

TRABAJO DE FIN DE GRADO

Aplicación Web de Gestión Inmobiliaria / Real Estate Management Application



Carlos gómez colmenero

UO257386 – DNI 53520720L

[1. CAPÍTULO 1. ¿QUÉ ES ESTE TRABAJO? 6](#_Toc67673907)

[2. CAPÍTULO 2. PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN 7](#_Toc67673908)

[2.1. Inicio del Plan de Sistemas de Información 7](#_Toc67673909)

[2.1.1. Análisis de la Necesidad del PSI 7](#_Toc67673910)

[2.1.2. Identificación del Alcance del PSI 7](#_Toc67673911)

[2.1.3. Determinación de responsables 8](#_Toc67673912)

[2.2. Definición y Organización del PSI 8](#_Toc67673913)

[2.2.1. Especificación del Ámbito y Alcance 8](#_Toc67673914)

[2.2.2. Organización del PSI 10](#_Toc67673915)

[2.3. Estudio de la Información Relevante 11](#_Toc67673916)

[2.3.1. Selección y Análisis de Antecedentes 11](#_Toc67673917)

[3. CAPÍTULO 3. DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA 13](#_Toc67673918)

[3.1.1. Identificación de las Necesidades de Infraestructura Tecnológica 13](#_Toc67673919)

[3.1.1.1. Necesidades de Despliegue de la Aplicación 13](#_Toc67673920)

[3.1.1.1.1. Amazon Web Services (AWS) 14](#_Toc67673921)

[3.1.1.1.2. Microsoft Azure 14](#_Toc67673922)

[3.1.1.1.3. Google Cloud 15](#_Toc67673923)

[3.1.1.2. Necesidades de Hardware 15](#_Toc67673924)

[3.1.1.3. Necesidades de Desarrollo 15](#_Toc67673925)

[3.1.1.3.1. Spring 16](#_Toc67673926)

[3.1.1.3.2. NodeJS 16](#_Toc67673927)

[3.1.1.4. Necesidades de Almacenamiento 17](#_Toc67673928)

[3.1.1.4.1. MySQL 17](#_Toc67673929)

[3.1.1.4.2. MongoDB 17](#_Toc67673930)

[3.1.2. Selección de la Arquitectura Tecnológica 17](#_Toc67673931)

[3.1.2.1. Selección de Despliegue de la Aplicación 17](#_Toc67673932)

[3.1.2.2. Selección de Desarrollo 18](#_Toc67673933)

[3.1.2.3. Selección de Almacenamiento 19](#_Toc67673934)

[4. CAPÍTULO 4. ESTUDIO DE VIABILIDAD DEL SISTEMA 19](#_Toc67673935)

[4.1. Sistema 1: Idealista (https://www.idealista.com/) 19](#_Toc67673936)

[4.2. Sistema 2: Pisos.com (https://www.pisos.com/) 22](#_Toc67673937)

[4.3. Sistema 3: Fotocasa (https://www.fotocasa.es/es/) 23](#_Toc67673938)

[4.4. Sistema 4: Inmoweb (https://www.inmoweb.es/) 24](#_Toc67673939)

[5. CAPÍTULO 5. PLANIFICACIÓN Y GESTIÓN DEL TFG 25](#_Toc67673940)

[5.1. Planificación del Proyecto 25](#_Toc67673941)

[5.1.1. Identificación de Interesados 25](#_Toc67673942)

[5.1.2. OBS y PBS 25](#_Toc67673943)

[5.1.3. Planificación Inicial. WBS 25](#_Toc67673944)

[5.1.4. Riesgos 25](#_Toc67673945)

[5.1.4.1. Plan de Gestión de Riesgos 25](#_Toc67673946)

[5.1.4.2. Identificación de Riesgos 25](#_Toc67673947)

[5.1.4.3. Registro de Riesgos 25](#_Toc67673948)

[5.1.5. Presupuesto Inicial 25](#_Toc67673949)

[5.1.5.1. Presupuesto de Costes 25](#_Toc67673950)

[5.1.5.2. Presupuesto de Cliente 25](#_Toc67673951)

[6. CAPÍTULO 6. ANÁLISIS DEL SISTEMA DE INFORMACIÓN 25](#_Toc67673952)

[6.1. Definición del Sistema 25](#_Toc67673953)

[6.1.1. Determinación del Alcance del Sistema 25](#_Toc67673954)

[6.2. Establecimiento de Requisitos 26](#_Toc67673955)

[6.2.1. Obtención de los Requisitos del Sistema 26](#_Toc67673956)

[6.2.1.1. Requisitos Funcionales 26](#_Toc67673957)

[6.2.1.1.1. Requisitos de la Gestión de Perfiles 26](#_Toc67673958)

[6.2.1.1.2. Requisitos de la Gestión de Propiedades 28](#_Toc67673959)

[6.2.1.1.3. Requisitos de la Gestión de Propietarios 33](#_Toc67673960)

[6.2.1.1.4. Visualización de datos de las Propiedades y Propietarios 34](#_Toc67673961)

[6.2.1.1.5. Requisitos del Filtro de Propiedades 37](#_Toc67673962)

[6.2.1.1.6. Requisitos del sistema de Seguimiento 39](#_Toc67673963)

[6.2.1.1.7. Requisitos del sistema de Comunicación entre Agente y Cliente 39](#_Toc67673964)

[6.2.1.1.8. Requisitos del sistema de Geolocalización 39](#_Toc67673965)

[6.2.1.2. Requisitos No Funcionales 40](#_Toc67673966)

[6.2.2. Identificación de Actores del Sistema 40](#_Toc67673967)

[6.2.2.1 Perfiles 40](#_Toc67673968)

[6.2.2.1.1. Usuario No Registrado 41](#_Toc67673969)

[6.2.2.1.2. Usuario Registrado 41](#_Toc67673970)

[6.2.2.1.3. Agente Registrado 41](#_Toc67673971)

[6.2.2.1.4. Super-agente Registrado 41](#_Toc67673972)

[6.2.3. Especificación de Casos de Uso 41](#_Toc67673973)

[6.2.3.1. Caso N.1 Crear Perfil Usuario 41](#_Toc67673974)

[6.2.3.2. Caso N.2 Editar Perfil Usuario 42](#_Toc67673975)

[6.2.3.3. Caso N.3 Eliminar Perfil Usuario 42](#_Toc67673976)

[6.2.3.4. Caso N.4 Crear Perfil Agente 43](#_Toc67673977)

[6.2.3.5. Caso N.5 Editar Perfil Agente 43](#_Toc67673978)

[6.2.3.6. Caso N.6 Eliminar Perfil Agente 44](#_Toc67673979)

[6.2.3.7. Caso N.7 Inicio de Sesión 44](#_Toc67673980)

[6.2.3.8. Caso N.8 Ver Estadísticas de Ventas 45](#_Toc67673981)

[6.2.3.9. Caso N.9 Crear Propiedad 45](#_Toc67673982)

[6.2.3.10. Caso N.10 Editar Propiedad 46](#_Toc67673983)

[6.2.3.11. Caso N.11 Eliminar Propiedad 46](#_Toc67673984)

[6.2.3.12. Caso N.12 Ver Propiedades 47](#_Toc67673985)

[6.2.3.13. Caso N.13 Ver detalles de Propiedades 47](#_Toc67673986)

[6.2.3.14. Caso N.14 Filtrar Propiedades 48](#_Toc67673987)

[6.2.3.15. Caso N.15 Enviar Mensaje 48](#_Toc67673988)

[6.2.3.16. Caso N.16 Ver Mensajes Recibidos 49](#_Toc67673989)

[6.2.3.17. Caso N.17 Añadir a Seguimientos 49](#_Toc67673990)

[6.2.3.18. Caso N.18 Eliminar de Seguimientos 50](#_Toc67673991)

[6.2.3.19. Caso N.19 Ver Seguimientos 50](#_Toc67673992)

[6.3. Identificación de los Subsistemas de Análisis 50](#_Toc67673993)

[6.3.1. Descripción de los Subsistemas 50](#_Toc67673994)

[6.3.2. Descripción de los Interfaces entre Subsistemas 53](#_Toc67673995)

[6.4. Análisis de los Casos de Uso 54](#_Toc67673996)

[6.4.1.1. Nomenclatura de los Casos de Uso 54](#_Toc67673997)

[6.4.1.2. Caso N.1 Crear Perfil Usuario 54](#_Toc67673998)

[6.4.1.3. Caso N.2 Editar Perfil Usuario 55](#_Toc67673999)

[6.4.1.4. Caso N.3 Eliminar Perfil Usuario 55](#_Toc67674000)

[6.4.1.5. Caso N.4 Crear Perfil Agente 56](#_Toc67674001)

[6.4.1.6. Caso N.5 Editar Perfil Agente 57](#_Toc67674002)

[6.4.1.7. Caso N.6 Eliminar Perfil Agente 57](#_Toc67674003)

[6.4.1.8. Caso N.7 Inicio de Sesión 58](#_Toc67674004)

[6.4.1.9. Caso N.8 Ver Estadísticas de Ventas 59](#_Toc67674005)

[6.4.1.10. Caso N.9 Crear Propiedad 59](#_Toc67674006)

[6.4.1.11. Caso N.10 Editar Propiedad 60](#_Toc67674007)

[6.4.1.12. Caso N.11 Eliminar Propiedad 60](#_Toc67674008)

[6.4.1.13. Caso N.12 Ver Propiedades 61](#_Toc67674009)

[6.4.1.14. Caso N.13 Ver detalles de Propiedades 61](#_Toc67674010)

[6.4.1.15. Caso N.14 Filtrar Propiedades 62](#_Toc67674011)

[6.4.1.16. Caso N.15 Enviar Mensaje 62](#_Toc67674012)

[6.4.1.17. Caso N.16 Ver Mensajes Recibidos 63](#_Toc67674013)

[6.4.1.18. Caso N.17 Añadir a Seguimiento 63](#_Toc67674014)

[6.4.1.19. Caso N.18 Eliminar de Seguimientos 64](#_Toc67674015)

[6.4.1.20. Caso N.19 Ver Seguimientos 64](#_Toc67674016)

[6.4.1.21. Matriz de Trazabilidad 66](#_Toc67674017)

[6.5. Análisis de Módulos 67](#_Toc67674018)

[6.5.1. Diagrama de Módulos 67](#_Toc67674019)

[6.5.2. Descripción de las Módulos 67](#_Toc67674020)

[6.5.2.1. Subsistema de Administración 67](#_Toc67674021)

[6.5.2.2. Subsistema de Utilidad 68](#_Toc67674022)

[6.5.2.3. Subsistema de Base de Datos 68](#_Toc67674023)

[6.5.2.4. Subsistema de Gestión del Sistema 70](#_Toc67674024)

[6.5.2.5. Subsistema de Gestión de Cuentas 70](#_Toc67674025)

[6.5.2.6. Subsistema de Mensajería 71](#_Toc67674026)

[6.5.2.7. Subsistema de Gestión de Propiedades 71](#_Toc67674027)

[6.5.2.8. Subsistema de Seguimiento 72](#_Toc67674028)

[6.6. Definición de Interfaces de Usuario 72](#_Toc67674029)

[6.6.1. Definición del aspecto de la interfaz 72](#_Toc67674030)

[6.6.1.1. Pantalla Principal 72](#_Toc67674031)

[6.6.1.2. Inicio en Sesión en el Sistema 73](#_Toc67674032)

[6.6.1.3. Gestión de Usuarios 73](#_Toc67674033)

[6.6.1.4. Gestión de Agentes 74](#_Toc67674034)

[6.6.1.5. Conversaciones 76](#_Toc67674035)

[6.6.1.6. Estadísticas y Contacto 77](#_Toc67674036)

[6.6.1.7. Gestión de Propiedades 78](#_Toc67674037)

[6.6.1.8. Gestión de Seguimientos 83](#_Toc67674038)

[6.6.2. Descripción del Comportamiento de la Interfaz 84](#_Toc67674039)

[6.6.2.1. Validación de Datos 84](#_Toc67674040)

[6.6.2.2. Protección de Cuentas 84](#_Toc67674041)

[6.6.2.3. Manejo de Errores 85](#_Toc67674042)

[6.6.3. Diagrama de Navegabilidad 86](#_Toc67674043)

[6.7. Especificación del Plan de Pruebas 87](#_Toc67674044)

[7. CAPÍTULO 7. DISEÑO DEL SISTEMA DE INFORMACIÓN 87](#_Toc67674045)

[7.1. Diseño de Casos de Uso Reales 87](#_Toc67674046)

[7.1.1. Caso N.1 Crear Perfil Usuario 87](#_Toc67674047)

[7.1.1.1. Diagrama de Interacción 87](#_Toc67674048)

[7.1.1.2. Diagrama de Rutas 87](#_Toc67674049)

[7.1.1.3. Diagrama de Rutas 88](#_Toc67674050)

[7.1.2. Caso N.2 Editar Perfil Usuario 88](#_Toc67674051)

[7.1.2.1. Diagrama de Interacción 88](#_Toc67674052)

[7.1.2.2. Diagrama de Rutas 89](#_Toc67674053)

[7.1.2.3. Diagrama de Actividad 89](#_Toc67674054)

[7.1.3. Caso N.3 Eliminar Perfil Usuario 90](#_Toc67674055)

[7.1.3.1. Diagrama de Interacción 90](#_Toc67674056)

[7.1.3.2. Diagrama de Rutas 90](#_Toc67674057)

[7.1.3.3. Diagrama de Actividad 91](#_Toc67674058)

[7.1.4. Caso N.4 Crear Perfil Agente 91](#_Toc67674059)

[7.1.4.1. Diagrama de Interacción 91](#_Toc67674060)

[7.1.4.2. Diagrama de Rutas 92](#_Toc67674061)

[7.1.4.3. Diagrama de Actividad 92](#_Toc67674062)

[7.1.5. Caso N.5 Editar Perfil Agente 93](#_Toc67674063)

[7.1.5.1. Diagrama de Interacción 93](#_Toc67674064)

[7.1.5.2. Diagrama de Rutas 93](#_Toc67674065)

[7.1.5.3. Diagrama de Actividad 94](#_Toc67674066)

[7.1.6. Caso N.6 Eliminar Perfil Agente 95](#_Toc67674067)

[7.1.6.1. Diagrama de Interacción 95](#_Toc67674068)

[7.1.6.2. Diagrama de Rutas 95](#_Toc67674069)

[7.1.6.3. Diagrama de Actividad 96](#_Toc67674070)

[7.1.7. Caso N.7 Inicio de Sesión 96](#_Toc67674071)

[7.1.7.1. Diagrama de Interacción 96](#_Toc67674072)

[7.1.7.2. Diagrama de Rutas 97](#_Toc67674073)

[7.1.7.3. Diagrama de Actividad 97](#_Toc67674074)

[7.1.8. Caso N.8 Ver Estadísticas de Ventas 97](#_Toc67674075)

[7.1.8.1. Diagrama de Interacción 97](#_Toc67674076)

[7.1.8.2. Diagrama de Rutas 98](#_Toc67674077)

[7.1.8.3. Diagrama de Actividad 98](#_Toc67674078)

[7.1.9. Caso N.9 Crear Propiedad 99](#_Toc67674079)

[7.1.9.1. Diagrama de Interacción 99](#_Toc67674080)

[7.1.9.2. Diagrama de Rutas 99](#_Toc67674081)

[7.1.9.3. Diagrama de Actividad 100](#_Toc67674082)

[7.1.10. Caso N.10 Editar Propiedad 101](#_Toc67674083)

[7.1.10.1. Diagrama de Interacción 101](#_Toc67674084)

[7.1.10.2. Diagrama de Rutas 101](#_Toc67674085)

[7.1.10.3. Diagrama de Actividad 102](#_Toc67674086)

[7.1.11. Caso N.11 Eliminar Propiedad 103](#_Toc67674087)

[7.1.11.1. Diagrama de Interacción 103](#_Toc67674088)

[7.1.11.2. Diagrama de Rutas 103](#_Toc67674089)

[7.1.11.3. Diagrama de Actividad 104](#_Toc67674090)

[7.1.12. Caso N.12 Ver Propiedades 104](#_Toc67674091)

[7.1.12.1. Diagrama de Interacción 104](#_Toc67674092)

[7.1.12.2. Diagrama de Rutas 105](#_Toc67674093)

[7.1.12.3. Diagrama de Actividad 105](#_Toc67674094)

[7.1.13. Caso N.13 Ver detalles de Propiedades 105](#_Toc67674095)

[7.1.13.1. Diagrama de Interacción 105](#_Toc67674096)

[7.1.13.2. Diagrama de Rutas 106](#_Toc67674097)

[7.1.13.3. Diagrama de Actividad 106](#_Toc67674098)

[7.1.14. Caso N.14 Filtrar Propiedades 107](#_Toc67674099)

[7.1.14.1. Diagrama de Interacción 107](#_Toc67674100)

[7.1.14.2. Diagrama de Rutas 107](#_Toc67674101)

[7.1.14.3. Diagrama de Actividad 108](#_Toc67674102)

[7.1.15. Caso N.15 Enviar Mensaje 109](#_Toc67674103)

[7.1.15.1. Diagrama de Interacción 109](#_Toc67674104)

[7.1.15.2. Diagrama de Rutas 109](#_Toc67674105)

[7.1.15.3. Diagrama de Actividad 110](#_Toc67674106)

[7.1.16. Caso N.16 Ver Mensajes Recibidos 111](#_Toc67674107)

[7.1.16.1. Diagrama de Interacción 111](#_Toc67674108)

[7.1.16.2. Diagrama de Rutas 111](#_Toc67674109)

[7.1.16.3. Diagrama de Actividad 112](#_Toc67674110)

[7.1.17. Caso N.17 Añadir a Seguimientos 112](#_Toc67674111)

[7.1.17.1. Diagrama de Interacción 112](#_Toc67674112)

[7.1.17.2. Diagrama de Rutas 113](#_Toc67674113)

[7.1.17.3. Diagrama de Actividad 113](#_Toc67674114)

[7.1.18. Caso N.18 Eliminar de Seguimientos 114](#_Toc67674115)

[7.1.18.1. Diagrama de Interacción 114](#_Toc67674116)

[7.1.18.2. Diagrama de Rutas 114](#_Toc67674117)

[7.1.18.3. Diagrama de Actividad 115](#_Toc67674118)

[7.1.19. Caso N.19 Ver Seguimientos 115](#_Toc67674119)

[7.1.19.1. Diagrama de Interacción 115](#_Toc67674120)

[7.1.19.2. Diagrama de Rutas 116](#_Toc67674121)

[7.1.19.3. Diagrama de Actividad 116](#_Toc67674122)

# CAPÍTULO 1. ¿QUÉ ES ESTE TRABAJO?

El objetivo del presente proyecto es el desarrollo de una aplicación web principalmente para escritorio, cuya principal finalidad es la gestión de un negocio de tipo inmobiliario, actuando a modo de mostrador y facilitando la comunicación entre el dueño de la empresa y un posible comprador. Esta aplicación se encontrará preparada para un entorno real, pues pretende solucionar el alcance limitado que tiene una pequeña empresa, haciendo que más personas puedan acceder a ella sin necesidad de personarse en el establecimiento físico.

Para realizar el siguiente trabajo, se hará uso de diferentes tecnologías de utilidad, como el entorno de ejecución Node, el Sistema de Gestión de Bases de Datos MongoDB y otras tecnologías como es el caso de sistemas de geolocalización.

Se pondrán en uso diferentes conocimientos obtenidos durante el Grado en Ingeniería Informática impartido en la Escuela de Ingeniería Informática, Universidad de Oviedo.

**Palabras Clave:** Aplicación, Inmobiliaria, Agente, Venta, Node, MongoDB.

# CAPÍTULO 2. PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN

## Inicio del Plan de Sistemas de Información

### Análisis de la Necesidad del PSI

El cliente, que es el dueño de la inmobiliaria, desea un sistema informático capaz de gestionar tanto inmuebles como las operaciones oportunas para una agencia inmobiliaria de tamaño pequeño, así como cualquier tipo de transferencia (ya sea compra, venta o alquiler) de estos. Dicho sistema será desarrollado de cara a un entorno de despliegue real, y será gestionado por un usuario (el dueño de la empresa), el cual tendrá un rol de administrador sobre cualquier cuenta de usuario que los clientes deseen crear.

La aplicación permitirá a los usuarios tanto publicar propiedades que los agentes deseen vender o alquilar como adquirir otras en las que estén interesados. La aplicación se encontrará enlazada con diferentes herramientas (como geolocalización), la cual permitirá geolocalizar las diferentes propiedades. Del mismo modo, la aplicación permitirá utilizar un sistema de mensajería integrado para facilitar la comunicación entre el vendedor y el interesado.

En las diferentes cuentas (ya sean perfiles de agente, super-agente o de comprador) podrán obtenerse diferentes estadísticas relacionadas con la interacción de los usuarios (búsquedas, solicitudes etc.), así como diferentes apartados relacionados con los interesados, como pueden ser solicitudes de más información para determinados inmuebles o contraofertas en el caso de que el precio sea negociable.

### Identificación del Alcance del PSI

Actualmente, el cliente posee su aplicación desplegada en Inmoweb, un software inmobiliario que se ocupa de las labores de gestión, almacenamiento y soporte en la nube de una aplicación web. No obstante, en la reunión inicial, cliente ha declarado que dicho software ofrece demasiadas funcionalidades que el cliente califica como “innecesarias”, así como otras que requiere no se encuentran disponibles. Se han establecido los siguientes objetivos:

* ***[OBJ. 1]*** Permitir la compra/venta/alquiler de inmuebles. Esta característica, naturalmente, es la más importante a la hora de planificar e implementar el proyecto, pues será de gran importancia que el sistema permita realizar dichas operaciones de la manera más sencilla, pero al mismo tiempo la manera más completa.
* ***[OBJ. 2]*** Permitir la comunicación entre el agente y el comprador con el fin de obtener información extra del inmueble y/o resolver dudas. Esta comunicación será fundamental y debe profundizarse mucho en ella, pues al final dentro de una adquisición tan importante como es un inmueble, debemos hacer que el cliente tenga acceso a toda la información posible, para así asegurar su comodidad y su confianza.
* ***[OBJ. 3]*** Acceso directo a los inmuebles, pues ciertos clientes no quieren pasar por el proceso de crear una cuenta y completar un amplio filtro para poder ojear los diferentes inmuebles que se encuentran en oferta en una zona determinada. Del mismo modo, una vez el usuario tenga su cuenta creada, debería sugerírsele ciertos inmuebles que le puedan interesar.
* ***[OBJ. 4]*** Al crear una cuenta, un comprador solo podrá crear una cuenta de tipo usuario. Una cuenta de tipo agente solo podrá ser creada por otra cuenta del mismo tipo.
* ***[OBJ. 5]*** Establecer un mecanismo que permita al cliente hacer un seguimiento de los precios de los inmuebles en los que esté interesado. Este seguimiento puede ser por diferentes características que resulten relevantes, como puede ser inmuebles concretos, zonas u otros.
* ***[OBJ. 6]*** Permitir obtener la geolocalización aproximada del inmueble en un mapa. Esta es una característica fundamental en los sistemas modernos, pues permite obtener una imagen visual de la zona en la que el inmueble se encuentra

### Determinación de responsables

* El tutor se ocupará de la supervisión tanto en la fase de documentación como la de desarrollo.
* El alumno proyectante se ocupará de la redacción de este documento, así como de la implementación del sistema web en solitario y bajo la supervisión del tutor.
* El cliente se ocupará de asegurar que se cumplen sus necesidades y requisitos, así como comprobar que la aplicación satisface los mismos.
* Un grupo de personas de un entorno cercano al proyectante, así como otros alumnos de la escuela se encargarán de labores de pruebas.

## Definición y Organización del PSI

### Especificación del Ámbito y Alcance

El presente proyecto se desarrollará en diferentes fases:

1. **Gestión de cuentas de usuario/agente/super-agente**

En primer lugar, se diseñará un sistema que permita a los usuarios crea una cuenta de cliente, desde la cual tengan permitido poner en venta su inmueble. Naturalmente, un usuario no registrado sólo podrá ver los inmuebles ofertados, pero no podrá realizar ningún tipo de operación sin crearse una cuenta en primer lugar. Por otra parte, se creará una cuenta de super-agente capaz de crear cuentas de todo tipo. Dicha cuenta no podrá ser eliminada. De este modo, el super-agente podrá crear una cuenta de agente si es necesario. Finalmente, existirán cuentas de agente, las cuales podrán gestionar la totalidad de la página web.

Respecto a las restricciones, solo existirá una cuenta de super-agente, la cual no podrá ser borrada. No obstante, pueden existir tantas cuentas de agente y usuario como sean necesarias.

**Objetivos de la fase:**

* **Crear una jerarquía de perfiles.**
* **Permitir al super-agente manejar cuentas tanto de agente como de usuario.**
* **Permitir al agente manejar cuentas de usuario.**
* **Permitir a un cliente crear una cuenta de usuario.**

1. **Expositor y filtro de inmuebles**

Existirá un sistema de exposición de los inmuebles, en el cual se podrán ver diferentes datos relativos a los mismos, pero en un formato más compacto. Al acceder a dicho inmueble, podrá verse una versión extendida de su información, con diferente información del cliente, por si el usuario deseara contactar con él.

De la misma manera, existirá un sistema de filtración y búsqueda, el cual permitirá al usuario obtener unos resultados más acordes a sus necesidades. Dicho filtro se compondrá por diferentes campos de importancia.

**Objetivos de la fase:**

* **Poder mostrar información reducida de un inmueble en un panel.**
* **Poder acceder a una descripción más amplia desde el panel anterior.**
* **Buscar inmuebles concretos.**
* **Filtrar inmuebles.**

1. **Comunicación agente-comprador.**

Se implementarán diferentes mecanismos que permitan al posible comprador establecer un canal de comunicación directa con el agente. Dichos canales serán:

* Dirección de correo electrónico:

Dentro de la descripción del producto, el usuario tendrá acceso al correo electrónico del agente, con el cual podrá enviar un mensaje automáticamente.

* Sistema de mensajería:

Dentro de la propia página web existirá un sistema de mensajería, el cual permitirá al usuario comunicarse con el agente si tiene alguna duda concreta que no requiera el uso del teléfono móvil.

* Número de teléfono:

Al igual que con el correo electrónico, el usuario podrá obtener el teléfono móvil del agente, con el cual podrá comunicarse rápidamente.

**Objetivos de la fase:**

* **Implementar el sistema de mensajería.**
* **Facilitar la comunicación entre el comprador y el agente.**

1. **Seguimiento de los precios.**

Deberá implementarse un mecanismo que permita, ya sea mediante notificaciones dentro de la página, como mediante e-mails, obtener mensajes automáticos cuando un inmueble en el cual están interesados se revalorice. Dicho sistema será automático y podrá apagarse cuando el usuario lo desee. De la misma forma, un usuario podrá crear una lista de seguimientos, donde guardará todos aquellos inmuebles en los que tenga interés.

**Objetivos de la fase:**

* **Sistema de seguimiento de precios.**
* **Seguimientos.**

1. **Geolocalización.**

El sistema contará con un sistema de geolocalización el cual permita al usuario obtener una localización aproximada del inmueble que desea visitar. Dicho sistema de geolocalización se implementará con una API, la cual insertará un mapa dinámico en la propia descripción del inmueble, mostrando un punto que represente el lugar en el cual se encuentra el inmueble en el cual está interesado.

**Objetivos de la fase:**

* **Sistema de geolocalización.**

### Organización del PSI

Estos serán los diferentes equipos encargados de llevar a cabo la tarea de realización del trabajo, así como de sus funciones principales:

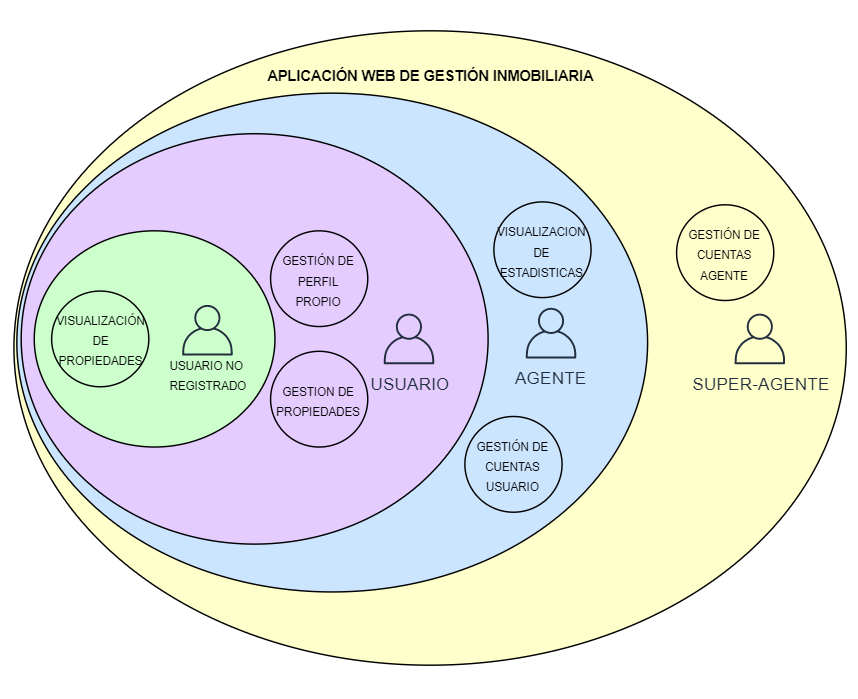
|  |  |
| --- | --- |
| **USUARIO** | FUNCION/ES |
| **EQUIPO DE SUPERVISION** | |
| **Dirección de la Escuela** | Supervisión de las fases del proyecto. |
| **Tutor del Proyecto** | Supervisión de que se cumplen los objetivos declarados por el alumno. |
| **Cliente** | Debe asegurar que se cumplan sus requisitos.  Supervisar que se consiguen los objetivos deseados. |
| **EQUIPO DE DESARROLLO Y DISEÑO** | |
| **Alumno** | Creación de los módulos software.  Elaboración de la documentación.  Diseño y análisis del sistema.  Pruebas iniciales del sistema.  Comprobar el correcto funcionamiento del sistema. |
| **EQUIPO DE PRUEBAS** | |
| **Alumnos de la Universidad / Usuarios Seleccionados** | Realizar pruebas de usabilidad, comprobar accesibilidad, comprobar funcionamiento del sistema. |

## Estudio de la Información Relevante

### Selección y Análisis de Antecedentes

Tras una reunión con el cliente, se han aclarado los siguientes conceptos:

* Actualmente, el cliente posee la información web en un software inmobiliario denominado Inmoweb. Este portal presenta un sistema de gestión total y absoluto de muchos de los aspectos relacionados con la gestión de procesos inmobiliarios. No obstante, debido a el pequeño tamaño de la empresa, mucha de esta funcionalidad resulta inútil, por lo que el cliente está pagando por unos servicios que no necesita.
* De la misma manera, el cliente considera que hay información importante que no se trata en dicho sistema. Debido a que el actual sistema no es una aplicación web, sino un software completo inmobiliario de pago no es posible obtener ninguna información relevante del mismo de cara a la implementación de nuestra aplicación web.
* Se establecerá un sistema de perfiles jerárquico de cara al desarrollo de la aplicación, donde un perfil de tipo Agente tendrá control sobre las propiedades, mientras los perfiles de tipo Usuario solo tendrán control sobre sí mismos. Los permisos y el alcance de cada perfil quedan explicados con más detalle en la figura inferior.
* Respecto a la legislación que será aplicada durante el proceso de desarrollo, así como en el resultado final del sistema, se seguirán las siguientes normas y legislaciones:
  + RGPD (Reglamento General de Protección de Datos).
  + LOPD (Ley Orgánica de Protección de Datos).



*Figura 1: Acceso de los diferentes Perfiles a los módulos de la aplicación*

Asimismo, en ficha reunión se ha aclarado que existirán los siguientes tipos de propiedades, así como las características concretas que debe cumplir cada uno:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **TIPO DE PROPIEDAD** | | | |
| **PROPIEDAD** |  | VIVIENDA | LOCAL | SUELO |
| Nombre |  |  |  |
| Tipo |  |  |  |
| Código |  |  |  |
| Descripción |  |  |  |
| Dirección |  |  |  |
| Población |  |  |  |
| Superficie |  |  |  |
| Núm. Habitaciones |  |  |  |
| Núm. Baños |  |  |  |
| Garaje |  |  |  |
| Piscina |  |  |  |
| Terraza |  |  |  |
| Trastero |  |  |  |
| Jardín |  |  |  |
| Calefacción |  |  |  |
| Aire Acondicionado |  |  |  |
| Ascensor |  |  |  |
| Amueblado |  |  |  |
| Permite Animales |  |  |  |
| Precio |  |  |  |
| Gastos de Comunidad |  |  |  |
| Escaparate |  |  |  |
| Aparcamiento |  |  |  |
| Zona de Carga y Descarga |  |  |  |
| Extintores |  |  |  |
| Núm. Aseos |  |  |  |
| Iluminación |  |  |  |
| Situación |  |  |  |
| Acceso a Agua |  |  |  |
| Acceso a Luz |  |  |  |
| Superficie Edificable |  |  |  |

De la misma manera, se ha acordado que los diferentes dueños de las propiedades tendrán los siguientes datos:

* Nombre y apellidos.
* Teléfono.
* Correo electrónico
* DNI/NIF/NIE
* Domicilio

# CAPÍTULO 3. DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA

### Identificación de las Necesidades de Infraestructura Tecnológica

#### Necesidades de Despliegue de la Aplicación

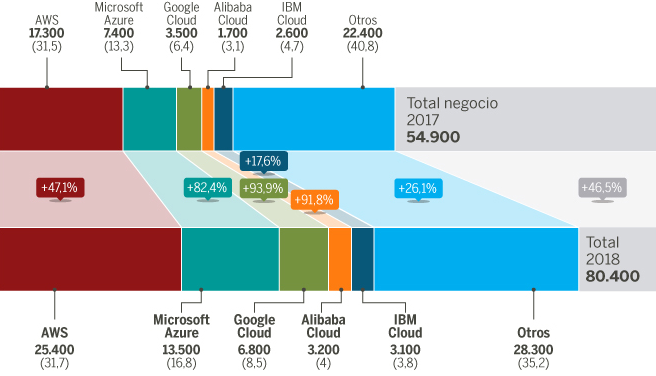
Para poder llevar a cabo el despliegue de la aplicación en la web, así como de establecer la dirección pública a la cual se deberán conectar los usuarios, es necesario tener acceso a unos servicios de computación en la web. La computación en la web es un paradigma que nos permite obtener diversos servicios desplegados en un servidor web remoto, de forma que todo aquel con conexión a internet pueda tener acceso a dicho servicio desde cualquier parte del mundo.

Otra opción sería el uso de un servidor local. No obstante, el costo de ese servidor local es demasiado elevado como para ser costeado por una pequeña empresa. Otro motivo por el cual no sería una buena opción en este caso sería el hecho de que para gestionar dicho servidor es necesario un conocimiento de informática por encima de la media, el cual nuestro cliente no posee.

En conclusión, utilizar los servicios de computación en la nube es la mejor opción, pues permite al cliente liberarse de la carga de mantener el hardware, por no hablar de que tendría un coste menor.

##### Amazon Web Services (AWS)

Es una colección de servicios de computación en la nube propiedad de Amazon. Es utilizado por una gran cantidad de aplicaciones. De hecho, casi el 32% de los servicios desplegados en la nube se encuentran alojados en los servidores de Amazon, como podemos ver en la Figura 2.



*Figura 2: Porcentaje de uso de los Servicios de Computación en la Nube en 2019 (Fuente: https://www.expansion.com/economia-digital/companias/2019/02/27/5c6ef3b2ca4741474b8b45c5.html)*

Proporciona todos los servicios de computación en la nube que necesitamos, y viene respaldada por una de las mayores empresas del mundo. Entre otros servicios, nos aporta sistemas de almacenamiento en la nube, diversos tipos de Bases de Datos y una elevada potencia de computación.

Entre sus diferentes ventajas esta AWS Lambda, el cual es un servicio que permite ejecutar una serie de algoritmos de forma automática, permitiendo reducir el consumo, acelerando el proceso de carga de la aplicación web.

Más información: https://aws.amazon.com/es/

##### Microsoft Azure

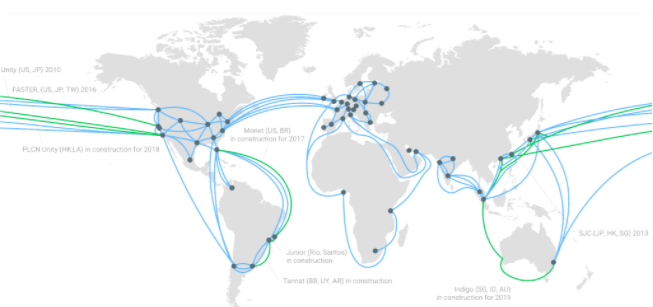
Similar al caso de Amazon Web Services, Azure viene respaldada por otra gran empresa tecnológica, en este caso Microsoft. Presenta una gran cantidad de similitudes con AWS, pero uno de los factores más importantes es que presenta más de 25 tipos de servidores que se encuentran específicamente diseñados para diferentes tareas, asegurando siempre que el sistema web se ejecute de la manera más eficiente posible.

También destaca por presentar una herramienta de inteligencia artificial la cual facilita la ejecución de la forma más optima posible.

Más información: https://azure.microsoft.com/es-es

##### Google Cloud

Otra de las grandes, en este caso creado por Google. El mismo Google declara que su mayor ventaja es el precio, pues es notablemente más barato que sus competidores. Por otra parte, tiene la infraestructura más potente en lo que a cableado se refiere, permitiendo transferencias de hasta 10Terabits por segundo.



*Figura 3: Red de Google Cloud (Fuente: https://kinsta.com/es/blog/google-cloud-hosting/)*

Más información: <https://cloud.google.com/>

#### Necesidades de Hardware

Respecto a las necesidades de Hardware de cara a su utilización por el cliente, un ordenador básico de oficina sería suficiente para gestionar la aplicación web. Es imprescindible una conexión a Internet de intensidad media para ser capaz de establecer una comunicación entre la máquina del cliente y la aplicación desplegada en la web.

#### Necesidades de Desarrollo

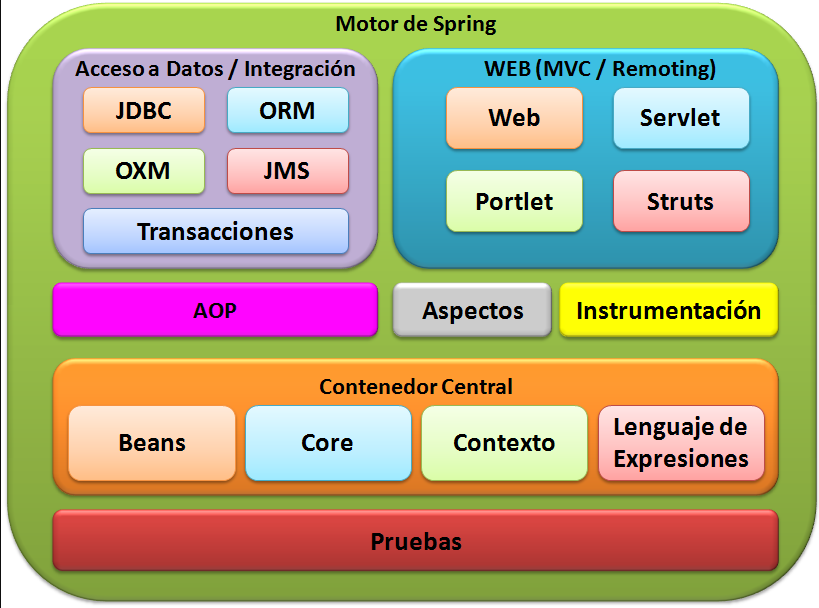
Ahora debemos cual será la tecnología que se utilizará. Dado el uso de los diversos lenguajes de programación de desarrollo en la web del lado del servidor, tanto Java como JavaScript son las mejores opciones para tener en cuenta, pues son los lenguajes más utilizados en este tipo de aplicaciones. Por otra parte, son los lenguajes más recomendados por los profesores de la universidad para el desarrollo de sitios web.

##### Spring

Spring es un conocido framework de desarrollo. Concretamente, utilizaríamos SpringBoot, el cual es una es una extensión preconfigurada de este.

A efectos prácticos, SpringBoot permite desarrollar una aplicación web saltándose todos los pasos de preconfigurar la misma. Spring es ideal para trabajar con el MVC (o Modelo Vista Controlador). También es muy adecuado para desarrollar aplicaciones que consuman APIs desde dispositivos móviles, Smart TVs y otros.

En concreto, en el caso de SpringBoot, presenta una aplicación fácilmente manejable, permitiendo dividir la aplicación en módulos muy diferenciados preparados para solventar diferentes necesidades. Es muy apto para aplicaciones escalables, pues su fácil ampliación permite la fácil comprensión del código.



*Figura 4: Estructura de Spring (Fuente: https://blognextstar.wordpress.com/2013/02/08/spring-framework/)*

##### NodeJS

Es un entorno en tiempo de ejecución multiplataforma, también de código abierto. En este caso, está basado en JavaScript. Es un entorno específicamente diseñado para ser muy eficiente en servidores web, por lo que puede ser de gran utilidad en nuestro desarrollo. Su código es fácilmente escalable, pues utiliza clusters para trabajar con las direcciones. También tiene Node Package Manager, el cual es un gestor de paquetes que permite instalar diferentes módulos con distintas funcionalidades con una simple línea de código. También es muy apta para empresas pequeñas, pues permite ser manejada por pocas personas, ya que el front-end y el back-end se pueden desarrollar con un mismo lenguaje. Empresas muy conocidas como PayPal o Netflix utilizan Node hoy en día, pues permite reducir en gran medida el coste de esfuerzo de las máquinas sin afectar a la seguridad, ya que su funcionamiento en clusters permite aislar las peticiones en métodos, haciéndolas, además, sencillas y fáciles de comprender.

#### Necesidades de Almacenamiento

Finalmente, respecto al sistema de Gestión de Bases de Datos (SGBD) a elegir, elegiremos un sistema relacional o documental, pues son los más aptos para la tarea que debemos llevar a cabo.

##### MySQL

Es un SGBD relacional desarrollado por Oracle Corporation y es conocida por ser una de las más populares del mundo. Es desarrollado por una empresa privada, la cual ofrece diferentes licencias: una de carácter gratuito y una de pago para empresas. Naturalmente, la versión de empresas ofrece ciertas herramientas extra, como asistencia técnica o herramientas de motorización.

Es un SGBD fácil de usar y muy rápido, presentando también diversas capas de seguridad y siendo altamente eficiente en lo que a memoria se refiere. También necesita pocos recursos.

Otras bases de datos similares a MySQL son PostgreSQL o SQLite, por ejemplo.

##### MongoDB

Es un SGBD NoSQL orientado a documentos. Eso significa que, en lugar de guardar los datos en tablas, como es el caso de MySQL, lo guarda en documentos en formato Binary JSON (o BSON). Es un SGBD muy apropiado en producción y presenta muchas funcionalidades. Es utilizada por grandes empresas como MTV o Network.

Es muy utilizada por su versatilidad, pues permite modificar los objetos ya existentes en la base de datos sin necesidad de reestructurar toda. De esta forma, si es necesario añadir atributos o elementos a una colección no es necesario recrear esta. Es especialmente apta para proyectos web que quieran estas tres características:

* **Escalabilidad:** Ser capaz de ampliarse fácilmente.
* **Disponibilidad:** Encontrarse disponible para su uso en cualquier situación.
* **Flexibilidad:** Ser capaz de modificarse fácilmente sin necesidad de grandes cambios.

### Selección de la Arquitectura Tecnológica

#### Selección de Despliegue de la Aplicación

Para seleccionar los servicios en la nube apropiados, tendremos diferentes factores en cuenta. En primer lugar, hablaremos de las tarifas:

Respecto a Amazon Web Services, debemos tener en cuenta que necesitamos una máquina EC2 que nos permita mantener una instancia de la aplicación web. Dicha máquina no requerirá de una elevada potencia de memoria, pues considerando el tamaño de la empresa no esperamos un elevado número de accesos a la misma. Por ello con una instancia de 1GiB será suficiente. Dicha instancia tiene un coste de 0,0047 USD por hora. Por otra parte, debemos considerar que el precio del almacenamiento en la nube es de 0,024 USD por cada GB que consumamos.

Fuentes:

* <https://aws.amazon.com/es/ec2/pricing/on-demand/>
* https://aws.amazon.com/es/s3/pricing/

Pasando al caso de Azure, el precio por una maquina similar (0,75GB en este caso) sería de 0,018 USD por hora. Este precio viene con 20GB de almacenamiento de carácter temporal, por lo que acabaríamos viéndonos obligados a adquirir almacenamiento. El almacenamiento es de 0,06 USD por cada GB.

Fuentes:

* <https://azure.microsoft.com/es-es/pricing/calculator/>

Finalmente, en el caso de Google Cloud, el precio por una maquina seria de 0,031 USD por hora, mientras que en el caso de almacenamiento sería de 0,0042 USD por GB.

Fuentes:

* <https://cloud.google.com/compute/vm-instance-pricing>

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Amazon Web Services** | **Microsoft Azure** | **Google Cloud** |
| **Precio Despliegue/h** | 0,0047 $ | 0,018 $ | 0,031 $ |
| **Precio Almacenamiento/h** | 0,024 $ | 0,06 $ | 0,0042 $ |
| **TOTAL** | **0,0287 $** | **0,078 $** | **0,0352 $** |

Podemos observar que las mejores alternativas a considerar serían AWS y Google Cloud. Por ello, vamos a elegir la primera opción, pues, además de ser la más económica, presenta las siguientes ventajas:

* Su dinamismo de precios permite que los usuarios paguen por el uso que le dan a la aplicación, por lo que nos aseguramos siempre de desembolsar lo mínimo necesario.
* Es fácilmente escalable, pues si en algún momento requerimos de un cambio en alguna de las tecnologías todos los servicios contratados seguirán siendo los mismos.
* Otras ventajas podrían ser el tener una gran empresa detrás como es Amazon, su seguridad y su excelente servicio técnico.

#### Selección de Desarrollo

Las dos alternativas que hemos presentado son igual de válidas a la hora de desarrollar una aplicación web, pero hemos optado por Node JS como la opción más optima por las siguientes razones:

* El lenguaje JavaScript hace la aplicación muy dinámica, permitiendo modificarla o ampliarla con nueva funcionalidad cuando sea necesario. El motivo de esto, como se indicó con anterioridad, es el hecho de que Node JS actúa en forma de clusters, permitiendo personalizar las peticiones web dentro de un método, y haciendo la aplicación más comprensible y cómoda. Por ello, si se desean modificar ciertas funcionalidades, Node JS permite realizar dicha modificación de una manera rápida y sencilla.
* Esta específicamente diseñado para trabajar con aplicaciones web.
* Su encapsulamiento de las rutas en clusters hace que la aplicación sea muy comprensible para todo aquel programador que se vea involucrado.
* La facilidad y comodidad de NPM nos permite añadir nueva funcionalidad con una sola línea de código, lo cual lo convierte en un excelente entorno.
* Es muy apropiado para proyectos que requieran de ejecución en tiempo real.

#### Selección de Almacenamiento

A la hora de elegir el SGBD adecuado, también debemos fijarnos en el precio. MySQL está destinado a medianas y grandes empresas que pueden permitirse un desembolso considerable en su SGBD.

Por otra parte, el precio de MongoDB es variado, pero considerando nuestras necesidades el precio que mejor se ajustaría sería el de 2GB de RAM y 10GB de Almacenamiento. Este coste sería de 0,08 USD por hora.

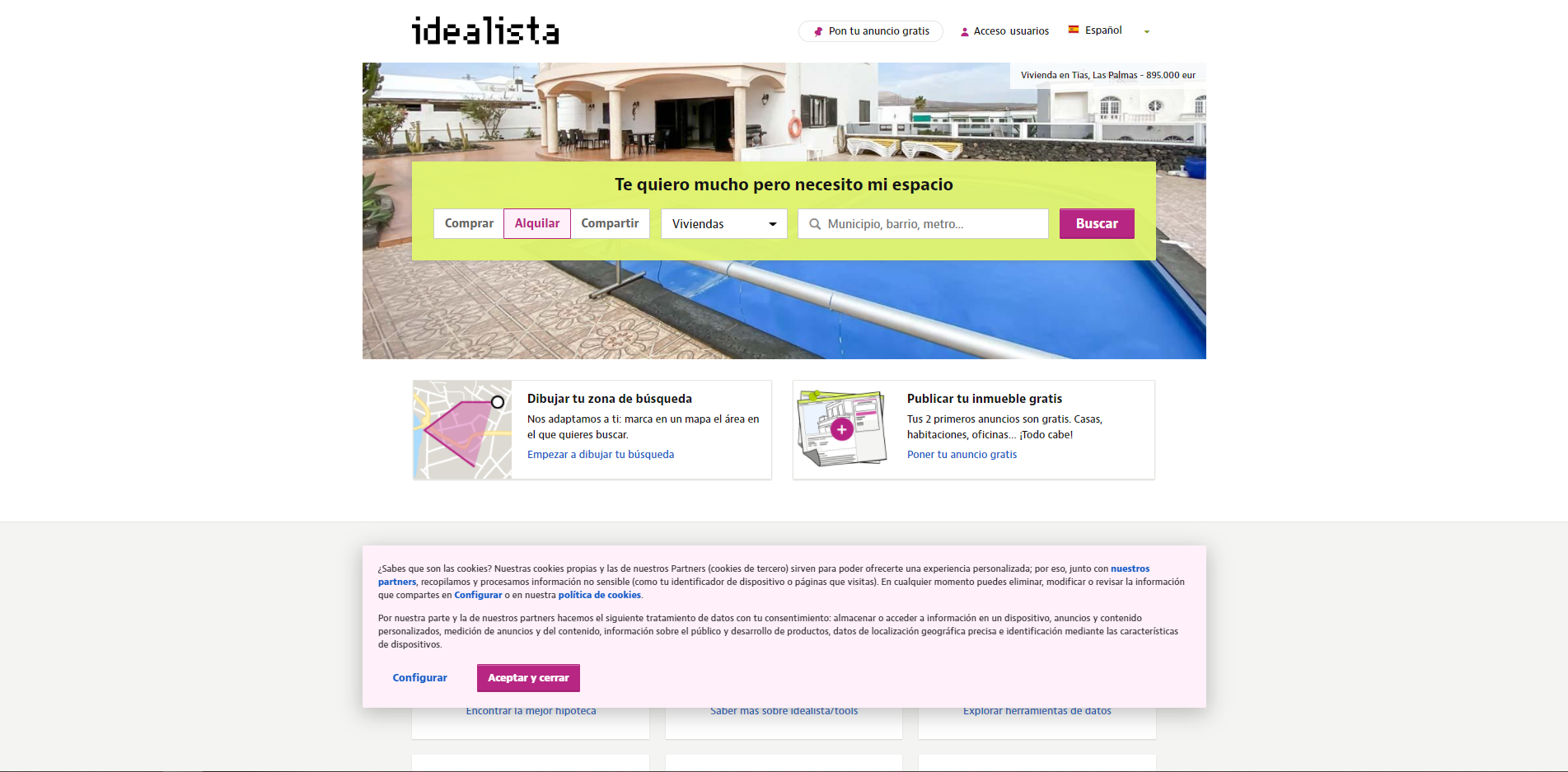
También, MongoDB se integra muy bien con nuestra anterior elección, gracias a NPM, su sistema de instalación de paquetes. Mediante un par de sencillas instalaciones, podremos utilizar MongoDB junto a NodeJS. Debido a esto, será sencillo integrarlos juntos y trabajar con ambos.

# CAPÍTULO 4. ESTUDIO DE VIABILIDAD DEL SISTEMA

## Sistema 1: Idealista (https://www.idealista.com/)

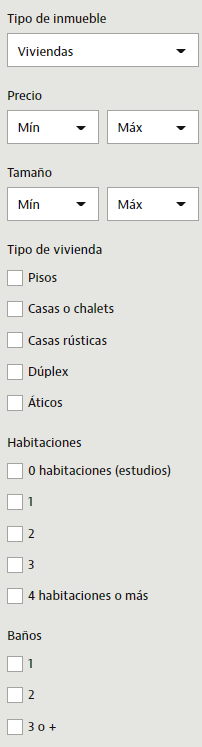
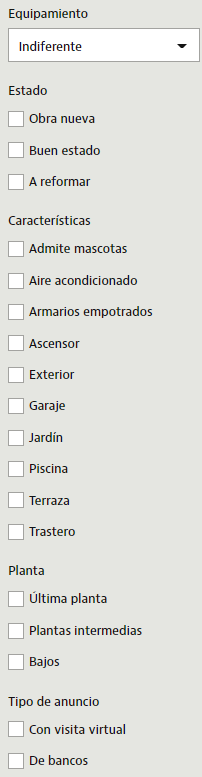
Idealista es uno de los portales inmobiliarios más conocidos de España. Nada más entrar en su página nos ofrece la oportunidad de acceder directamente a una serie de resultados dependiendo de unos pequeños campos que introduzcamos. Tan solo hay que aportar si se desea comprar, alquilar o compartir, el tipo de inmueble que se desea y una zona. Por otra parte, está el área de clientes, la cual nos permite obtener una funcionalidad más personalizada.

*Figura 4: Logotipo de idealista.com*



*Figura 5: Pantalla Principal de idealista.com*

Tras rellenar el filtro inicial con nuestros campos deseados, si introducimos una ciudad, como por ejemplo Oviedo, se nos ofrece la opción de seleccionar una zona de esta, para así obtener un resultado más personalizado. Una vez en la lista de inmuebles ofrecidos, tenemos un filtro más extenso que nos permite perfilar nuestro interés hasta el más mínimo detalle.



*Figuras 6 y 7: Filtro de búsqueda de idealista.com*

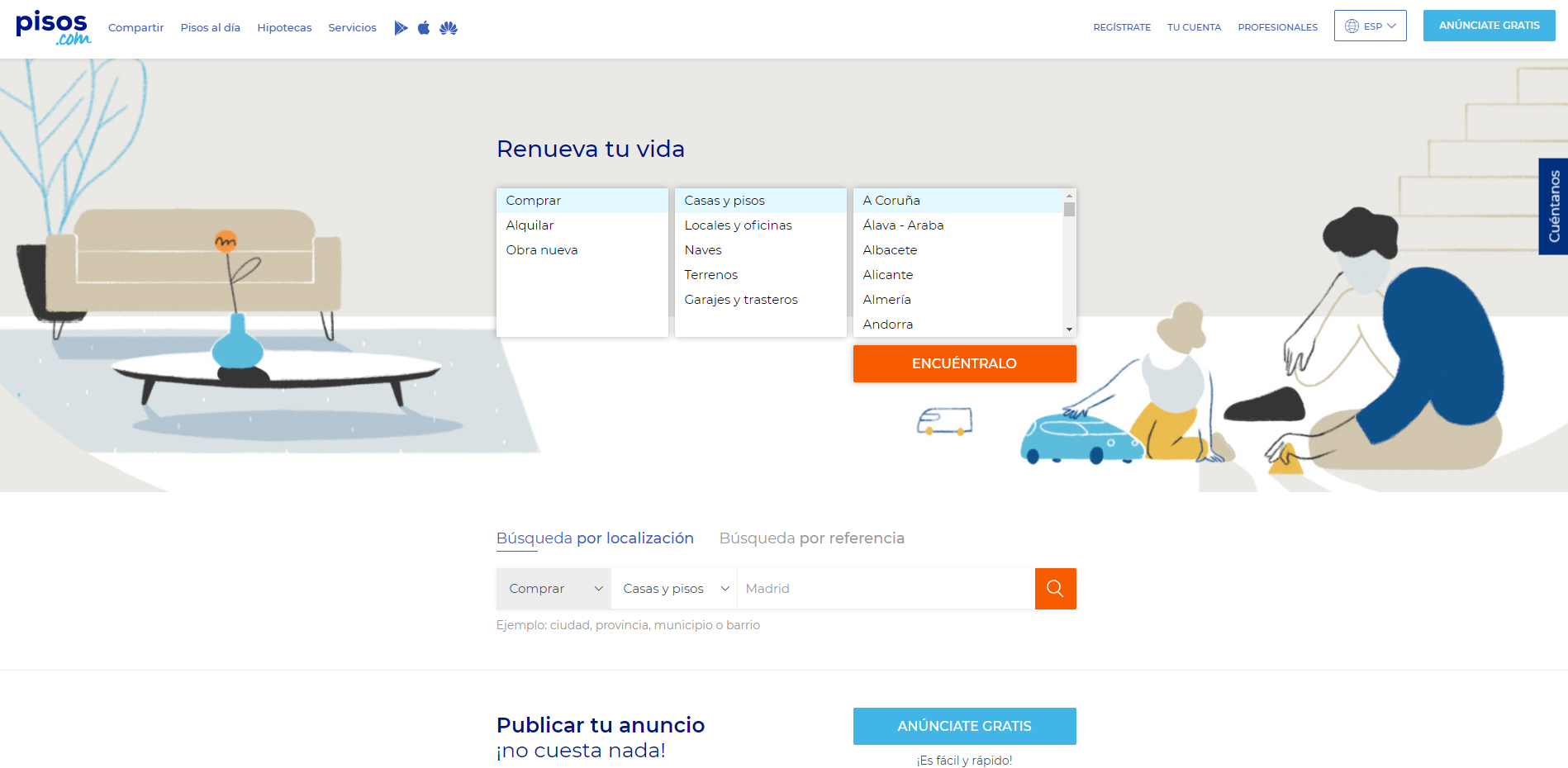
Una vez accedemos a la presentación del inmueble, obtenemos una visual de toda la información general del inmueble, seguido de varias opciones interesantes como contactar con el vendedor, guardar en favoritos, ver en el mapa y compartir.

Para vender un inmueble, primero se te solicita una información general, siendo la localidad, la calle, el número y el tipo de oferta (venta o alquiler). Una vez se introduce la calle aparece un mapa con la localización de esta. El vendedor debe indicar si la localización que aparece en el mapa es la misma que indico. Una vez pasado el primer filtro, aparece una serie de campos mucho más específicos para indicar exactamente como es el inmueble que ofrece.

## Sistema 2: Pisos.com (https://www.pisos.com/)

Pisos.com es probablemente la web más rápida y sencilla de todas estas. Se puede apreciar que se ofrece una página web muy integrada para personas jóvenes que desean hacer estos trámites de una manera rápida y sencilla. Como ocurría en el caso anterior, lo primero que obtenemos es un filtro de búsqueda rápida para empezar a perfilar los deseos del cliente.

*Figura 8: Logotipo de pisos.com*

**

*Figura 9: Pantalla Principal de Pisos.com*

Tras pasar el primer filtro, nos permite elegir, en este caso, el concejo de Asturias en el que deseamos buscar inmueble. Una vez seleccionado, nos ofrece una elección de barrio y, tras elegir, nos aparecen las ofertas disponibles, junto con un filtro algo más extenso.



*Figura 10: Filtro de búsqueda de pisos.com*

Al acceder a un inmueble, obtenemos una vista mucho más gráfica que en el caso de Idealista, pues hace mucho más uso de imágenes e iconos. Dentro de la presentación podemos obtener opciones de contacto con el anunciante, de notificar si hay cambios, de añadir a favoritos, de compartir y de pedir más información, entre otros.

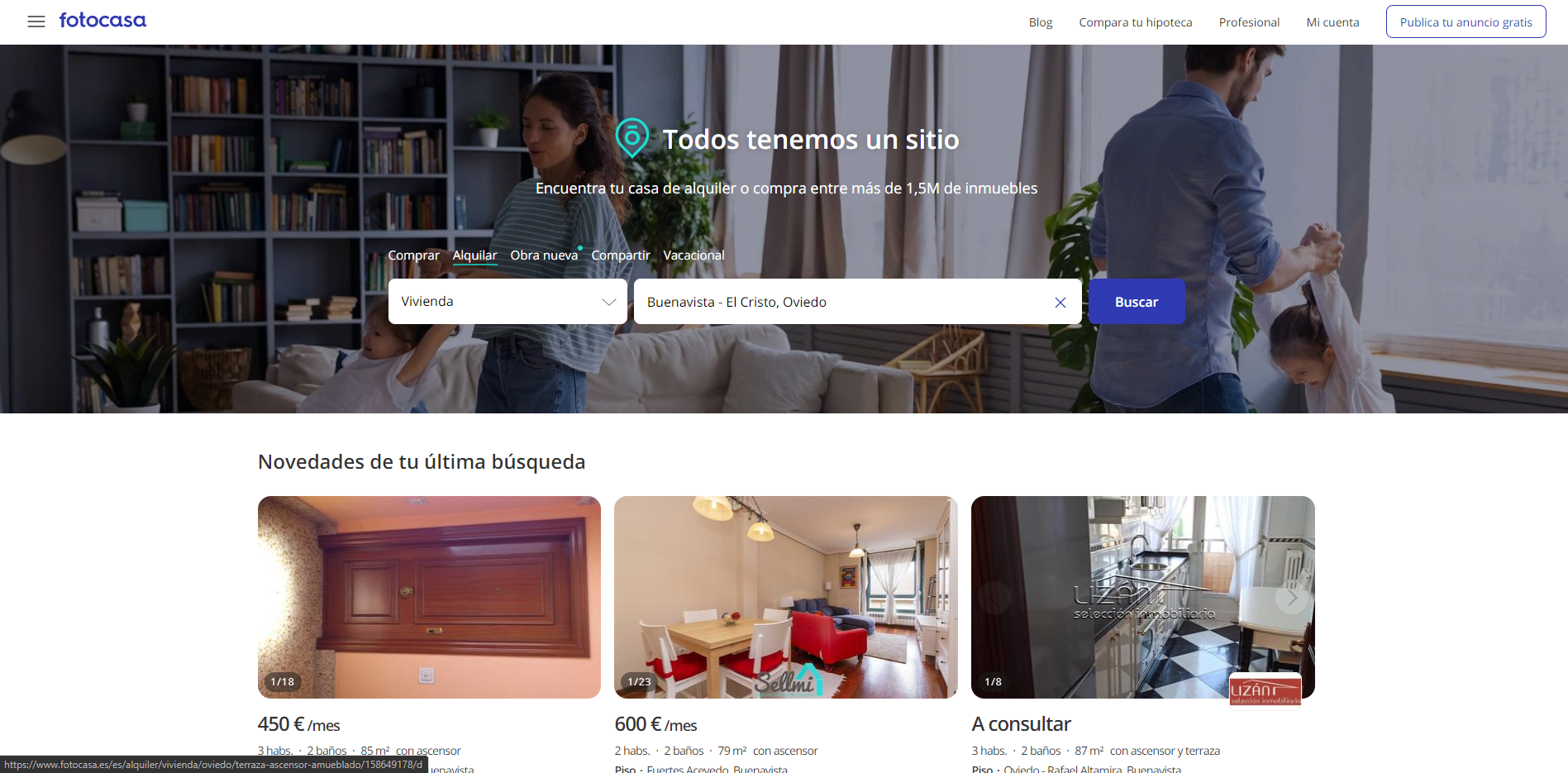
La venta del inmueble es mucho más rápida que en los demás portales. Sencillamente se ofrecen tres pestañas:

* **Datos Generales:** Aquí se introduce toda la información que se desee mostrar del inmueble.
* **Fotos y vídeos:** En ella se aporta toda la información multimedia deseada.
* **Confirmación:** Proceso de permisos y confirmación mediante e-mail.

## Sistema 3: Fotocasa (https://www.fotocasa.es/es/)

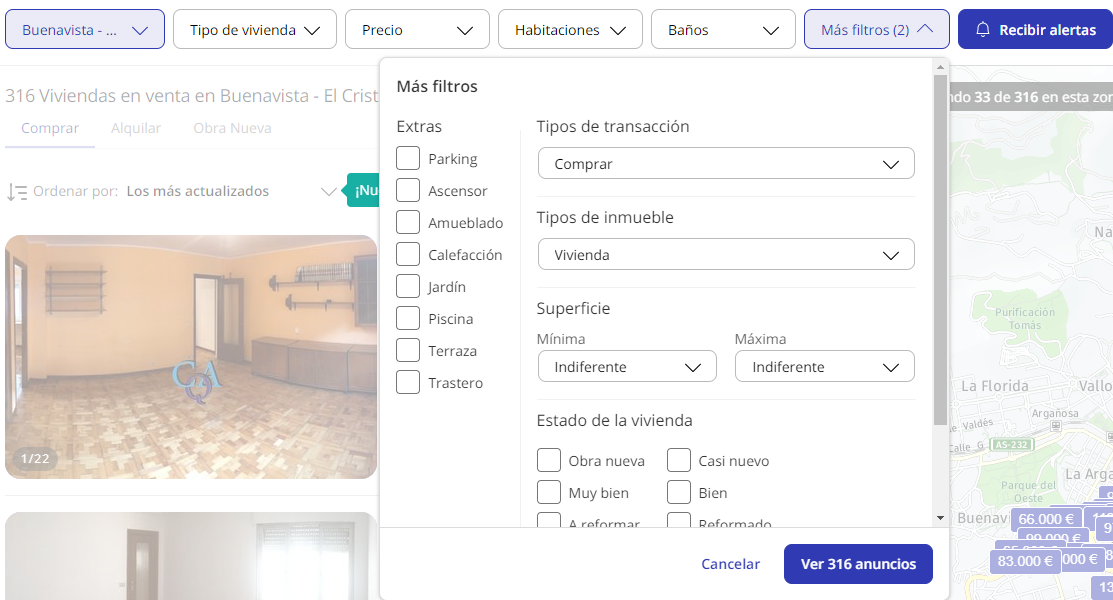
Fotocasa se coloca entre las dos opciones anteriores. Tiene el toque moderno de pisos.com pero el detalle y la atención de idealista. Como en los casos anteriores, lo primero que vemos al acceder al portal es un filtro rápido de información.

*Figura 11: Logotipo de fotocasa.es*

**

*Figura 12: Pantalla Principal de Fotocasa.es*

En este caso no se obliga al cliente a aportar un barrio de la ciudad en la que desea buscar, sino que se le aporta como opción. El filtro de búsqueda ampliado tiene, según mi punto de vista, la información justa y necesaria.



*Figura 13: Filtro de búsqueda de fotocasa.es*

La presentación de los inmuebles es algo más simple que en los casos anteriores, pero ofreciendo siempre la opción de solicitar más datos al anunciante. También ofrece la opción de compartir, guardar en favoritos y contactar con el vendedor.

En este caso todo el formulario de venta se ofrece en la misma pantalla. Se debe introducir una serie de campos amplios, así como la localización de la vivienda en el mapa.

## Sistema 4: Inmoweb (https://www.inmoweb.es/)

Inmoweb es un portal inmobiliario que permite crear y mantener tu propia página web. Dicho portal actúa como un modo de gestión de administrador, dejando la propia aplicación únicamente para usuarios y clientes. Al no ser una aplicación Web, no merece mucha atención, pero sí es importante mencionarlo, pues es un gran competidor en el mercado inmobiliario.

**Conclusión**

|  |  |  |
| --- | --- | --- |
| **SISTEMA** | **VENTAJAS** | **INCONVENIENTES** |
| **S.1 - Idealista** | **Filtro muy completo.**  **Seguimientos.** | **Difícil de entender.** |
| **S.2 – Pisos.com** | **Buen uso de imágenes simbólicas**  **Procesos rápidos y con poco esfuerzo** | **Demasiado simple** |
| **S.3 - Fotocasa** | **Filtro extenso y detallado.**  **Buen uso de imágenes.** | **Toda la información en una pantalla (Puede confundir al cliente)** |

# CAPÍTULO 5. PLANIFICACIÓN Y GESTIÓN DEL TFG

En este capítulo se dejará constancia de todos los puntos importantes del proyecto desde un punto de vista de planificación. Se incluirán diferentes apartados de importancia como riegos, presupuestos o planificación de tareas.

## Planificación del Proyecto

### Identificación de Interesados

### OBS y PBS

### Planificación Inicial. WBS

### Riesgos

#### Plan de Gestión de Riesgos

#### Identificación de Riesgos

#### Registro de Riesgos

### Presupuesto Inicial

#### Presupuesto de Costes

#### Presupuesto de Cliente

# CAPÍTULO 6. ANÁLISIS DEL SISTEMA DE INFORMACIÓN

## Definición del Sistema

### Determinación del Alcance del Sistema

Ya que el proyecto se va a desarrollar con la finalidad de utilizarlo en un entorno de explotación real, el proceso de desarrollo de este será el mismo, así como el proceso de documentación se realizará como si de un proyecto completo se tratara.

No obstante, se debe tener en cuenta que para poder realizar algunas tareas como el despliegue de la aplicación necesitaríamos de una financiación económica de la cual no se dispone. Debido a ello, se realizará el desarrollo de la aplicación y se ejecutará en un servidor local montado en la misma máquina del desarrollo. Este servidor no será el utilizado con finalidad a la hora de publicar nuestra aplicación, pero servirá durante el proceso de desarrollo y la exposición y presentación del proyecto final.

Algo similar ocurre con la Base de Datos en MongoDB. Pese a ser de pago, es posible obtener una versión gratuita con 510MB de memoria. Este espacio es más que suficiente para probar la aplicación (ya que, al almacenarse los datos en documentos, estos ocupan poco trabajo). El mayor problema en cuanto a almacenamiento serían las fotografías de los inmuebles. Para solucionar ese problema, utilizaremos un módulo de NPM derivado de express llamado express-fileupload, el cual se encarga de obtener las fotografías y guardarlas en una carpeta dentro del proyecto de la aplicación con un ID único, guardando dicho ID en la base de datos. Una vez necesitemos obtener esa imagen, podemos utilizar el ID obtenido de la colección y obtener la imagen correspondiente de la carpeta.

## Establecimiento de Requisitos

### Obtención de los Requisitos del Sistema

Los requisitos que se obtengan seguirán la siguiente nomenclatura:

* **Índice:** Cada requisito tendrá un único índice, el cual estará clasificado jerárquicamente, desde los más genéricos hasta los más concretos y específicos. En el caso de ser funcionales, vendrán acompañados por una F, mientras que, si son no funcionales, aparecerán con NF.
* **Nombre:** Nombre del requisito. Permitirá agrupar los requisitos en función de su objetivo.
* **Descripción:** Texto explicando la funcionalidad del requisito de manera extendida. Deberá explicar de forma clara y concisa qué condiciones se están imponiendo.

#### Requisitos Funcionales

##### Requisitos de la Gestión de Perfiles

Aquí quedan incluidos todos los requisitos del sistema referentes a la gestión de los diferentes tipos de perfiles por parte del sistema informático.

|  |  |  |
| --- | --- | --- |
| **GESTIÓN DE PERFILES** | | |
| **Índice** | **Nombre** | **Descripción** |
| RF.1 | Gestión de usuarios | El sistema permitirá la gestión de cuentas tanto de tipo agente como de tipo usuario. |
| >RF.1.1 | Creación de usuarios | El sistema permitirá la creación de cuentas de usuario y de agente, mediante la introducción de diferentes datos. |
| >>RF.1.1.1 | Restricciones | Una cuenta de tipo agente solo podrá ser creada por una de tipo super-agente. |
| >>RF.1.1.2 | Datos | Se deberá introducir un nombre. |
| >>RF.1.1.3 | Datos | Se deberá introducir unos apellidos. |
| >>RF.1.1.4 | Datos | Se deberá introducir un correo electrónico. |
| >>RF.1.1.5 | Datos | Se deberá introducir una contraseña. |
| >>>RF.1.1.5.1 | Restricciones | La contraseña deberá tener al menos 8 caracteres o dígitos. |
| >>>RF.1.1.5.2 | Restricciones | La contraseña deberá tener, al menos, una letra minúscula. |
| >>>RF.1.1.5.3 | Restricciones | La contraseña deberá tener, al menos, una letra mayúscula. |
| >>>RF.1.1.5.4 | Restricciones | La contraseña deberá tener, al menos, un dígito. |
| >>RF.1.1.6 | Verificación de correo electrónico | El sistema se encargará de verificar la autenticidad del correo electrónico. |
| >>>RF.1.1.6.1 | Modo de verificación | El sistema enviará un correo electrónico desde el cual el cliente podrá verificar su dirección. |
| >RF.1.2 | Modificación de usuarios | El sistema permitirá la modificación de las cuentas, tanto de tipo usuario como agente. |
| >>RF.1.2.1 | Datos | Se podrá modificar el nombre. |
| >>RF.1.2.2 | Datos | Se podrán modificar los apellidos. |
| >>RF.1.2.3 | Datos | Se podrá modificar la contraseña. |
| >>RF.1.2.3.1 | Restricciones | El sistema solicitará la contraseña al usuario dos veces, para asegurar que este no se equivoque. |
| >RF.1.3 | Eliminación de usuarios | El sistema permitirá la eliminación de los usuarios |
| >>RF.1.3.1 | Restricciones | Sólo una cuenta de tipo agente podrá borrar cualquier cuenta de tipo usuario. |
| >>RF.1.3.1 | Restricciones | Una cuenta de tipo usuario podrá borrarse a sí misma. |
| >RF.1.4 | Recuperación de contraseña | El sistema permitirá al usuario reestablecer su contraseña si éste la olvida. |
| >>RF.1.4.1 | Recuperación de contraseña | Para reestablecer la contraseña, el sistema enviará un correo electrónico con un enlace desde el cual podrá actualizarla. |

##### Requisitos de la Gestión de Propiedades

Aquí se detallan todos los procesos que el sistema deberá abarcar en lo que se refiere a gestión de las propiedades (añadir, modificar y eliminar).

|  |  |  |
| --- | --- | --- |
| **GESTIÓN DE INMUEBLES** | | |
| **Índice** | **Nombre** | **Descripción** |
| RF.2 | Gestor de Inmuebles | El sistema permitirá gestionar los diferentes tipos de propiedad. |
| >RF.2.1 | Tipos de Inmuebles | El sistema permitirá añadir tres tipos de propiedad. |
| >>RF.2.1.1 | Tipo de Inmuebles | El sistema permitirá añadir propiedades de tipo “Vivienda”. |
| >>>RF.2.1.1.1 | Datos | Se deberá incluir un nombre. |
| >>>RF.2.1.1.2 | Datos | Se deberá incluir un tipo. |
| >>>>RF.2.1.1.2.1 | Restricciones | El tipo podrá ser Alquiler o Venta. |
| >>>RF.2.1.1.3 | Datos | Se deberá incluir un código identificativo. |
| >>>RF.2.1.1.4 | Datos | Se deberá incluir una descripción. |
| >>>RF.2.1.1.5 | Datos | Se deberá incluir una dirección. |
| >>>RF.2.1.1.6 | Datos | Se deberá incluir una localización. |
| >>>RF.2.1.1.7 | Datos | Se deberá incluir la superficie que ocupa la propiedad. |
| >>>RF.2.1.1.8 | Datos | Se deberá incluir el número de habitaciones. |
| >>>RF.2.1.1.9 | Datos | Se deberá incluir el número de baños. |
| >>>RF.2.1.1.10 | Datos | Se deberá incluir si la propiedad tiene garaje o no. |
| >>>RF.2.1.1.11 | Datos | Se deberá incluir si la propiedad tiene piscina o no. |
| >>>RF.2.1.1.12 | Datos | Se deberá incluir si la propiedad tiene terraza o no. |
| >>>RF.2.1.1.13 | Datos | Se deberá incluir si la propiedad tiene jardín o no. |
| >>>RF.2.1.1.14 | Datos | Se deberá incluir si la propiedad tiene calefacción o no. |
| >>>RF.2.1.1.15 | Datos | Se deberá incluir si la propiedad tiene aire acondicionado o no. |
| >>>RF.2.1.1.16 | Datos | Se deberá incluir si la propiedad tiene ascensor o no. |
| >>>RF.2.1.1.17 | Datos | Se deberá incluir si la propiedad esta amueblada o no. |
| >>>RF.2.1.1.18 | Datos | Se deberá incluir si la propiedad permite mascotas o no. |
| >>>RF.2.1.1.19 | Datos | Se deberá incluir el precio de la propiedad. |
| >>>RF.2.1.1.20 | Datos | Se deberán incluir los honorarios del agente. |
| >>>RF.2.1.1.21 | Datos | Se deberá incluir el precio de gasto de comunidad de la propiedad. |
| >>>RF.2.1.1.22 | Datos | Se deberá incluir si la propiedad tiene trastero o no. |
| >>RF.2.1.2 | Tipo de Inmuebles | El sistema permitirá añadir propiedades de tipo “Local”. |
| >>>RF.2.1.2.1 | Datos | Se deberá incluir un nombre. |
| >>>RF.2.1.2.2 | Datos | Se deberá incluir un tipo. |
| >>>>RF.2.1.2.2.1 | Restricciones | El tipo podrá ser Alquiler o Venta. |
| >>>RF.2.1.2.3 | Datos | Se deberá incluir un código identificativo. |
| >>>RF.2.1.2.4 | Datos | Se deberá incluir una descripción. |
| >>>RF.2.1.2.5 | Datos | Se deberá incluir una dirección. |
| >>>RF.2.1.2.6 | Datos | Se deberá incluir una localización. |
| >>>RF.2.1.2.7 | Datos | Se deberá incluir la superficie que ocupa la propiedad. |
| >>>RF.2.1.2.8 | Datos | Se deberá incluir si el local tiene escaparates o no. |
| >>>RF.2.1.2.9 | Datos | Se deberá incluir si el local tiene aparcamientos o no. |
| >>>RF.2.1.2.10 | Datos | Se deberá incluir si el local tiene zona de carga y descarga. |
| >>>RF.2.1.2.11 | Datos | Se deberá incluir si el local posee un sistema de extinción de incendios. |
| >>>RF.2.1.2.12 | Datos | Se deberá incluir el número de aseos que posee el local. |
| >>>RF.2.1.2.13 | Datos | Se deberá incluir si el local tiene iluminación o no. |
| >>>RF.2.1.2.14 | Datos | Se deberá incluir si el local tiene calefacción. |
| >>>RF.2.1.2.15 | Datos | Se deberá incluir si el local tiene aire acondicionado. |
| >>>RF.2.1.2.16 | Datos | Se deberá incluir el precio del local. |
| >>>RF.2.1.2.17 | Datos | Se deberán incluir los honorarios del agente. |
| >>>RF.2.1.2.18 | Datos | Se deberá incluir el precio de gasto de comunidad del local. |
| >>RF.2.1.3 | Tipo de Inmuebles | El sistema permitirá añadir propiedades de tipo “Suelo”. |
| >>>RF.2.1.3.1 | Datos | Se deberá incluir un nombre. |
| >>>RF.2.1.3.2 | Datos | Se deberá incluir un tipo. |
| >>>>RF.2.1.3.2.1 | Restricciones | El tipo podrá ser Alquiler o Venta. |
| >>>RF.2.1.3.3 | Datos | Se deberá incluir un código identificativo. |
| >>>RF.2.1.3.4 | Datos | Se deberá incluir una descripción. |
| >>>RF.2.1.3.5 | Datos | Se deberá incluir la superficie del suelo. |
| >>>RF.2.1.3.6 | Datos | Se deberá incluir la situación geográfica del suelo. |
| >>>RF.2.1.3.7 | Datos | Se deberá incluir el precio total del suelo. |
| >>>RF.2.1.3.8 | Datos | Se deberán incluir los honorarios del agente. |
| >>>RF.2.1.3.9 | Datos | Se deberá incluir si el suelo tiene acceso a agua. |
| >>>RF.2.1.3.10 | Datos | Se deberá incluir si el suelo tiene acceso a luz. |
| >>>RF.2.1.3.11 | Datos | Se deberá incluir la superficie edificable del suelo. |
| >>>RF.2.1.3.12 | Datos | Se deberá incluir la edificabilidad del suelo. |
| >>>RF.2.1.3.13 | Datos | Se deberá incluir la calificación urbanística del suelo. |
| >RF.2.2 | Tipos de Inmuebles | El sistema permitirá modificar tres tipos de propiedad. |
| >>RF.2.2.1 | Tipo de Inmuebles | El sistema permitirá modificar propiedades de tipo “Vivienda”. |
| >>>RF.2.2.1.1 | Datos | Se podrá modificar el nombre. |
| >>>RF.2.2.1.2 | Datos | Se podrá modificar el tipo. |
| >>>RF.2.2.1.3 | Datos | No se podrá modificar el código identificativo |
| >>>RF.2.2.1.4 | Datos | Se podrá modificar la descripción. |
| >>>RF.2.2.1.5 | Datos | Se podrá modificar la dirección. |
| >>>RF.2.2.1.6 | Datos | Se podrá modificar la localización. |
| >>>RF.2.2.1.7 | Datos | Se podrá modificar la superficie que ocupa la propiedad. |
| >>>RF.2.2.1.8 | Datos | Se podrá modificar el número de habitaciones. |
| >>>RF.2.2.1.9 | Datos | Se podrá modificar el número de baños. |
| >>>RF.2.2.1.10 | Datos | Se podrá modificar si la propiedad tiene garaje o no. |
| >>>RF.2.2.1.11 | Datos | Se podrá modificar si la propiedad tiene piscina o no. |
| >>>RF.2.2.1.12 | Datos | Se podrá modificar si la propiedad tiene terraza o no. |
| >>>RF.2.2.1.13 | Datos | Se podrá modificar si la propiedad tiene jardín o no. |
| >>>RF.2.2.1.14 | Datos | Se podrá modificar si la propiedad tiene calefacción o no. |
| >>>RF.2.2.1.15 | Datos | Se podrá modificar si la propiedad tiene aire acondicionado o no. |
| >>>RF.2.2.1.16 | Datos | Se podrá modificar si la propiedad tiene ascensor o no. |
| >>>RF.2.2.1.17 | Datos | Se podrá modificar si la propiedad esta amueblada o no. |
| >>>RF.2.2.1.18 | Datos | Se podrá modificar si la propiedad permite mascotas o no. |
| >>>RF.2.2.1.19 | Datos | Se podrá modificar el precio de la propiedad. |
| >>>RF.2.2.1.20 | Datos | Se podrán modificar los honorarios del agente. |
| >>>RF.2.2.1.21 | Datos | Se podrá modificar el precio de gasto de comunidad de la propiedad. |
| >>>RF.2.2.1.22 | Datos | Se podrá modificar si la propiedad tiene trastero o no. |
| >>RF.2.2.2 | Tipo de Inmuebles | El sistema permitirá modificar propiedades de tipo “Local”. |
| >>>RF.2.2.2.1 | Datos | Se podrá modificar el nombre. |
| >>>RF.2.2.2.2 | Datos | Se podrá modificar el tipo. |
| >>>RF.2.2.2.3 | Datos | No se podrá modificar el código identificativo. |
| >>>RF.2.2.2.4 | Datos | Se podrá modificar la descripción. |
| >>>RF.2.2.2.5 | Datos | Se podrá modificar la dirección. |
| >>>RF.2.2.2.6 | Datos | Se podrá modificar la localización. |
| >>>RF.2.2.2.7 | Datos | Se podrá modificar la superficie que ocupa la propiedad. |
| >>>RF.2.2.2.8 | Datos | Se podrá modificar si el local tiene escaparates o no. |
| >>>RF.2.2.2.9 | Datos | Se podrá modificar si el local tiene aparcamientos o no. |
| >>>RF.2.2.2.10 | Datos | Se podrá modificar si el local tiene zona de carga y descarga. |
| >>>RF.2.2.2.11 | Datos | Se podrá modificar si el local posee un sistema de extinción de incendios. |
| >>>RF.2.2.2.12 | Datos | Se podrá modificar el número de aseos que posee el local. |
| >>>RF.2.2.2.13 | Datos | Se podrá modificar si el local tiene iluminación o no. |
| >>>RF.2.2.2.14 | Datos | Se podrá modificar si el local tiene calefacción. |
| >>>RF.2.2.2.15 | Datos | Se podrá modificar si el local tiene aire acondicionado. |
| >>>RF.2.2.2.16 | Datos | Se podrá modificar el precio del local. |
| >>>RF.2.2.2.17 | Datos | Se podrán modificar los honorarios del agente. |
| >>>RF.2.2.2.18 | Datos | Se podrá modificar el precio de gasto de comunidad del local. |
| >>RF.2.2.3 | Tipo de Inmuebles | El sistema permitirá modificar propiedades de tipo “Suelo”. |
| >>>RF.2.2.3.1 | Datos | Se podrá modificar el nombre. |
| >>>RF.2.2.3.2 | Datos | Se podrá modificar el tipo. |
| >>>RF.2.2.3.3 | Datos | No se podrá modificar el código identificativo. |
| >>>RF.2.2.3.4 | Datos | Se podrá modificar la descripción. |
| >>>RF.2.2.3.5 | Datos | Se podrá modificar la superficie del suelo. |
| >>>RF.2.2.3.6 | Datos | Se podrá modificar la situación geográfica del suelo. |
| >>>RF.2.2.3.7 | Datos | Se podrá modificar el precio total del suelo. |
| >>>RF.2.2.3.8 | Datos | Se podrán modificar los honorarios del agente. |
| >>>RF.2.2.3.9 | Datos | Se podrá modificar si el suelo tiene acceso a agua. |
| >>>RF.2.2.3.10 | Datos | Se podrá modificar si el suelo tiene acceso a luz. |
| >>>RF.2.2.3.11 | Datos | Se podrá modificar la superficie edificable del suelo. |
| >>>RF.2.2.3.12 | Datos | Se podrá modificar la edificabilidad del suelo. |
| >>>RF.2.2.3.13 | Datos | Se podrá modificar la calificación urbanística del suelo. |
| >RF.2.3 | Tipos de Inmuebles | El sistema permitirá eliminar tres tipos de propiedad. |
| >>RF.2.3.1 | Tipo de Inmuebles | El sistema permitirá eliminar propiedades de tipo “Vivienda”. |
| >>RF.2.3.2 | Tipo de Inmuebles | El sistema permitirá eliminar propiedades de tipo “Local”. |
| >>RF.2.3.3 | Tipo de Inmuebles | El sistema permitirá eliminar propiedades de tipo “Suelo”. |

##### Requisitos de la Gestión de Propietarios

En este apartado se detalla todo lo referido a la información de los propietarios de los inmuebles. También se detallan las operaciones que pueden realizarse con los mismos.

|  |  |  |
| --- | --- | --- |
| **GESTIÓN DE PROPIETARIOS** | | |
| **Índice** | **Nombre** | **Descripción** |
| RF.3 | Gestión de Propietarios | El sistema deberá gestionar lo referido a el propietario de un inmueble. |
| >RF.3.1 | Añadir Propietarios | El sistema permitirá añadir la información de un propietario a una propiedad. |
| >>RF.3.1.1 | Datos | Se deberá añadir el nombre. |
| >>RF.3.1.2 | Datos | Se deberá añadir los apellidos. |
| >>RF.3.1.3 | Datos | Se deberá añadir un número de teléfono. |
| >>RF.3.1.4 | Datos | Se deberá añadir un correo electrónico. |
| >>RF.3.1.3 | Datos | Se deberá añadir un documento identificativo. |
| >>>RF.3.1.3.1 | Restricciones | El documento identificativo podrá ser el NIF, NIE o DNI. |
| >>RF.3.1.4 | Datos | Se deberá añadir una dirección. |
| >RF.3.2 | Modificar Propietarios | El sistema permitirá modificar la información de un propietario a una propiedad. |
| >>RF.3.2.1 | Datos | Se podrá modificar el nombre. |
| >>RF.3.2.2 | Datos | Se podrán modificar los apellidos. |
| >>RF.3.2.3 | Datos | Se podrá modificar el número de teléfono. |
| >>RF.3.2.4 | Datos | Se podrá modificar el correo electrónico. |
| >>RF.3.2.3 | Datos | Se podrá modificar el documento identificativo. |
| >>RF.3.2.4 | Datos | Se podrá modificar la dirección. |
| >RF.3.3 | Eliminar Propietarios | El sistema permitirá eliminar la información de un propietario junto a su propiedad. |

##### Visualización de datos de las Propiedades y Propietarios

Aquí se incluye toda la información relativa a la información a mostrar de las diferentes propiedades.

|  |  |  |
| --- | --- | --- |
| **VISUALIZACION DE DATOS** **DE PROPIEDADES Y PROPIETARIOS** | | |
| **Índice** | **Nombre** | **Descripción** |
| RF.4 | Muestreo de Propiedades | El sistema permitirá mostrar la información tanto de las diferentes propiedades como de sus propietarios. |
| >RF.4.1 | Muestreo de propietarios | El sistema permitirá mostrar la información de los propietarios junto a la de sus propiedades. |
| >>RF.4.1.1 | Restricciones | Solo los agentes y el super-agente podrán ver la información de los propietarios. |
| >>RF.4.1.2 | Datos de propietarios | El sistema mostrara diferentes datos de los propietarios. |
| >>>RF.4.1.2.1 | Datos | Permitirá mostrar el nombre del propietario. |
| >>>RF.4.1.2.2 | Datos | Permitirá mostrar los apellidos del propietario. |
| >>>RF.4.1.2.3 | Datos | Permitirá mostrar el documento identificativo del propietario. |
| >>>RF.4.1.2.4 | Datos | Permitirá mostrar el domicilio del propietario. |
| >>>RF.4.1.2.5 | Datos | Permitirá mostrar el teléfono del propietario. |
| >>>RF.4.1.2.6 | Datos | Permitirá mostrar el correo electrónico del propietario. |
| >RF.4.2 | Tipos de Inmuebles | El sistema permitirá mostrar la información tres tipos de propiedad. |
| >>RF.4.2.1 | Tipo de Inmuebles | El sistema permitirá mostrar la información de las propiedades de tipo “Vivienda”. |
| >>>RF.4.2.1.1 | Datos | Permitirá mostrar su nombre. |
| >>>RF.4.2.1.2 | Datos | Permitirá mostrar su tipo. |
| >>>RF.4.2.1.3 | Datos | Permitirá mostrar su código identificativo. |
| >>>>RF.4.2.1.3.1 | Restricciones | El código identificativo sólo podrá ser visto por una cuenta de tipo “agente” o “super-agente” |
| >>>RF.4.2.1.4 | Datos | Permitirá mostrar su descripción. |
| >>>RF.4.2.1.5 | Datos | Permitirá mostrar su dirección. |
| >>>RF.4.2.1.6 | Datos | Permitirá mostrar su localización. |
| >>>RF.4.2.1.7 | Datos | Permitirá mostrar la superficie que ocupa la propiedad. |
| >>>RF.4.2.1.8 | Datos | Permitirá mostrar su número de habitaciones. |
| >>>RF.4.2.1.9 | Datos | Permitirá mostrar su número de baños. |
| >>>RF.4.2.1.10 | Datos | Permitirá mostrar si la propiedad tiene garaje o no. |
| >>>RF.4.2.1.11 | Datos | Permitirá mostrar si la propiedad tiene piscina o no. |
| >>>RF.4.2.1.12 | Datos | Permitirá mostrar si la propiedad tiene terraza o no. |
| >>>RF.4.2.1.13 | Datos | Permitirá mostrar si la propiedad tiene jardín o no. |
| >>>RF.4.2.1.14 | Datos | Permitirá mostrar si la propiedad tiene calefacción o no. |
| >>>RF.4.2.1.15 | Datos | Permitirá mostrar si la propiedad tiene aire acondicionado o no. |
| >>>RF.4.2.1.16 | Datos | Permitirá mostrar si la propiedad tiene ascensor o no. |
| >>>RF.4.2.1.17 | Datos | Permitirá mostrar si la propiedad esta amueblada o no. |
| >>>RF.4.2.1.18 | Datos | Permitirá mostrar si la propiedad permite mascotas o no. |
| >>>RF.4.2.1.19 | Datos | Permitirá mostrar el precio de la propiedad. |
| >>>RF.4.2.1.20 | Datos | Permitirá mostrar los honorarios del agente. |
| >>>>RF.4.2.1.20.1 | Restricciones | Los honorarios del agente sólo podrán ser vistos por una cuenta de tipo “agente” o “super-agente” |
| >>>RF.4.2.1.21 | Datos | Permitirá mostrar el precio de gasto de comunidad de la propiedad. |
| >>>RF.4.4.1.22 | Datos | Permitirá mostrar si la propiedad tiene trastero o no. |
| >>RF.4.2.2 | Tipo de Inmuebles | El sistema permitirá mostrar la información de las propiedades de tipo “Local”. |
| >>>RF.4.2.2.1 | Datos | Permitirá mostrar su nombre. |
| >>>RF.4.2.2.2 | Datos | Permitirá mostrar su tipo. |
| >>>RF.4.2.2.3 | Datos | Permitirá mostrar su código identificativo. |
| >>>>RF.4.2.2.3.1 | Restricciones | El código identificativo sólo podrá ser visto por una cuenta de tipo “agente” o “super-agente”. |
| >>>RF.4.2.2.4 | Datos | Permitirá mostrar su descripción. |
| >>>RF.4.2.2.5 | Datos | Permitirá mostrar su dirección. |
| >>>RF.4.2.2.6 | Datos | Permitirá mostrar su localización. |
| >>>RF.4.2.2.7 | Datos | Permitirá mostrar la superficie que ocupa la propiedad. |
| >>>RF.4.2.2.8 | Datos | Permitirá mostrar si el local tiene escaparates o no. |
| >>>RF.4.2.2.9 | Datos | Permitirá mostrar si el local tiene aparcamientos o no. |
| >>>RF.4.2.2.10 | Datos | Permitirá mostrar si el local tiene zona de carga y descarga. |
| >>>RF.4.2.2.11 | Datos | Permitirá mostrar si el local posee un sistema de extinción de incendios. |
| >>>RF.4.2.2.12 | Datos | Permitirá mostrar el número de aseos que posee el local. |
| >>>RF.4.2.2.13 | Datos | Permitirá mostrar si el local tiene iluminación o no. |
| >>>RF.4.2.2.14 | Datos | Permitirá mostrar si el local tiene calefacción. |
| >>>RF.4.2.2.15 | Datos | Permitirá mostrar si el local tiene aire acondicionado. |
| >>>RF.4.2.2.16 | Datos | Permitirá mostrar el precio del local. |
| >>>RF.4.2.2.17 | Datos | Permitirá mostrar los honorarios del agente. |
| >>>>RF.4.2.2.17.1 | Restricciones | Los honorarios del agente sólo podrán ser vistos por una cuenta de tipo “agente” o “super-agente”. |
| >>>RF.4.2.2.18 | Datos | Permitirá mostrar el precio de gasto de comunidad del local. |
| >>RF.4.2.3 | Tipo de Inmuebles | El sistema permitirá mostrar la información de las propiedades de tipo “Suelo”. |
| >>>RF.4.2.3.1 | Datos | Permitirá mostrar su nombre. |
| >>>RF.4.2.3.2 | Datos | Permitirá mostrar su tipo. |
| >>>RF.4.2.3.3 | Datos | Permitirá mostrar su código identificativo. |
| >>>>RF.4.2.3.3.1 | Restricciones | El código identificativo sólo podrá ser visto por una cuenta de tipo “agente” o “super-agente”. |
| >>>RF.4.2.3.4 | Datos | Permitirá mostrar su descripción. |
| >>>RF.4.2.3.5 | Datos | Permitirá mostrar la superficie del suelo. |
| >>>RF.4.2.3.6 | Datos | Permitirá mostrar la situación geográfica del suelo. |
| >>>RF.4.2.3.7 | Datos | Permitirá mostrar el precio total del suelo. |
| >>>RF.4.2.3.8 | Datos | Permitirá mostrar los honorarios del agente. |
| >>>>RF.4.2.3.8.1 | Restricciones | Los honorarios del agente sólo podrán ser vistos por una cuenta de tipo “agente” o “super-agente”. |
| >>>RF.4.2.3.9 | Datos | Permitirá mostrar si el suelo tiene acceso a agua. |
| >>>RF.4.2.3.10 | Datos | Permitirá mostrar si el suelo tiene acceso a luz. |
| >>>RF.4.2.3.11 | Datos | Permitirá mostrar la superficie edificable del suelo. |
| >>>RF.4.2.3.12 | Datos | Permitirá mostrar la edificabilidad del suelo. |
| >>>RF.4.2.3.13 | Datos | Permitirá mostrar la calificación urbanística del suelo. |

##### Requisitos del Filtro de Propiedades

Aquí se declara toda la funcionalidad del filtro que permitirá al usuario obtener unos resultados más específicos en relación con las propiedades.

|  |  |  |
| --- | --- | --- |
| **FILTRO DE PROPIEDADES** | | |
| **Índice** | **Nombre** | **Descripción** |
| RF.5 | Filtrado de Propiedades | El sistema permitirá filtrar las diferentes propiedades por diferentes apartados. |
| >RF.5.1 | Datos | El sistema permitirá filtrar por nombre en las propiedades de tipo “Vivienda”, “Local” y “Suelo”. |
| >RF.5.2 | Datos | El sistema permitirá filtrar por código en las propiedades de tipo “Vivienda”, “Local” y “Suelo”. |
| >RF.5.3 | Datos | El sistema permitirá filtrar por superficie en las propiedades de tipo “Vivienda”, “Local” y “Suelo”. |
| >>RF.5.4.1 | Restricciones | Solo un agente o super-agente podrá filtrar por código. |
| >RF.5.5 | Datos | El sistema permitirá filtrar por precio en las propiedades de tipo “Vivienda”, “Local” y “Suelo”. |
| >RF.5.6 | Datos | El sistema permitirá filtrar por dirección en las propiedades de tipo “Vivienda” y “Local”. |
| >RF.5.7 | Datos | El sistema permitirá filtrar por población en las propiedades de tipo “Vivienda” y “Local”. |
| >RF.5.8 | Datos | El sistema permitirá filtrar si tiene calefacción o no en las propiedades de tipo “Vivienda” y “Local”. |
| >RF.5.9 | Datos | El sistema permitirá filtrar si tiene aire acondicionado o no en las propiedades de tipo “Vivienda” y “Local”. |
| >RF.5.10 | Datos | El sistema permitirá filtrar por número de habitaciones en las propiedades de tipo “Vivienda”. |
| >RF.5.11 | Datos | El sistema permitirá filtrar por número de baños en las propiedades de tipo “Vivienda”. |
| >RF.5.12 | Datos | El sistema permitirá filtrar si tiene garaje o no en las propiedades de tipo “Vivienda”. |
| >RF.5.13 | Datos | El sistema permitirá filtrar si tiene piscina o no en las propiedades de tipo “Vivienda”. |
| >RF.5.14 | Datos | El sistema permitirá filtrar si tiene terraza o no en las propiedades de tipo “Vivienda”. |
| >RF.5.15 | Datos | El sistema permitirá filtrar si tiene jardín o no en las propiedades de tipo “Vivienda”. |
| >RF.5.16 | Datos | El sistema permitirá filtrar si tiene ascensor o no en las propiedades de tipo “Vivienda”. |
| >RF.5.17 | Datos | El sistema permitirá filtrar si esta amueblado o no en las propiedades de tipo “Vivienda”. |
| >RF.5.18 | Datos | El sistema permitirá filtrar si permite animales o no en las propiedades de tipo “Vivienda”. |
| >RF.5.19 | Datos | El sistema permitirá filtrar si tiene trastero o no en las propiedades de tipo “Vivienda”. |
| >RF.5.20 | Datos | El sistema permitirá filtrar si posee escaparate o no en las propiedades de tipo “Local”. |
| >RF.5.21 | Datos | El sistema permitirá filtrar si posee aparcamiento o no en las propiedades de tipo “Local”. |
| >RF.5.22 | Datos | El sistema permitirá filtrar si posee zona de carga y descarga o no en las propiedades de tipo “Local”. |
| >RF.5.23 | Datos | El sistema permitirá filtrar si posee un sistema de extinción de incendios o no en las propiedades de tipo “Local”. |
| >RF.5.24 | Datos | El sistema permitirá filtrar por número de aseos en las propiedades de tipo “Local”. |
| >RF.5.25 | Datos | El sistema permitirá filtrar si posee iluminación o no en las propiedades de tipo “Local”. |
| >RF.5.26 | Datos | El sistema permitirá filtrar por situación en las propiedades de tipo “Suelo”. |
| >RF.5.27 | Datos | El sistema permitirá filtrar por acceso a agua en las propiedades de tipo “Suelo”. |
| >RF.5.28 | Datos | El sistema permitirá filtrar por acceso a luz en las propiedades de tipo “Suelo”. |
| >RF.5.29 | Datos | El sistema permitirá filtrar por superficie edificable en las propiedades de tipo “Suelo”. |

##### Requisitos del sistema de Seguimiento

Este apartado detallará los requisitos que debe cumplir el sistema de seguimientos de los usuarios.

|  |  |  |
| --- | --- | --- |
| **GESTIÓN DE SEGUIMIENTOS** | | |
| **Índice** | **Nombre** | **Descripción** |
| RF.6 | Seguimientos | Un cliente de tipo usuario podrá guardar las propiedades en las que esté interesado. |
| >RF.6.1 | Notificación | El sistema notificará al usuario cuando un inmueble de sus seguimientos baje de precio. |

##### Requisitos del sistema de Comunicación entre Agente y Cliente

Este sistema declarará las restricciones del sistema de mensajería y otros canales de comunicación que se facilitarán a los usuarios. De esta manera podrán comunicarse con el agente de una manera sencilla.

|  |  |  |
| --- | --- | --- |
| **GESTIÓN DE CANALES DE COMUNICACIÓN** | | |
| **Índice** | **Nombre** | **Descripción** |
| RF.7 | Comunicación Agente-Cliente | El sistema permitirá al usuario establecer un canal de comunicación con el agente. |
| >RF.7.1 | Canal de Comunicación | El sistema implementara un sistema de mensajería dentro de la aplicación. |
| >>RF.7.1.1 | Restricciones | El sistema permitirá al usuario enviar mensajes al agente y viceversa. |
| >>RF.7.1.2 | Restricciones | El sistema permitirá al usuario recibir mensajes del agente y viceversa. |
| >RF.7.2 | Canal de Comunicación | El sistema ofrecerá un correo electrónico para que el usuario pueda comunicarse con el agente. |
| >RF.7.3 | Canal de Comunicación | El sistema ofrecerá número de teléfono para que el usuario pueda establecer un canal de comunicación directo con el agente. |

##### Requisitos del sistema de Geolocalización

Este apartado describirá las restricciones que deberá cumplir el sistema en cuanto a lo que la geolocalización del inmueble se refiere.

|  |  |  |
| --- | --- | --- |
| **GESTIÓN DE GEOLOCALIZACIÓN** | | |
| **Índice** | **Nombre** | **Descripción** |
| RF.8 | Sistema de Geolocalización | El sistema permitirá al agente añadir las coordenadas representando la localización geográfica del inmueble. |
| >RF.8.1 | Restricciones | Las coordenadas se obtendrán automáticamente aportando la dirección. |
| >RF8.2 | Error | En el caso de no poder cargar la localización, el sistema mostrará un mensaje de error. |
| >RF.8.3 | Sistema de Geolocalización | El sistema mostrará al usuario la localización del inmueble en un mapa dinámico junto a la información de la propiedad. |

#### Requisitos No Funcionales

|  |  |  |
| --- | --- | --- |
| **REQUISITOS NO FUNCIONALES** | | |
| **Índice** | **Nombre** | **Descripción** |
| RNF.1 | Usabilidad | El tiempo de aprendizaje del sistema debe ser menor de 3 horas. |
| RNF.2 | Usabilidad | El sistema debe contar con un manual de usuario bien estructurado. |
| RNF.3 | Usabilidad | El sistema debe mostrar mensajes de error que sean informativos y fácilmente comprensibles. |
| RNF.4 | Seguridad | Todas las contraseñas deberán encontrarse encriptadas con el algoritmo RSA. |
| RNF.5 | Requisitos | El sistema será desarrollado para PC. |
| RNF.6 | Desarrollo | La interfaz de usuario será implementada para navegadores web con HTML5 y JavaScript |
| RNF.7 | Desarrollo | El sistema será desarrollado utilizando la herramienta Web Storm (Versión de 2020.3.2) |
| RNF.8 | Desarrollo | El sistema utilizará la base de datos MongoDB (Versión 4) |
| RNF.9 | Desarrollo | El sistema será desarrollado utilizando el entorno NodeJS (Versión 15.0.1) |
| RNF.10 | Desarrollo | El sistema será desarrollado utilizando el framework Bootstrap (Versión 4) |
| RNF.11 | Desarrollo | El sistema será desarrollado utilizando la biblioteca JQuery (Versión 3.2.0) |
| RNF.12 | Pruebas | El sistema será probado utilizando Selenium (Versión 3.14) |

### Identificación de Actores del Sistema

Se analizarán los distintos actores que intervendrán en cualquiera de los procesos del sistema. Identificaremos como actor a todo elemento que interactúa con el sistema desde fuera, o que es algo o alguien sobre el cual interactúa el sistema.

#### 6.2.2.1 Perfiles

Los perfiles podrán ser de diferentes tipos. Un perfil es cualquier actor físico que, identificado o no, interactúe con el sistema. Para ver con más detalle ver Figura 1.

##### Usuario No Registrado

Un usuario no registrado podrá acceder a la aplicación para observar los diferentes inmuebles, así como acceder a su pantalla descriptiva. Ese será el único movimiento que se le permitirá hacer (obviando el de registrarse y el iniciar sesión).

##### Usuario Registrado

Un usuario registrado tendrá permitido realizar cualquier operación relacionada con la edición y eliminación de su propio perfil, así como ver las diferentes propiedades ofertadas. También podrá visualizar la información de su propia cuenta. También podrá enviar y recibir mensajes.

##### Agente Registrado

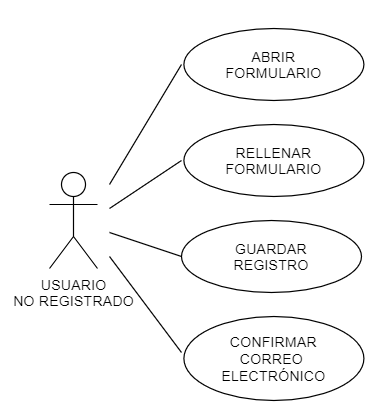
Un agente registrado podrá realizar las operaciones que se mencionaron en el apartado anterior, así como crear, modificar y eliminar cualquier perfil de usuario que desee, crear, modificar y eliminar propiedades y ver las estadísticas de ventas de su perfil.

##### Super-agente Registrado

Un super-agente registrado podrá realizar las operaciones que se mencionaron en el apartado anterior, así como crear, modificar y eliminar cualquier perfil de usuario o agente que desee, crear, modificar y eliminar propiedades y ver las estadísticas de ventas de su perfil.

### Especificación de Casos de Uso

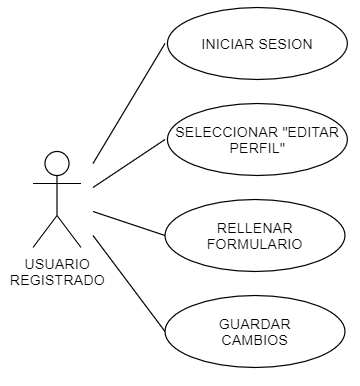
#### Caso N.1 Crear Perfil Usuario



*Figura 13: Esquema de Caso de Uso – Crear Perfil Usuario*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Crear Perfil Usuario |
| **Descripción** | |
| Un usuario no registrado creará su cuenta en el sistema. | |

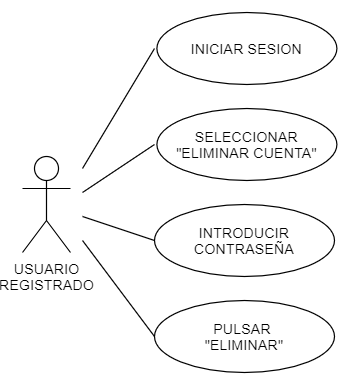
#### Caso N.2 Editar Perfil Usuario



*Figura 14: Esquema de Caso de Uso – Editar Perfil Usuario*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Editar Perfil Usuario |
| **Descripción** | |
| Un usuario con una cuenta desea editar su información personal. | |

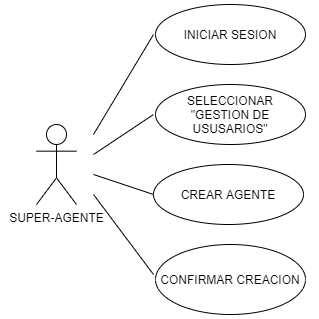
#### Caso N.3 Eliminar Perfil Usuario



*Figura 15: Esquema de Caso de Uso – Eliminar Perfil Usuario*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Eliminar Perfil Usuario |
| **Descripción** | |
| Un usuario con una cuenta desea eliminar su información de la cuenta. | |

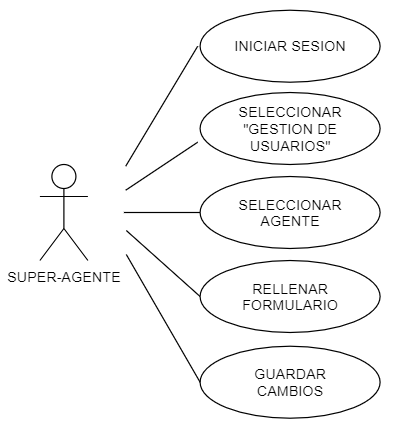
#### Caso N.4 Crear Perfil Agente



*Figura 16: Esquema de Caso de Uso – Crear Perfil Agente*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Crear Perfil Agente |
| **Descripción** | |
| El super-agente necesita crear una cuenta de tipo agente. | |

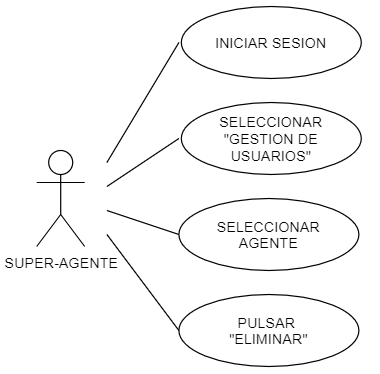
#### Caso N.5 Editar Perfil Agente



*Figura 17: Esquema de Caso de Uso – Editar Perfil Agente*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Editar Perfil Agente |
| **Descripción** | |
| El super-agente necesita modificar los datos de una cuenta de tipo agente. | |

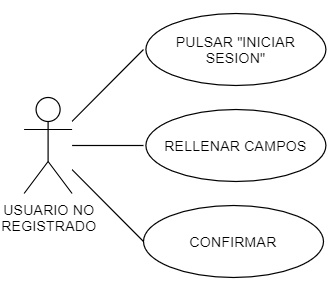
#### Caso N.6 Eliminar Perfil Agente



*Figura 18: Esquema de Caso de Uso – Eliminar Perfil Agente*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Eliminar Perfil Agente |
| **Descripción** | |
| El super-agente necesita eliminar una cuenta de tipo agente del sistema. | |

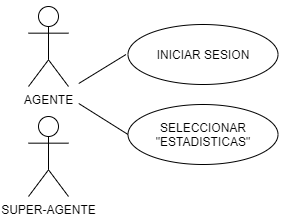
#### Caso N.7 Inicio de Sesión



*Figura 19: Esquema de Caso de Uso – Inicio de Sesión*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Inicio de Sesión |
| **Descripción** | |
| Un usuario necesita iniciar sesión en el sistema para acceder a su información personal. | |

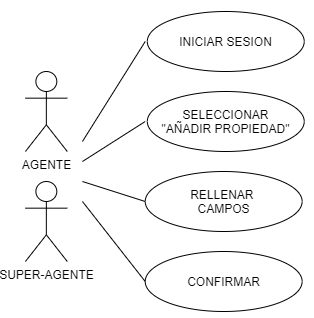
#### Caso N.8 Ver Estadísticas de Ventas



*Figura 20: Esquema de Caso de Uso – Ver Estadísticas de Ventas*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Ver Estadísticas de Ventas |
| **Descripción** | |
| Un agente o super-agente desea ver las estadísticas de ventas de sus propiedades. | |

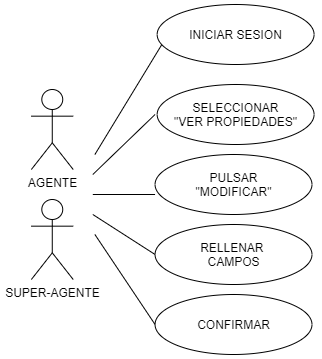
#### Caso N.9 Crear Propiedad



*Figura 21: Esquema de Caso de Uso – Crear Propiedad*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Crear Propiedad |
| **Descripción** | |
| Un agente o super-agente desea crear y publicar una propiedad en la aplicación. | |

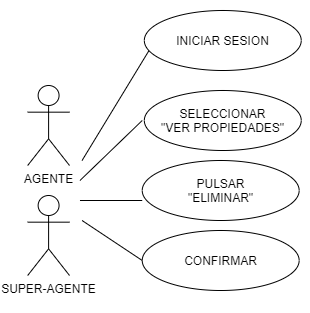
#### Caso N.10 Editar Propiedad



*Figura 22: Esquema de Caso de Uso – Editar Propiedad*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Editar Propiedad |
| **Descripción** | |
| Un agente o super-agente desea editar una propiedad ya publicada en la aplicación. | |

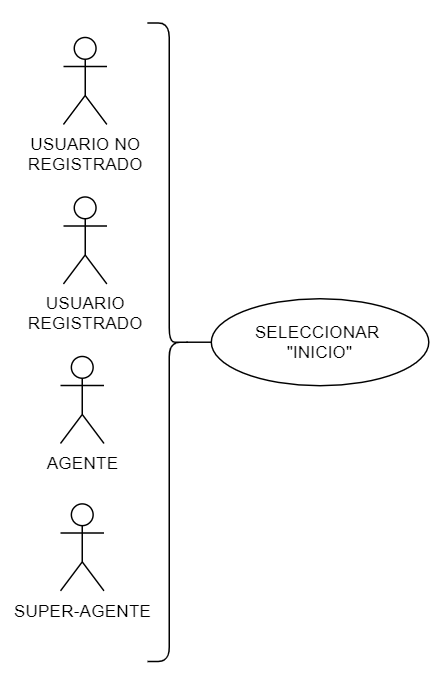
#### Caso N.11 Eliminar Propiedad



*Figura 23: Esquema de Caso de Uso – Eliminar Propiedad*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Eliminar Propiedad |
| **Descripción** | |
| Un agente o super-agente desea eliminar una propiedad ya publicada en la aplicación. | |

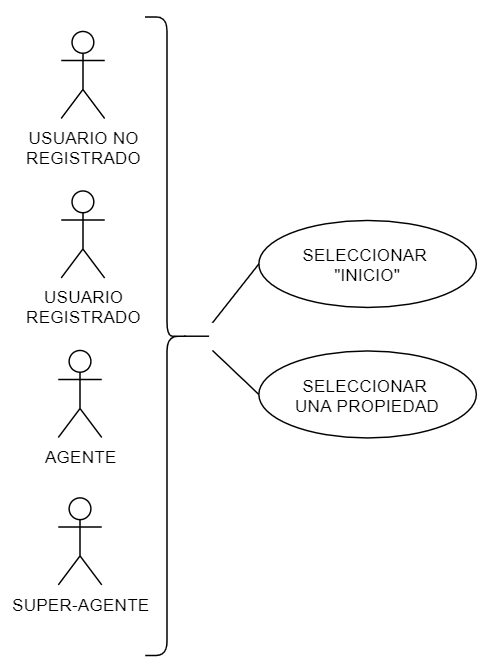
#### Caso N.12 Ver Propiedades



*Figura 24: Esquema de Caso de Uso – Ver Propiedades*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Ver Propiedades |
| **Descripción** | |
| Cualquier tipo de cuenta desea ver las propiedades ofertadas en el sistema. | |

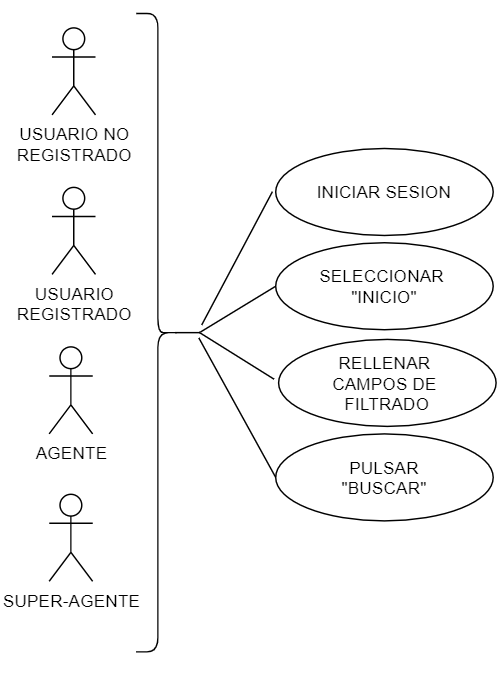
#### Caso N.13 Ver detalles de Propiedades



*Figura 25: Esquema de Caso de Uso – Ver detalles de Propiedad*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Ver detalles de Propiedades |
| **Descripción** | |
| Cualquier tipo de cuenta desea ver las propiedades específicas de una de las propiedades ofertadas en el sistema. | |

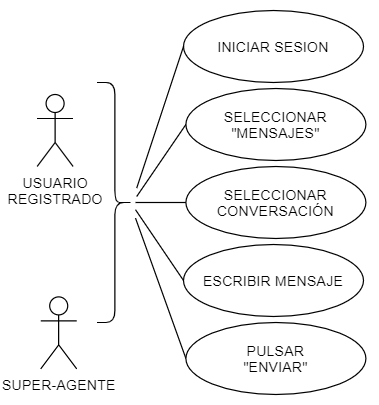
#### Caso N.14 Filtrar Propiedades



*Figura 26: Esquema de Caso de Uso – Filtrar Propiedades*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Filtrar Propiedades |
| **Descripción** | |
| Cualquier tipo de cuenta desea filtrar las propiedades existentes en el sistema para así obtener una búsqueda más acorde con sus necesidades. | |

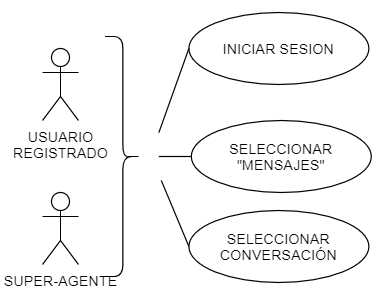
#### Caso N.15 Enviar Mensaje



*Figura 27: Esquema de Caso de Uso – Enviar Mensaje*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Enviar Mensaje |
| **Descripción** | |
| Tanto un usuario como el super-agente desean enviar un mensaje mediante el sistema de mensajería de la aplicación. | |

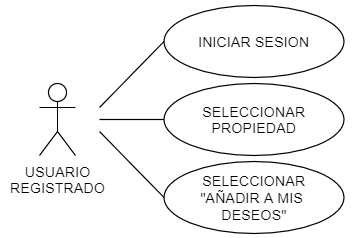
#### Caso N.16 Ver Mensajes Recibidos



*Figura 28: Esquema de Caso de Uso – Ver Mensajes Recibidos*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Ver Mensajes Recibidos |
| **Descripción** | |
| Tanto un usuario como un agente desean recibir un mensaje mediante el sistema de mensajería de la aplicación. | |

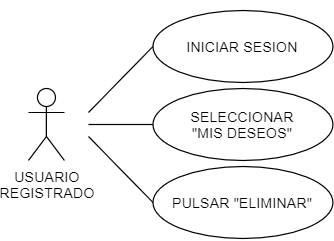
#### Caso N.17 Añadir a Seguimientos



*Figura 29: Esquema de Caso de Uso – Añadir Seguimientos*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Añadir a seguimientos |
| **Descripción** | |
| Un usuario registrado desea añadir una propiedad en la que está interesado a sus seguimientos. | |

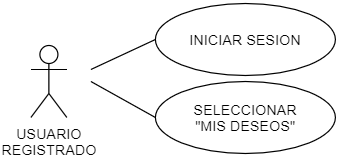
#### Caso N.18 Eliminar de Seguimientos



*Figura 30: Esquema de Caso de Uso – Eliminar de Seguimientos*

|  |
| --- |
| **Nombre Del Caso de Uso** |
| Eliminar de Seguimientos |
| **Descripción** | |
| Un usuario registrado desea eliminar una propiedad en la que ya no esté interesado de Seguimientos. | |

#### Caso N.19 Ver Seguimientos



*Figura 31: Esquema de Caso de Uso – Ver Seguimientos*

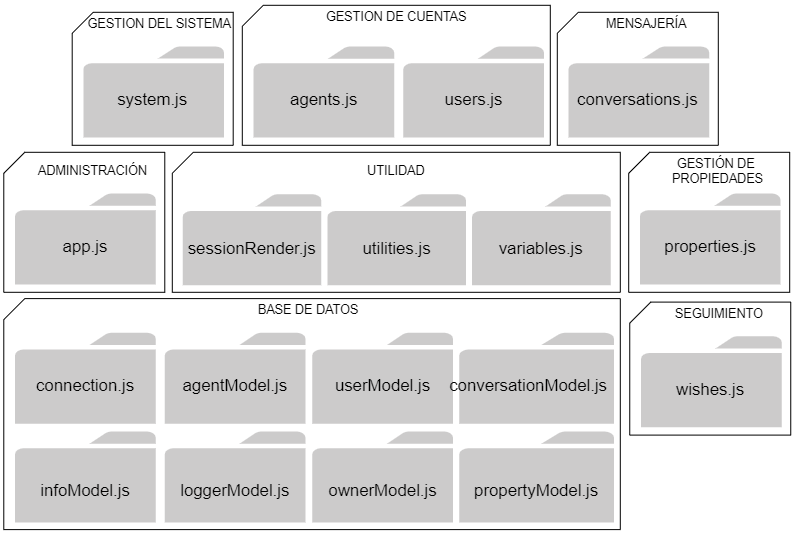
|  |
| --- |
| **Nombre Del Caso de Uso** |
| Ver Seguimientos |
| **Descripción** | |
| Un usuario registrado desea ver todas las propiedades que tiene añadidas a Seguimientos. | |

## Identificación de los Subsistemas de Análisis

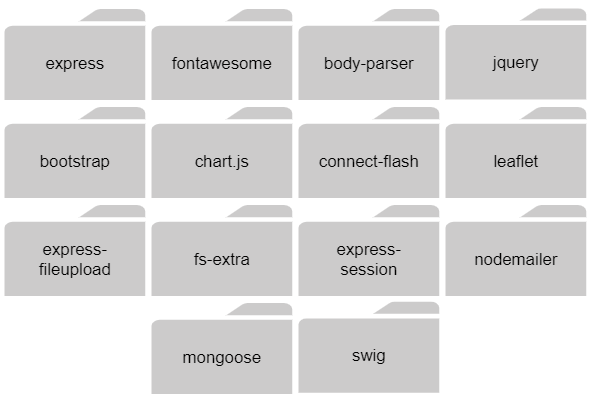
### Descripción de los Subsistemas

En primer lugar, debemos tener en cuenta que el sistema está siendo desarrollado siguiendo la organización en clusters con rutas, por lo que no se está siguiendo el tradicional paradigma orientado a objetos, sino que se está siguiendo una arquitectura modular, agrupando las rutas relacionadas en módulos. Esta apreciación será importante de cara al análisis de los sistemas, pues debemos tener en cuenta que ya no trabajamos con diagramas y estructuras de clases sino de módulos. Estos serían los subsistemas que compondrán la integridad del proyecto de desarrollo:

* + **Subsistema de Administración:** Dentro de este subsistema se encuentra el módulo relacionado con la configuración y despliegue del sistema. Es, por así decirlo, el núcleo encargado de indicarle a cada módulo con qué paquetes trabajará. Está compuesto por un único modulo, **app.js.** Este módulo es el único objeto creado en el sistema y es creado por el módulo express, del que hablaremos en futuros subsistemas. Se encarga de preparar la aplicación para ser lanzada y de implementar todos los paquetes necesarios para su correcto funcionamiento y asignarlos a los demás módulos.
* **Subsistema de Utilidad**: Es el encargado de proporcionar al resto de módulos las funciones, interfaces y variables globales que éste requiera.
  + **sessionRender.js:** El módulo encargado de cargar las diferentes vistas (o interfaces del sistema).
  + **utilities.js:** Este módulo se encarga de contener todas las funciones de ayuda para el resto de los módulos.
  + **variables.js:** Contiene todas las variables globales en un único modulo, para así ser modificadas con más facilidad.
* **Subsistema de Gestión de Usuarios:** Este subsistema se encarga de gestionar todo lo necesario para la gestión y trabajo con las cuentas de cualquier tipo. Se compone de los siguientes módulos:
  + **agents.js:** Este módulo se encarga de gestionar todo lo relativo a la creación, edición y eliminación de cuentas de tipo agente por parte del super-agente.
  + **users.js:** Se ocupa de lo relativo a las cuentas de usuario, las cuales pueden ser creadas por cualquier usuario que lo desee.
* **Subsistema de Base de Datos:** Encargado de hacer de intermediario entre la aplicación y la Base de Datos. Está compuesto por diversos módulos, que actúan como modelos intermediarios a la Base de Datos:
  + **connection.js:** Establece la conexión con la base de datos.
  + **agentModel:** Este modelo nos permite trabajar con las cuentas de tipo agente o super-agente en la Base de Datos.
  + **userModel:** Nos permite trabajar con las cuentas de tipo usuario en la Base de Datos.
  + **infoModel:** Nos permite trabajar con los documentos de información del sistema (datos de contacto, por ejemplo).
  + **conversationModel:** Nos permite trabajar con las conversaciones entre los agentes y los usuarios (chat).
  + **loggerModel:** Con el podemos trabajar en Base de Datos con la información estadística que se guarda.
  + **ownerModel:** Desde este módulo podemos tratar con los propietarios de las diferentes propiedades disponibles.
  + **propertyModel:** Con el podemos trabajar con las propiedades que el usuario desee publicar, editar o eliminar.
* **Subsistema de mensajería:** Se ocupa de todo lo relativo al canal de mensajería integrado en la aplicación, el cual se establece entre un agente y un usuario registrado interesado en una propiedad. Se encuentra compuesto por un único modulo, **conversations.js.**
* **Subsistema de seguimiento:** Se ocupa de toda la lógica referente a las listas de seguimiento de los usuarios. Al igual que en el caso anterior, se compone de un único modulo llamado **wishes.js.**
* **Subsistema de Gestión del Sistema:** Se encarga de permitir al administrador realizar diferentes acciones, como modificar la a información de contacto y obtener estadísticas. Está compuesto por un único modulo, **system.js.**
* **Subsistema de Gestión de Propiedades:** Se encarga de todas las tareas relacionadas con las propiedades. Está compuesto por un único módulo, **properties.js.**
* **Paquetes:** Cada uno de los paquetes de funcionalidad que hemos ido añadiendo al sistema representaría un subsistema. Estos paquetes son:
  + **express:** Es un framework que nos permite un desarrollo mucho más dinámico y sencillo de la aplicación.
  + **mongoose:** Es un paquete que nos permite crear los modelos indicados en el subsistema de Base de Datos, para así utilizarlos de una manera sencilla y segura.
  + **fontawesome:** Permite obtener diferentes imágenes en formato icono.
  + **body-parser:** Nos permite transferir información entre las interfaces y los módulos de una forma cómoda y segura.
  + **bootstrap**: Un framework que nos facilita el proceso de crear interfaces responsive (adaptables según el tamaño de pantalla).
  + **chart.js:** Este paquete nos permite diseñar una página de estadísticas con diferentes elementos como gráficos de línea y de donut.
  + **connect-flash**: Nos permite guardar en una memoria flash una serie de mensajes de error y, al mostrarlos, borrarlos automáticamente sin necesidad de limpiarla cada llamada.
  + **express-fileupload:** Nos permite recoger las imágenes de las propiedades que han sido seleccionadas por el usuario y añadirlas a una carpeta.
  + **fs-extra:** Complementario al anterior, este paquete nos permite crear un directorio con un id identificativo de cada propiedad. De la misma manera, se encarga de eliminar el directorio y su contenido si eliminamos la propiedad, y de actualizarlo si modificamos sus imágenes.
  + **express-session:** Nos permite trabajar con un objeto sesión, el cual se carga al iniciar sesión un usuario.
  + **jquery:** Nos permite trabajar con código dentro de las interfaces, para así realizar tareas como mostrar un error si ciertos campos no existen y otras funcionalidades.
  + **leaflet:** Muestra el mapa dinámico con la geolocalización del inmueble en cuestión.
  + **nodemailer:** Se encarga de crear y enviar el mensaje de correo electrónico con diferentes funcionalidades, como cambiar la contraseña o verificar el correo electrónico, entre otros.
  + **swig:** Permite trabajar con los objetos de sesión desde dentro de las interfaces, para así, por ejemplo, mostrar u ocultar campos dependiendo del usuario que este en sesión.



*Figura 32: Subsistemas – Visión Genérica, Parte 1*

**

*Figura 33: Subsistemas – Visión Genérica, Parte 2*

### Descripción de los Interfaces entre Subsistemas

Respecto a las interfaces entre los subsistemas, debemos decir que la inmensa mayoría de estos se comunicaran localmente dentro de la máquina.

No obstante, la conexión a la Base de Datos remota debe ser considerada como una interfaz de importancia. Esta conexión sigue los protocolos TCP/IP para conectarse a ella, por lo que tendríamos otra interfaz que considerar.

Por otra parte, estos subsistemas se comunicarán mediante la web con algunos de los paquetes que instalamos, pues requieren de una conexión a internet para cumplir con sus funciones correctamente. Estos paquetes son:

* **leaflet:** Requiere de una conexión para cargar el mapa con la dirección entregada.
* **nodemailer:** Para enviar el correo electrónico correctamente.

Por ello, podremos decir que la conexión a estos dos paquetes trabajará con una API externa, la cual será interpretada como una interfaz desde el punto de vista de diseño.

## Análisis de los Casos de Uso

#### Nomenclatura de los Casos de Uso

Para describir los diferentes casos de uso, se utilizarán los siguientes elementos explicativos:

* **Descripción de alto nivel:** A modo de resumen, explicará en un par de líneas cual es el caso de uso.
* **Requisitos:** Referido a la anterior sección *Requisitos del Sistema,* una lista de qué requisitos se ven involucrados en el caso de uso.
* **Actores:** Una lista de qué actores intervienen en el proceso de ejecución del caso de uso.
* **Precondiciones:** Que condiciones deben cumplirse para que ese caso de uso pueda llevarse a cabo
* **Postcondiciones:** Que condiciones aparecen o se modifican al final de la ejecución del caso de uso.
* **Flujo de eventos:** Una lista de eventos o acciones definidos en orden temporal explicando las diferentes fases por las que pasa la ejecución correcta del caso de uso.
* **Rutas alternativas:** Una serie de movimientos que pueden causar resultados alternativos derivados del flujo de eventos.

#### Caso N.1 Crear Perfil Usuario

|  |  |
| --- | --- |
| **CREAR PERFIL USUARIO** | |
| **Descripción de alto nivel** |  |
|  | Un usuario no registrado desea registrarse en la aplicación. |
| **Requisitos** |  |
|  | RF.1, RF.1.1 |
| **Actores** |  |
|  | Usuario no Registrado |
| **Precondiciones** |  |
|  | * El usuario debe haber seleccionado la opción “Registrarse”, de la barra de navegación. |
| **Postcondiciones** |  |
|  | * Se registra el usuario en la aplicación |
| **Flujo de eventos** |  |
|  | 1. El usuario selecciona la opción “Registrarse”. 2. El usuario rellena los campos con sus datos. 3. El usuario selecciona el botón de ”Registrarme”. 4. El sistema envía un correo electrónico de verificación al usuario. 5. El usuario accede a su cuenta de correo electrónico y accede al link en su interior. 6. El sistema redirige al usuario a la página de inicio de sesión. La cuenta se ha creado correctamente. |
| **Rutas alternativas** | * El usuario cancela el registro. * El usuario no valida su correo electrónico. * El usuario introduce un correo electrónico ya registrado (Se mostrará un mensaje de error). |

#### Caso N.2 Editar Perfil Usuario

|  |  |
| --- | --- |
| **EDITAR PERFIL USUARIO** | |
| **Descripción de alto nivel** |  |
|  | Un usuario no registrado desea modificar la información de su perfil. |
| **Requisitos** |  |
|  | RF.1, RF.1.2, RF.1.2.3.2 |
| **Actores** |  |
|  | Usuario Registrado |
| **Precondiciones** |  |
|  | * El usuario debe haber seleccionado la opción “Editar Perfil”, de la barra de navegación. |
| **Postcondiciones** |  |
|  | * Se modifican los datos de la aplicación |
| **Flujo de eventos** |  |
|  | 1. EL usuario inicia sesión en la aplicación. 2. El usuario hace click en su correo electrónico en la barra de navegación. 3. El usuario hace click en la opción “Editar Perfil”, de la barra de navegación. 4. EL usuario modifica los campos que desee editar. 5. El usuario presiona el botón de “Modificar Perfil” |
| **Rutas alternativas** | * El usuario cancela el cambio. * El usuario introduce campos erróneos o vacíos. |

#### Caso N.3 Eliminar Perfil Usuario

|  |  |
| --- | --- |
| **ELIMINAR PERFIL USUARIO** | |
| **Descripción de alto nivel** |  |
|  | Un usuario registrado decide eliminar su cuenta del sistema. |
| **Requisitos** |  |
|  | RF.1.3, RF.1.3.1 |
| **Actores** |  |
|  | Usuario Registrado |
| **Precondiciones** |  |
|  | * El usuario deberá estar registrado en el sistema. * EL usuario deberá haber iniciado sesión en el sistema. |
| **Postcondiciones** |  |
|  | * El sistema eliminará por completo sus datos. |
| **Flujo de eventos** |  |
|  | 1. El usuario hace click en su correo electrónico mostrado en la barra de navegación. 2. El usuario selecciona la opción de “Eliminar Cuenta” 3. El sistema muestra una ventana. 4. El usuario introducirá su contraseña en el campo. 5. El usuario pulsará la opción “Eliminar Cuenta”. |
| **Rutas alternativas** | * El usuario pulsa el botón “Cancelar”, cancelando la operación. * El usuario introduce una contraseña incorrecta. * Una cuenta de agente decide eliminar la cuenta de usuario.  1. El agente selecciona la opción de gestión de usuarios. 2. El agente elimina la cuenta del usuario. |

#### Caso N.4 Crear Perfil Agente

|  |  |
| --- | --- |
| **CREAR PERFIL AGENTE** | |
| **Descripción de alto nivel** |  |
|  | Un agente desea crear una cuenta de su tipo. |
| **Requisitos** |  |
|  | RF.1, RF.1.1, RF.1.1.1 |
| **Actores** |  |
|  | Agente |
| **Precondiciones** |  |
|  | * Un agente debe haber iniciado sesión. * El agente debe crear el perfil manualmente. |
| **Postcondiciones** |  |
|  | * Se registra el nuevo agente en la aplicación. |
| **Flujo de eventos** |  |
|  | 1. El agente hará click en “Perfiles”, en la barra de navegación. 2. Hará click en la opción “Gestión de Perfiles”, en la barra de navegación. 3. El agente hará click en “Añadir Agente”. 4. El agente rellenará los campos con los datos. 5. El agente pulsará en el botón “Crear Agente”. 6. El perfil de tipo Agente se creará |
| **Rutas alternativas** | * El agente cancela la creación del perfil. * El agente rellena los datos incorrectamente. |

#### Caso N.5 Editar Perfil Agente

|  |  |
| --- | --- |
| **EDITAR PERFIL AGENTE** | |
| **Descripción de alto nivel** |  |
|  | Un usuario no registrado desea modificar la información de su perfil. |
| **Requisitos** |  |
|  | RF.1, RF.1.2, RF.1.2.3.2 |
| **Actores** |  |
|  | Agente |
| **Precondiciones** |  |
|  | * El agente debe haber seleccionado la opción “Editar Perfil”, de la barra de navegación. |
| **Postcondiciones** |  |
|  | * Se modifican los datos de la aplicación |
| **Flujo de eventos** |  |
|  | 1. EL agente inicia sesión en la aplicación. 2. El agente hace click en su correo electrónico en la barra de navegación. 3. El agente hace click en la opción “Editar Perfil”, de la barra de navegación. 4. EL agente modifica los campos que desee editar. 5. El agente presiona el botón de “Modificar Perfil” |
| **Rutas alternativas** | * El agente cancela el cambio. * El agente introduce campos erróneos o vacíos. |

#### Caso N.6 Eliminar Perfil Agente

|  |  |
| --- | --- |
| **ELIMINAR PERFIL AGENTE** | |
| **Descripción de alto nivel** |  |
|  | Un agente registrado decide eliminar su cuenta del sistema. |
| **Requisitos** |  |
|  | RF.1.3, RF.1.3.2 |
| **Actores** |  |
|  | Agente |
| **Precondiciones** |  |
|  | * El agente deberá estar registrado en el sistema. * EL agente deberá haber iniciado sesión en el sistema. |
| **Postcondiciones** |  |
|  | * El sistema eliminará por completo sus datos. |
| **Flujo de eventos** |  |
|  | 1. El agente hace click en su correo electrónico mostrado en la barra de navegación. 2. El agente selecciona la opción de “Eliminar Cuenta” 3. El sistema muestra una ventana. 4. El agente introducirá su contraseña en el campo. 5. El agente pulsará la opción “Eliminar Cuenta”. |
| **Rutas alternativas** | * El agente pulsa el botón “Cancelar”, cancelando la operación. * El agente introduce una contraseña incorrecta. * Su perfil es el único de tipo agente existente en el sistema. |

#### Caso N.7 Inicio de Sesión

|  |  |
| --- | --- |
| **INICIO DE SESIÓN** | |
| **Descripción de alto nivel** |  |
|  | Un usuario que no ha iniciado sesión decide acceder a su cuenta. |
| **Requisitos** |  |
|  | RF.1.4, RF.1.5 |
| **Actores** |  |
|  | Usuario no Registrado |
| **Precondiciones** |  |
|  | * El usuario debe estar registrado en el sistema |
| **Postcondiciones** |  |
|  | * El sistema guarda al usuario en la sesión. * El usuario entra en su cuenta, |
| **Flujo de eventos** |  |
|  | 1. El usuario no registrado selecciona la opción “Iniciar Sesión”. 2. El usuario rellena los campos con su correo electrónico y su contraseña. 3. El usuario selecciona el botón de ”Iniciar Sesión”. 4. En el caso de que los datos sean correctos, el usuario Accede a su cuenta personal en el sistema. 5. El usuario accede a su cuenta dentro del sistema. |
| **Rutas alternativas** | * El usuario cancela el inicio de sesión. * Los datos no son correctos (Se mostrará un mensaje de error). * El usuario ha olvidado su contraseña   1. El usuario selecciona la opción se “¿Has olvidado tu contraseña?”   2. El sistema envía un correo electrónico con un enlace.   3. El usuario abre el enlace recibido.   4. EL usuario introduce su nueva contraseña.   5. El sistema envía al usuario de nuevo a la pantalla de inicio de sesión. |

#### Caso N.8 Ver Estadísticas de Ventas

|  |  |
| --- | --- |
| **VER ESTADÍSTICAS DE VENTAS** | |
| **Descripción de alto nivel** |  |
|  | Un agente registrado quiere obtener información estadística de la empresa. |
| **Requisitos** |  |
|  | RF.1.6.1 |
| **Actores** |  |
|  | Agente |
| **Precondiciones** |  |
|  | * El agente deberá estar registrado en el sistema. * EL agente deberá haber iniciado sesión en el sistema. |
| **Postcondiciones** |  |
|  | * El sistema mostrará la información estadística. |
| **Flujo de eventos** |  |
|  | 1. El agente hará click en la opción “Estadísticas”, en la barra de navegación. 2. El sistema mostrará una pantalla de estadísticas. |
| **Rutas alternativas** |  |

#### Caso N.9 Crear Propiedad

|  |  |
| --- | --- |
| **CREAR PROPIEDAD** | |
| **Descripción de alto nivel** |  |
|  | Un agente desea crear una propiedad para ofertar. |
| **Requisitos** |  |
|  | RF.2, RF.2.1 |
| **Actores** |  |
|  | Agente |
| **Precondiciones** |  |
|  | * La cuenta deberá ser de tipo Agente. |
| **Postcondiciones** |  |
|  | * Se añade la nueva propiedad. |
| **Flujo de eventos** |  |
|  | 1. El agente selecciona la opción “Propiedades”, barra de navegación. 2. El agente seleccionará la opción “Añadir Propiedad”, en la barra de navegación. 3. El agente añadirá los campos que desee. 4. El agente pulsará el botón “Agregar”. |
| **Rutas alternativas** | * El agente cancela la operación. * El agente introduce campos erróneos (Mostrará un error). |

#### Caso N.10 Editar Propiedad

|  |  |
| --- | --- |
| **EDITAR PROPIEDAD** | |
| **Descripción de alto nivel** |  |
|  | Un agente desea editar una propiedad para ofertar. |
| **Requisitos** |  |
|  | RF.2, RF.2.2 |
| **Actores** |  |
|  | Agente |
| **Precondiciones** |  |
|  | * La cuenta deberá ser de tipo Agente. |
| **Postcondiciones** |  |
|  | * Se modifica la nueva propiedad. |
| **Flujo de eventos** |  |
|  | 1. El agente selecciona la opción “Propiedades”, barra de navegación. 2. El agente seleccionará la opción “Ver Propiedades”, en la barra de navegación. 3. El agente selecciona el botón “Modificar”, junto a la propiedad que desee modificar. 4. El agente editará los campos que desee. 5. El agente pulsará el botón “Modificar”. |
| **Rutas alternativas** | * El agente cancela la operación. * El agente introduce campos erróneos (Mostrará un error). |

#### Caso N.11 Eliminar Propiedad

|  |  |
| --- | --- |
| **EDITAR PROPIEDAD** | |
| **Descripción de alto nivel** |  |
|  | Un agente desea editar una propiedad para ofertar. |
| **Requisitos** |  |
|  | RF.2, RF.2.3 |
| **Actores** |  |
|  | Agente |
| **Precondiciones** |  |
|  | * La cuenta deberá ser de tipo Agente. |
| **Postcondiciones** |  |
|  | * Se elimina la nueva propiedad. |
| **Flujo de eventos** |  |
|  | 1. El agente selecciona la opción “Propiedades”, barra de navegación. 2. El agente seleccionará la opción “Ver Propiedades”, en la barra de navegación. 3. El agente selecciona el botón “Eliminar”, junto a la propiedad que desee modificar. 4. El sistema eliminará la propiedad. |
| **Rutas alternativas** | * El agente cancela la operación. |

#### Caso N.12 Ver Propiedades

|  |  |
| --- | --- |
| **VER PROPIEDADES** | |
| **Descripción de alto nivel** |  |
|  | Un usuario (tanto registrado como no) o un agente desea ver las propiedades ofertadas en la aplicación |
| **Requisitos** |  |
|  | RF.4.2 |
| **Actores** |  |
|  | Usuario Registrado, Usuario No Registrado, Agente |
| **Precondiciones** |  |
|  | * La cuenta deberá haber seleccionado la opción “Inicio”. |
| **Postcondiciones** |  |
|  | * Se muestra una lista de propiedades ofertadas en la aplicación. |
| **Flujo de eventos** |  |
|  | 1. La cuenta selecciona la opción de “Inicio” en la barra de navegación. 2. El sistema muestra una lista de inmuebles ofertados. |
| **Rutas alternativas** | 1. Al iniciar la aplicación se muestra la lista de propiedades por defecto. |

#### Caso N.13 Ver detalles de Propiedades

|  |  |
| --- | --- |
| **VER DETALLES DE PROPIEDADES** | |
| **Descripción de alto nivel** |  |
|  | Cualquier perfil desea ver los detalles de una propiedad. |
| **Requisitos** |  |
|  | RF.4, RF.4.2 |
| **Actores** |  |
|  | Usuario Registrado, Agente, Usuario no Registrado. |
| **Precondiciones** |  |
|  | * La cuenta deberá haber seleccionado al menos la opción de “Inicio” en la barra de navegación. |
| **Postcondiciones** |  |
|  | * El sistema mostrará los detalles de la propiedad. |
| **Flujo de eventos** |  |
|  | 1. El usuario hace click en la propiedad mostrada en la sección de “Inicio”. 2. El sistema mostrará los detalles concretos de la propiedad. |
| **Rutas alternativas** |  |

#### Caso N.14 Filtrar Propiedades

|  |  |
| --- | --- |
| **FILTRAR PROPIEDADES** | |
| **Descripción de alto nivel** |  |
|  | Un usuario (tanto registrado como no) o un agente desea filtrar las propiedades ofertadas en la aplicación |
| **Requisitos** |  |
|  | RF.5 |
| **Actores** |  |
|  | Usuario Registrado, Usuario No Registrado, Agente |
| **Precondiciones** |  |
|  | * La cuenta deberá haber seleccionado la opción “Inicio”. * La cuenta deberá haber rellenado, al menos, uno de los campos del filtro. |
| **Postcondiciones** |  |
|  | * Se muestra una lista de todas aquellas propiedades ofertadas en la aplicación que cumplan con los requisitos declarados en el filtro. |
| **Flujo de eventos** |  |
|  | 1. La cuenta selecciona la opción de “Inicio” en la barra de navegación. 2. La cuenta rellena el filtro de acuerdo con sus necesidades. 3. El sistema muestra una lista de inmuebles ofertados de acuerdo con los requisitos declarados en el filtro. |
| **Rutas alternativas** | * Al iniciar la aplicación se muestra la lista de propiedades por defecto. * La cuenta puede rellenar el filtro. |

#### Caso N.15 Enviar Mensaje

|  |  |
| --- | --- |
| **ENVIAR MENSAJE** | |
| **Descripción de alto nivel** |  |
|  | Un agente o usuario registrado envían un mensaje. |
| **Requisitos** |  |
|  | RF.7, RF.7.1, RF.7.1.1 |
| **Actores** |  |
|  | Usuario Registrado, Agente |
| **Precondiciones** |  |
|  | * Una cuenta de cualquier tipo deberá haber iniciado sesión. |
| **Postcondiciones** |  |
|  | * Se envía el mensaje. |
| **Flujo de eventos** |  |
|  | 1. La cuenta deberá haber seleccionado la opción “Mensajes”, en la barra de navegación. 2. La cuenta deberá seleccionar la propiedad sobre la cual se desea enviar el mensaje. 3. La cuenta deberá escribir el mensaje. 4. La cuenta deberá pulsar el botón “Enviar”. |
| **Rutas alternativas** | * Se cancela la operación. * Se intenta enviar un mensaje vacío. * En el caso de las cuentas de tipo Usuario, en la pantalla de detalles del sistema tendrán que pulsar un botón que los llevará directamente a una conversación. |

#### Caso N.16 Ver Mensajes Recibidos

|  |  |
| --- | --- |
| **VER MENSAJES RECIBIDOS** | |
| **Descripción de alto nivel** |  |
|  | Un agente o usuario registrado reciben un mensaje. |
| **Requisitos** |  |
|  | RF.7, RF.7.1, RF.7.1.2 |
| **Actores** |  |
|  | Usuario Registrado, Agente |
| **Precondiciones** |  |
|  | * Una cuenta de cualquier tipo deberá haber iniciado sesión. |
| **Postcondiciones** |  |
|  | * Se recibe el mensaje. |
| **Flujo de eventos** |  |
|  | 1. La cuenta deberá haber seleccionado la opción “Mensajes”, en la barra de navegación. 2. La cuenta deberá seleccionar la propiedad sobre la cual se desea enviar el mensaje. 3. La cuenta deberá recibir el mensaje si el emisor del mensaje ha pulsado la opción “Enviar”. |
| **Rutas alternativas** |  |

#### Caso N.17 Añadir a Seguimiento

|  |  |
| --- | --- |
| **AÑADIR A SEGUIMIENTO** | |
| **Descripción de alto nivel** |  |
|  | Un usuario desea añadir a sus seguimientos una propiedad. |
| **Requisitos** |  |
|  | RF.6, RF.6.1 |
| **Actores** |  |
|  | Usuario Registrado |
| **Precondiciones** |  |
|  | * Una cuenta de tipo usuario deberá haber iniciado sesión. |
| **Postcondiciones** |  |
|  | * Se añade la propiedad a sus seguimientos. |
| **Flujo de eventos** |  |
|  | 1. El usuario deberá acceder a la vista de los detalles de cualquier propiedad que no se encuentre en sus seguimientos. 2. El usuario deberá pulsar el botón “Añadir a Seguimiento”. 3. La propiedad se añadirá a sus seguimientos. |
| **Rutas alternativas** |  |

#### Caso N.18 Eliminar de Seguimientos

|  |  |
| --- | --- |
| **ELIMINAR DE SEGUIMIENTO** | |
| **Descripción de alto nivel** |  |
|  | Un usuario desea eliminar de sus seguimientos una propiedad. |
| **Requisitos** |  |
|  | RF.6, RF.6.2 |
| **Actores** |  |
|  | Usuario Registrado |
| **Precondiciones** |  |
|  | * Una cuenta de tipo usuario deberá haber iniciado sesión. |
| **Postcondiciones** |  |
|  | * Se eliminar la propiedad de sus seguimientos. |
| **Flujo de eventos** |  |
|  | 1. El usuario deberá seleccionar la opción “Mis Seguimientos” de la barra de navegación. 2. El usuario deberá pulsar el botón “Eliminar” junto a la propiedad que desee eliminar de sus seguimientos. 3. La propiedad se eliminará de sus seguimientos. |
| **Rutas alternativas** | * El usuario podrá eliminar de sus seguimientos pulsando el botón “Eliminar de Seguimiento”, en la lista de los detalles de cualquier propiedad que se encuentre en sus seguimientos. |

#### Caso N.19 Ver Seguimientos

|  |  |
| --- | --- |
| **VER SEGUIMIENTOS** | |
| **Descripción de alto nivel** |  |
|  | Un usuario desea ver las propiedades de sus seguimientos. |
| **Requisitos** |  |
|  | RF.6, RF.6.4 |
| **Actores** |  |
|  | Usuario Registrado |
| **Precondiciones** |  |
|  | * Una cuenta de tipo usuario deberá haber iniciado sesión. |
| **Postcondiciones** |  |
|  | * Se mostrarán las propiedades de sus seguimientos. |
| **Flujo de eventos** |  |
|  | 1. El usuario deberá seleccionar la opción “Mis Seguimientos” de la barra de navegación. 2. El sistema mostrará sus seguimientos. |
| **Rutas alternativas** |  |

#### Matriz de Trazabilidad

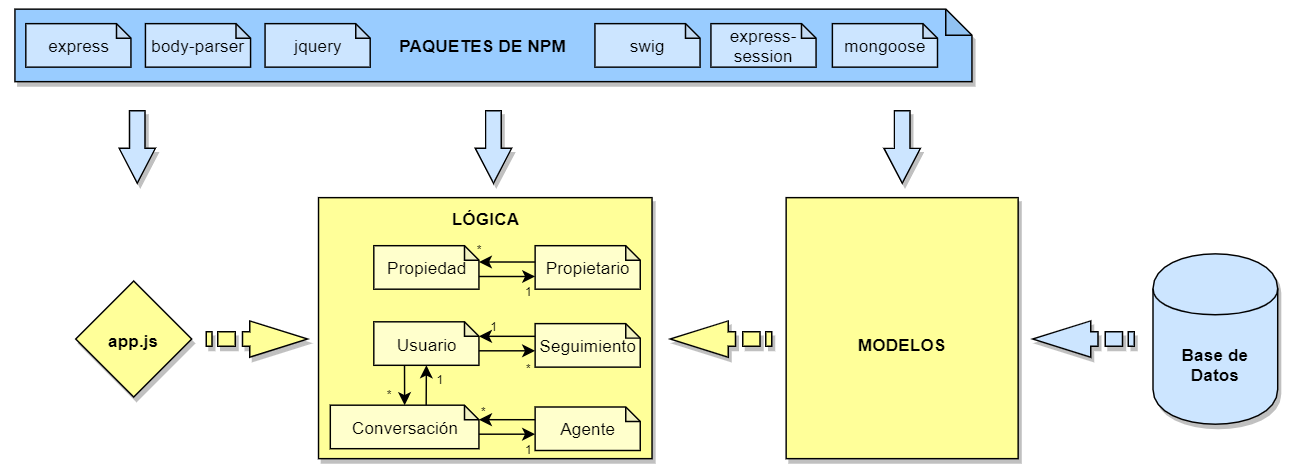
Para no extender la matriz de una manera desproporcionada, en la columna “Requisitos” solo aparecerán aquellos que intervengan al menos una vez.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **CASO DE USO** | | | | | | | | | | | | | | | | | | | |
| **REQUISITO** |  | N.1 | N.2 | N.3 | N.4 | N.5 | N.6 | N.7 | N.8 | N.9 | N.10 | N.11 | N.12 | N.13 | N.14 | N.15 | N.16 | N.17 | N.18 | N.19 |
| RF.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.1.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.1.1.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.1.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.1.2.3.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.1.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.1.3.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.1.3.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.1.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.1.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.1.6.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.2.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.2.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.2.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.4.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.6.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.6.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.6.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.7.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.7.1.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF.7.1.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Análisis de Módulos

### Diagrama de Módulos

El sistema tendrá una distribución similar a la siguiente:



*Figura 34: Subsistemas – Diagrama de Módulos*

Se puede observar que, en primer lugar, es el módulo app.js el encargado de lanzar el sistema y enviárselo a los módulos de lógica. Los modelos se ocupan de obtener la información de la Base de Datos y transmitirla a los módulos de lógica, los cuales se ocupan de procesar la información y mostrarla al usuario, si es necesario. Asimismo, los paquetes de NPM sirven a todos los módulos de la aplicación.

### Descripción de las Módulos

#### Subsistema de Administración

|  |
| --- |
| **Nombre del Módulo** |
| app.js |
| **Descripción** | |
| Módulo encargado de iniciar la aplicación, así como implementar los módulos externos y enviárselos al resto. | |
| **Responsabilidades** | |
| * Mantener la sincronicidad entre módulos. * Lanzar la aplicación. * Establecer el protocolo de lanzamiento HTTPS. * Controlar accesos indebidos. * Preparar la Base de Datos. | |
| **Rutas Propuestas** | |
| * */* | |

#### Subsistema de Utilidad

|  |
| --- |
| **Nombre del Módulo** |
| sessionRender.js |
| **Descripción** | |
| Módulo encargado de mostrar la vista que se le indica por parámetro. | |
| **Responsabilidades** | |
| * Mostrar la vista. * Cargar los datos de la sesión en la vista. | |
| **Rutas Propuestas** | |
|  | |

|  |
| --- |
| **Nombre del Módulo** |
| utilities.js |
| **Descripción** | |
| Módulo encargado de contener las funciones de uso común para el resto de los módulos. | |
| **Responsabilidades** | |
| * Contener las funciones comunes para el resto de las partes del sistema. | |
| **Rutas Propuestas** | |
|  | |

|  |
| --- |
| **Nombre del Módulo** |
| variables.js |
| **Descripción** | |
| Módulo encargado de contener las variables globales para todo el sistema. | |
| **Responsabilidades** | |
| * Contener las variables globales usadas por el resto del sistema. | |
| **Rutas Propuestas** | |
|  | |

#### Subsistema de Base de Datos

|  |
| --- |
| **Nombre del Módulo** |
| connection.js |
| **Descripción** | |
| Módulo encargado de crear y mantener la conexión con la Base de Datos | |
| **Responsabilidades** | |
| * Crear y mantener la conexión con la Base de Datos. | |
| **Rutas Propuestas** | |
|  | |

#### 

|  |
| --- |
| **Nombre del Módulo** |
| agentModel.js |
| **Descripción** | |
| Este módulo crea el esquema de Base de Datos apropiado para gestionar tanto agentes como el super-agente en la Base de Datos. | |
| **Responsabilidades** | |
| * Preparar la Base de Datos para trabajar tanto con agentes como el super-agente. | |
| **Rutas Propuestas** | |
|  | |

|  |
| --- |
| **Nombre del Módulo** |
| userModel.js |
| **Descripción** | |
| Este módulo crea el esquema de Base de Datos apropiado para gestionar los usuarios en la Base de Datos. | |
| **Responsabilidades** | |
| * Preparar la Base de Datos para trabajar con usuarios. | |
| **Rutas Propuestas** | |
|  | |

|  |
| --- |
| **Nombre del Módulo** |
| conversationModel.js |
| **Descripción** | |
| Este módulo crea el esquema de Base de Datos apropiado para gestionar las conversaciones entre agentes y usuarios en la Base de Datos. | |
| **Responsabilidades** | |
| * Preparar la Base de Datos para trabajar con las conversaciones entre agentes y usuarios. | |
| **Rutas Propuestas** | |
|  | |

|  |
| --- |
| **Nombre del Módulo** |
| infoModel.js |
| **Descripción** | |
| Este módulo crea el esquema de Base de Datos apropiado para gestionar la información del sistema almacenada en la Base de Datos. | |
| **Responsabilidades** | |
| * Preparar la Base de Datos para trabajar con la información del sistema. | |
| **Rutas Propuestas** | |
|  | |

|  |
| --- |
| **Nombre del Módulo** |
| loggerModel.js |
| **Descripción** | |
| Este módulo crea el esquema de Base de Datos apropiado para gestionar los registros de actividad de los usuarios almacenados en la Base de Datos. | |
| **Responsabilidades** | |
| * Preparar la Base de Datos para trabajar con los registros de actividad de los usuarios. | |
| **Rutas Propuestas** | |
|  | |

|  |
| --- |
| **Nombre del Módulo** |
| ownerModel.js |
| **Descripción** | |
| Este módulo crea el esquema de Base de Datos apropiado para gestionar los propietarios de las diferentes propiedades en la Base de Datos. | |
| **Responsabilidades** | |
| * Preparar la Base de Datos para trabajar con los propietarios de las propiedades. | |
| **Rutas Propuestas** | |
|  | |

|  |
| --- |
| **Nombre del Módulo** |
| propertyModel.js |
| **Descripción** | |
| Este módulo crea el esquema de Base de Datos apropiado para gestionar las propiedades en la Base de Datos. | |
| **Responsabilidades** | |
| * Preparar la Base de Datos para trabajar con propiedades. | |
| **Rutas Propuestas** | |
|  | |

#### Subsistema de Gestión del Sistema

|  |
| --- |
| **Nombre del Módulo** |
| system.js |
| **Descripción** | |
| Módulo encargado de gestionar la información referida al sistema en cuestión. También se encarga de permitir a un usuario cualquiera registrarse e iniciar sesión. | |
| **Responsabilidades** | |
| * Controlar la información de contacto del cliente. * Preparar y mostrar estadísticas al cliente. | |
| **Rutas Propuestas** | |
| * */home* * */info/contact* * */info/contact/edit* * */info/statistics* | |

#### Subsistema de Gestión de Cuentas

|  |
| --- |
| **Nombre del Módulo** |
| agents.js |
| **Descripción** | |
| Módulo encargado de trabajar con las cuentas de tipo ‘agente’. | |
| **Responsabilidades** | |
| * Mostrar los agentes creados por el super-agente. * Añadir agentes. * Editar agentes. * Filtrar agentes. * Eliminar agentes | |
| **Rutas Propuestas** | |
| * */agents* * */agents/add* * */agents/edit* * */agents/delete* * */signin* * */login* | |

|  |
| --- |
| **Nombre del Módulo** |
| users.js |
| **Descripción** | |
| Módulo encargado de trabajar con las cuentas de tipo ‘usuario’. | |
| **Responsabilidades** | |
| * Añadir usuario (Registrarse). * Editar usuario. * Eliminar usuario. * Permitir a un usuario recuperar su contraseña. * Permitir a un usuario verificar su correo electrónico. | |
| **Rutas Propuestas** | |
| * */user/verification* * */recover* * */user/edit* * */user/delete* | |

#### Subsistema de Mensajería

|  |
| --- |
| **Nombre del Módulo** |
| conversations.js |
| **Descripción** | |
| Módulo encargado de establecer la conversación entre el usuario y el agente. | |
| **Responsabilidades** | |
| * Crear una conversación. * Mostrar los mensajes de la conversación. * Enviar mensajes. * Listar conversaciones. | |
| **Rutas Propuestas** | |
| * /TODO/ | |

#### Subsistema de Gestión de Propiedades

|  |
| --- |
| **Nombre del Módulo** |
| properties.js |
| **Descripción** | |
| Módulo encargado de trabajar con las diferentes propiedades ofertadas. | |
| **Responsabilidades** | |
| * Mostrar los detalles de una propiedad. * Mostrar la pantalla principal de propiedades. * Filtrar propiedades. * Editar una propiedad. * Añadir una propiedad. * Eliminar una propiedad. * Listar propiedades añadidas. | |
| **Rutas Propuestas** | |
| * */properties/details* * */properties/edit* * */properties/delete* * */properties/add* * */properties* * */myproperties* | |

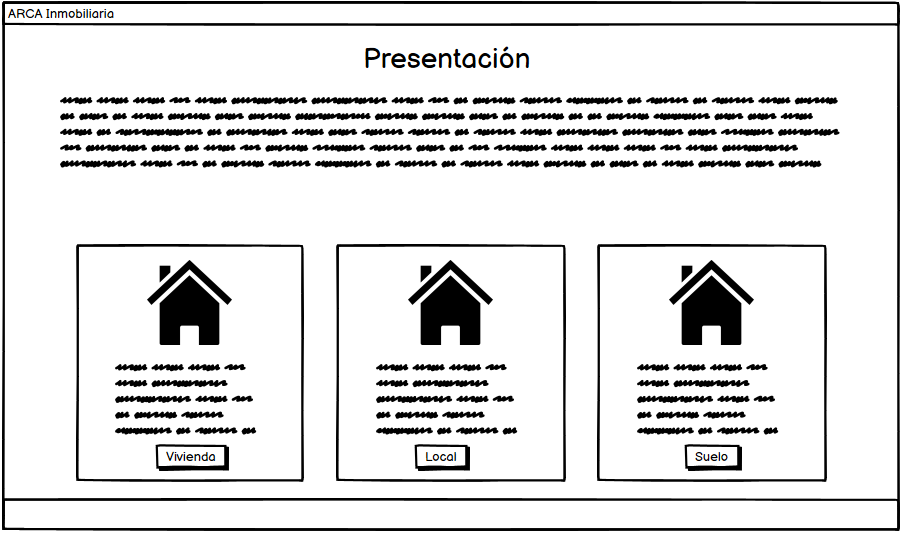
#### Subsistema de Seguimiento

|  |
| --- |
| **Nombre del Módulo** |
| wishes.js |
| **Descripción** | |
| Módulo encargado de trabajar con las propiedades seguidas por un usuario. | |
| **Responsabilidades** | |
| * Mostrar las propiedades seguidas. * Seguir una propiedad. * Eliminar una propiedad de seguimientos. | |
| **Rutas Propuestas** | |
| * */wishes* * */wishes/delete* | |

## Definición de Interfaces de Usuario

### Definición del aspecto de la interfaz

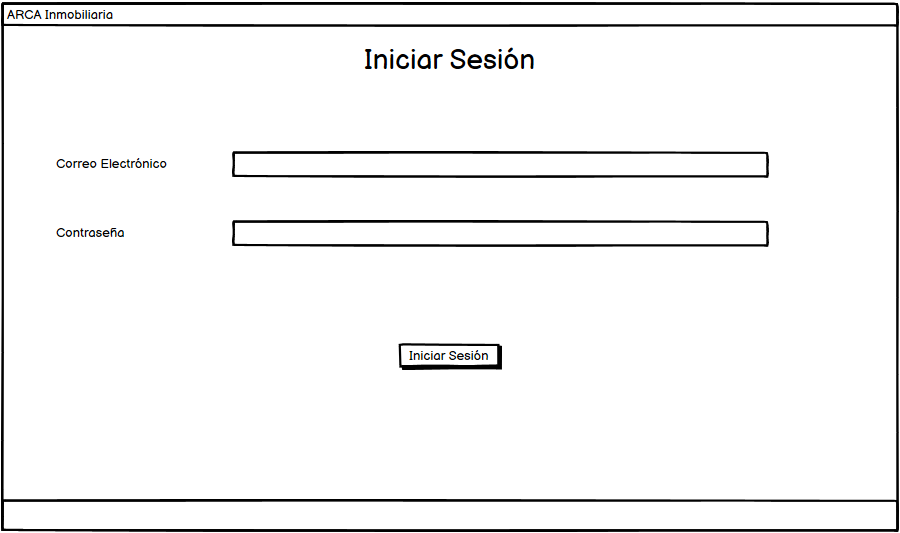
#### Pantalla Principal

Esta sería la pantalla principal que vería un usuario al acceder a la página web.

*Figura 35: Interfaces – Pantalla Principal*

#### Inicio en Sesión en el Sistema

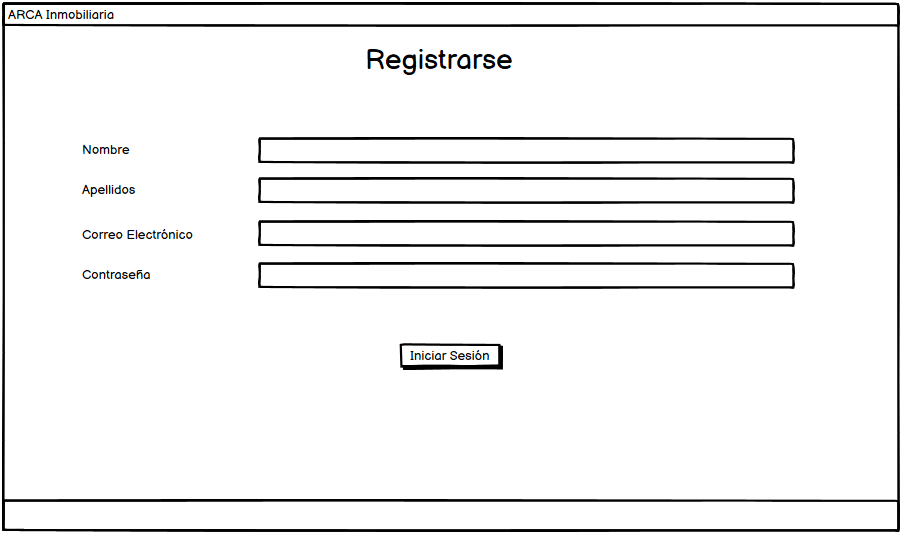
Una vez el usuario se ha registrado en el sistema, podrá iniciar sesión en un formulario similar a este.



*Figura 36: Interfaces – Iniciar Sesión*

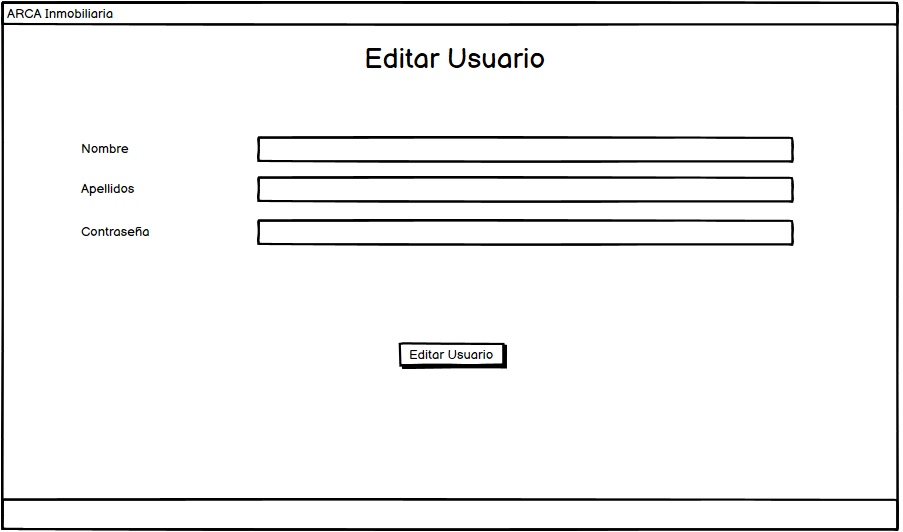
#### Gestión de Usuarios

Esta será la interfaz que permitirá al usuario crear su perfil dentro de la aplicación.



*Figura 37: Interfaces – Registro*

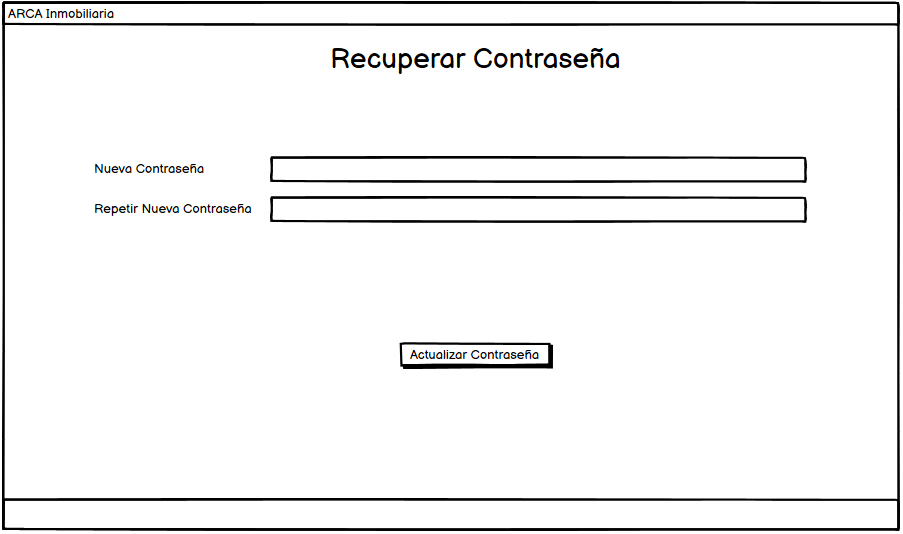
Asimismo, un usuario podrá modificar los datos que introdujo para registrarse mediante un formulario similar a este.



*Figura 38: Interfaces – Editar Usuario*

Como podemos observar, es muy similar al formulario de registro, salvo que el correo electrónico no se puede modificar. Por otra parte, no existirá pantalla de eliminación de usuario, pues una vez el mismo seleccione la opción de eliminarla, aparecerá un pop-up con un campo para introducir su contraseña de nuevo.

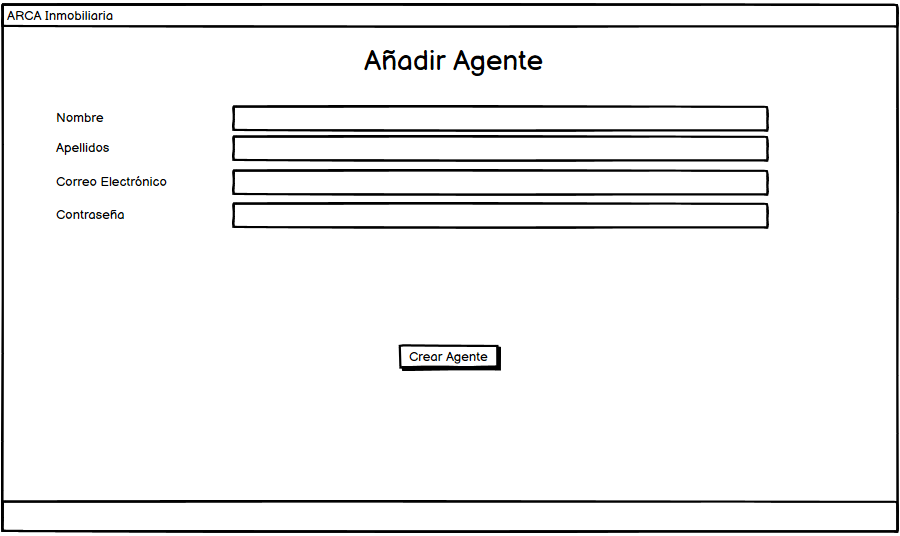
Por otra parte, si un usuario perdiera su contraseña y quisiera recuperarla, tras acceder al enlace enviado por correo vería una pantalla similar a la siguiente.



*Figura 39: Interfaces – Recuperar Contraseña*

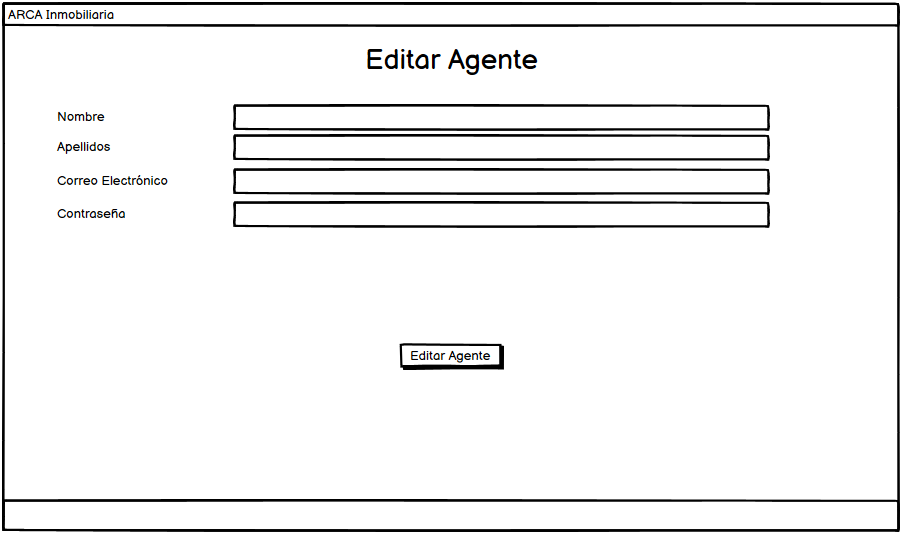
#### Gestión de Agentes

Una cuenta de super-usuario podrá añadir un agente tras completar el siguiente formulario.



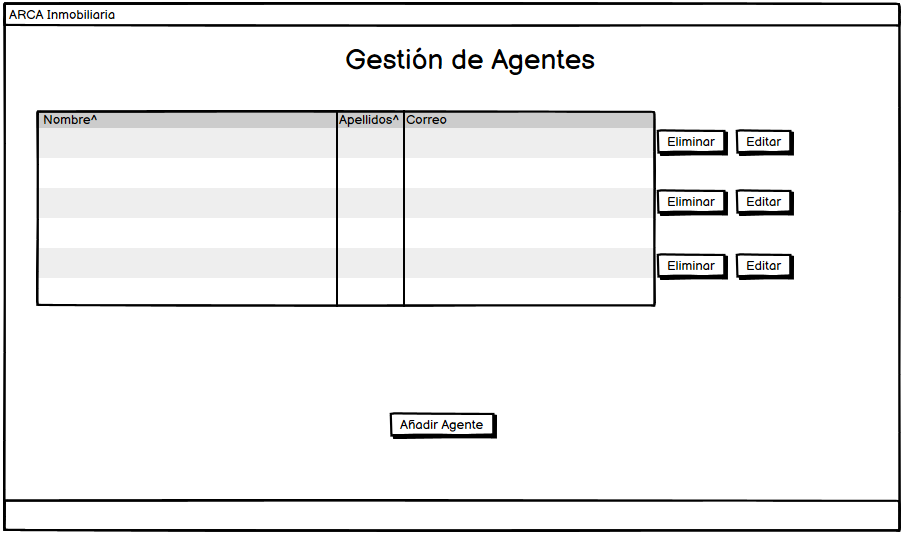
*Figura 40: Interfaces – Añadir Agente*

Asimismo, el formulario de edición del agente será similar al anterior, pues podrá editar todos los campos, en este caso incluido el correo electrónico.



*Figura 41: Interfaces – Editar Agente*

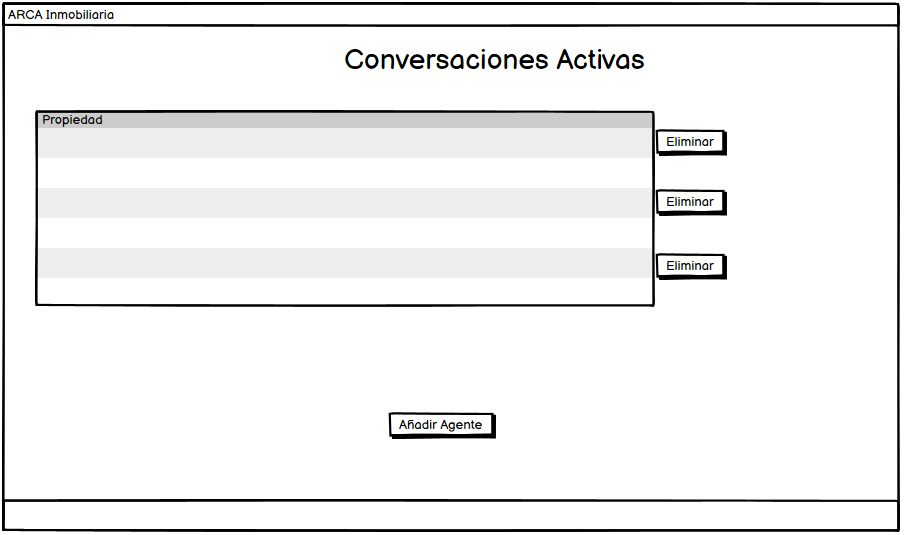
Finalmente, el super-agente podrá gestionar los diferentes agentes creados mediante un formulario como este.



*Figura 42: Interfaces – Gestión de Agentes*

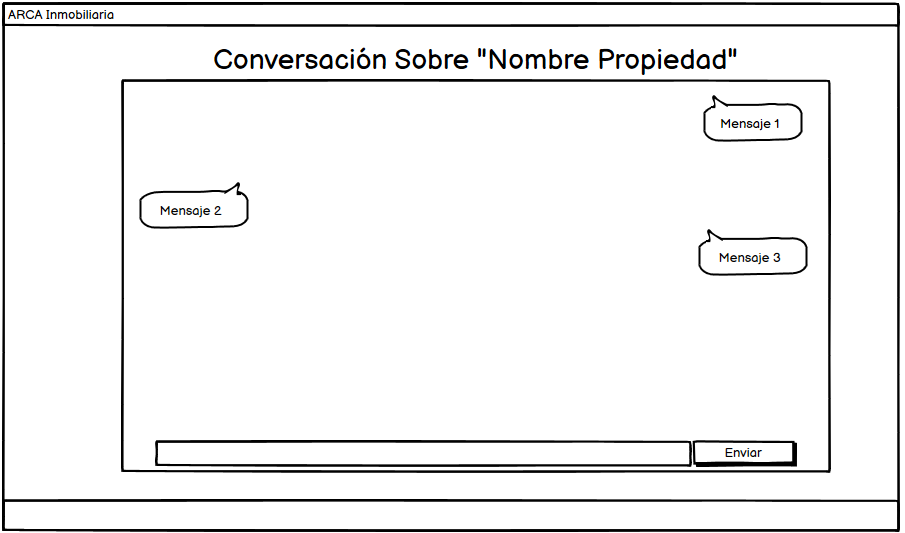
#### Conversaciones

Una cuenta podrá ver las conversaciones que tiene activas mediante una interfaz como la siguiente.



*Figura 43: Interfaces – Conversaciones*

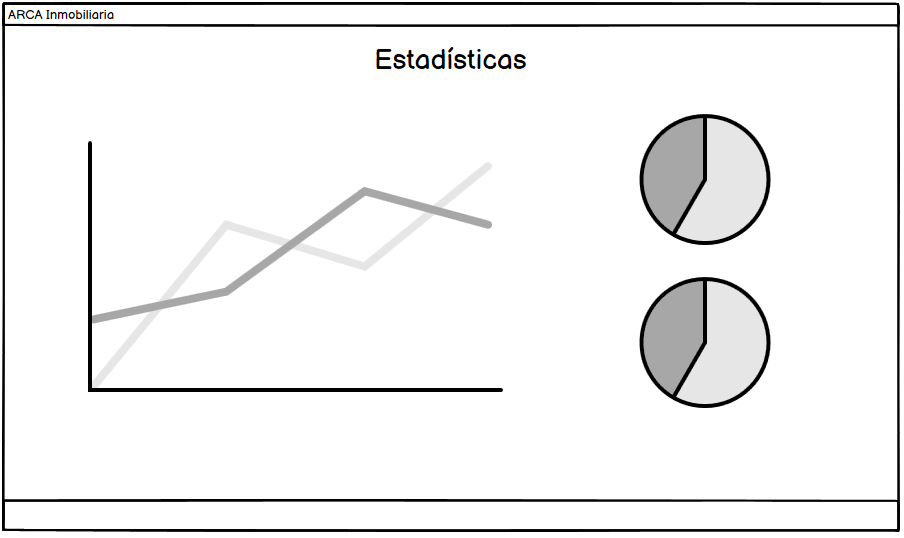
Una vez accedida a una conversación, esta será la pantalla de uno de los chats.



*Figura 44: Interfaces – Chat*

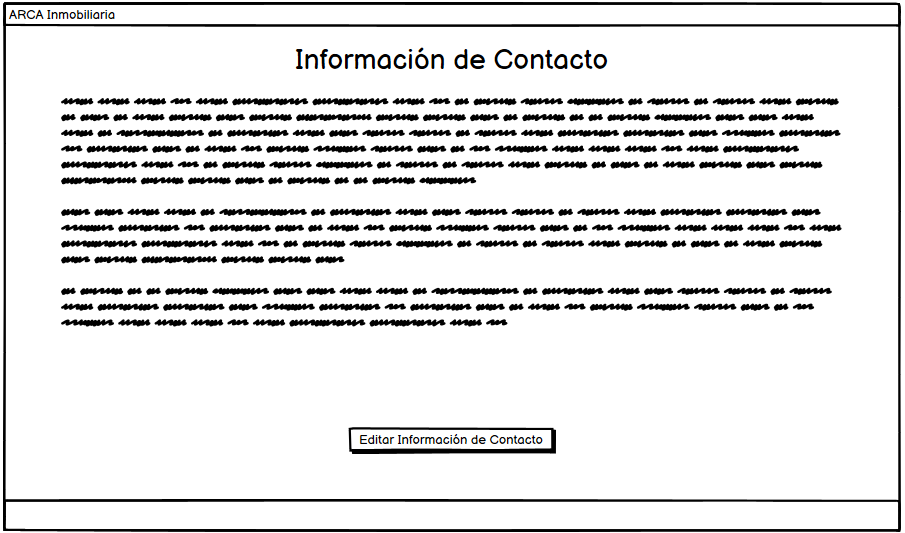
#### Estadísticas y Contacto

Esta sería la pantalla de estadísticas que le aparecerían a los agentes y al super-agente.



*Figura 45: Interfaces – Estadísticas*

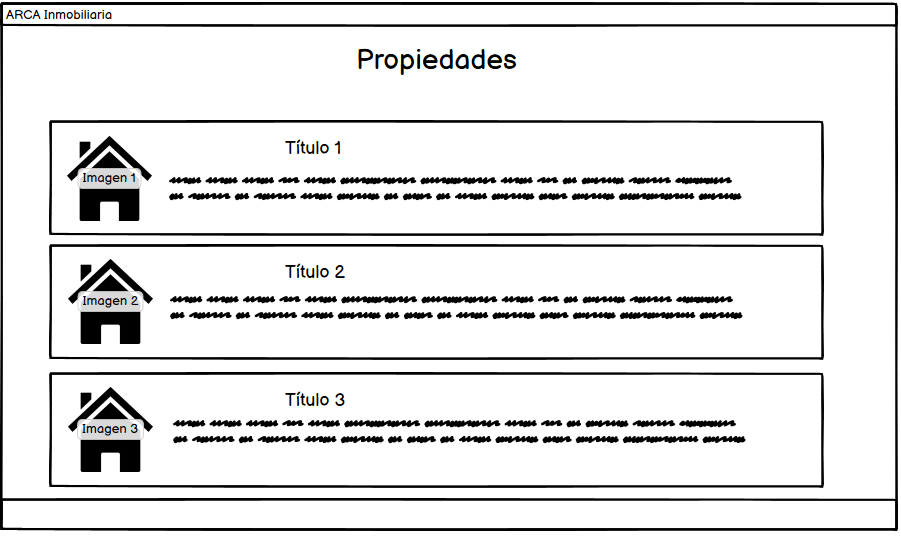
Asimismo, si un usuario accediera a la opción de “Contactar”, le aparecería una interfaz similar a esta.



*Figura 46: Interfaces – Información de Contacto*

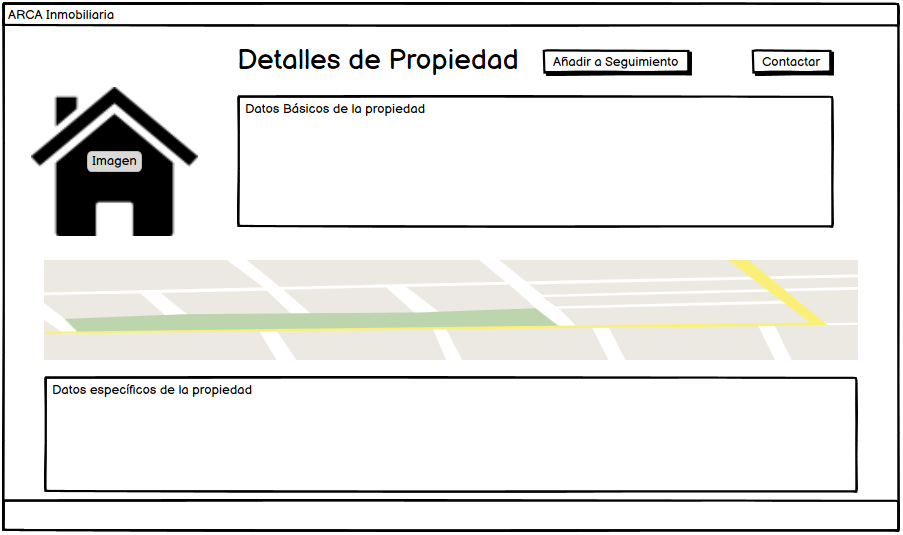
#### Gestión de Propiedades

La pantalla principal desde la que un cliente podrá ver las propiedades ofertadas será como la siguiente:



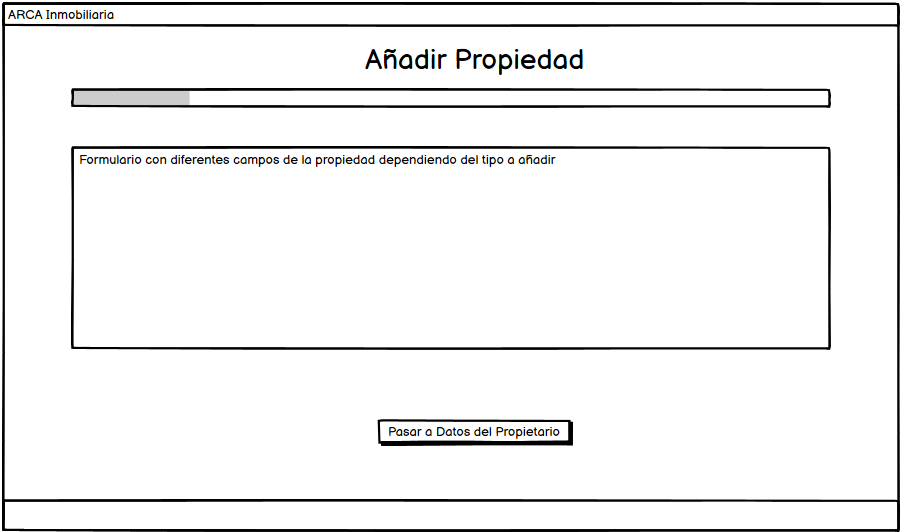
*Figura 47: Interfaces – Lista de Propiedades*

Una vez el cliente desee ver los detalles de una propiedad en concreto, podrá verlos mediante una pantalla como la siguiente.



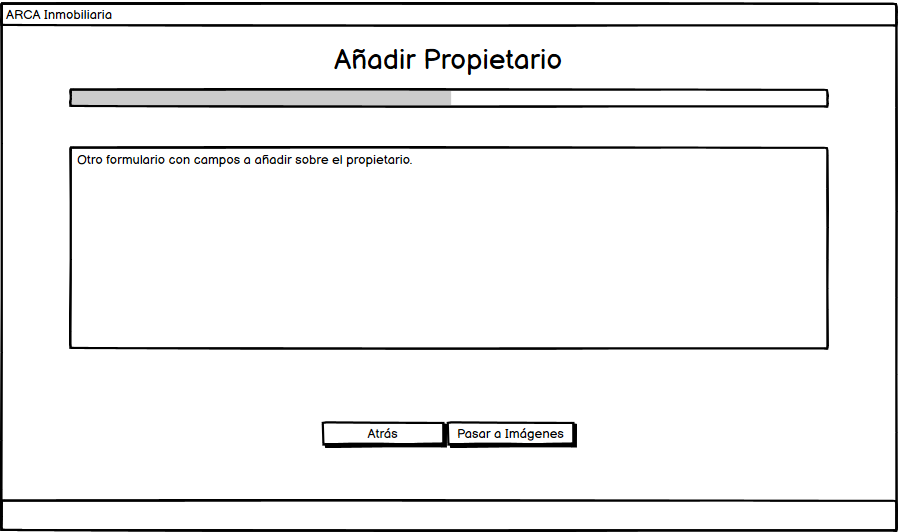
*Figura 48: Interfaces – Detalles*

Por otra parte, una vez un agente desee añadir una propiedad. Deberá rellenar una serie de formularios que le aparecerán acompañados de una barra de progreso, para que pueda ver el estado en el que se encuentra el progreso de adición. La primera de las ventanas le permitiría añadir datos sobre la propiedad en sí.



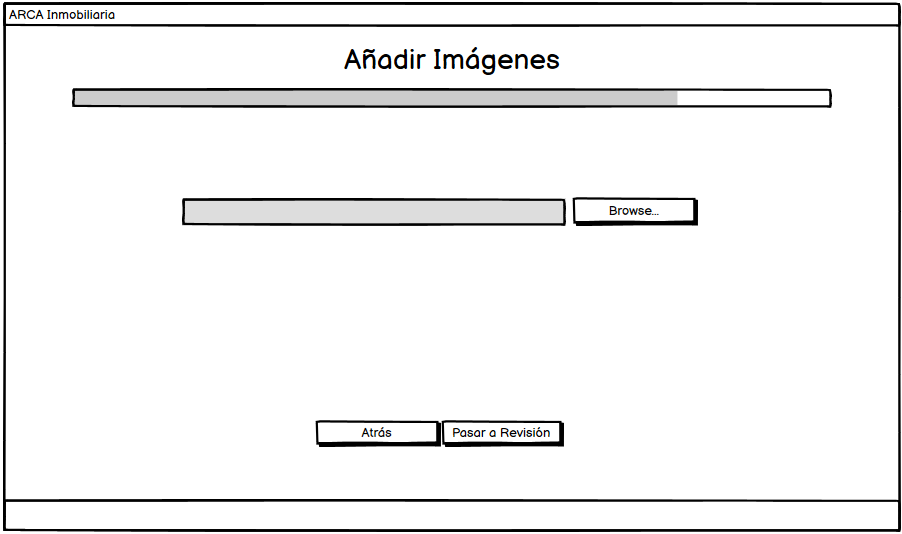
*Figura 49: Interfaces – Añadir Propiedad 1*

Una vez rellenados los campos correctamente, pasaría a la ventana de añadir datos del propietario.



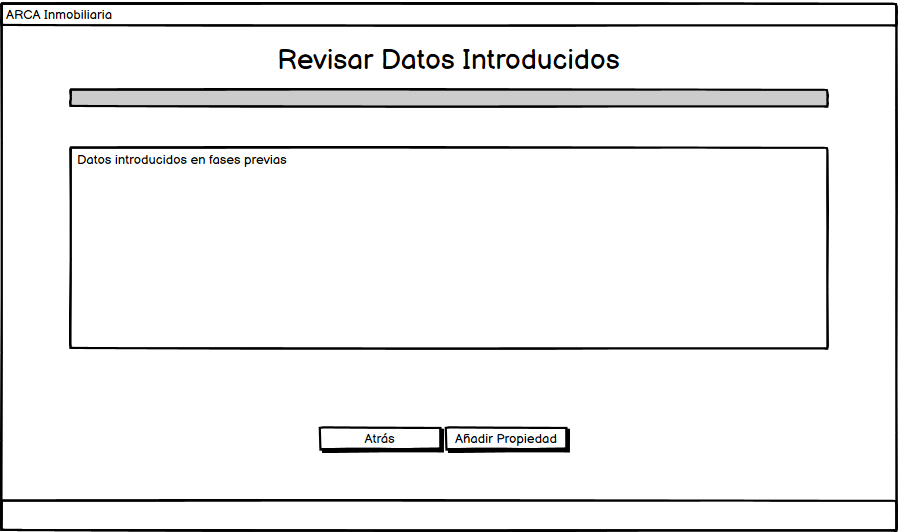
*Figura 50: Interfaces – Añadir Propiedad 2*

Continuando el proceso, pasaríamos a la ventana de añadir imágenes de la propiedad.



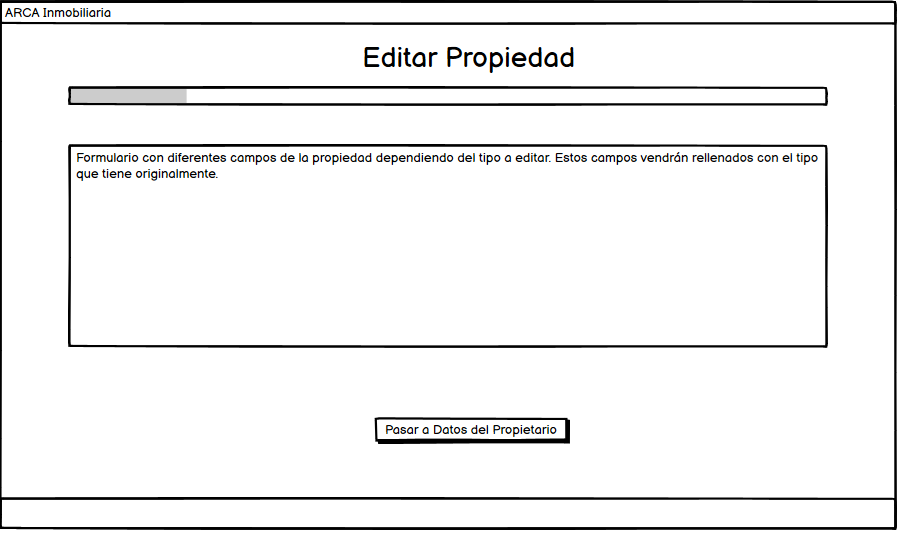
*Figura 51: Interfaces – Añadir Propiedad 3*

Finalmente, el sistema mostraría al agente los datos que se van a añadir al sistema, pudiendo volver para modificarlos en cualquier momento.

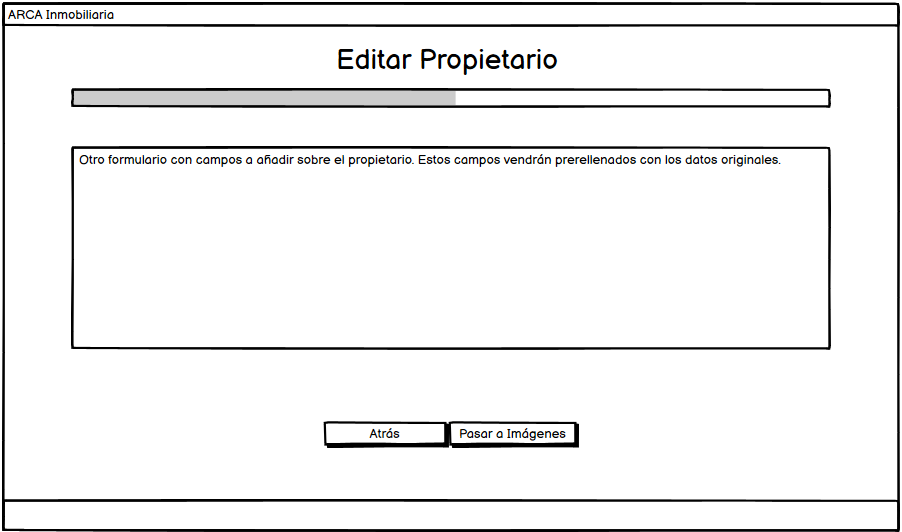


*Figura 52: Interfaces – Añadir Propiedad 4*

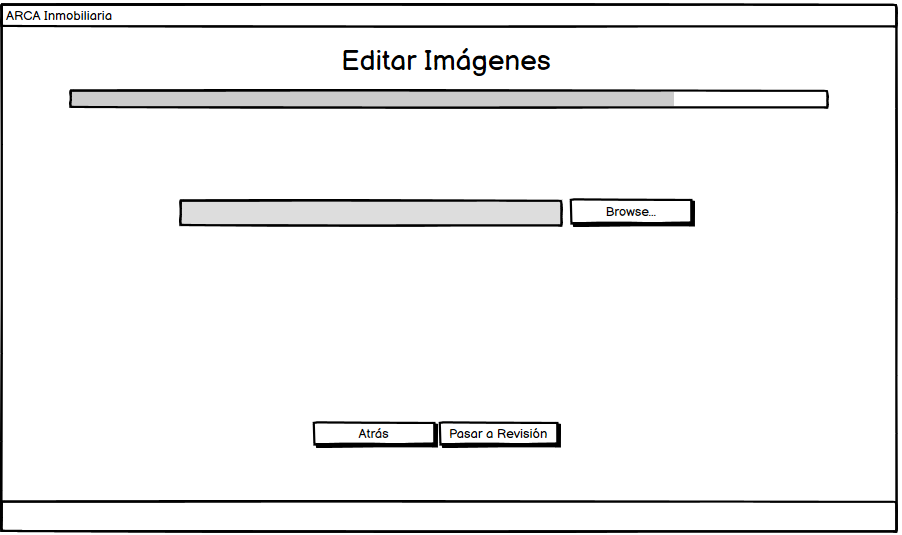
Podemos ver por otra parte, que el formulario de editar una propiedad sería prácticamente el mismo.



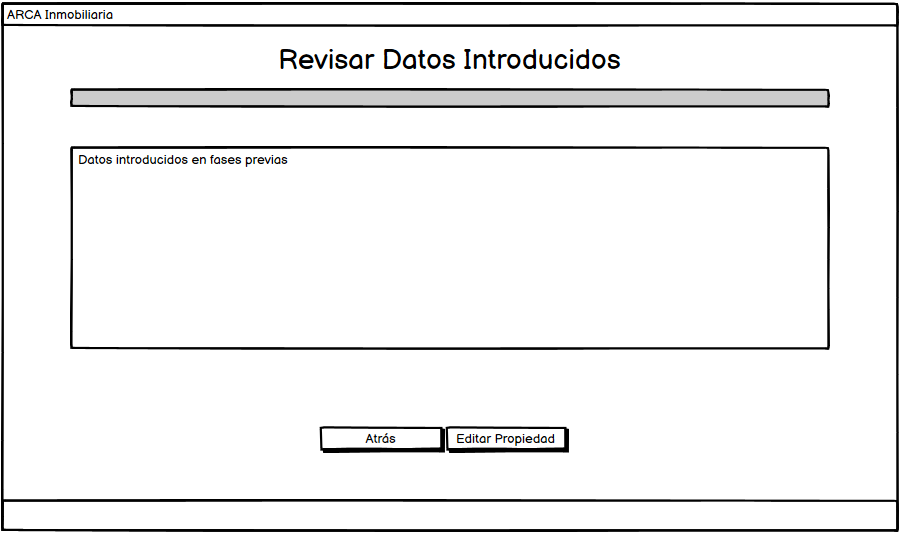
*Figura 53: Interfaces – Editar Propiedad 1*



*Figura 54: Interfaces – Editar Propiedad 2*

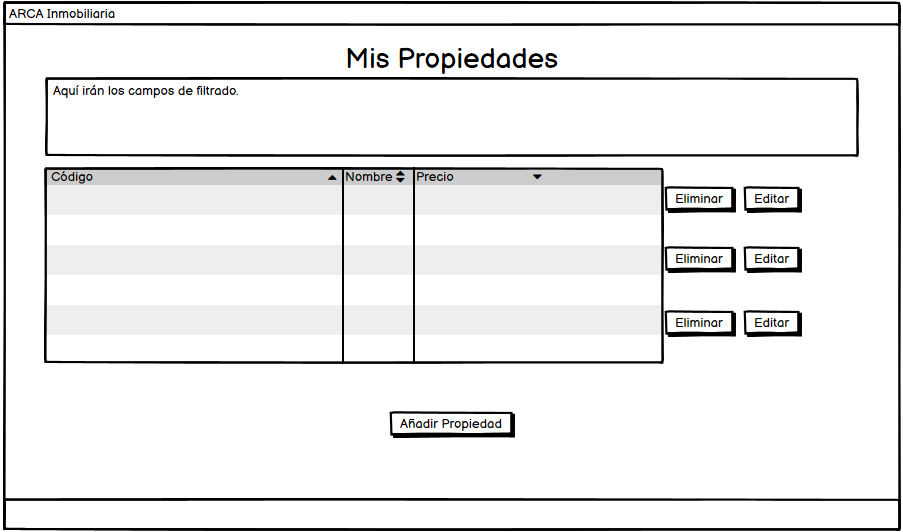


*Figura 55: Interfaces – Editar Propiedad 3*



*Figura 56: Interfaces – Editar Propiedad 4*

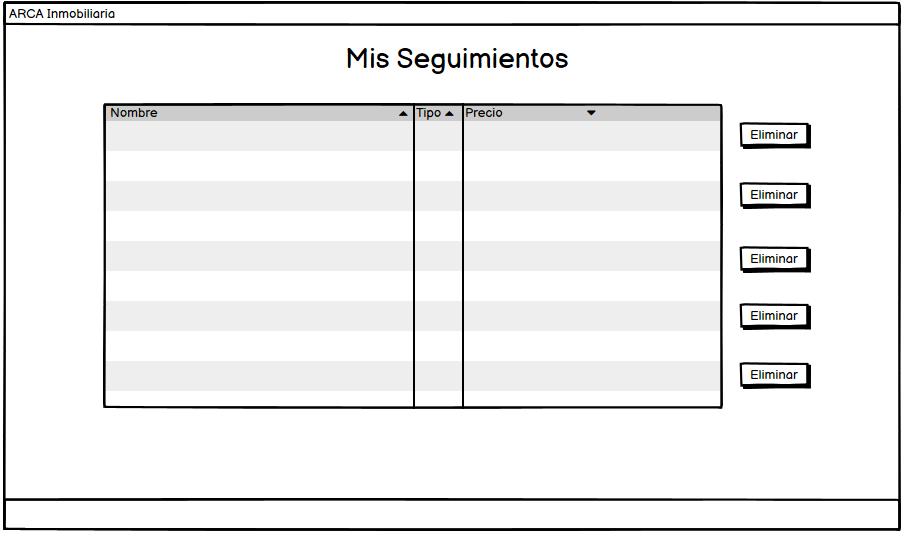
Por otra parte, un agente podrá ver las propiedades que ha añadido con una interfaz como esta.



*Figura 57: Interfaces – Mis Propiedades*

#### Gestión de Seguimientos

Un usuario podrá ver la lista de las propiedades que tiene en seguimiento mediante la siguiente interfaz.



*Figura 58: Interfaces – Mis Seguimientos*

### Descripción del Comportamiento de la Interfaz

Para describir el comportamiento de la interfaz, debemos tener en cuenta en qué situación (o situaciones) se originan los contratiempos que obligan al sistema a mostrar ciertos errores o a bloquear ciertas acciones.

#### Validación de Datos

En primer lugar, debemos tener en cuenta que todas aquellas peticiones que conllevan la introducción de datos personales por parte del usuario son enviadas mediante el método POST. Este método nos permite enviar información de una manera más segura que la contraposición, GET, pues POST está diseñado para enviar la información introducida de un modo oculto o en segundo plano, mientras que GET envía los datos utilizando la URL.

Por otra parte, cabe destacar que todas las contraseñas se encuentran debidamente encriptadas para que no puedan ser robadas por posibles atacantes.

Finalmente, gracias a los esquemas creados con el paquete de NPN *mongoose*, podemos indicarle al sistema qué tipos deben tener los campos que vamos a introducir, por lo que, si el sistema se encuentra con algún dato inesperado, mostrará un error y no permitirá continuar con la tarea.

#### Protección de Cuentas

Si un usuario desea registrarse, su cuenta aparecerá desactivada hasta que valide su correo electrónico accediendo al enlace que se le envía automáticamente por correo electrónico. Similar ocurre en el caso de que un usuario desee recuperar su contraseña, pues accederá a un enlace de modificación que se le envíe por correo electrónico. Si el usuario cambia la contraseña correctamente, se le mostrará un mensaje de éxito. Si no la introduce correctamente, se mostrará un error. Si dicho enlace caduca, deberá solicitar el proceso de nuevo.

Por otra parte, si el usuario desea realizar acciones críticas, como por ejemplo eliminar su contraseña, el sistema mostrará una pantalla desde la cual deberá introducir su contraseña. Si es correcta, la acción se llevará a cabo. En caso contrario, se mostrará un mensaje de error.

#### Manejo de Errores

Mediante el paquete de NPM llamado *connect-flash* podemos crear una memoria flash (la cual nos permite crear un error y eliminarlo automáticamente una vez mostrado), con la cual mostrar diversos mensajes, tanto de error como de éxito, en la parte superior en la pantalla de forma que sean claramente visibles por el usuario. Estos errores se mostrarán en la práctica totalidad de la aplicación:

* Cuando una tarea se realiza correctamente (inicio de sesión correcto, añadir una propiedad correctamente etc.) mostrará un mensaje de éxito en color verde con un texto explicativo.
* Cuando una tarea no llega a realizarse correctamente (campos introducidos incorrectos o vacíos, error en la Base de Datos etc.) mostrará un mensaje de error de color rojo con, de la misma manera, un texto descriptivo.

De la misma manera, los mensajes contenidos en los errores contienen únicamente texto explicativo, y no mostrarán al usuario mas información de la estrictamente necesaria.

### Diagrama de Navegabilidad

*Figura 59: Diagrama de Navegabilidad*

## Especificación del Plan de Pruebas

# CAPÍTULO 7. DISEÑO DEL SISTEMA DE INFORMACIÓN

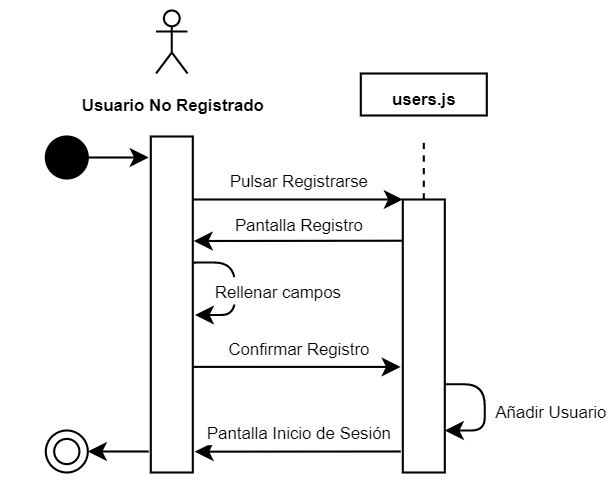
## Diseño de Casos de Uso Reales

Para mostrar el comportamiento de cada uno de los casos de uso, utilizaremos tres tipos diferentes de diagramas:

* **Diagrama de Interacción:** Muestra la interacción entre los diferentes componentes del sistema.
* **Diagrama de Rutas:** Muestra la secuencia de rutas que sigue el sistema para realizar todas las tareas oportunas.
* **Diagrama de Actividad:** Muestra las actividades que se ejecutarán para llevar a cabo la actividad del caso de uso.

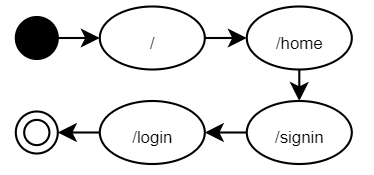
### Caso N.1 Crear Perfil Usuario

#### Diagrama de Interacción



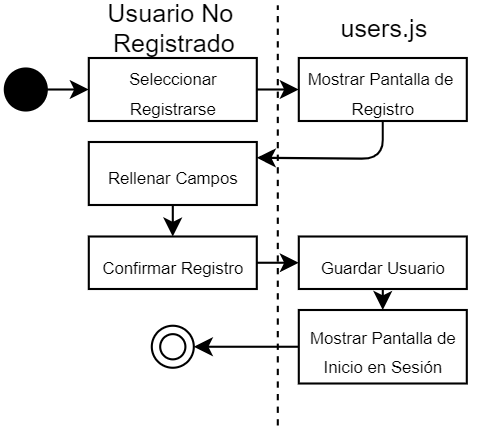
*Figura 60: Crear Perfil Usuario – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 61: Crear Perfil Usuario – Diagrama de Rutas*

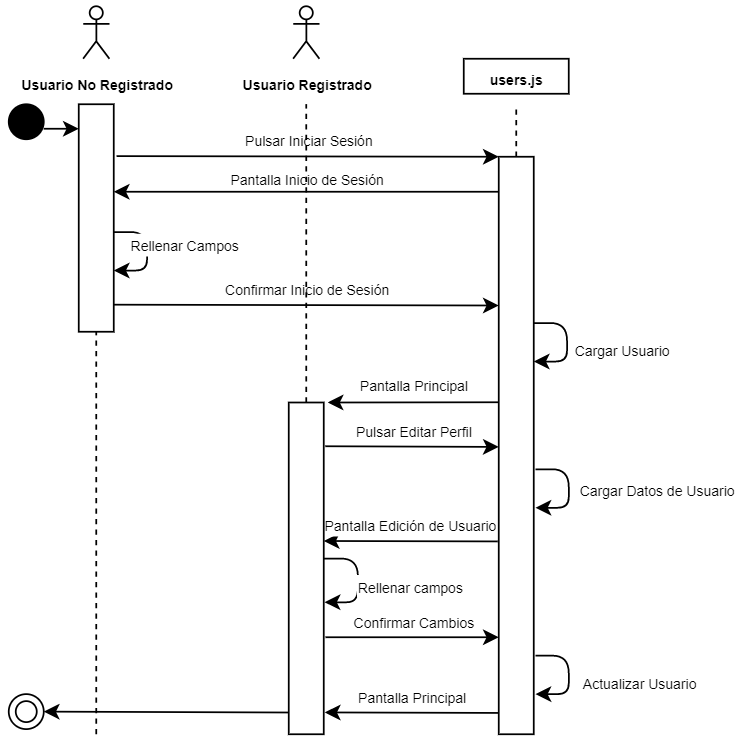
#### Diagrama de Actividad



*Figura 62: Crear Perfil Usuario – Diagrama de Actividad*

### Caso N.2 Editar Perfil Usuario

#### Diagrama de Interacción



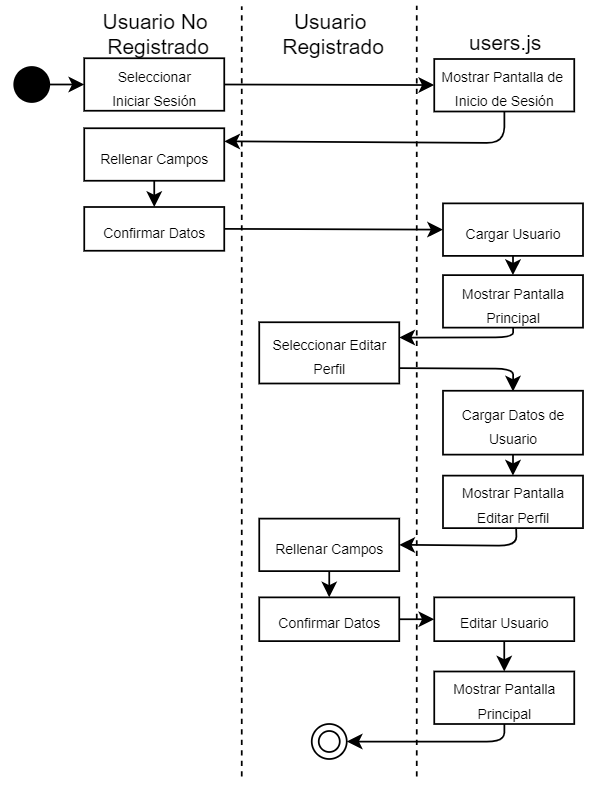
*Figura 63: Editar Perfil Usuario – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 64: Editar Perfil Usuario – Diagrama de Rutas*

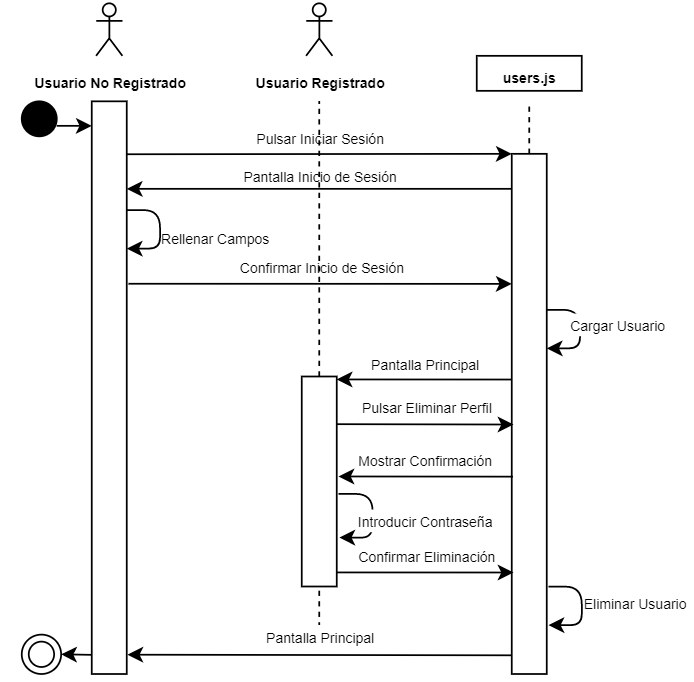
#### Diagrama de Actividad



*Figura 65: Editar Perfil Usuario – Diagrama de Actividad*

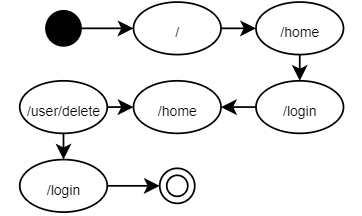
### Caso N.3 Eliminar Perfil Usuario

#### Diagrama de Interacción



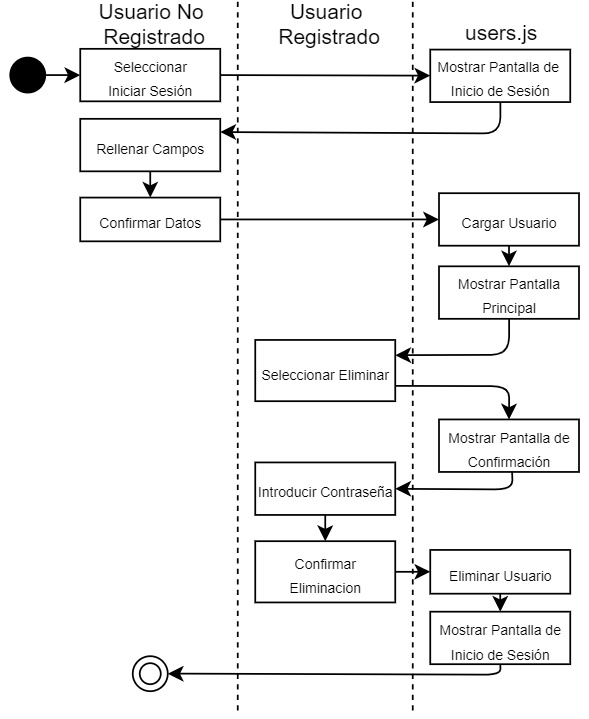
*Figura 66: Eliminar Perfil Usuario – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 67: Eliminar Perfil Usuario – Diagrama de Rutas*

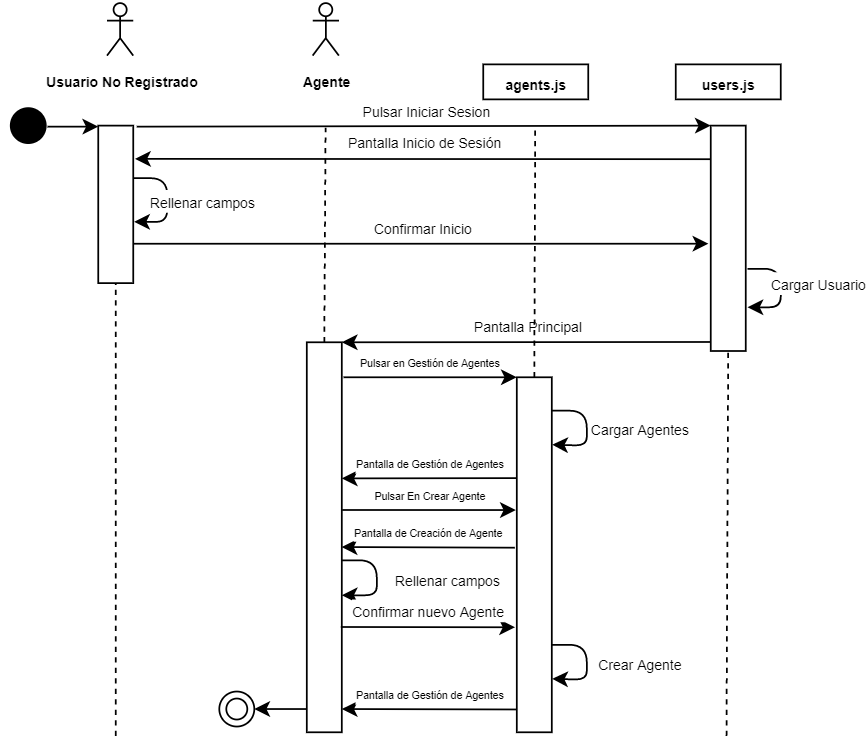
#### Diagrama de Actividad



*Figura 68: Eliminar Perfil Usuario – Diagrama de Actividad*

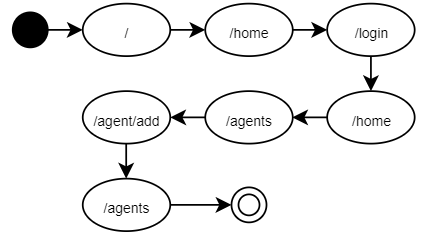
### Caso N.4 Crear Perfil Agente

#### Diagrama de Interacción



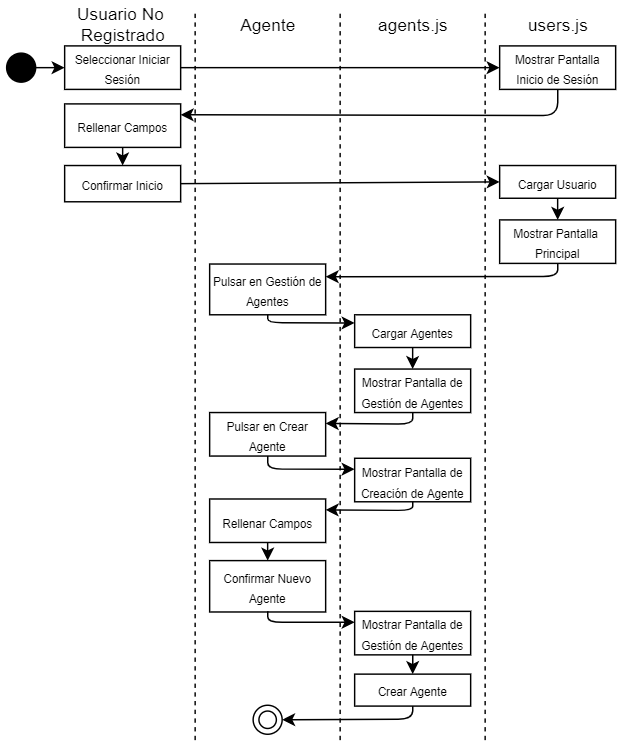
*Figura 69: Crear Perfil Agente – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 70: Crear Perfil Agente – Diagrama de Rutas*

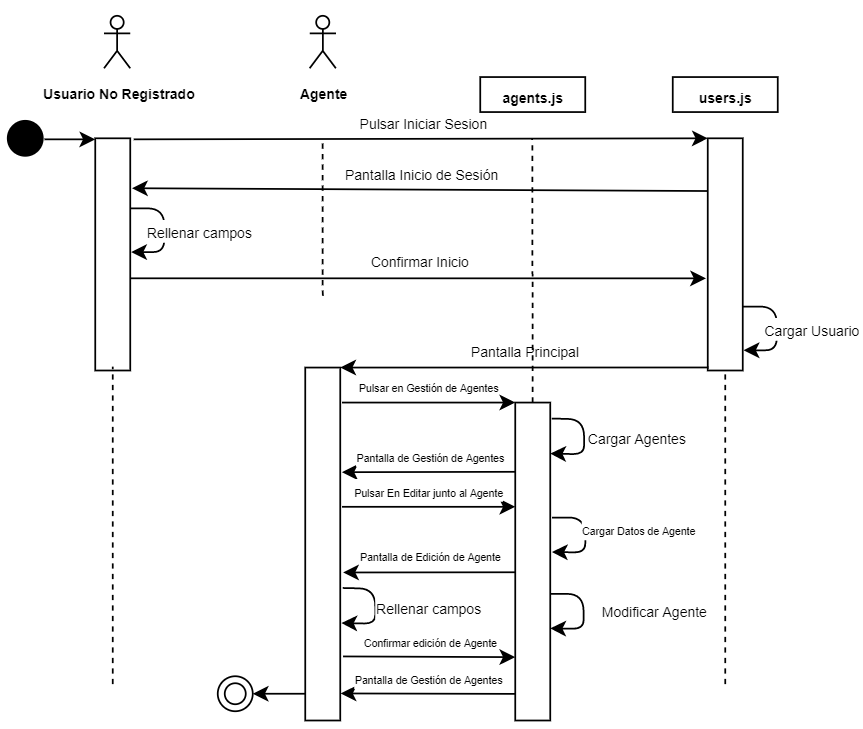
#### Diagrama de Actividad



*Figura 71: Crear Perfil Agente – Diagrama de Actividad*

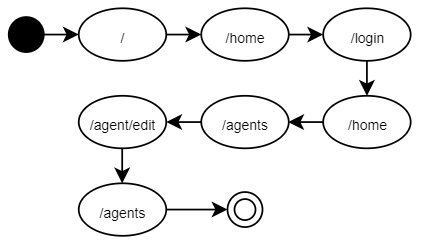
### Caso N.5 Editar Perfil Agente

#### Diagrama de Interacción



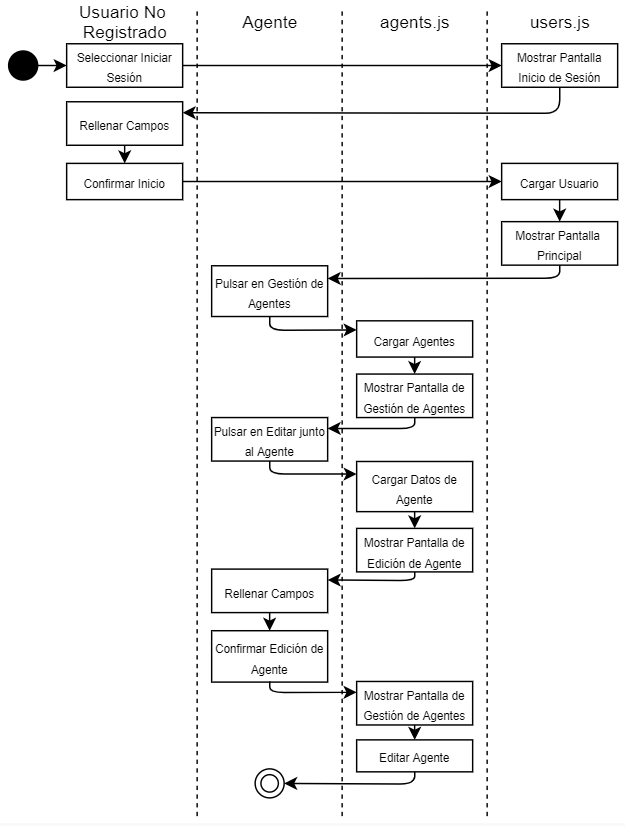
*Figura 72: Editar Perfil Agente – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 73: Crear Perfil Agente – Diagrama de Rutas*

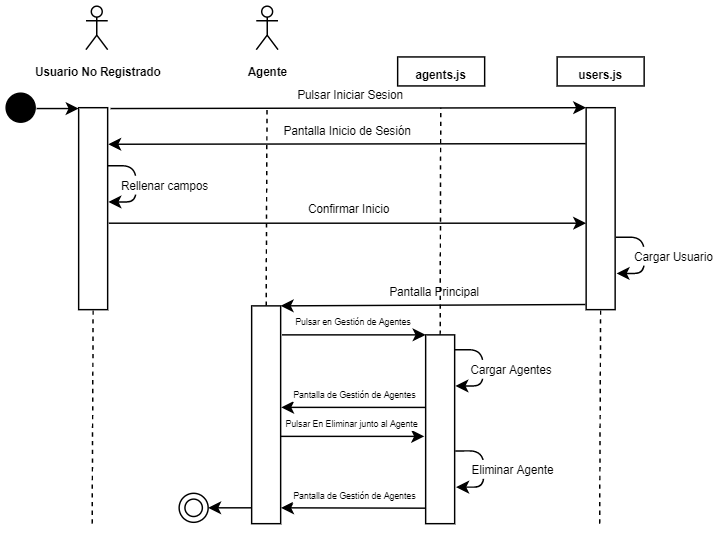
#### Diagrama de Actividad



*Figura 74: Crear Perfil Agente – Diagrama de Actividad*

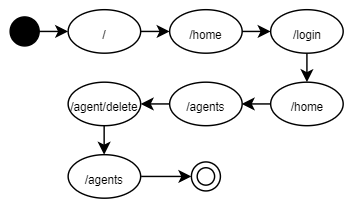
### Caso N.6 Eliminar Perfil Agente

#### Diagrama de Interacción



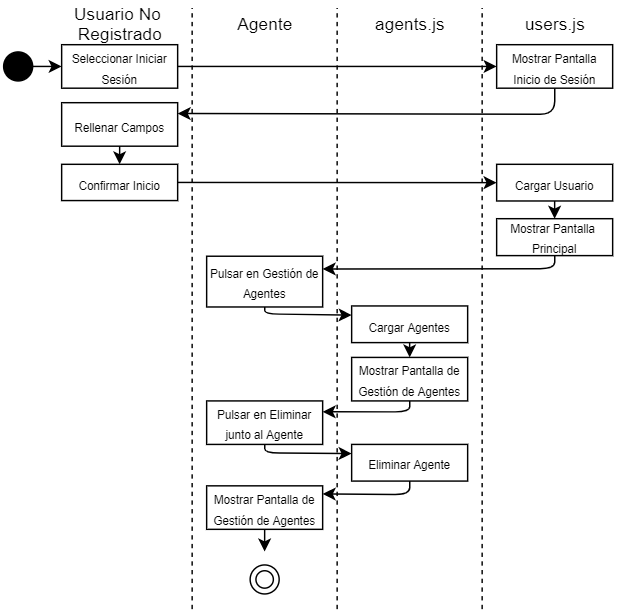
*Figura 75: Eliminar Perfil Agente – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 76: Eliminar Perfil Agente – Diagrama de Rutas*

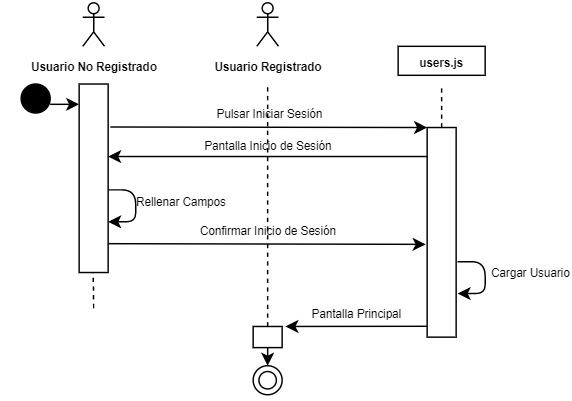
#### Diagrama de Actividad



*Figura 77: Eliminar Perfil Agente – Diagrama de Actividad*

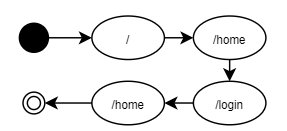
### Caso N.7 Inicio de Sesión

#### Diagrama de Interacción



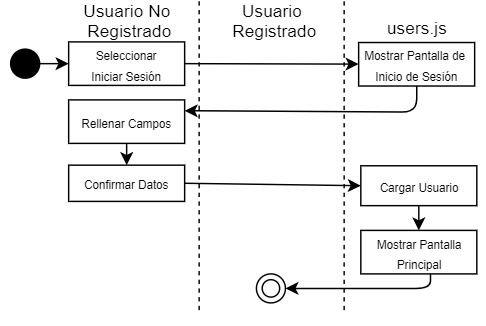
*Figura 78: Inicio de Sesión – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 79: Inicio de Sesión – Diagrama de Rutas*

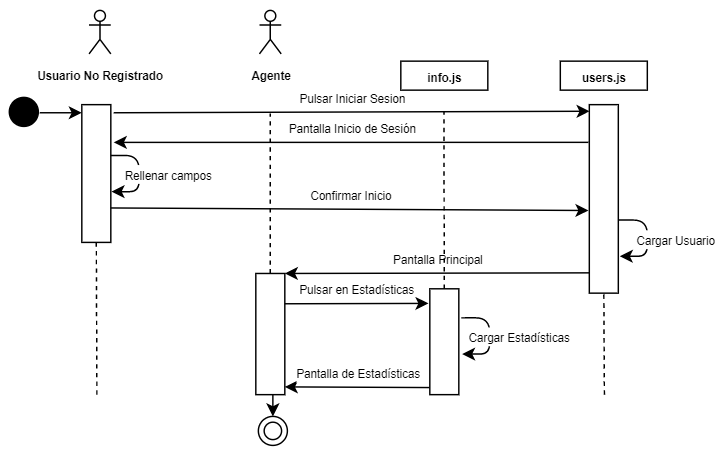
#### Diagrama de Actividad



*Figura 80: Inicio de Sesión – Diagrama de Actividad*

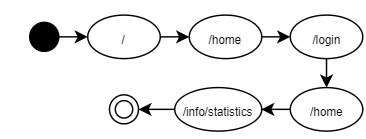
### Caso N.8 Ver Estadísticas de Ventas

#### Diagrama de Interacción



*Figura 81: Ver Estadísticas de Ventas – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 82: Ver Estadísticas de Ventas – Diagrama de Rutas*

#### Diagrama de Actividad



*Figura 83: Ver Estadísticas de Ventas – Diagrama de Actividad*

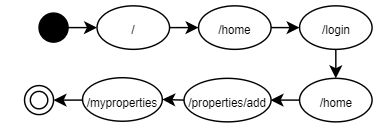
### Caso N.9 Crear Propiedad

#### Diagrama de Interacción



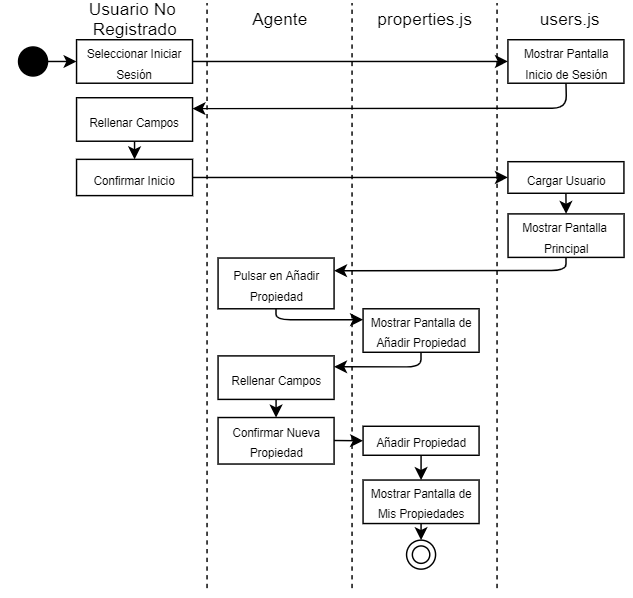
*Figura 84: Crear Propiedad – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 85: Crear Propiedad – Diagrama de Rutas*

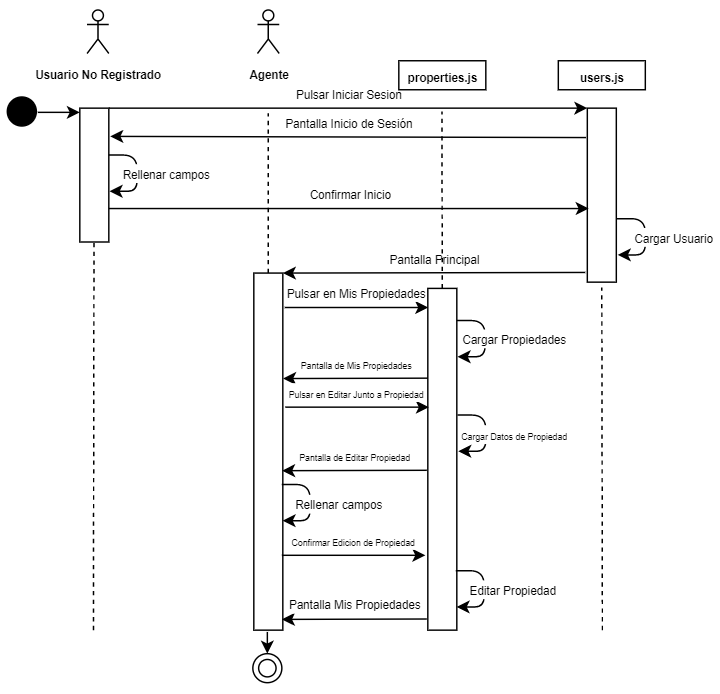
#### Diagrama de Actividad



*Figura 86: Crear Propiedad – Diagrama de Actividad*

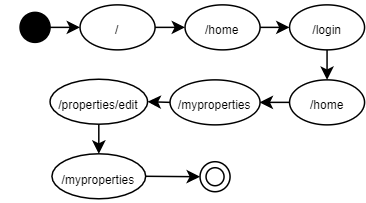
### Caso N.10 Editar Propiedad

#### Diagrama de Interacción



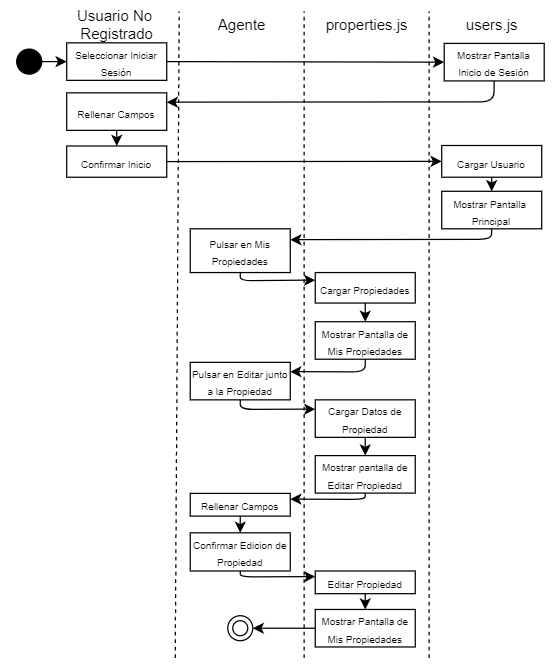
*Figura 87: Editar Propiedad – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 88: Editar Propiedad – Diagrama de Rutas*

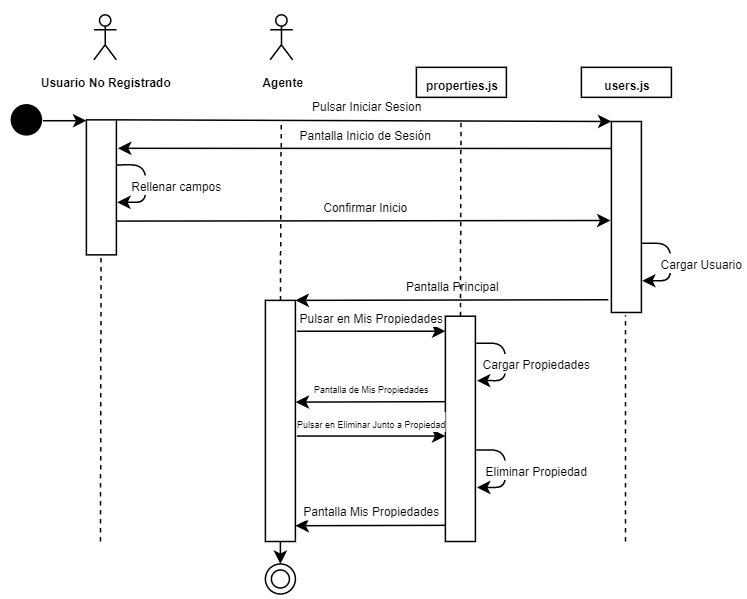
#### Diagrama de Actividad



*Figura 89: Editar Propiedad – Diagrama de Actividad*

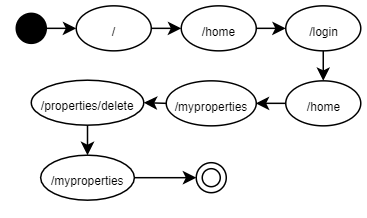
### Caso N.11 Eliminar Propiedad

#### Diagrama de Interacción



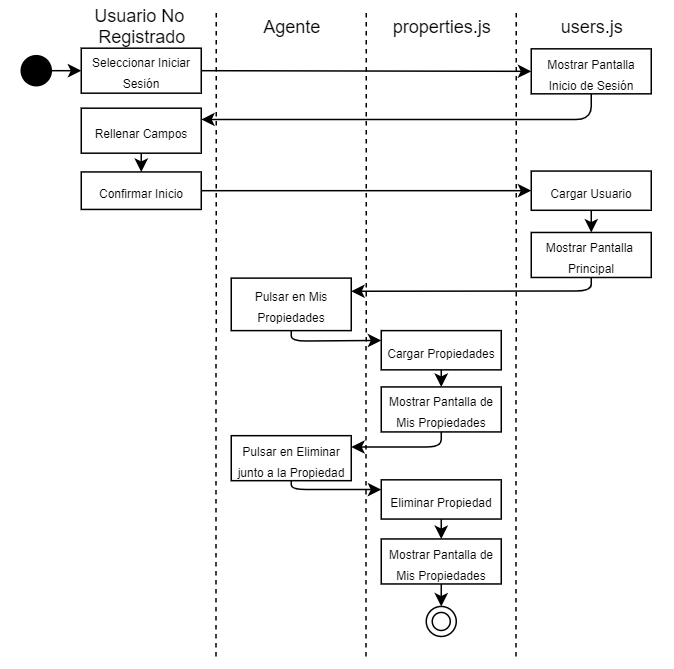
*Figura 90: Eliminar Propiedad – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 91: Eliminar Propiedad – Diagrama de Rutas*

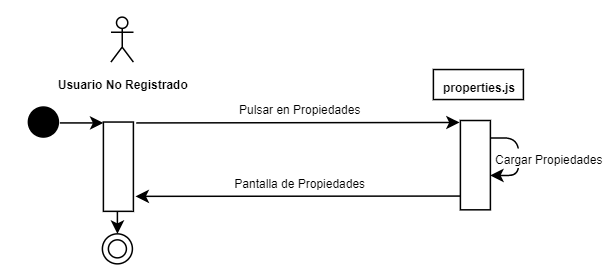
#### Diagrama de Actividad



*Figura 92: Eliminar Propiedad – Diagrama de Actividad*

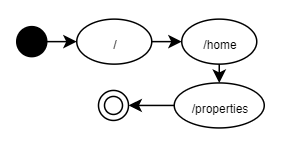
### Caso N.12 Ver Propiedades

#### Diagrama de Interacción



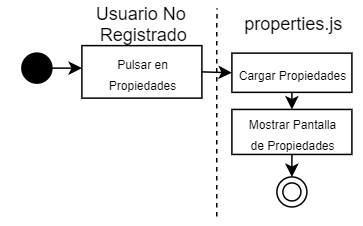
*Figura 93: Ver Propiedades – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 94: Ver Propiedades – Diagrama de Rutas*

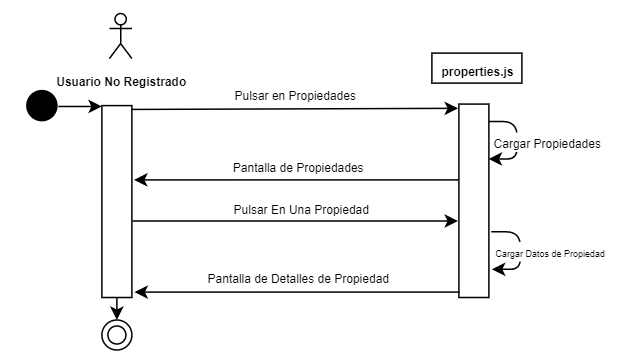
#### Diagrama de Actividad



*Figura 95: Ver Propiedades – Diagrama de Actividad*

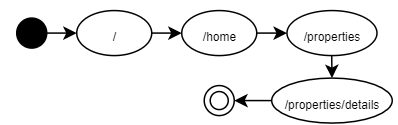
### Caso N.13 Ver detalles de Propiedades

#### Diagrama de Interacción



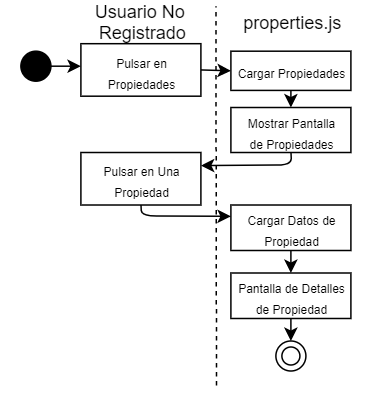
*Figura 96: Ver detalles de Propiedades – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 97: Ver detalles de Propiedades – Diagrama de Rutas*

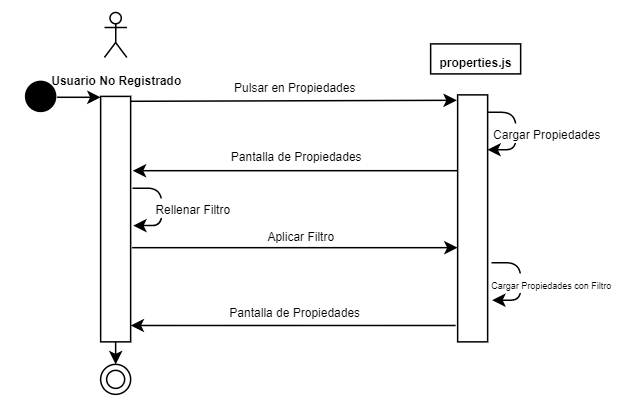
#### Diagrama de Actividad



*Figura 98: Ver detalles de Propiedades – Diagrama de Actividad*

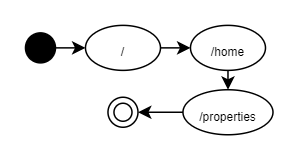
### Caso N.14 Filtrar Propiedades

#### Diagrama de Interacción



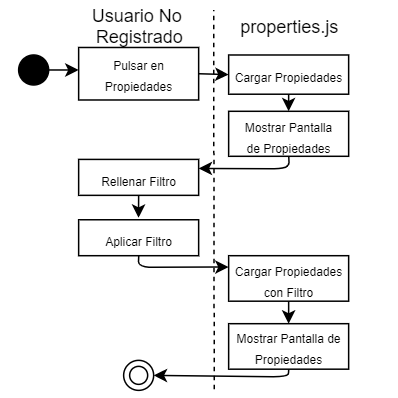
*Figura 99: Filtrar Propiedades – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 100: Filtrar Propiedades – Diagrama de Rutas*

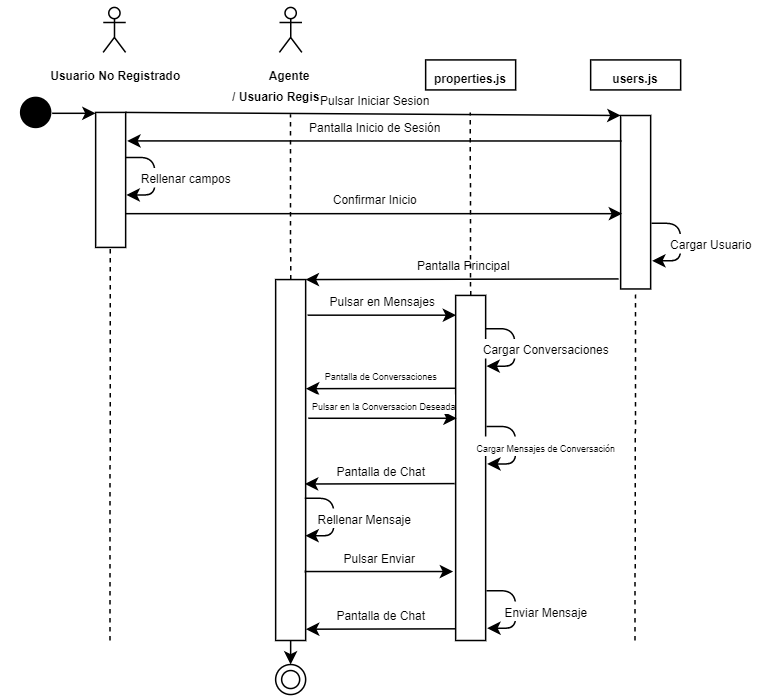
#### Diagrama de Actividad



*Figura 101: Filtrar Propiedades – Diagrama de Actividad*

### Caso N.15 Enviar Mensaje

#### Diagrama de Interacción



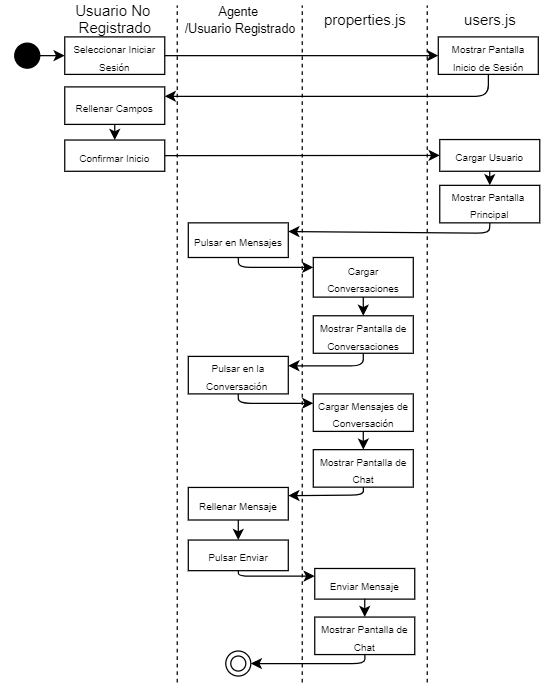
*Figura 102: Enviar Mensaje – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 103: Enviar Mensaje – Diagrama de Rutas*

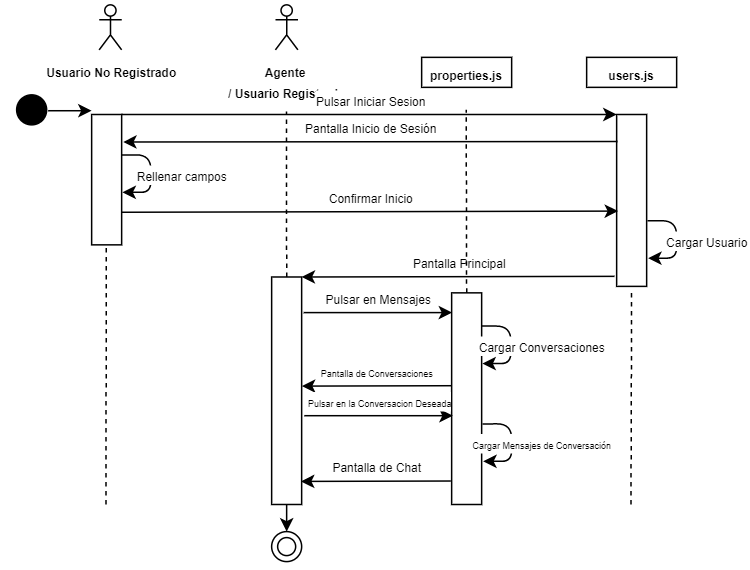
#### Diagrama de Actividad



*Figura 104: Enviar Mensaje – Diagrama de Actividad*

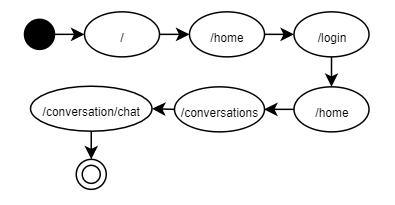
### Caso N.16 Ver Mensajes Recibidos

#### Diagrama de Interacción



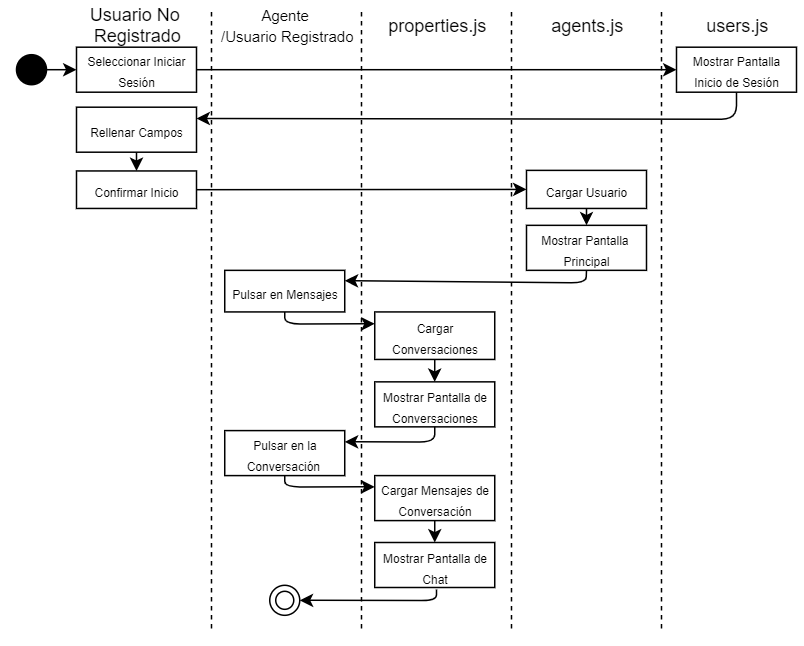
*Figura 105: Ver Mensajes Recibidos – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 106: Ver Mensajes Recibidos – Diagrama de Rutas*

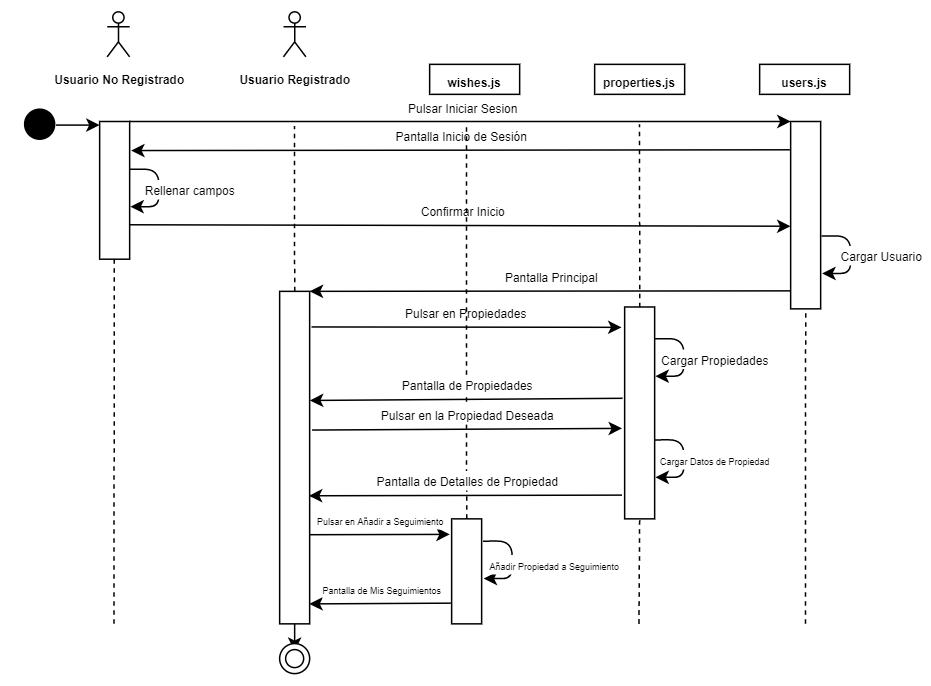
#### Diagrama de Actividad



*Figura 107: Ver Mensajes Recibidos – Diagrama de Actividad*

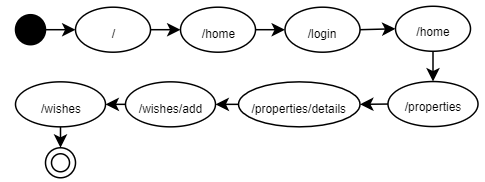
### Caso N.17 Añadir a Seguimientos

#### Diagrama de Interacción



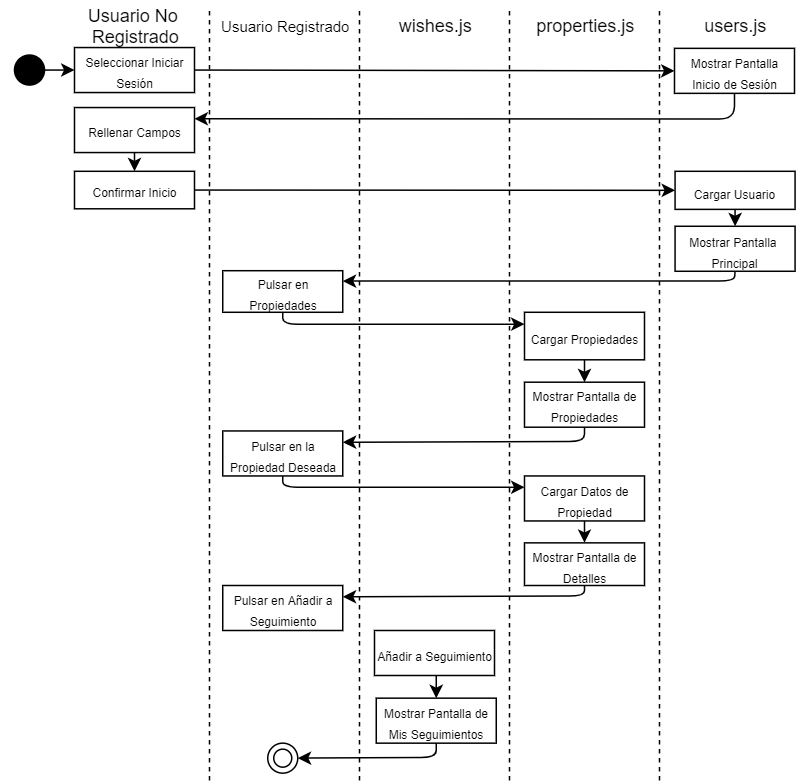
*Figura 108: Añadir a Seguimientos – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 109: Añadir a Seguimientos – Diagrama de Rutas*

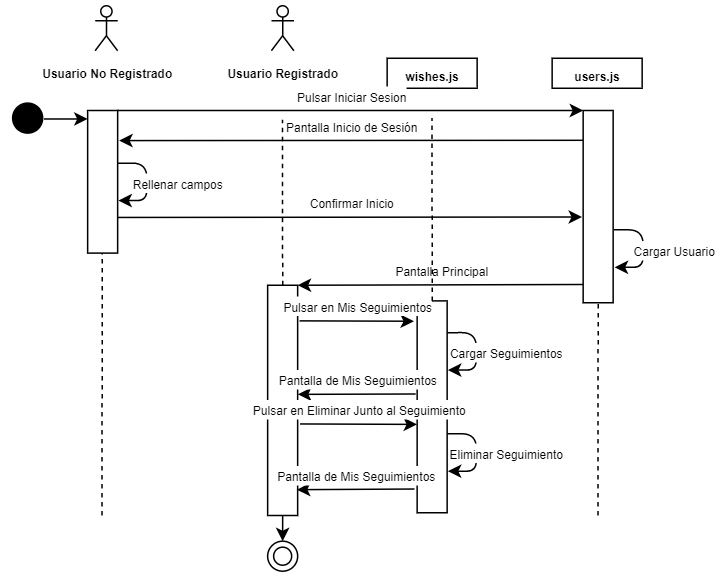
#### Diagrama de Actividad



*Figura 110: Añadir a Seguimientos – Diagrama de Actividad*

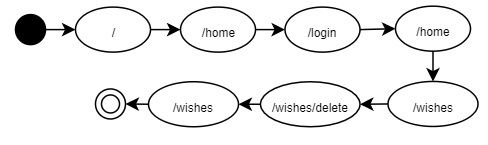
### Caso N.18 Eliminar de Seguimientos

#### Diagrama de Interacción



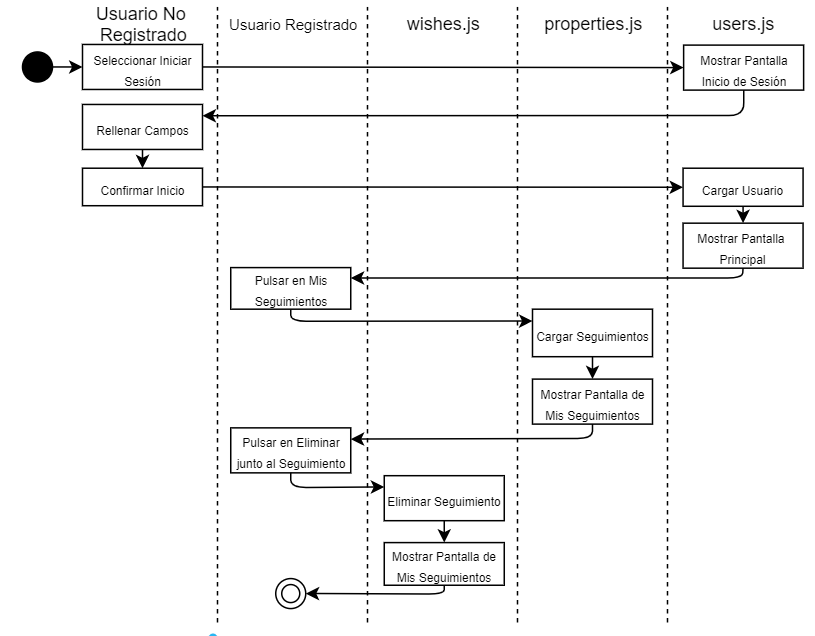
*Figura 111: Eliminar de Seguimientos – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 112: Eliminar de Seguimientos – Diagrama de Rutas*

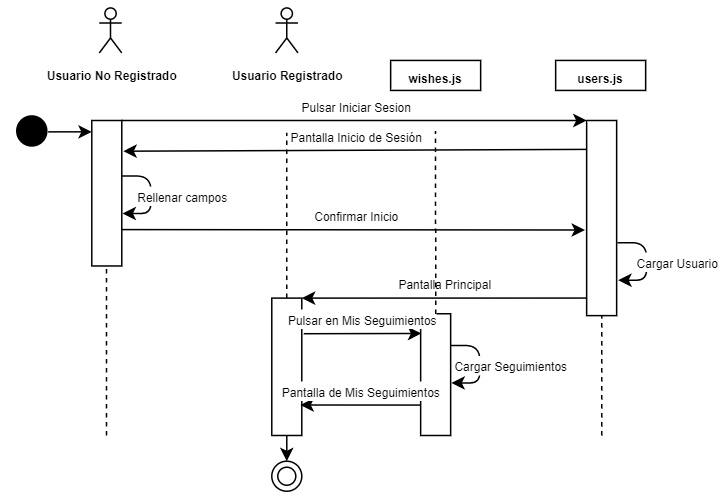
#### Diagrama de Actividad



*Figura 113: Eliminar de Seguimientos – Diagrama de Actividad*

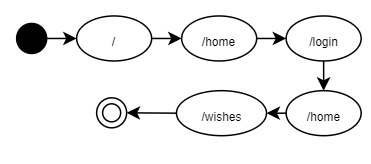
### Caso N.19 Ver Seguimientos

#### Diagrama de Interacción



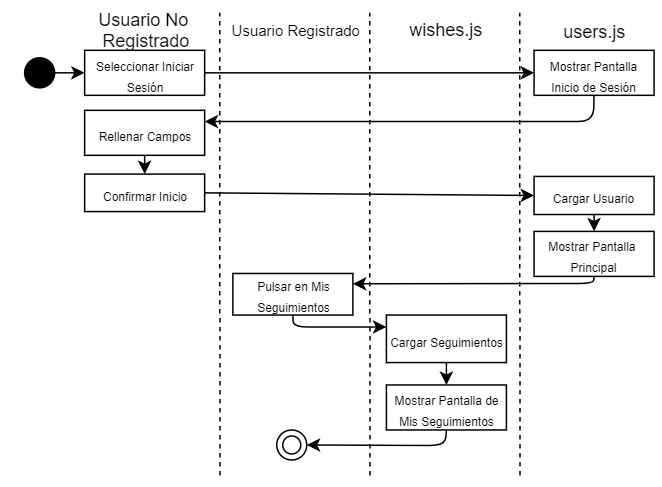
*Figura 114: Ver Seguimientos – Diagrama de Interacción*

#### Diagrama de Rutas



*Figura 115: Ver Seguimientos – Diagrama de Rutas*

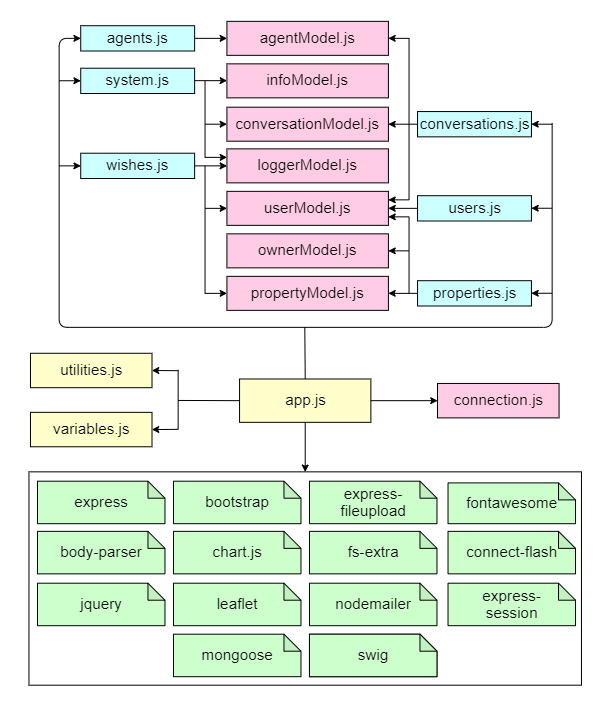
#### Diagrama de Actividad



*Figura 116: Ver Actividad – Diagrama de Actividad*

## Diseño de Módulos

### Diagrama De Módulos



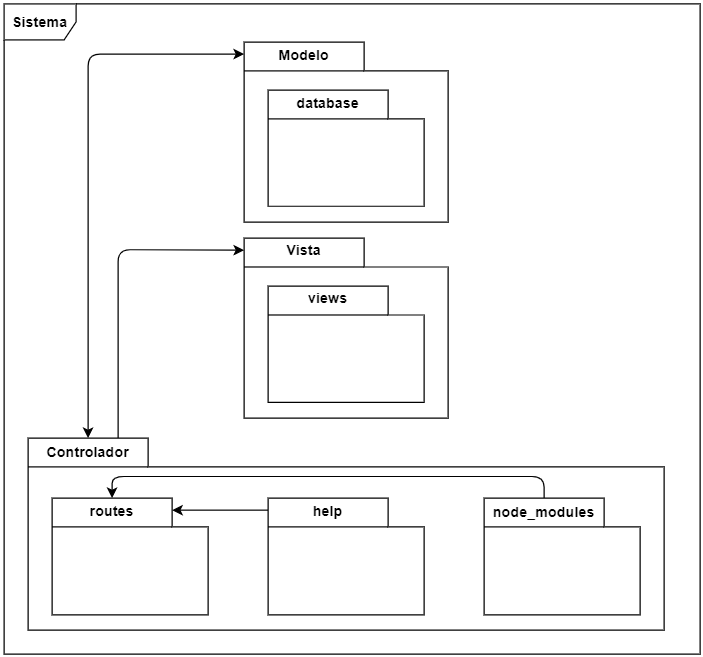
*Figura 117: Diagrama de Módulos*

## Diseño de la Arquitectura de Módulos del Sistema

### Diseño de Módulos del Sistema

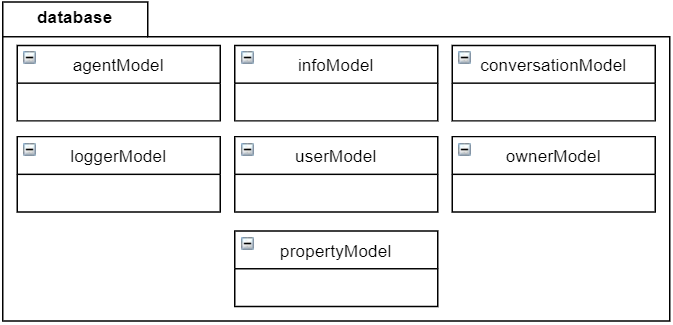
#### Diagrama de Paquetes

Esta sería la estructura de paquetes desde un punto de vista genérico.



*Figura 118: Diagrama de Paquetes*

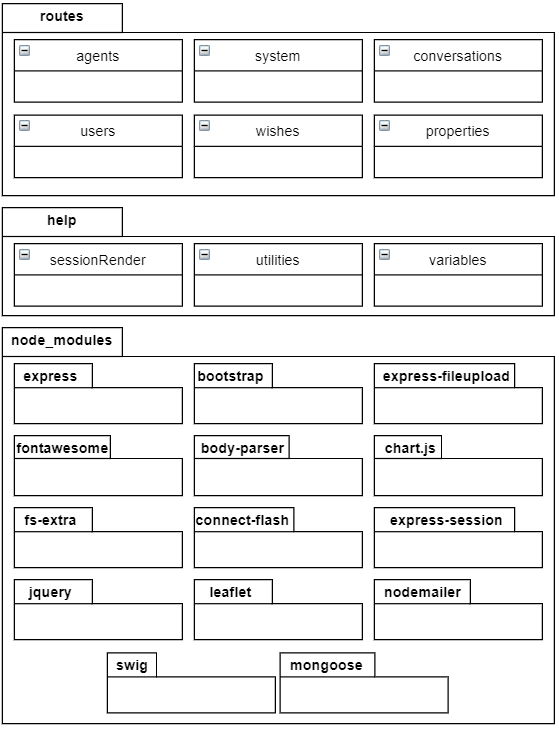
Como podemos observar, la arquitectura del sistema seguiría un carácter y organización similar al **Modelo Vista Controlador** (o MVC). Dentro del modelo podemos observar que existe un único paquete, denominado **database**



*Figura 119: Diagrama de Paquetes – Modelo*

Este paquete contiene todos los módulos que definen los diferentes datos que se añadirán a la Base de Datos, así como los tipos que dichos datos deben tener para ser aceptados por la misma. Mantiene una relación de dos direcciones con el Controlador, pues estos dos paquetes trabajan entre si solicitando y enviándose datos para obtener la información de la Base de Datos.

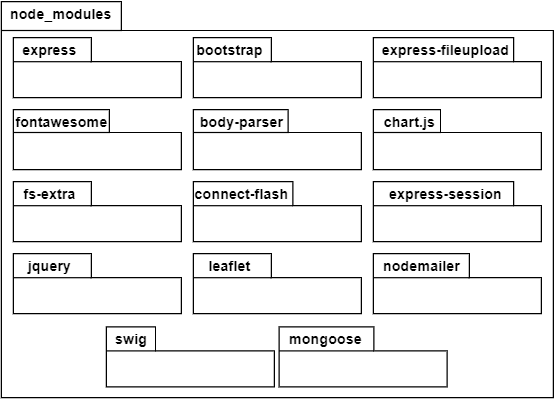
Por otra parte, la Vista se compone únicamente por el paquete **views**, el cual contiene todos los documentos encargados del muestreo de información al usuario. Al igual que con el Modelo, se encuentra conectado con el Controlador, del cual hablaremos más adelante.



*Figura 119: Diagrama de Paquetes – Controlador*

Finalmente, el Controlador se compone por tres paquetes. El primero, denominado **routes**, tiene todos los módulos de lógica del Sistema. Se encarga de trabajar con la información de la Base de Datos y mostrar al usuario los datos necesarios. En este paquete, los módulos se encuentran agrupados por funcionalidad, con el objetivo de asegurar que cada módulo solo trabaja con un tipo de dato en particular.

Por otra parte, el paquete **help** contiene todos los módulos de ayuda y apoyo, como pueden ser funcionalidades comunes genéricas o variables usadas globalmente.

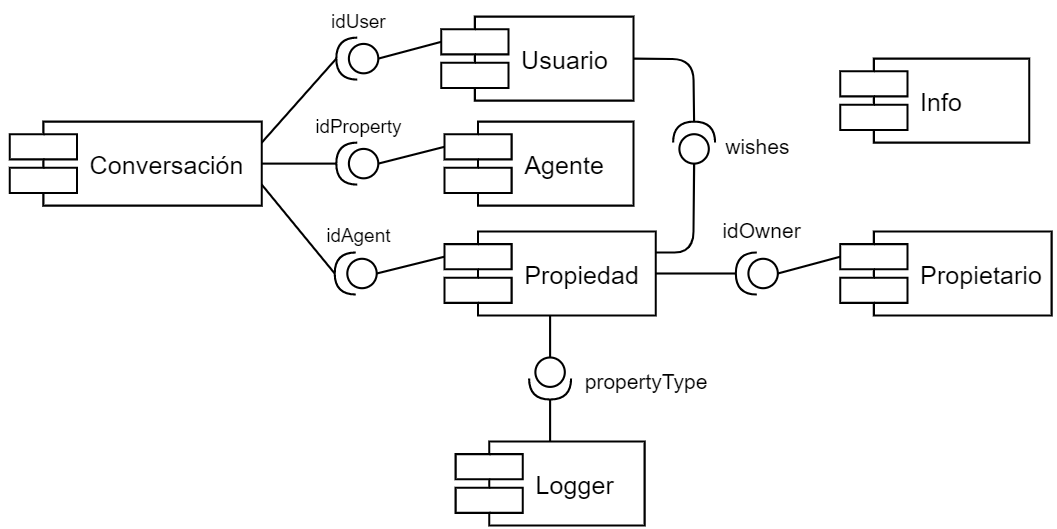


*Figura 120: Diagrama de Paquetes – Controlador II*

En ultimo lugar, el paquete **node\_modules** contiene todos los paquetes de funcionalidad instalados mediante NPM.

### Diseño de Comunicaciones entre Módulos

#### Diagrama de Componentes

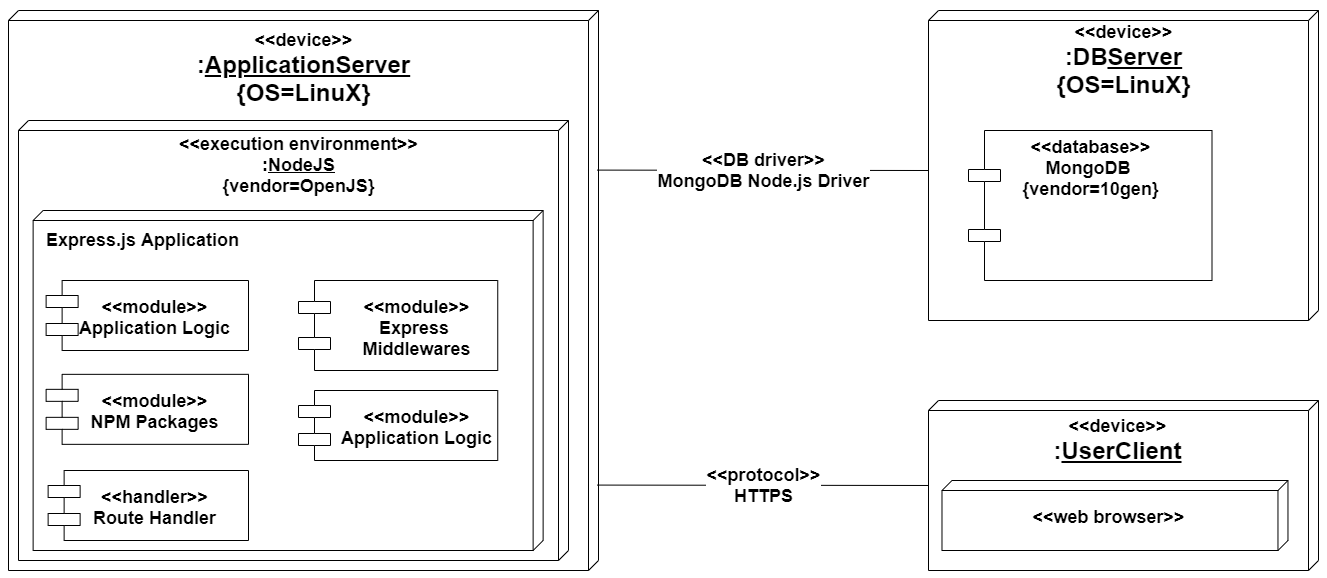


*Figura 121: Diagrama de Componentes*

En este diagrama se pueden observar las diferentes dependencias de los elementos que componen el Sistema. Todos los elementos del sistema corresponden a un documento de información contenido en la Base de Datos. Se puede observar que el único componente que es totalmente independiente de los demás es el componente Info, pues contiene exclusivamente información del sistema editado por el agente. Por otra parte, es posible observar que las conversaciones son los componentes más dependientes de otros, pues se encarga de relacionar un usuario con un agente, mediante una propiedad. Podemos ver también que un usuario se encuentra relacionado con una propiedad mediante las listas de deseos. También, a la hora de obtener datos del Logger (el modulo encargado de obtener y generar estadísticas), podemos ver que almacena el tipo de la propiedad, pues dichas estadísticas dependen, en algunos casos, del tipo.

Finalmente, podemos ver que la propiedad se encuentra relacionada con un componente propietario, el cual no tiene un módulo propio en el sistema. Esto es debido a que, para ahorrar espacio en la base de datos, es posible “reciclar” un propietario, de forma que, si un propietario tiene varias propiedades, estas se encuentren relacionadas con un único componente, y no con varios con idénticas propiedades.

#### Diagrama de Despliegue



*Figura 122: Diagrama de Despliegue*

Podemos observar que el proceso de despliegue se compone por tres elementos principales:

* **Servidor de la Aplicación:** En él, la aplicación se despliega en el servidor en la nube deseado. En su interior se encuentra el entorno de ejecución Node.js. A su vez, en su interior se encuentra nuestra aplicación desarrollada, compuesta por diferentes módulos importantes.
* **Servidor de la Base de Datos:** En este se encuentra contenida la base de datos de tipo MongoDB, la cual contiene toda la información del sistema almacenada (propiedades, usuarios etc.). Este servidor se conecta al servidor de la aplicación por medio del Driver de MongoDB con Node.js.
* **Máquina del Usuario:** Este dispositivo representa la máquina del usuario que desea acceder a la aplicación, para la cual necesita acceder a su buscador web (o web browser) y acceder a la aplicación mediante un protocolo de seguridad.

### Revisión de la Interfaz de Usuario

// TODO

#### Diseño detallado de la UI y Especificación de Datos

#### Navegación Entre Ventanas

#### Mensajes de Error

## Diseño Físico de Datos

### Descripción del SGBD Usado

Como bien hablamos anteriormente, MongoDB es el SGBD (o Sistema de Gestión de Bases de Datos) que se utilizará. A la hora de hablar de él en lo relativo al tratamiento de datos, deben tenerse en cuenta algunos puntos de gran importancia.

#### Copias de Seguridad

Respecto a la realización y mantenimiento de datos en copias de seguridad, tendremos en cuenta que MongoDB ofrece una herramienta, denominada mongodump, la cual nos permite realizar copias de seguridad periódicas, ya sea online o fuera de la línea.

Para realizar y mantener dichas copias de seguridad, la empresa IBM (o International Business Machines Corporation) ofrece una documentación muy completa, con diferentes casos de fallos o errores que se puedan cometer y como afrontarlos. Dicha documentación puede encontrarse en el siguiente enlace.

*https://developer.ibm.com/es/technologies/data-management/tutorials/ba-mongodb-backupscripts/#:~:text=Para%20realizar%20la%20copia%20de,ubicación%20predeterminada%20%2Fbin%2Fdump.*

Mediante esta documentación, es posible configurar un servidor MongoDB para que ejecute scripts periódicos de copia de seguridad, así asegurando la permanencia de los datos.

#### Purgado de Datos

Respecto a la política de purgado de datos, debemos tener en cuenta que, al trabajar con una empresa de pequeño tamaño, es poco probable que alcancemos un tamaño de datos que el sistema no permita.

No obstante, se procurará mantener la mayor cantidad de información en el menor espacio posible, para así minimizar los posibles riesgos.

#### Mantenimiento de Datos

Respecto a la política de limpiado de datos y eliminación de duplicados, el sistema se encargará de controlar que la Base de Datos no contenga datos duplicados ni información innecesaria en el sistema.

#### Restablecimiento de Datos

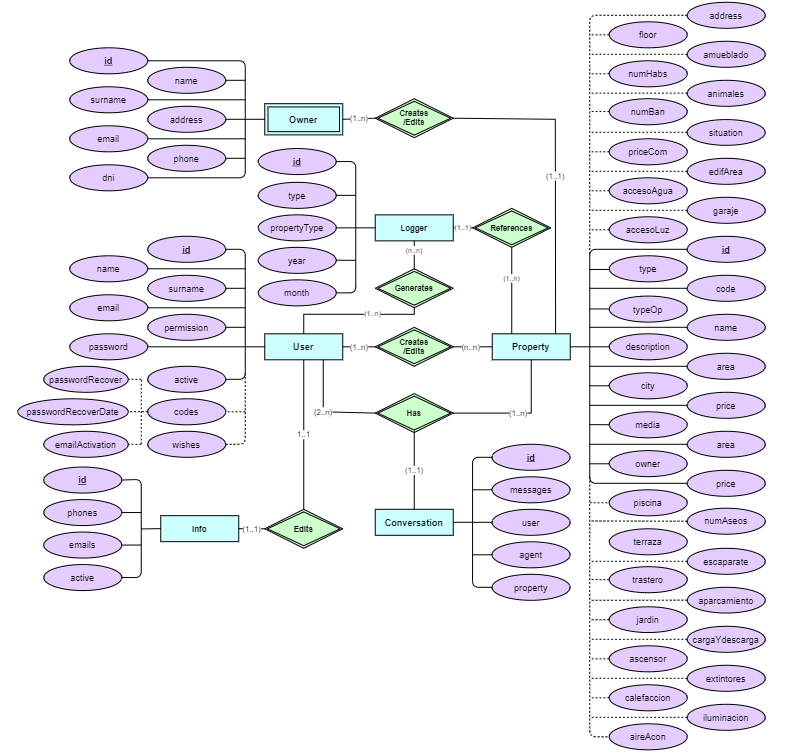
Como bien se indicó con anterioridad, es posible obtener una copia de seguridad periódica de los datos de la Base de Datos, por lo que la recuperación de esta es muy sencilla.

### Integración del SGBD en Nuestro Sistema

Para realizar la integración del SGBD hemos utilizado un paquete de NPM denominado *mongoose.*

*Mongoose* es un paquete NPM que nos permite crear un modelo que contenga un esquema de datos, el cual nos permite declarar qué datos tendrá un elemento en la base de datos, así como su tipo. También nos permite indicar si dicho campo será obligatorio u opcional. Como podemos observar, *mongoose* se encarga de controlar qué datos se intentan introducir en el sistema.

### Diagrama E-R



## Especificación Técnica del Plan de Pruebas

### Pruebas Unitarias

### Pruebas de Integración y del Sistema

### Pruebas de Usabilidad y Accesibilidad

### Pruebas de Accesibilidad

### Pruebas de Rendimiento