



# **Proyecto Integrado**

## **Portal del Viajero - Alquiler rural**

Martín Candela, Juan

Dorante Lucas, David

Proyecto Integrado Ciclo Superior DAW

Dpto. de Informática – IES AlMudeyne

# ÍNDICE

<b>ÍNDICE.....</b>	<b>2</b>
<b>INTRODUCCIÓN.....</b>	<b>6</b>
<b>ESTUDIO DE VIABILIDAD.....</b>	<b>7</b>
Descripción del Sistema Actual:.....	7
Descripción del Sistema Nuevo:.....	8
Identificación de Requisitos del Sistema:.....	9
Requisitos de información:.....	9
Tabla de correspondencia de tipos de datos:.....	9
Alojamiento:.....	10
• Descripción:.....	10
• Lista de campos: [nombre_campo   tipo_dato]:.....	10
• Otras observaciones:.....	10
Perfil:.....	10
• Descripción:.....	10
• Lista de campos: [nombre_campo   tipo_dato]:.....	10
• Otras observaciones:.....	10
Usuario:.....	11
• Descripción:.....	11
• Lista de campos: [nombre_campo   tipo_dato]:.....	11
• Otras observaciones:.....	11
AlojamientoComodidadAlojamiento:.....	11
• Descripción:.....	11
• Lista de campos: [nombre_campo   tipo_dato]:.....	11
• Otras observaciones:.....	11
AlquilerAlojamiento:.....	12
• Descripción:.....	12
• Lista de campos: [nombre_campo   tipo_dato]:.....	12
• Otras observaciones:.....	12
ComodidadAlojamiento:.....	12
• Descripción:.....	12
• Lista de campos: [nombre_campo   tipo_dato]:.....	12
• Otras observaciones:.....	12
TipoComodidad:.....	13
• Descripción:.....	13
• Lista de campos: [nombre_campo   tipo_dato]:.....	13
• Otras observaciones:.....	13
ImagenAlojamiento:.....	13
• Descripción:.....	13
• Lista de campos: [nombre_campo   tipo_dato]:.....	13
• Otras observaciones:.....	13

UbicacionAlojamiento:	14
• Descripción:	14
• Lista de campos: [nombre_campo   tipo_dato]:	14
• Otras observaciones:	14
ValoracionAlojamiento:	14
• Descripción:	14
• Lista de campos: [nombre_campo   tipo_dato]:	14
• Otras observaciones:	14
Requisitos funcionales:	15
Registro de usuarios:	15
Inicio de sesión:	15
Gestión de casas rurales:	15
Gestión de reservas:	15
Gestión de clientes:	16
Búsqueda y filtro de las casa rurales:	16
Notificaciones:	16
Otros Requisitos:	17
Control de versiones:	17
Portabilidad de la aplicación:	17
Balanceo de carga:	17
Seguridad de la aplicación:	17
Descripción de la solución:	18
Planificación del proyecto:	20
Equipo de trabajo:	20
Diseñador Web:	20
Nombre del puesto de trabajo:	20
Descripción de puesto de trabajo:	20
Requisitos del puesto:	20
Aptitudes Valorables:	20
Sueldo mensual:	20
Trabajadores que ocupan este puesto:	20
Desarrollador Web:	21
Nombre del puesto de trabajo:	21
Descripción de puesto de trabajo:	21
Requisitos del puesto:	21
Aptitudes Valorables:	21
Sueldo mensual:	21
Trabajadores que ocupan este puesto:	21
Jefe de Proyecto:	22
Nombre del puesto de trabajo:	22
Descripción de puesto de trabajo:	22
Requisitos del puesto:	22
Aptitudes Valorables:	22
Sueldo mensual:	22

Trabajadores que ocupan este puesto:.....	22
Planificación temporal:.....	23
Fase 1: Preparación y planificación.....	23
Periodo 1:.....	23
Tareas:.....	23
Responsables:.....	23
Periodo 2:.....	23
Tareas:.....	23
Responsables:.....	23
Fase 2: Desarrollo.....	24
Tareas:.....	24
Responsables:.....	24
Fase 3: Pruebas y Depuración.....	25
Tareas:.....	25
Responsables:.....	25
Fase 4: Despliegue y puesta en marcha.....	26
Tareas:.....	26
Responsables:.....	26
Fase 5: Mantenimiento y Soporte.....	27
Tareas:.....	27
Responsables:.....	27
Estudio del coste del proyecto:.....	28
Costes de Personal.....	28
Jefe de Proyecto:.....	28
Horas:.....	28
Salario:.....	28
Diseñador Web:.....	28
Horas:.....	28
Salario:.....	28
Desarrollador Web:.....	29
Horas:.....	29
Salario:.....	29
Coste total:.....	29
<b>ANÁLISIS DEL SISTEMA DE INFORMACIÓN.....</b>	<b>30</b>
Identificación del entorno tecnológico:.....	30
Entorno Tecnológico de Desarrollo.....	30
Lenguajes de Programación:.....	30
Frontend:.....	30
Backend:.....	30
Frameworks y Bibliotecas:.....	30
Angular:.....	31
Spring:.....	31
Base de Datos:.....	31
Herramientas de Desarrollo:.....	31

Entorno de Desarrollo Integrado:.....	31
Herramientas de control de versiones:.....	31
Servidores de aplicaciones:.....	31
Entorno Tecnológico de Explotación.....	32
Contenedores Docker:.....	32
Imágenes Docker:.....	32
Orquestador de Contenedores y balanceo de carga:.....	32
Seguridad:.....	32
Modelado de datos:.....	33
Modelo Entidad Relación y esquema de base de datos:.....	33
Identificación de los usuarios participantes y finales:.....	34
Usuarios Participantes.....	34
Administrador del Sistema:.....	34
Usuarios Finales.....	34
Gestores de Casas Rurales:.....	34
Clientes:.....	34
Otros sistemas.....	35
Sistema de Pago en Línea:.....	35
Servicios de Geolocalización:.....	35
Plataforma de Correo Electrónico:.....	35
Diagramas de Análisis:.....	36
Diagramas de Casos de Uso.....	36
Diagrama de Estados.....	40
Diagrama de Secuencia o Interacción.....	41
Portal del Viajero:.....	41
Tablas de decisión.....	42
Tabla de decisión para Gestión de Casas Rurales:.....	42
Tabla de decisión para Gestión de Clientes:.....	42
Tabla de decisión para Gestión de Usuarios y Perfiles:.....	43

# INTRODUCCIÓN

La gestión de alojamientos rurales ha sufrido una creciente demanda en los últimos años. Sin embargo, a pesar de la popularidad de las casas rurales, tanto para vacaciones como para eventos especiales, la gestión de reservas de alojamientos rurales ha enfrentado muchos problemas.

Es por ello que en este documento se establecen las bases para el desarrollo de un proyecto integrado cuyo objetivo es crear una aplicación web para la gestión de reservas de casas en entornos rurales. Este proyecto surge con la finalidad y el objetivo de brindar a los usuarios una plataforma que les permita conocer la disponibilidad de casas rurales de acuerdo a sus necesidades específicas, así como realizar reservas de manera efectiva durante períodos determinados.

Para cubrir estas necesidades, se han conseguido identificar requisitos y funcionalidades claves que se abordarán en el desarrollo de la aplicación. Todas estas incluyen la gestión de casas rurales, la disponibilidad de las casas rurales en tiempo real, así también como la gestión de clientes y sus reservas.

Se llevará a cabo este proyecto de manera eficiente, ya que se establecerán etapas claras de desarrollo, que abarcarán desde la preparación y planificación inicial hasta el despliegue y mantenimiento continuo de la aplicación. Además, se emplearán tecnologías modernas, incluyendo contenedores Docker para garantizar la portabilidad y escalabilidad del sistema.

El objetivo final de este proyecto es proporcionar una herramienta que sea sólida, robusta y fácil de usar, y que a la vez satisfaga las necesidades tanto de los usuarios finales como de la empresa, facilitando la gestión de reservas de casas rurales y mejorando la experiencia general de los usuarios en este sector.

# ESTUDIO DE VIABILIDAD

## Descripción del Sistema Actual:

A continuación, se expondrán algunas de las características que implementa el sistema actual de reservas de casas rurales:

En el sistema actual se puede **iniciar sesión** con una **cuenta** que previamente habremos de haber **registrado**.

El sistema también dispone de un **buscador** con varias opciones, en el cual podremos **filtrar** por **provincia, rango de precio, tamaño, número de habitaciones, número de personas, las características de dicha casa rural, las fechas que el cliente desee...** . Cada sistema posee filtros parecidos, muy útiles a la hora de que el consumidor encuentre su alojamiento ideal.

Dispone de **alojamientos** que se pueden **alquilar** y que **cumplen** con los **requisitos solicitados por el cliente**. En caso de **no** ser solicitado **ningún requisito**, se muestran **todos los anuncios disponibles**, junto con las imágenes de la casa rural, la información de dicha casa rural y enlaces de contacto con el propietario o gestor.

También dispone de **sistemas de ordenamiento de búsquedas**, que incluyen un ordenamiento basado en el **precio/noche** del alojamiento. Aunque también se pueden encontrar otros.

El sistema provee a la cuenta que posee el perfil de gestor de alojamientos la opción de **publicar anuncios de alojamientos** a arrendar, en el cual puede poner los datos de dicho alojamiento.

La persona que entre con perfil de cliente o cómo anónimo, tiene la posibilidad de **acceder a los anuncios** de los alojamientos que se pueden alquilar e incluso se le da la opción de alquilarla, pero para ello, previamente el cliente tiene que estar registrado.

La persona que entre como **propietario o gestor** en la página web, tiene la posibilidad de **publicar los alojamientos** que quiera poner en alquiler, pudiendo modificar las publicaciones, añadirlas o incluso eliminarlas.

La persona que acceda a la página web como **administrador**, tiene la posibilidad de dar de baja a clientes o propietarios además.

## Descripción del Sistema Nuevo:

Este sistema, se enfocará en un **nicho de alquileres rurales**. Esto quiere decir que toda la aplicación estará **ambientada** en entornos rurales (imágenes de casas rurales, montaña, tonos verdes...). Por lo tanto, este será un punto de diferenciación clave entre las apariencias del sistema actual y el sistema nuevo.

Se pretende que la interfaz de usuario sea lo más **sencilla** posible, para ello se llevará a cabo un desarrollo muy enfocado en la usabilidad de la aplicación, teniendo en cuenta que el objetivo principal de la mayoría de personas que consuman dicha aplicación será encontrar una **casa rentable** en cuanto a **comodidades/precio** para llevar a cabo un **alquiler vacacional**. De esta manera se podrá determinar qué elementos se deben añadir y en qué parte se deben posicionar. También, contaremos con un sistema que indique que clases de instalaciones y comodidades ofrece el alojamiento (nº baños, cocina, etc...).

El sistema contará con la posibilidad de registrarse como **gestor de alojamientos** o **cliente**, siendo los primeros los **propietarios** de las casas que podrán ser **alquiladas** y siendo los **segundos** los que las **podrán alquilar**. **Los gestores de alojamientos no podrán llevar a cabo ningún tipo de alquiler vacacional a los pisos que se consideren de su propiedad** (esto se hace así para evitar conflictos en cuánto a los alquileres). En caso de encontrarse registrado, accederás al sistema como **usuario anónimo**, teniendo la posibilidad de acceder a las opciones de alojamiento pero siendo incapaz de efectuar un alquiler.

Cómo **gestor de alojamientos**, se contará con una **sección única** de la aplicación (que solo será renderizada si el perfil usado es el de gestor) en la que se podrá **administrar las viviendas** que tiene ingresadas para que los clientes puedan alquilarlas. Desde este apartado, se podrá ocultar la visibilidad del alojamiento, editar la descripción, imágenes, precio, eliminarlas, etc...

Se hará uso de un **buscador con varios filtros** para llevar a cabo búsquedas de casas que estén disponibles para ser alquiladas. Se pretenden poner filtros como la **ubicación de la casa**, las **fechas** entre las que se quiere alquilar la casa, **número de personas** y el **precio máximo y mínimo**.

El sistema también contará con una pantalla de inicio desde la que se podrán visualizar las opciones mejor valoradas de alojamientos. Además del buscador anteriormente mencionado y otra información relevante.



## Identificación de Requisitos del Sistema:

### Requisitos de información:

Para llevar a cabo la recopilación de requisitos de información se tendrán en cuenta las relaciones entre las distintas **entidades** que se presentarán en este proyecto.

La base de datos se generará de manera **automática** con el framework de Spring (que compone gran parte del backend de la aplicación). Esto quiere decir que gracias a la configuración que se le aporta, la aplicación es capaz de **generar tablas** y campos con solo levantar el backend. Lo que quiere decir que la lista de campos será tomada directamente de la lista de atributos que tendrán las clases mapeadas de JPA en la parte backend de la aplicación, **aunque en el diagrama entidad/relación se podrá apreciar con mayor detalle el tipo de dato que se almacenará en BBDD junto con sus restricciones.**

### Tabla de correspondencia de tipos de datos:

TIPO EN JAVA	TIPO EN MYSQL SERVER
boolean, java.lang.Boolean	TINYINT(1)
int, java.lang.Integer	INTEGER
long, java.lang.Long	BIGINT
float, java.lang.Float	FLOAT
double, java.lang.Double	DOUBLE
short, java.lang.Short	SMALLINT
byte, java.lang.Byte	SMALLINT
java.lang.Number	DECIMAL(38)
java.math.BigInteger	BIGINT
java.math.BigDecimal	DECIMAL(38)
java.lang.String	VARCHAR(255)
char, java.lang.Character	CHAR(1)
byte[], java.lang.Byte[], java.sql.Blob	BLOB(64000)
char[], java.lang.Character[], java.sql.Clob	TEXT(64000)
java.sql.Date	DATE
java.sql.Time	TIME
java.sql.Timestamp	DATETIME

## Alojamiento:

- **Descripción:**

Esta entidad guardará toda la información referente a los alojamientos que se encuentran registrados en la aplicación web.

- **Lista de campos:** [nombre\_campo | tipo\_dato]

1. id | Long - (*obligatorio*)
2. txtNombre | String - (*obligatorio*)
3. txtDescripcion | String - (*obligatorio*)
4. numPlazaMin | Integer - (*obligatorio*)
5. numPlazaMax | Integer - (*obligatorio*)
6. numPrecioPlaza | Double - (*obligatorio*)
7. numCamas | Integer - (*obligatorio*)
8. numBanyos | Integer - (*obligatorio*)
9. alquileresAlojamiento | Set<AlquilerAlojamiento> - (*opcional*)
10. idUsuario | Usuario - (*obligatorio*)
11. idImagenesAlojamiento | Set<ImagenAlojamiento> - (*obligatorio*)
12. idUbicacion | UbicacionAlojamiento - (*obligatorio*)
13. idValoracionesAlojamiento | Set<ValoracionAlojamiento> - (*opcional*)
14. idAlojamientoComodidades | Set<AlojamientoComodidadAlojamiento> - (*opcional*)

- **Otras observaciones:**

1. Esta entidad es la principal de toda la aplicación.
2. La tabla en BBDD de esta entidad es la que más relaciones tiene con las otras.
3. La asociación de esta tabla con la tabla de usuarios se realiza mediante la tabla de alquileres.

## Perfil:

- **Descripción:**

Esta entidad guardará toda la información referente a los perfiles de los usuarios de la página web.

- **Lista de campos:** [nombre\_campo | tipo\_dato]:

1. id | Long - (*obligatorio*)
2. codPerfil | String - (*obligatorio*)
3. txtPerfil | String - (*obligatorio*)
4. idUsuarios | Set<Usuario> - (*obligatorio*)

- **Otras observaciones:**

1. La función principal de esta entidad será determinar los permisos de los usuarios dentro de la aplicación. Los perfiles de la aplicación serán los que determinen si el usuario es cliente, gestor o administrador.

Usuario:

- **Descripción:**

Esta entidad guardará toda la información referente a los usuarios de la aplicación web, es indiferente del rol que posea el usuario.

- **Lista de campos: [nombre\_campo | tipo\_dato]:**

1. id | Long - (*obligatorio*)
2. txtNombreUsuario | String - (*obligatorio*)
3. txtDescripcion | String - (*opcional*)
4. txtDni | String - (*obligatorio*)
5. numTelefono | Integer - (*obligatorio*)
6. txtEmail | String - (*obligatorio*)
7. txtPassword | String - (*obligatorio*)
8. idPerfil | Perfil - (*obligatorio*)
9. idAlquileres | Set<AlquilerAlojamiento> - (*opcional*)
10. idAlojamientos | Set<Alojamiento> - (*opcional*)
11. idValoracionesAlojamientos | Set<ValoracionAlojamiento> - (*opcional*)

- **Otras observaciones:**

1. En esta tabla se podrán encontrar todos los usuarios, desde aquellos que poseen permisos de administración hasta aquellos que son meros clientes. Todo es distinguido a través del perfil asignado a cada usuario.

AlojamientoComodidadAlojamiento:

- **Descripción:**

Esta entidad guardará toda la información referente a las relaciones existentes entre la entidad de Alojamiento y ComodidadAlojamiento.

- **Lista de campos: [nombre\_campo | tipo\_dato]:**

1. id | Long - (*obligatorio*)
2. idComodidadAlojamiento | ComodidadAlojamiento - (*obligatorio*)
3. idAlojamiento | Alojamiento - (*obligatorio*)

- **Otras observaciones:**

1. Esta tabla es producto de la relación (N,M) entre las entidades de Alojamiento y ComodidadAlojamiento.

## AlquilerAlojamiento:

- **Descripción:**

Esta entidad guardará toda la información referente a los alquileres de alojamientos, guardará tanto el alojamiento cómo el cliente que alquila el alojamiento.

- **Lista de campos: [nombre\_campo | tipo\_dato]:**

1. id | Long - (*obligatorio*)
2. fechaInicioAlquiler | LocalDate - (*obligatorio*)
3. fechaFinAlquiler | LocalDate - (*obligatorio*)
4. idUsuario | Usuario - (*obligatorio*)
5. idAlojamiento | Alojamiento - (*obligatorio*)
6. precioTotalAlquiler | Double - (*obligatorio*)
7. numPlazasReservadas | Integer - (*obligatorio*)

- **Otras observaciones:**

1. El usuario gestor del alojamiento ya es un atributo del alojamiento alquilado, por lo que no es necesario incluirlo explícitamente en esta entidad.

## ComodidadAlojamiento:

- **Descripción:**

Esta entidad almacena una serie de comodidades que se pueden referenciar en los alojamientos. Pueden abarcar desde instalaciones interiores hasta actividades que se pueden realizar.

- **Lista de campos: [nombre\_campo | tipo\_dato]:**

1. id | Long - (*obligatorio*)
2. codigoComodidad | String - (*obligatorio*)
3. txtNombre | String - (*obligatorio*)
4. txtDescripcion | String - (*obligatorio*)
5. idAlojamientoComodidades | Set<Alojamiento> - (*opcional*)
6. idTipoComodidad | TipoComodidad - (*obligatorio*)

- **Otras observaciones:**

1. Se pretende que los datos de esta entidad se carguen en el front como una lista de comodidades que le puedes añadir cómo atributo a los alojamientos.
2. Los alojamientos podrán tener ninguna o varias comodidades y una misma comodidad se podrá dar en ninguno o varios alojamientos. Por lo que este dato generará una tabla adicional debido a su relación (N, M) con la entidad alojamiento. Sin embargo, como se especificó al principio, no se explicará la tabla generada por la relación entre esas dos entidades.

## TipoComodidad:

- **Descripción:**

Esta entidad almacena los distintos tipos de comodidades, desde comodidad del tipo instalación hasta comodidad del tipo actividad.

- **Lista de campos:** [nombre\_campo | tipo\_dato]:

1. id | Long - (*obligatorio*)
2. codigoTipoComodidad | String - (*obligatorio*)
3. txtNombre | String - (*obligatorio*)
4. idComodidadAlojamientos | Set<ComodidadAlojamiento> - (*opcional*)

- **Otras observaciones:**

1. La función principal de esta tabla será diferenciar cuál es el tipo de comodidad que estaremos guardando. De esta manera, nos ahorramos crear tablas independientes para instalaciones exteriores, instalaciones interiores, actividades, paisaje... Entre otros detalles que se pueden añadir como items al anuncio de un alojamiento

## ImagenAlojamiento:

- **Descripción:**

Esta entidad almacenará una serie de imágenes asociadas a los alojamientos.

- **Lista de campos:** [nombre\_campo | tipo\_dato]:

1. id | Long - (*obligatorio*)
2. datosImagen | Byte[] - (*obligatorio*)
3. idAlojamiento | Alojamiento - (*obligatorio*)
4. numOrden | Integer - (*obligatorio*)

- **Otras observaciones:**

1. El atributo "numOrden" indicará el orden dentro de la lista de imágenes que pueda tener un alojamiento.

## UbicacionAlojamiento:

- **Descripción:**

Esta entidad almacena la ubicación del alojamiento.

- **Lista de campos:** [nombre\_campo | tipo\_dato]:

1. id | Long - (*obligatorio*)
2. idAlojamiento | Alojamiento - (*obligatorio*)
3. codigoPostal | Integer - (*obligatorio*)
4. ciudad | String - (*obligatorio*)
5. provincia | String - (*obligatorio*)
6. lineaDireccion | String (*obligatorio*)
7. longitud | String - (*opcional*)
8. latitud | String - (*opcional*)

- **Otras observaciones:**

1. Los valores de “longitud” y “latitud” son opcionales, los gestores pueden ubicar su alojamiento en un mapa que se pueda visualizar por los usuarios. Para ello, estos dos valores son determinantes.

## ValoracionAlojamiento:

- **Descripción:**

Esta entidad almacena la información referente a las distintas valoraciones que puede tener un alojamiento.

- **Lista de campos:** [nombre\_campo | tipo\_dato]:

1. id | Long - (*obligatorio*)
2. idAlojamiento | Alojamiento - (*obligatorio*)
3. txtMensaje | String - (*obligatorio*)
4. txtAsunto | String (*obligatorio*)
5. idUsuario | Usuario - (*obligatorio*)
6. puntuacion | Integer - (*obligatorio*)

- **Otras observaciones:**

1. Se ha decidido optar por una valoración con un formato asunto-autor-puntuación-mensaje dónde un usuario podrá escribir y puntuar su experiencia en las casas rurales.

## Requisitos funcionales:

Para llevar a cabo los requisitos funcionales, se seguirán una serie de criterios los cuáles se verán a continuación:

### Registro de usuarios:

Los usuarios podrán registrarse o bien como clientes o bien como gestor de alojamientos, siendo necesario especificar dicho perfil. Para ello se habrá de proporcionar ciertos datos como el correo electrónico, la contraseña, etc...

### Inicio de sesión:

Los usuarios podrán iniciar sesión introduciendo su correo electrónico y su contraseña. De este modo, los clientes podrán efectuar sus alquileres vacacionales y los gestores podrán poner en alquiler sus viviendas. A su vez, si no se inicia sesión en la aplicación, se podrá únicamente visualizar el contenido de la aplicación (anuncios, ofertas...) pero no consumirlo.

### Gestión de casas rurales:

Los gestores tendrán permisos para agregar nuevas casas rurales disponibles, junto con sus características, ubicación, precio o disponibilidad... Estos, también tendrán permisos para editar y suprimir casas rurales existentes siempre y cuándo sean de su propiedad.

### Gestión de reservas:

Una vez el cliente haya realizado un alquiler, se le debe de enviar una confirmación de la reserva al cliente por correo electrónico junto con la factura.

En cuanto al gestor de alojamientos, a este se le notificará, a la hora en la que un usuario haya realizado una reserva sobre algún alojamiento en posesión del gestor. Además, los gestores podrán decidir si aceptar o no la reserva de este alojamiento por parte del cliente, incluyendo la capacidad de poder cancelar cualquier reserva que no se encuentre activa y que tenga una fecha a futuro.

Los administradores, podrán ver una lista de todas las reservas realizadas, incluyendo los detalles de dichos alquileres, tanto por parte del cliente como por parte del gestor de la casa rural, teniendo la opción de cancelar reservas existentes en el caso en el que sea necesario.

### Gestión de clientes:

Los gestores podrán ver una lista de todos los alquileres que han sido efectuados sobre sus viviendas. Pudiendo de esta manera obtener los datos de usuario público de esos clientes.

En cuanto a los administradores, estos podrán ver la lista completa de los clientes que se encuentran almacenados en la aplicación, y podrán modificar cada uno de sus atributos.

### Búsqueda y filtro de las casa rurales:

Los clientes podrán buscar casas rurales y se les dará la opción de establecer un periodo de tiempo específico, un rango de precio detallado, características de la vivienda y ubicación del alojamiento buscado. Además, los clientes podrán ver las características de estas casas rurales junto con las imágenes que el propietario haya subido.

### Notificaciones:

Los clientes recibirán aviso a través de correo electrónico del alquiler de una casa rural o de la modificación de dicho alquiler.

Los gestores recibirán un aviso a través de correo electrónico cuando algún cliente trate de llevar a cabo la reserva de un alojamiento.



## Otros Requisitos:

### Control de versiones:

Cómo tecnología de control de versiones se usará Git, toda la aplicación y la información referente a esta se encontrará disponible en un repositorio online de Github.

De esta forma, se podrá trabajar de manera colaborativa, además de permitirnos el poder llevar a cabo el desarrollo de funcionalidades independientes de una manera más cómoda. Así, nos aseguramos de que la traza de desarrollo de la aplicación sea visible, se pueda apreciar y se encuentre en un lugar seguro y escalable.

### Portabilidad de la aplicación:

La aplicación contará con un soporte para ser desplegada en contenedores docker. Esto quiere decir que el despliegue de la aplicación será notoriamente similar desde cualquier sistema operativo, siendo el único requisito tener docker instalado.

### Balanceo de carga:

La aplicación contará con una tecnología que aumentará considerablemente el rendimiento de esta misma. En este caso, se usará Kubernetes para orquestar los distintos contenedores en los que la aplicación se encuentra desplegada.

### Seguridad de la aplicación:

En este caso, hemos decidido usar Spring Security para tener un control de usuarios y de perfiles. De esta manera, cada usuario tendrá su propio token de autenticación que determinará los permisos que tiene dentro del sistema.

## Descripción de la solución:

Para llevar a cabo un desarrollo óptimo y eficaz de la aplicación se realizarán las siguientes tareas:

1. Llevar a cabo un análisis completo de la aplicación a realizar. Esto incluye:
  - Buscar sistemas similares y comprobar la manera en que funcionan.
  - Plantear un sistema propio basado en el de la competencia.
  - Determinar toda la información que será necesaria almacenar.
  - Determinar una serie de funcionalidades que debe de realizar el sistema.
  - Llevar a cabo un análisis del tiempo que tomará completar el desarrollo de la aplicación.
  - Llevar a cabo un análisis de costes basándose en el tiempo de desarrollo y el puesto laboral de cada integrante juntos con sus horas empleadas en realizar la tarea.
  - Analizar el software requerido para construir la aplicación.
  - Realizar un diagrama que represente la modelización de los datos almacenados para estructurar una base de datos.
  - Realizar un análisis de usuarios participantes y finales.
  - Llevar a cabo diagramas de casos de uso de la aplicación para saber a qué decisiones se enfrentarán los usuarios y cómo estructurar los elementos de la aplicación para que estas decisiones se resuelvan de la manera más sencilla y usable posible.
  - Especificar el diseño, estructuración y navegación de la página
2. Una vez realizado el análisis y teniendo claro cuál es el objetivo, los medios y las herramientas que se emplearán para alcanzar el mismo. Se procederá a maquetar un prototipo de la página que se visualizará en el lado del cliente. Para esto, vamos a usar una herramienta llamada **Figma**. De esta manera, podremos establecer un diseño desde el que partir, para así no encontrarnos con ningún imprevisto a la hora de desarrollar el Front de la aplicación
3. Una vez realizado el prototipo de la página, se procederá a crear la estructura básica de las entidades mapeadas de JPA en el backend. Dicho de otra manera, representaremos a través de clases Java la estructura, relaciones y restricciones de la base de datos haciendo uso del framework de Spring Data y basándonos en el diagrama entidad/relación creado en la fase de análisis. A través de la configuración aportada en el backend, la propia aplicación creará una base de datos directamente a la hora de ser desplegada, por lo que no será necesario la creación de un script adicional (aunque de todas formas lo habrá).
4. Una vez terminada la estructura básica del backend, el prototipo del frontend en Figma y teniendo una base de datos que se ha generado automáticamente por el despliegue de la aplicación de Spring. Se procederá a realizar un desarrollo paralelo del frontend y el backend. Se tendrá un control de los cambios realizados en esta etapa desde Github, ya que se usará la tecnología de Git para controlar las

versiones y cambios realizados en la aplicación. Cabe recalcar que el frontend de la aplicación se desarrollará usando el framework de Angular, teniendo una orientación basada en componentes y se hará uso de la librería de Bootstrap para poder estructurar la página de manera correcta y que se pueda visualizar bien desde cualquier dispositivo.

5. Cuando se termine de desarrollar por completo el sistema, se procederá a probar todas y cada una de las funcionalidades creadas. Para, de esta manera, poder asegurarnos que cada componente de la aplicación se comunica bien con el resto y de la manera que se desea.
6. Una vez terminada completamente la aplicación y sus pruebas, se procederá a usar la tecnología de Docker para desplegar las aplicaciones en distintos contenedores. De esta manera, se dotará al sistema de una mayor portabilidad y escalabilidad.
7. Cuando las configuraciones de Docker se encuentren finalizadas, se realizará una configuración con Kubernetes para poder manejar el balanceo de carga de la aplicación. De esta manera, nos aseguraremos de tener varios servidores funcionando con la misma aplicación que puedan distribuir la carga de peticiones.
8. Por último, se llevará a cabo un análisis de accesibilidad y usabilidad de aplicación final. Concluyendo de esta manera con el desarrollo de la aplicación.

## Planificación del proyecto:

### **Equipo de trabajo:**

#### Diseñador Web:

##### **Nombre del puesto de trabajo:**

Diseñador Web.

##### **Descripción de puesto de trabajo:**

El diseñador web se encarga de dar vida al aspecto visual de los sitios web. Su trabajo consiste en crear la interfaz de usuario (UI) y la experiencia de usuario (UX). Esto implica asegurarse de que el sitio web sea atractivo, práctico, intuitivo y agradable para los visitantes. Algunas tareas específicas de este puesto son: Crear ideas de diseño y maquetas, escoger colores, fuentes, imágenes y otros elementos visuales, determinar la navegación y la ubicación de los elementos en la página, optimizar el diseño para que sea accesible y funcione bien en diversos dispositivos y navegadores y colaborar con desarrolladores web para implementar el diseño.

##### **Requisitos del puesto:**

- Certificación de cursos de diseño.
- Tener un título de Ciclo de Grado Superior en diseño o relacionados.
- Tener un año de experiencia laboral como mínimo.
- Tener experiencia con Figma o herramientas de desarrollo de wireframes y prototipos de páginas web.

##### **Aptitudes Valorables:**

- Creatividad.
- Pensamiento Crítico.
- Inteligencia emocional.
- Proactividad.
- Conocimiento de software y aplicaciones informáticas.

##### **Sueldo mensual:**

El sueldo mensual promedio del diseñador web será de unos 1500€/mes por trabajar un total de 40 horas semanales.

##### **Trabajadores que ocupan este puesto:**

- Juan Martín Candela
- David Dorante Lucas

### Desarrollador Web:

#### **Nombre del puesto de trabajo:**

Desarrollador Web.

#### **Descripción de puesto de trabajo:**

El desarrollador web se encarga de codificar todos los aspectos relacionados con una aplicación web. En este caso, podremos encontrarnos una diferencia entre aquellos que se dedican a programar el aspecto visual y funcional del lado del cliente y aquellos que se dedican a desarrollar código para ser ejecutado en un servidor. Deben de trabajar en equipo y colaborar para que todas las partes de la aplicación se comuniquen de manera correcta.

#### **Requisitos del puesto:**

- Tener conocimientos de lenguajes como Javascript y Java.
- Tener conocimientos en el uso del framework de Spring.
- Tener conocimientos en el uso de Angular.
- Tener conocimientos en el uso de HTML y CSS.
- Tener un título de Ciclo de Grado Superior en Desarrollo de Aplicaciones Web o similar.

#### **Aptitudes Valorables:**

- Resolutivo.
- Resolución de problemas.
- Cooperativo.
- Pensamiento Crítico.
- Proactividad.
- Conocimiento de software y aplicaciones informáticas.

#### **Sueldo mensual:**

El sueldo mensual promedio del desarrollador web será de unos 1300€/mes por trabajar un total de 40 horas semanales.

#### **Trabajadores que ocupan este puesto:**

- Juan Martín Candela
- David Dorante Lucas

### Jefe de Proyecto:

#### **Nombre del puesto de trabajo:**

Jefe de Proyecto.

#### **Descripción de puesto de trabajo:**

El Jefe de Proyecto se encarga de coordinar a un equipo de trabajo. De cara al producto final, él es quien se encarga de escoger qué tecnologías serán usadas para el desarrollo de una aplicación. También se encarga de hablar con los clientes finales de dicha aplicación y presentarles el progreso del proyecto.

#### **Requisitos del puesto:**

- Conocimientos de herramientas de gestión de proyectos
- Experiencia técnica y laboral dentro del puesto de trabajo de desarrollador muy avanzadas.
- Habilidades comunicativas efectivas.
- Tener un título de Ciclo de Grado Superior en Desarrollo de Aplicaciones Web o similar o un título de grado universitario.

#### **Aptitudes Valorables:**

- Análisis de datos.
- Capacidad de adaptación
- Liderazgo.
- Resolución de problemas.
- Cooperativo.
- Pensamiento Crítico.
- Proactividad.
- Conocimiento de software y aplicaciones informáticas.

#### **Sueldo mensual:**

El sueldo mensual promedio del desarrollador web será de unos 2000€/mes por trabajar un total de 40 horas semanales.

#### **Trabajadores que ocupan este puesto:**

- Juan Martín Candela
- David Dorante Lucas

## **Planificación temporal:**

En cuanto a la planificación temporal del proyecto, se prevé que el tiempo total invertido en proyecto sea de unos tres meses, contando con un margen adicional para posibles retrasos o contratiempos que se den durante su producción y desarrollo.

### **Fase 1: Preparación y planificación.**

Esta fase abarca las dos primeras semanas de trabajo y tendrá una cantidad de dos periodos en el que se llevarán a cabo diferentes tareas.

#### **Periodo 1:**

Tendrá lugar los primeros 3 días.

##### ***Tareas:***

- Se llevarán a cabo revisiones exhaustivas y un análisis detallado de los requisitos del proyecto, para comprender al completo las necesidades del cliente.
- Se programan reuniones con el cliente que sirven para aclarar las expectativas del mismo y establecer objetivos claros.

##### ***Responsables:***

Los responsables de llevar a cabo estas tareas son los Jefes de Proyecto.

#### **Periodo 2:**

Tendrá lugar los días restantes próximos a los días del *periodo 1*.

##### ***Tareas:***

- Se procede a la selección de tecnologías y herramientas adecuadas para llevar a cabo el desarrollo del proyecto, teniendo en cuenta los requisitos de los clientes y las capacidades que presenta el equipo.
- Se procede a hacer una lista de objetivos que tendrán que ser cumplidos en un periodo de tiempo preestablecido.
- Se realiza un reparto de todas las tareas y de qué personas serán responsables de cada parte de la aplicación.
- Se realizan diagramas de casos de uso, entidad/relación, etc... Todo esto con el objeto de clarificar los pasos a seguir para llegar al producto final.

##### ***Responsables:***

Los responsables de llevar a cabo este conjunto de tareas son los Jefes de Proyectos, ya que son los capacitados para desempeñar este tipo de tareas en ambos periodos.

## Fase 2: Desarrollo.

En esta fase, se llevará a cabo la codificación y desarrollo de la aplicación, tanto del aspecto visual orientado a la parte que con la que va a interactuar el cliente, como en la parte que se ejecutará en el servidor. Además, tendrá una duración de unos 2 meses.

### **Tareas:**

- El Diseñador Web se enfocará en el diseño de la interfaz de usuario (UI) y la experiencia de usuario (UX) para garantizar un diseño lo más atractivo y funcional posible, llevará a cabo la realización de Wireframes y prototipos de la página. Así como la navegación en la misma.
- El Desarrollador Web se encargará de codificar el apartado visual, es decir, el frontend. Para ello se hará uso de herramientas como el framework de Angular.
- El Desarrollador Web también es el responsable de codificar el backend de la aplicación web, utilizando el framework de Spring Boot para garantizar que el funcionamiento de la aplicación sea robusto y eficiente.
- Los Jefes de Proyecto, seguirán teniendo reuniones con el cliente para mostrar cómo va avanzando la aplicación y para actualizar cambios, en el caso del que cliente desee modificar o agregar cambios al producto final.

### **Responsables:**

Los responsables para llevar a cabo esta fase son:

- Diseñadores Web.
- Desarrolladores Web.
- Jefes de Proyectos.



### Fase 3: Pruebas y Depuración.

En esta fase se llevarán a cabo los procesos correspondientes para la depuración y pruebas del proyecto, comprobando el correcto funcionamiento de las funcionalidades de la aplicación. Este proceso, conlleva una duración de 1 semana.

#### **Tareas:**

- Se llevarán a cabo todo tipos de pruebas exhaustivas, sobre todo, pruebas de seguridad, funcionamiento y rendimiento para garantizar la calidad y fiabilidad de la aplicación.
- Identificación y corrección de cualquier error o fallo de funcionamiento encontrado durante las pruebas que se realizan en este periodo de pruebas.

#### **Responsables:**

Los encargados para llevar a cabo esta fase es el equipo de desarrollo al completo:

- Jefes de Proyectos.
- Diseñadores Web.
- Desarrolladores Web.

#### Fase 4: Despliegue y puesta en marcha.

En esta fase se llevará a cabo el despliegue de la aplicación y se comprobará que funciona todo los elementos que componen la aplicación correctamente. Para esta fase, se estima una duración de 1 semana.

##### **Tareas:**

- Se procede al despliegue de la aplicación web en un entorno de producción, asegurando que la aplicación tenga una configuración adecuada de servidores y una configuración adecuada en la base de datos.
- Se llevan a cabo pruebas finales en el entorno de producción para garantizar que el lanzamiento del producto final sea exitoso.

##### **Responsables:**

Los responsables para llevar a cabo esta fase es el equipo de desarrollo al completo:

- Jefes de proyectos.
- Diseñadores Web.
- Desarrolladores Web.

### Fase 5: Mantenimiento y Soporte.

Esta fase tendrá lugar tras la finalización del proyecto, para seguir mejorando la aplicación y seguir actualizando a petición del cliente. Esta fase se realizará de forma continua tras finalizar con el producto final.

#### **Tareas:**

- Se realizan actualizaciones de forma periódica para mejorar funcionalidades y corregir errores, asegurando así que la aplicación tenga un rendimiento óptimo.
- Se proporciona soporte técnico continuo para resolver los problemas que surjan y atender solicitudes de los usuarios de manera continua.

#### **Responsables:**

Esta fase la llevará a cabo todo el equipo de desarrollo:

- Jefes de Proyectos.
- Diseñadores Web.
- Desarrolladores Web.

## Estudio del coste del proyecto:

El estudio del proyecto implica una evaluación detallada de todos los recursos financieros necesarios para llevar a cabo todas las actividades planificadas y para alcanzar los objetivos establecidos. A continuación, se presentan una serie de grupos cada uno con sus respectivos costes:

### **Costes de Personal.**

Dentro de este grupo se incluyen los salarios y beneficios del equipo de trabajo que se dividen a su vez en tres grupos:

#### **Jefe de Proyecto:**

En nuestro caso, el proyecto contará con dos Jefes de Proyectos, los cuáles presentan las siguientes características salariales y laborales:

##### **Horas:**

Los jefes de proyectos trabajan un total de 8 horas al día, haciendo un cómputo de 40 horas semanales.

##### **Salario:**

En cuanto a los salarios cobrarán un total de 2000€ al mes, haciendo un cómputo de 500€ semanales y cobrando a 12'5€ la hora.

Los costes laborales se calculan en función de las horas de trabajo estimadas para cada miembro del equipo durante el período de seis meses.

#### **Diseñador Web:**

En nuestro caso, el proyecto contará con dos Diseñadores Webs, los cuáles presentan las siguientes características salariales y laborales:

##### **Horas:**

Los Diseñadores Webs, trabajan un total de 8 horas al día, haciendo un cómputo de 40 horas semanales.

##### **Salario:**

En cuanto a los salarios cobran un total de 1500€ al mes, haciendo un cómputo de 375€ semanales y cobrando a 9.38€ la hora.

Los costes laborales se calculan en función de las horas trabajadas estimadas para cada miembro del equipo durante el período de seis meses.

**Desarrollador Web:**

En nuestro caso, el proyecto contará con dos Desarrolladores Webs, los cuáles presentan las siguientes características laborales y salariales:

**Horas:**

Los Desarrolladores Webs, trabajan un total de 8 horas diarias, haciendo un cómputo de 40 horas semanales.

**Salario:**

En cuanto a los salarios, cobran un total de 1300€ al mes, haciendo un cómputo de 325€ semanales y cobrando a 8,13€ la hora.

Los costes laborales se calculan en función de las horas trabajadas estimadas para cada miembro del equipo durante el período de seis meses.

**Coste total:**

El proceso de creación de la aplicación web, tendrá un costo total aproximado de unos **28800€** en el periodo de los 3 meses que se necesita para la creación de dicha aplicación, y sin contar con las herramientas que se usan o bien gratuitas o de pago.

# ANÁLISIS DEL SISTEMA DE INFORMACIÓN

## Identificación del entorno tecnológico:

Para el diseño, la implementación y el funcionamiento exitoso de la aplicación web de gestión de reservas de casas rurales, es fundamental la identificación del entorno tecnológico, tanto de desarrollo como de explotación del sistema. A continuación, vamos a mostrar más en detalle cómo será el entorno tecnológico en ambas fases:

### Entorno Tecnológico de Desarrollo.

#### **Lenguajes de Programación:**

Para el desarrollo de la aplicación, vamos a usar una serie de lenguajes de programación, los cuales vamos a detallar a continuación:

#### **Frontend:**

Para la parte del frontend, vamos a usar dos lenguajes de marca: “**HTML**” y “**CSS**” y un lenguaje de programación: “**Javascript**”, los cuales nos van a permitir crear la página web que el cliente verá. Para ello, usaremos HTML para crear los elementos a mostrar en la página web, CSS para aplicar estilos a los elementos que creemos con HTML y por último, usaremos Javascript, para darle funcionalidad a aquellos elementos que hayamos creado previamente, usando como framework para aplicar todos estos lenguajes, **Angular**. De la misma manera, al hacer uso de **Angular** estaremos proveyendo a la aplicación una **orientación modularizada**.

#### **Backend:**

Para la parte del backend, vamos a usar un solo lenguaje el cual nos va a permitir llevar a cabo el desarrollo de la parte del servidor. Para ello, usaremos “**Java**” para llevar a cabo el desarrollo del backend. Además, estaremos haciendo uso del framework de “**Spring**”, en este caso, para facilitar procesos como el mapeo de objetos, creación de repositorios para las entidades, servicios, controladores, seguridad de la aplicación, despliegue, etc...

#### **Frameworks y Bibliotecas:**

Son el conjunto de herramientas, bibliotecas y convenciones que usaremos para llevar a cabo el desarrollo de la aplicación, ofreciendo una base sólida y reutilizable, proporcionando estructuras, patrones y componentes comunes.

### **Angular:**

Usamos **Angular** para llevar a cabo el desarrollo del frontend de la aplicación web, permitiendo la creación de interfaces de usuario para que sean dinámicas y altamente interactivas. Además, este framework hace que la aplicación sea exponencialmente más mantenible y escalable, aportando un enfoque orientado a módulos.

### **Spring:**

Usamos **Spring** para llevar a cabo el desarrollo del backend de la aplicación web, proporcionando un entorno de desarrollo que sea más ágil y eficiente. Además, **Spring** proporciona múltiples frameworks que nos ayudarán a la hora de conectar la aplicación con la base de datos, el frontend, el despliegue y para aportar seguridad a la aplicación. Para desarrollar la parte del backend con el framework de **Spring**, vamos a utilizar el IDE de **Spring Tools Suite 4**.

### **Base de Datos:**

En nuestra aplicación web, necesitamos de una base de datos, para poder guardar la información de los clientes, la información de las casas rurales y de las distintas reservas que se realicen. Para ello vamos a usar **MYSQL**, usando **DBEAVER**, para comprobar que los datos se manipulan de forma correcta.

### **Herramientas de Desarrollo:**

Son las herramientas que vamos a usar para llevar a cabo el desarrollo de la aplicación, y son las siguientes, dividiéndolas según sus características:

#### **Entorno de Desarrollo Integrado:**

Vamos a usar **Spring Tool Suite 4** para el desarrollo de la parte del backend de la aplicación web, y **Visual Studio Code** para el desarrollo del frontend de la aplicación de reservas de casas rurales.

#### **Herramientas de control de versiones:**

Vamos a usar **GIT** y **GITHUB** para el control de versiones del código fuente, y guardar los cambios que se produzcan en la nube, de forma que todos los miembros del grupo de trabajo puedan tener el código actualizado.

#### **Servidores de aplicaciones:**

Son los servidores que usaremos para ejecutar y desplegar la aplicación web en el entorno de desarrollo local, usando en nuestro caso, **Apache Tomcat**.

## **Entorno Tecnológico de Explotación.**

Para el entorno tecnológico de **Explotación** de la aplicación web del alquiler de casas rurales, vamos a usar **Docker**, obteniendo una mayor flexibilidad, portabilidad y eficiencia en la gestión y el despliegue de la aplicación, facilitando así su mantenimiento y escalabilidad en un entorno de producción.

### **Contenedores Docker:**

Empaquetamos la aplicación web en contenedores **Docker** para garantizar la portabilidad y la consistencia del entorno de ejecución.

### **Imágenes Docker:**

Para el frontend y el backend crearemos imágenes **Docker**. Estas imágenes van a contener todos los archivos y dependencias necesarias para ejecutar la aplicación de manera aislada.

### **Orquestador de Contenedores y balanceo de carga:**

Gestionamos y orquestamos contenedores Docker en un entorno de producción usando **kubernetes**. Estas herramientas facilitan la administración y el despliegue de la aplicación en múltiples y diversos contenedores, además de garantizar una comunicación efectiva entre todos los contenedores.

### **Seguridad:**

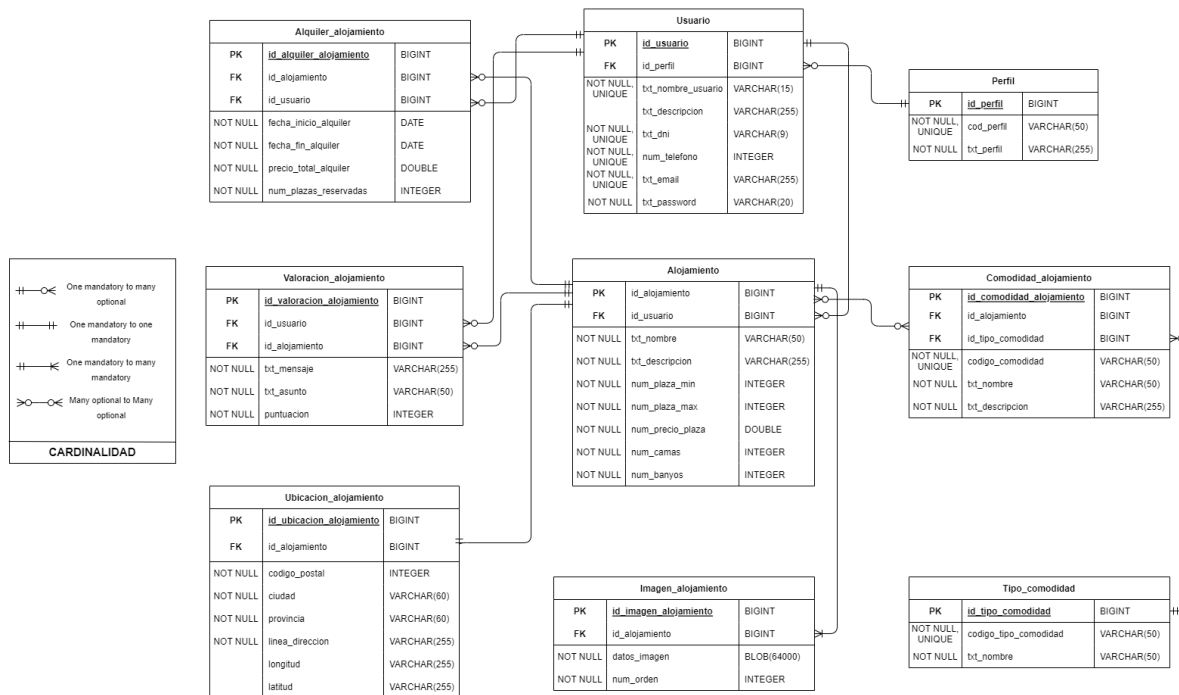
Se aplicarán las mejores prácticas de seguridad para contenedores Docker, como el uso de imágenes oficiales, el escaneo de vulnerabilidades y la gestión adecuada de credenciales, junto con permisos de accesos.

Además de todas estas medidas de seguridad, la propia aplicación también tendrá su propia capa de seguridad aportada por **Spring Security**.



## Modelado de datos:

### Modelo Entidad Relación y esquema de base de datos:



A continuación, se aportarán links para poder descargar y visualizar los archivos del modelo de entidad relación y el esquema completo de la Base de Datos:

[https://github.com/jmarcan286/porta1\\_del\\_viajero-PI-/tree/develop/public-resources](https://github.com/jmarcan286/porta1_del_viajero-PI-/tree/develop/public-resources)

## Identificación de los usuarios participantes y finales:

Para la gestión de reservas de casas rurales en el contexto de la aplicación web, es importante identificar a los usuarios participantes y finales que interactúan con la aplicación. A continuación se mostrará una identificación y explicación de cada uno de ellos, crucial para poder entender cómo interactúan con el sistema y qué funciones realizan:

### Usuarios Participantes.

Los usuarios participantes son aquellos que participan en la aplicación manteniendo la aplicación en funcionamiento.

#### **Administrador del Sistema:**

- Estos usuarios son responsables de administrar y mantener el sistema en funcionamiento.
- Se encargan de realizar funciones, como incluir la gestión de las propiedades disponibles, la atención a las reservas, la actualización de la disponibilidad o también se encargan de incluir la gestión de los clientes.
- Además, interactúan directamente con la interfaz de administración del sistema.

### Usuarios Finales.

Los usuarios finales son aquellos que interactúan directamente con la página web, pudiendo realizar reservas o obtener información sobre las casas rurales y sus propietarios, para poder contactar con ellos.

#### **Gestores de Casas Rurales:**

- Estos usuarios son los propietarios de las casas rurales que se encargan de ponerlas en alquiler.
- Se encargan de realizar funciones, como gestionar las casas rurales o propiedades disponibles, prestar atención a las reservas, actualizar la disponibilidad o no de las casas rurales y se encargan también de la gestión de los clientes, con los cuáles podrán llegar a un acuerdo o no para el alquiler.
- Además, interactúan con la interfaz de administración para gestionar las casas rurales que hay en la página web y las reservas.

#### **Clientes:**

- Los clientes son aquellos usuarios finales que desean alquilar una casa rural que se encuentre dentro de sus requisitos a través de nuestra página web.
- Se encargan de realizar funciones, como realizar una búsqueda de casas rurales que estén disponibles, realizar reservas, actualizar la información personal y visualizar el historial de reservas.
- Además, interactúan con la interfaz pública de la aplicación web para buscar, seleccionar y reservar casas rurales.

### Otros sistemas.

Cuando hablamos de otros sistemas, hacemos referencia a la utilización de herramientas que no funcionan dentro de la aplicación, es decir, que su sistema por el cual funciona es externo.

#### ***Sistema de Pago en Línea:***

- Es un sistema externo el cual proporciona servicios de procesamiento de pagos en línea.
- Además, la aplicación web puede integrarse con este sistema para permitir a los clientes realizar pagos de reservas utilizando tarjetas de crédito, transferencias bancarias u otros métodos de pago en línea.

#### ***Servicios de Geolocalización:***

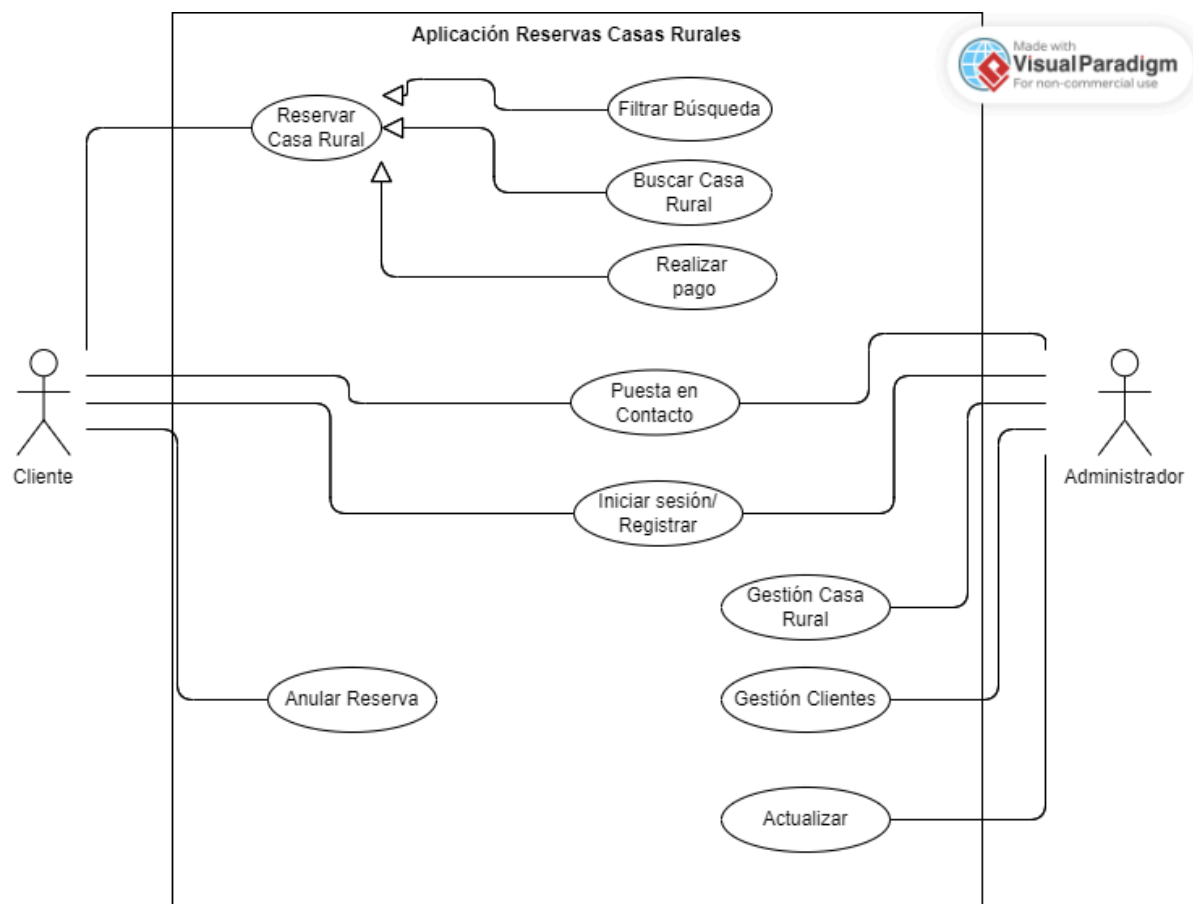
- Los servicios de geolocalización pueden ser utilizados para proporcionar funcionalidades de búsqueda que estén basadas en una ubicación especificada, permitiendo a los clientes encontrar casas rurales que estén cerca de ellos, o en la ubicación que los clientes especifiquen.
- La aplicación web podría integrarse con servicios como por ejemplo Google Maps para mostrar la ubicación de las casas rurales y proporcionar indicaciones de cómo llegar a dichas casas rurales.

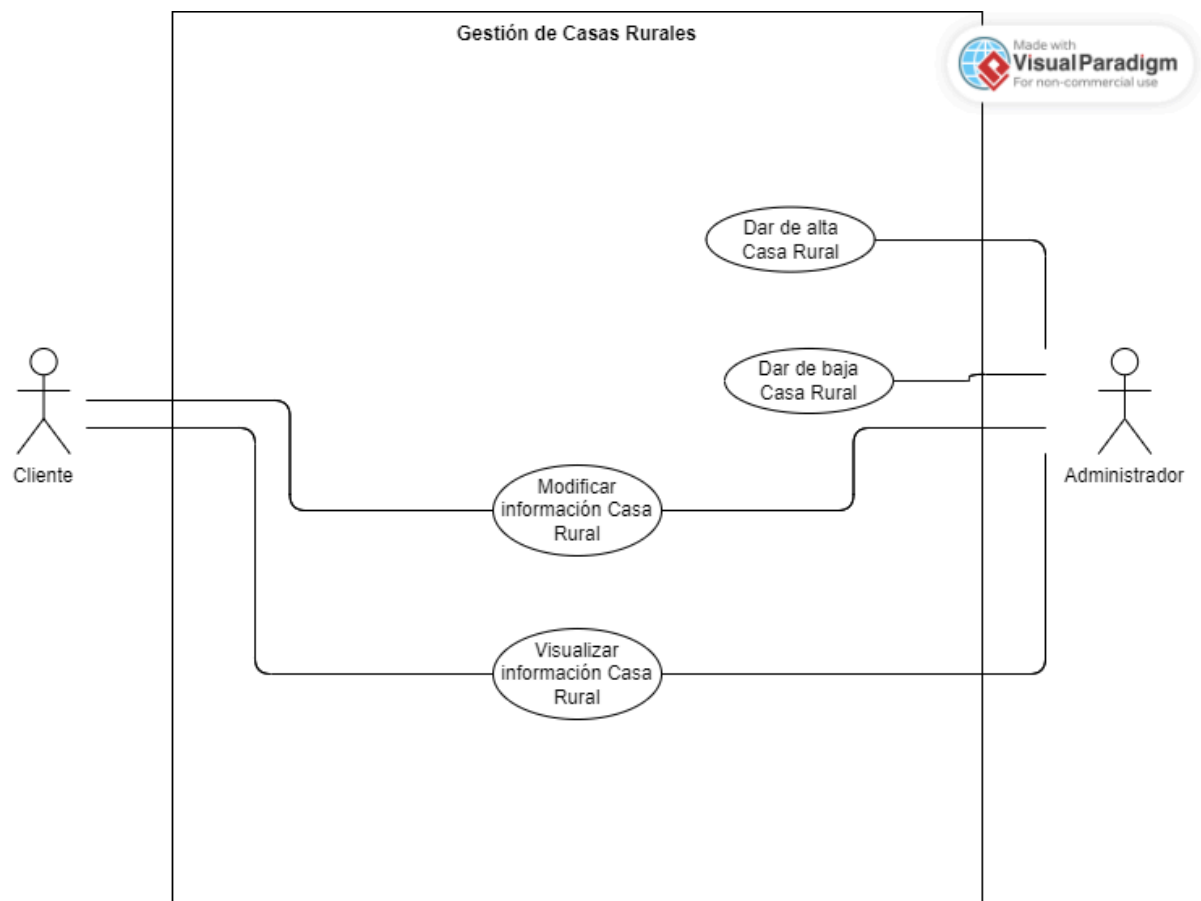
#### ***Plataforma de Correo Electrónico:***

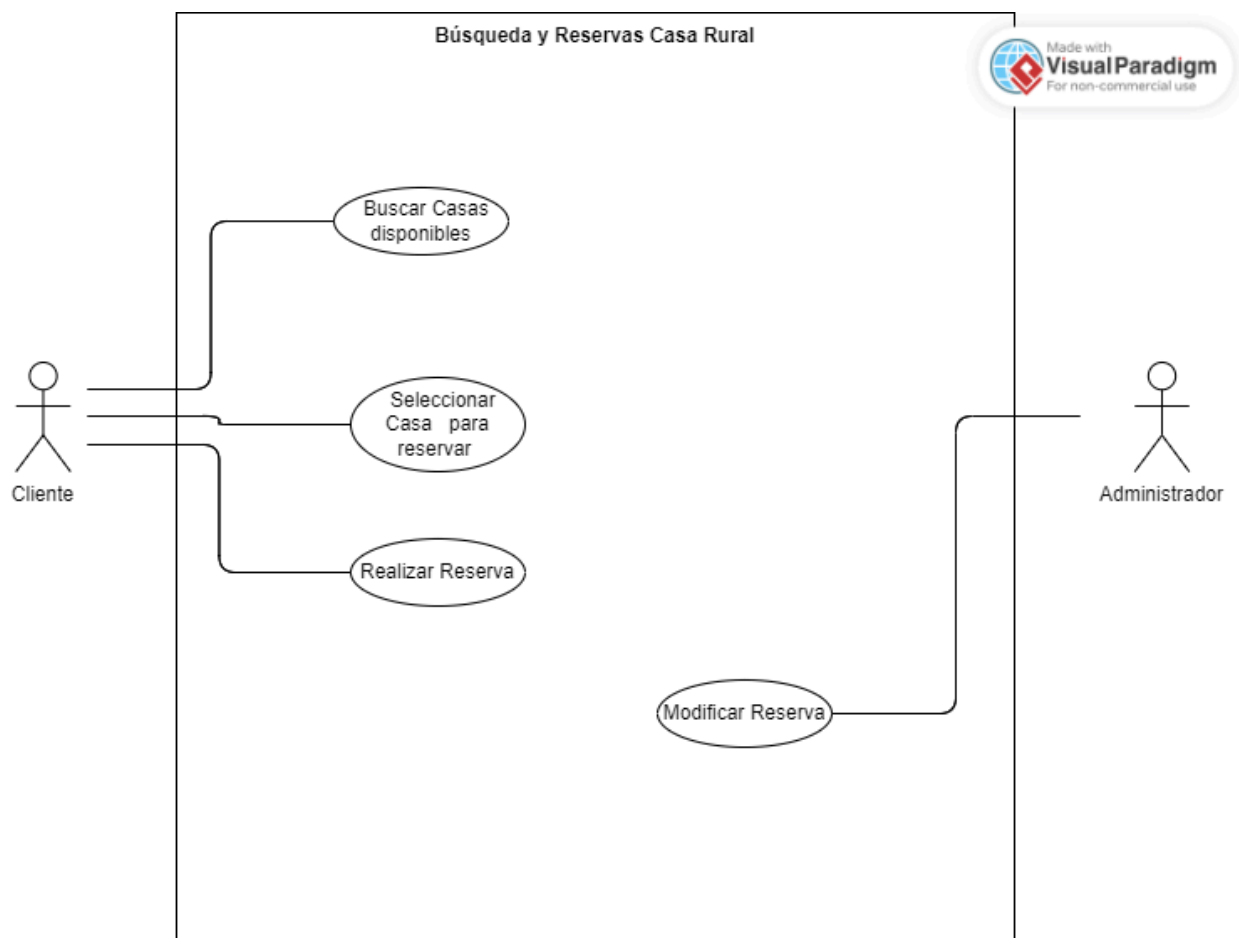
- Estas plataformas, se utilizan para enviar notificaciones por correo electrónico a los clientes, pueden ser confirmaciones de reservas, recordatorios de pago o actualizaciones sobre el estado de las reservas.
- Además, la aplicación web puede integrarse con servicios de correo electrónico como Gmail, Outlook o Hotmail, pudiendo así enviar automáticamente correos electrónicos a los clientes.

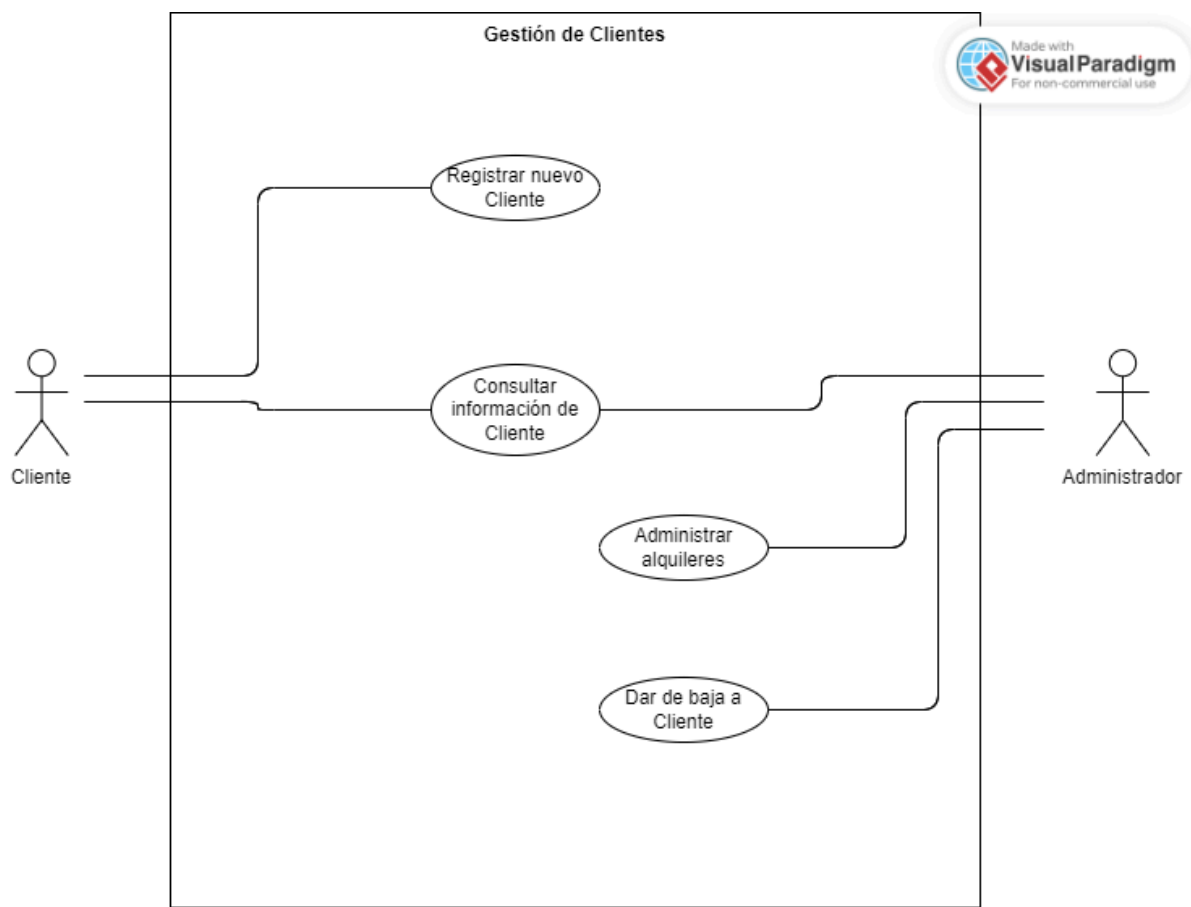
## Diagramas de Análisis:

### Diagramas de Casos de Uso.

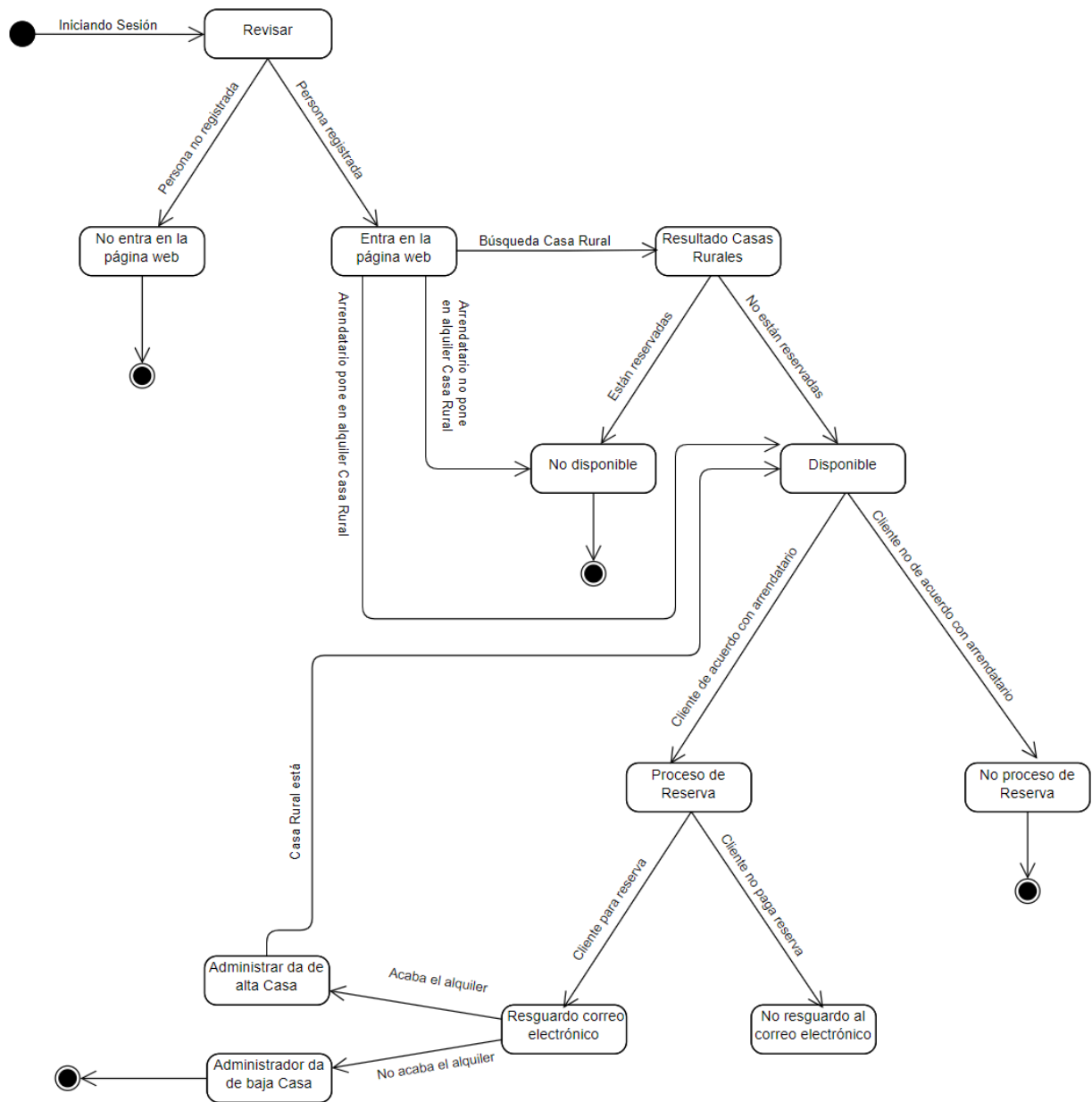








## Diagrama de Estados.

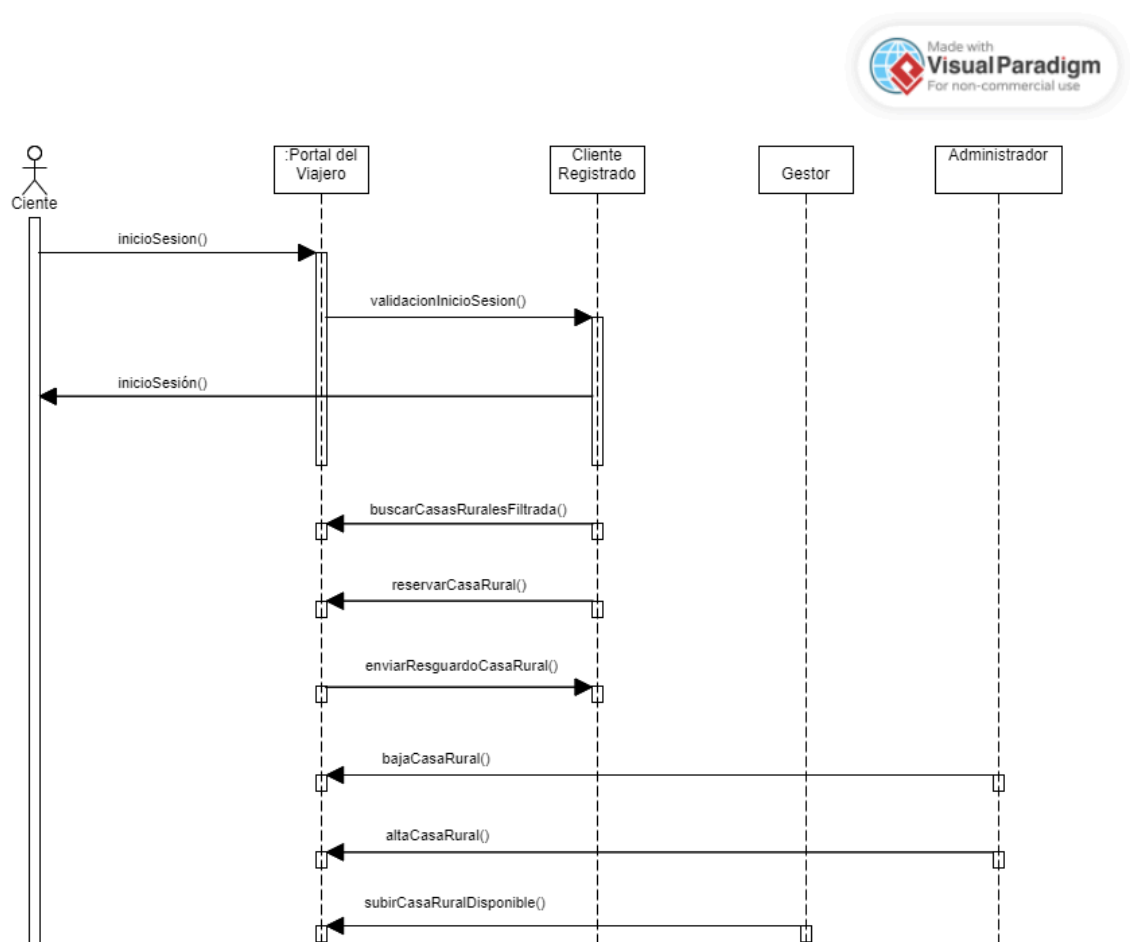




### Diagrama de Secuencia o Interacción.

El diagrama de secuencia es un tipo de diagrama usado para modelar la interacción entre objetos en un sistema según UML, aunque no están pensados para mostrar las lógicas de procedimientos que son complejos.

#### **Portal del Viajero:**



### Tablas de decisión.

**Tabla de decisión para Gestión de Casas Rurales:**

Condición	Acción
Casa Rural nueva agregada	Actualizar la página web para que se muestre las nuevas casas rurales agregadas recientemente.
Casa Rural desactivada por mantenimiento	Ocultar la Casa Rural de la lista de disponibles para alquilar.
Cliente solicita más información sobre Casa Rural	Mostrar detalles de la casa rural seleccionada al cliente.
Cliente alquila una Casa Rural	Actualizar la disponibilidad de la casa rural en la base de datos.

**Tabla de decisión para Gestión de Clientes:**

Condición	Acción
Nuevo cliente registrado	Almacenar los datos del cliente en la base de datos.
Cliente busca casas rurales disponibles	Aplicar filtros según las preferencias del cliente o también en el caso que no aplique filtros a la búsqueda, se hará una búsqueda de todas las casas rurales que hay en la página web.
Cliente alquila casa rural	Registrar el alquiler del cliente en la base de datos en la base de datos y modificar en la página web el anuncio de dicha casa rural para que no esté disponible para las fechas que estén alquiladas.
Cliente visita información sobre casa rural	En el anuncio de la casa rural podrá ver la información sobre la casa rural, pudiendo verla más en detalle si pincha en el anuncio.
Cliente modifica la información de su perfil	Cambiar alguno de los datos del cliente en base de datos.

**Tabla de decisión para Gestión de Usuarios y Perfiles:**

Condición	Acción
Nuevo usuario registrado como gestor	Otorga permisos de gestor en la aplicación y tendrá disponible todas las opciones del gestor de viviendas.
Usuario realiza una modificación en una casa	Registrar la modificación en la base de datos, para que en el caso de mostrar la información del cliente esté actualizado correctamente en la página también.
Usuario elimina una casa	Eliminar la casa rural de la base de datos, eliminandolo de la página web para evitar la confusión de los clientes.
Registra el perfil del usuario	Guardar perfil de usuario en la base de datos, ya que cada uno tendrá unas opciones disponibles o no.