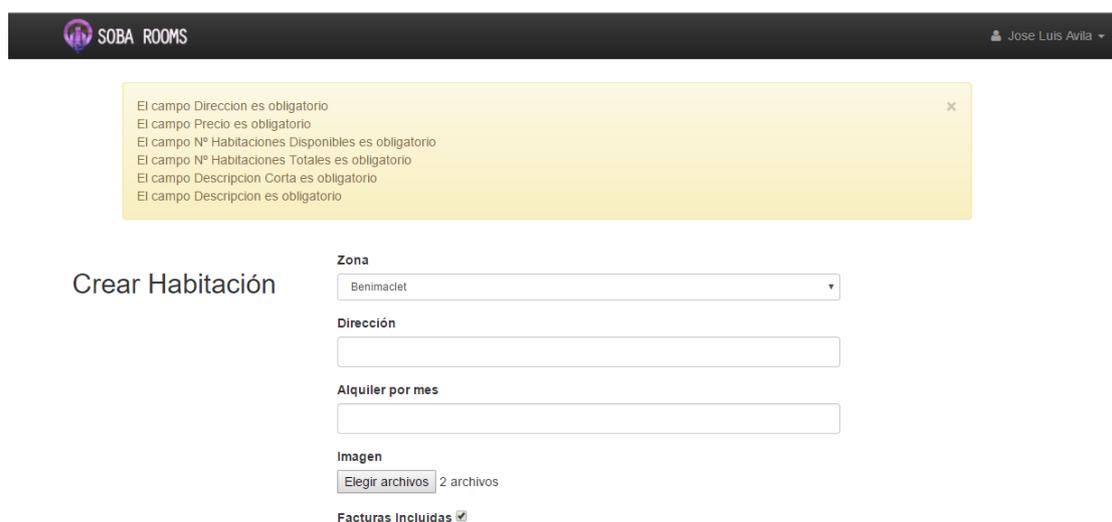
The screenshot shows the SOBA ROOMS website interface. At the top, there is a navigation bar with the logo and the user name 'Jose Luis Avila'. A green notification banner at the top center reads 'Habitación publicada correctamente'. Below this, the listing is titled 'Piso en zona de Benimaclet'. A large image shows a modern bedroom with a blue bed, a desk, and a window. To the right of the image, there is a price tag of '300 €/Mes' with 'Gastos Incluidos' and a '¡Contacta ahora!' button. Below the image, there are sections for 'Características', 'Equipamiento', and 'Universidades a menos de 4km'. The 'Equipamiento' section includes icons for 'Ascensor: Si', 'Internet: Si', 'Aire Acondicionado: Si', 'Calentamiento: No', 'Cerradura: Si', 'Amueblada: No', and 'Baño Compartido: No'. The 'Universidades' section is a table with two rows.

Universidad	Distancia
Universidad Politecnica de Valencia	3.20 Kms
Universidad de Valencia	3.50 Kms

Imagen 6.13 Habitación Creada

## Excepciones

Cuando se produce algo fallo en la creación de la habitación se nos informa de los fallos que se han producido como se muestra en la siguiente imagen.



The screenshot shows the 'Crear Habitación' form on the SOBA ROOMS website. A yellow error message box at the top lists several mandatory fields that are missing: 'El campo Dirección es obligatorio', 'El campo Precio es obligatorio', 'El campo Nº Habitaciones Disponibles es obligatorio', 'El campo Nº Habitaciones Totales es obligatorio', 'El campo Descripción Corta es obligatorio', and 'El campo Descripción es obligatorio'. Below the error message, the form fields are: 'Zona' (dropdown menu with 'Benimaclet' selected), 'Dirección' (text input), 'Alquiler por mes' (text input), 'Imagen' (button 'Elegir archivos' with '2 archivos' next to it), and 'Facturas Incluidas' (checkbox checked).

Imagen 6.14 Campos Obligatorios

### 6.4.3 Búsqueda Avanzada

Una vez se ha iniciado sesión, como redirigidos a la página principal de la aplicación, siendo esta una precondition de nuestro caso de uso procedemos realizar la prueba en sus distintos flujos.

#### Flujo Normal

Estando en la página principal de la aplicación se seleccionan los criterios con los que se desea realizar la búsqueda avanzada de la habitación.

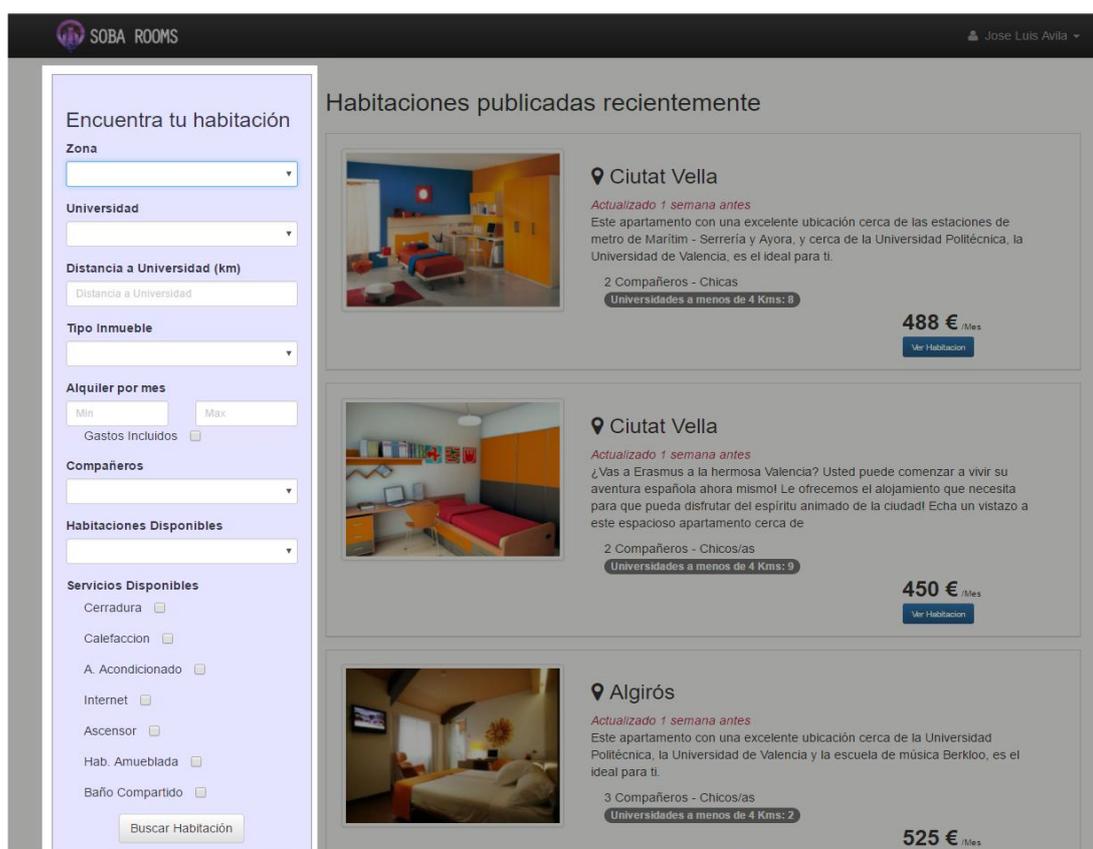


Imagen 6.15 Selección de criterios

Una vez se han seleccionado los criterios se pulsará el botón “Buscar Habitación” para realizar la búsqueda.

A continuación, se muestran los resultados de la búsqueda avanzada de habitaciones, donde se le indican además la cantidad de habitaciones encontradas.

## Diseño de un portal web para alquiler de habitaciones a estudiantes

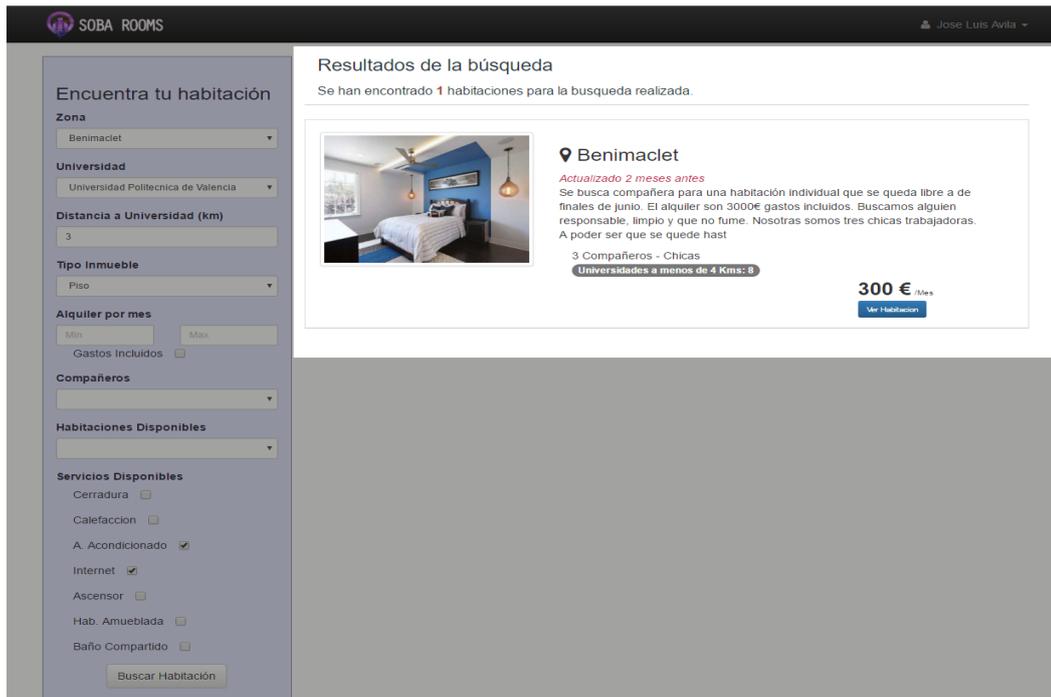


Imagen 6.16 Resultado búsqueda

### Excepciones

Cuando se realiza una búsqueda en la aplicación y no se obtienen resultados el sistema nos informa de este caso, permitiéndonos volver a realizar una búsqueda si así lo quisiéramos.

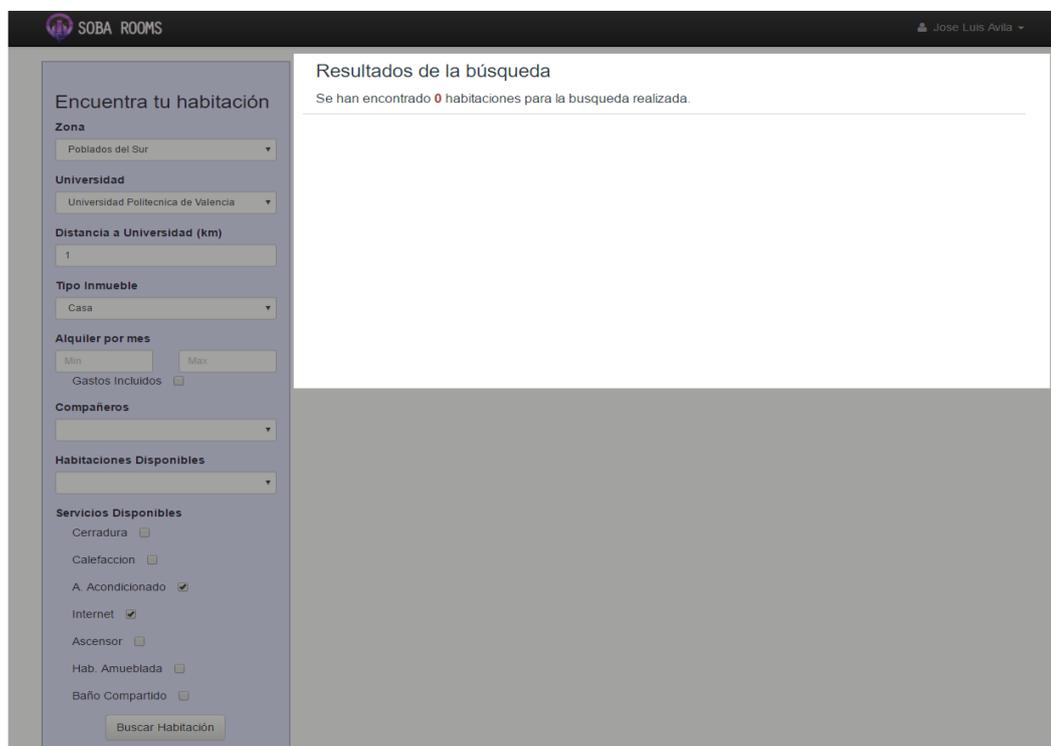


Imagen 6.17 Búsqueda sin resultados

#### 6.4.4 Ver Habitación

Haciendo uso de los resultados obtenidos en la prueba de búsqueda avanzada de habitaciones se probará el caso de uso de visualización de el detalle de una habitación.

#### Flujo Normal

En el resultado de la búsqueda avanzada se ha encontrado una habitación, se pulsa sobre el botón “Ver Habitación” de la habitación que se desea ver, obteniendo como resultado la página de detalle de esta.

Piso en zona de Benimaclet

☆ Añadir a Favoritas

**300 €** /Mes

Gastos Incluidos

¡Contacta ahora!

Asunto

Mensaje

Enviar Mensaje

Contacta con: Jose Luis Avila

Características

Tipo: Piso  
Habitaciones Totales: 4  
Habitaciones Disponibles: 3  
Se admiten: Solo Chicas

Equipamiento

Ascensor: Si Internet: Si Aire Acondicionado: Si Calefacción: No

Cerradura: Si Amueblada: No Baño Compartido: No

Universidades a menos de 4km

Universidad	Distancia
Universidad Politecnica de Valencia	3.20 Kms
Universidad de Valencia	3.50 Kms
Centro de Estudios Financieros	2.90 Kms
Escuela superior de Gestion Comercio y Marketing	3.60 Kms

Imagen 6.18 Ver Habitación

#### Excepciones

En caso de que se intente acceder a una habitación que no se encuentra ya disponible o no exista en nuestro sistema se redirecciona al usuario a la página principal de la aplicación.

## 7 Conclusiones

---

En este proyecto final de carrera se ha desarrollado un portal web de alquiler de habitaciones a estudiantes que permite a estos la búsqueda de habitaciones, así como también la posibilidad de ofertarlas.

La aplicación ofrece al usuario una interfaz de muy fácil manejo con un aspecto visual sencillo y practico que hace la hace perfecta para el uso del usuario inexperto.

Se ha optado por el diseño de tres capas haciendo uso del patrón modelo, vista, controlador ya que este se adapta perfectamente a la constante evolución que requiere una aplicación web tanto en la parte frontend como en el backend.

Dado que el público objetivo de la aplicación son los estudiantes universitarios se buscó dar ese punto diferente respecto a las clásicas web de alquiler de habitaciones ofreciéndole al estudiante la distancia de las habitaciones que se ofertan respecto a las diferentes universidades.

En el desarrollo de la aplicación se han reforzado muchos de los conocimientos adquiridos durante el Grado, además de ello se ha hecho uso de muchas herramientas y tecnologías que no se han estudiado durante el grado, por lo que se han adquirido nuevos conocimientos.

## 8 Bibliografía

---

- Laravel. Routing. [En línea] - <https://laravel.com/docs/5.2/routing>
- Laravel. Middleware. [En línea] - <https://laravel.com/docs/5.2/middleware>
- Laravel. Controllers. [En línea] - <https://laravel.com/docs/5.2/controllers>
- Laravel. Views. [En línea] - <https://laravel.com/docs/5.2/views>
- Laravel. Blade Templates. [En línea] - <https://laravel.com/docs/5.2/blade>
- Laravel. Application Structure. [En línea] - <https://laravel.com/docs/5.2/structure>
- Laravel. Artisan. [En línea] - <https://laravel.com/docs/5.2/artisan>
- Laravel. Eloquent. [En línea] - <https://laravel.com/docs/5.2/eloquent>
- Laravel. Migrations. [En línea] - <https://laravel.com/docs/5.2/migrations>
- Laravel. Seeding. [En línea] - <https://laravel.com/docs/5.2/seeding>
- Laragon [En línea] - <https://styde.net/laragon-un-entorno-de-desarrollo-para-laravel-en-windows/>
- Especificación de Requisitos según el estándar de IEEE 830. Gonzalo Méndez, Universidad Complutense de Madrid [En línea] - <http://www.fdi.ucm.es/profesor/Gmendez/docs/is0809/ieee830.pdf>
- Manual PHP. [En línea] - <http://php.net/manual/es/index.php>
- UML [En línea] - [https://es.wikipedia.org/wiki/Lenguaje\\_unificado\\_de\\_modelado](https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado)
- Introducción a XHTML [En línea] - <http://librosweb.es/libro/xhtml/>
- HTML5 y CSS3 - Aurby, Christophe. 2012. Cornellà de Llobregat, Barcelona: Ediciones ENI.
- Composer. [En línea] - [https://en.wikipedia.org/wiki/Composer\\_\(software\)](https://en.wikipedia.org/wiki/Composer_(software))