

TRABAJO DE FIN DE GRADO

Aplicación Web de Gestión Inmobiliaria / Real Estate Management Application



Carlos gómez colmenero

UO257386 – DNI 53520720L

Tabla de contenido

[1. Capítulo 1. ¿Qué es este trabajo? 3](#_Toc64381574)

[2. Capítulo 2. Planificación del Sistema de Información 3](#_Toc64381575)

[2.1. Inicio del Plan de Sistemas de Información 3](#_Toc64381576)

[2.1.1. Análisis de la Necesidad del PSI 3](#_Toc64381577)

[2.1.2. Identificación del Alcance del PSI 4](#_Toc64381578)

[2.1.3. Determinación de responsables 5](#_Toc64381579)

[2.2. Definición y Organización del PSI 5](#_Toc64381580)

[2.2.1. Especificación del Ámbito y Alcance 5](#_Toc64381581)

[2.2.2. Organización del PSI 7](#_Toc64381582)

[2.3. Estudio de la Información Relevante 8](#_Toc64381583)

[2.3.1. Selección y Análisis de Antecedentes 8](#_Toc64381584)

[3. Capítulo 3. Definición de la Arquitectura Tecnológica 9](#_Toc64381585)

[3.1.1. Identificación de las Necesidades de Infraestructura Tecnológica 9](#_Toc64381586)

[3.1.1.1. Necesidades de Hardware 9](#_Toc64381587)

[3.1.1.2. Necesidades de Despliegue de la Aplicación 9](#_Toc64381588)

[3.1.1.2.1. Amazon Web Services (AWS) 10](#_Toc64381589)

[3.1.1.2.2. Microsoft Azure 11](#_Toc64381590)

[3.1.1.2.3. Google Cloud 11](#_Toc64381591)

[3.1.1.3. Necesidades de Desarrollo 11](#_Toc64381592)

[3.1.1.3.1. SpringBoot 11](#_Toc64381593)

[3.1.1.3.2. NodeJS 12](#_Toc64381594)

[3.1.1.4. Necesidades de Almacenamiento 12](#_Toc64381595)

[3.1.1.4.1. MySQL 12](#_Toc64381596)

[3.1.1.4.2. MongoDB 12](#_Toc64381597)

[3.1.2. Selección de la Arquitectura Tecnológica 13](#_Toc64381598)

[3.1.2.1. Selección de Despliegue de la Aplicación 13](#_Toc64381599)

[3.1.2.2. Selección de Desarrollo 14](#_Toc64381600)

[3.1.2.3. Selección de Almacenamiento 14](#_Toc64381601)

[4. Capítulo 3. Estudio de Viabilidad del Sistema 15](#_Toc64381602)

[5. Capítulo 5. Planificación y Gestión del TFG 20](#_Toc64381603)

[5.1. Planificación del Proyecto 20](#_Toc64381604)

[5.1.1. Identificación de Interesados 20](#_Toc64381605)

[5.1.2. OBS y PBS 20](#_Toc64381606)

[5.1.3. Planificación Inicial. WBS 20](#_Toc64381607)

[5.1.4. Riesgos 20](#_Toc64381608)

[5.1.4.1. Plan de Gestión de Riesgos 20](#_Toc64381609)

[5.1.4.2. Identificación de Riesgos 20](#_Toc64381610)

[5.1.4.3. Registro de Riesgos 20](#_Toc64381611)

[5.1.5. Presupuesto Inicial 20](#_Toc64381612)

[5.1.5.1. Presupuesto de Costes 20](#_Toc64381613)

[5.1.5.2. Presupuesto de Cliente 20](#_Toc64381614)

# 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 cliente y un posible comprador. Esta aplicación se encuentra 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 Gestion 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

Se desea crear 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 obtener una localización exacta de la propiedad. 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 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 no posee ningún tipo de soporte web que le permita realizar estos procesos mediante la web. Tras una reunión con este, se ha llegado a 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 proyecto para desarrollar se dividirá en diferentes fases:

1. **Gestión de cuentas de usuario/administrador**

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-usuario capaz de crear cuentas de todo tipo. De este modo, el super-usuario podrá crear una cuenta de administrador si es necesario. Finalmente, existirán cuentas de administrador, las cuales podrán gestionar la totalidad de la página web.

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

**Objetivos de la fase:**

* **Crear una jerarquía de perfiles.**
* **Permitir al super-usuario manejar cuentas tanto de administrador como de usuario.**
* **Permitir al administrador 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 vendedor-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 deseos, donde guardará todos aquellos inmuebles en los que tenga interés.

**Objetivos de la fase:**

* **Sistema de seguimiento de precios.**
* **Lista de deseos.**

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:

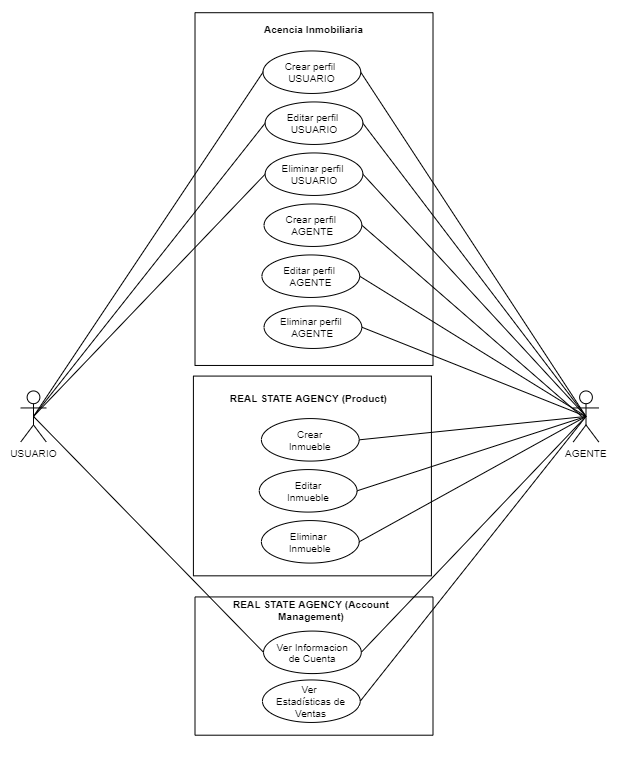
|  |  |
| --- | --- |
| **USUARIO** | FUNCION/ES |
| **EQUIPO DE SUPERVISION** | |
| **Dirección de la Escuela** | Supervisión de las fases del proyecto. |
| **Tutor del Proyecto** | Supervisión de que se cumplen los objetivos declarados por el alumno. |
| **Cliente** | Debe asegurar que se cumplan sus requisitos.  Supervisar que se consiguen los objetivos deseados. |
| **EQUIPO DE DESARROLLO Y DISEÑO** | |
| **Alumno** | Creación de los módulos software.  Elaboración de la documentación.  Diseño 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 anterior 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 1.
* 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: Permisos de Perfiles*

# Capítulo 3. Definición de la Arquitectura Tecnológica

### Identificación de las Necesidades de Infraestructura Tecnológica

#### Necesidades de Hardware

Respecto a las necesidades de Hardware, 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 Despliegue de la Aplicación

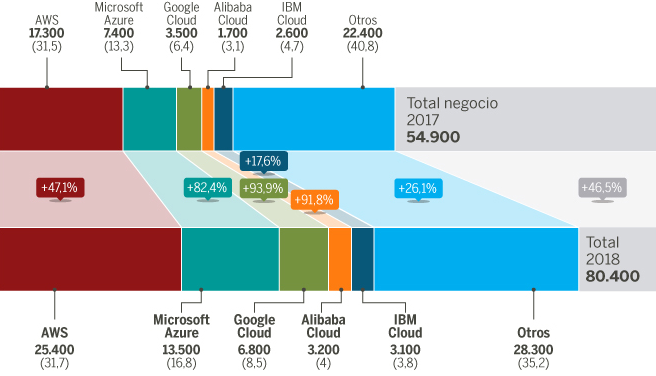
Para poder llevar a cabo el despliegue de la aplicacion en la web, así como de establecer la dirección publica a la cual se deberán conectar los usuarios, es necesario tener acceso a unos servicios de computación en la web. La computación en la web es un paradigma que nos permite obtener diversos servicios 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 nbe es la mejor opción, pues permite al cliente liberarse de la carga de mantener el sitio web, 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 mas importantes es que presenta mas de 25 tipos de servidores que se encuentran específicamente diseñados para diferentes tareas, asegurando siempre que el sistema web se ejecute de la manera mas eficiente posible.

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

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

##### Google Cloud

Otra de las grandes, en este caso acompañado por Google. El mismo Google declara que su mayor ventaja es el precio, pues es notablemente mas 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 Desarrollo

Ahora debemos cual será la tecnología que utilizaremos. 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 a tener en cuenta, pues son los lenguajes más utilizados en este tipo de aplicaciones.

##### SpringBoot

SpringBoot es una extensión preconfigurada del conocido framework de desarrollo Spring. 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.

##### 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 maquinas sin afectar a la seguridad.

#### 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 e formato BSON (similar a JSON). 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 numero 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) seria 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 mas optima por las siguientes razones:

* El lenguaje JavaScript hace la aplicación muy dinámica, permitiendo modificarla o ampliarla con nueva funcionalidad cuando sea necesario.
* 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 esta 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.

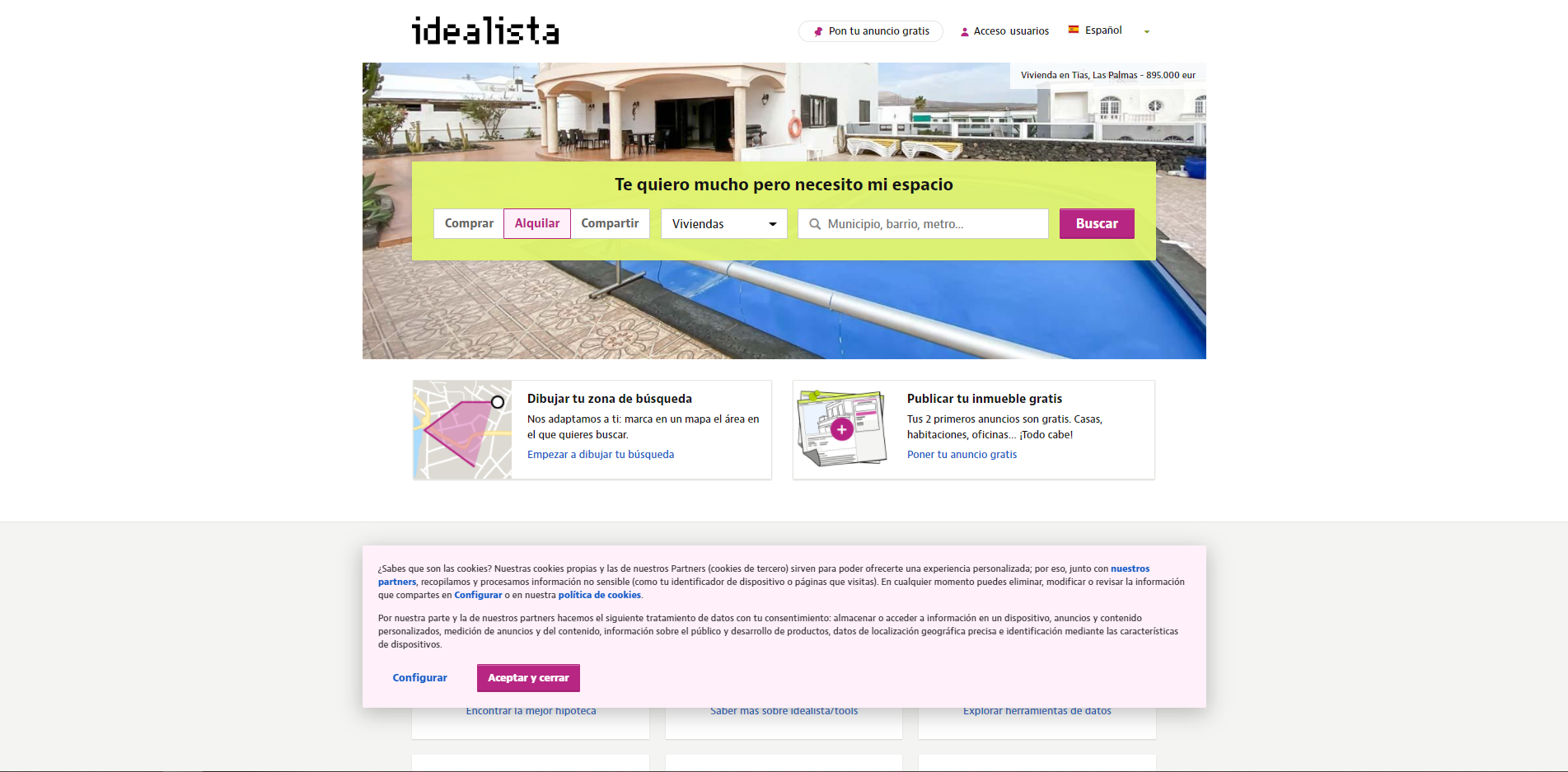
También, MongoDB se integra muy bien con nuestra anterior elección, NodeJS, por lo que será sencillo intefrarlos juntos y trabajar con ambos.

# 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*

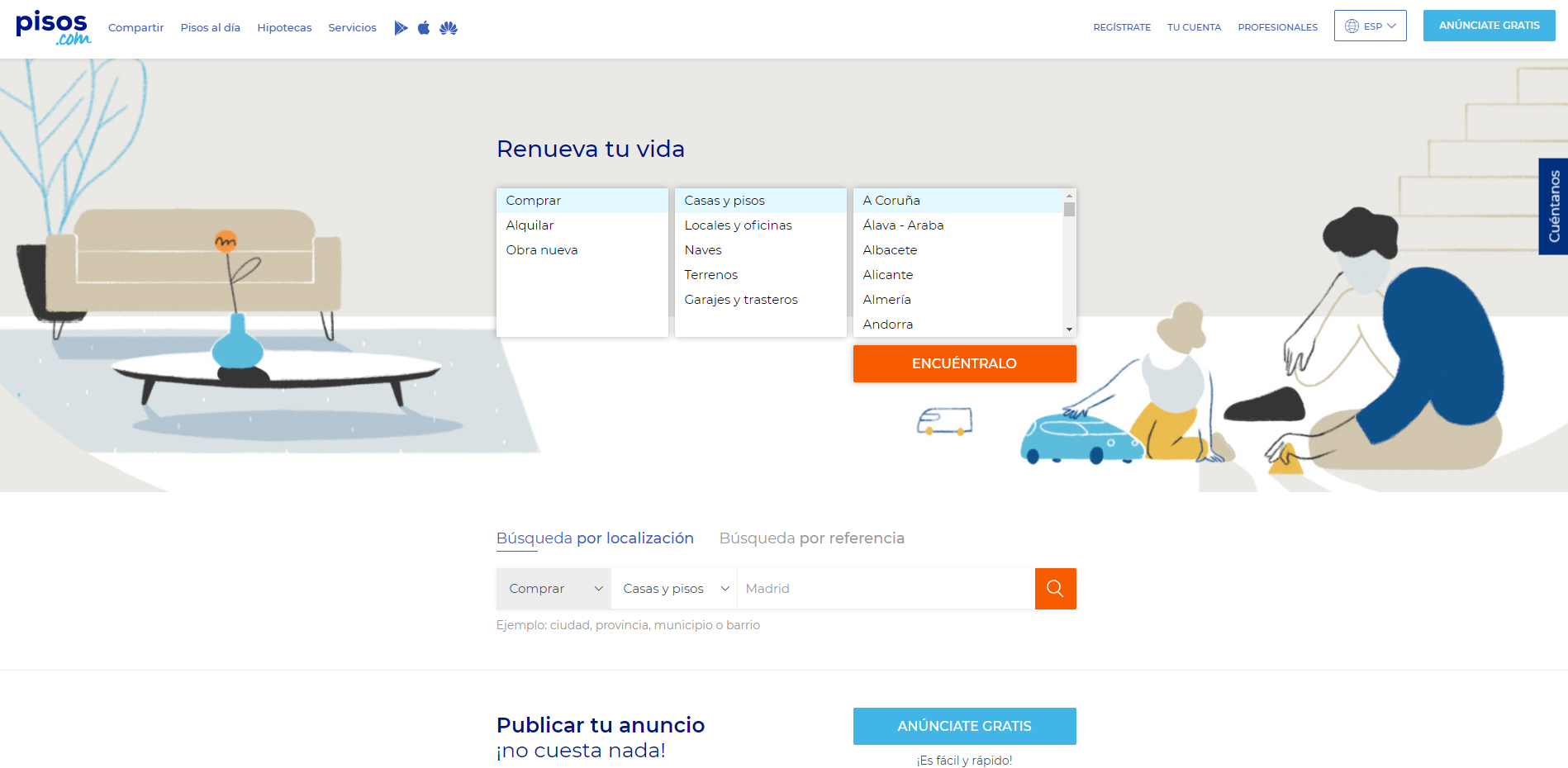
Una vez accedemos a la presentación del inmueble, obtenemos una visual de toda la información general del inmueble, seguido de varias opciones interesantes como contactar con el vendedor, guardar en favoritos, ver en el mapa y compartir.

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.

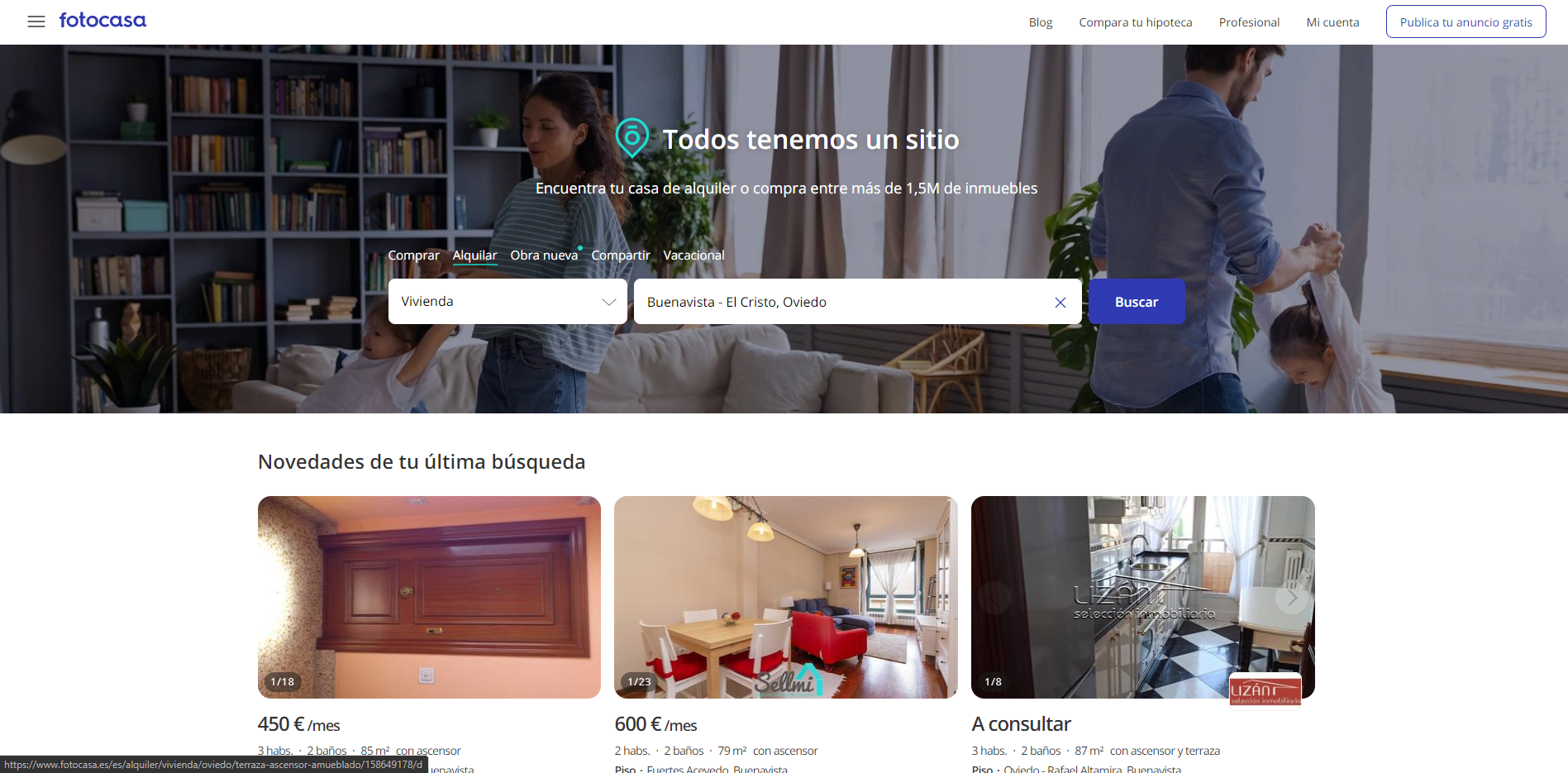
La venta del inmueble es mucho más rápida que en los demás portales. Sencillamente se ofrecen tres pestañas:

* **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.**  **Lista de deseos.** | **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 capitulo se dejara 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 nuestro 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, debemos tener en cuenta que para poder realizar algunas tareas como el despliegue de la aplicación necesitaríamos de una financiación económica de la cual no disponemos. Debido a ello, realizaremos el desarrollo de la aplicación y lo ejecutaremos en un servidor local montado en la misma maquina 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 mas 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 mas genéricos hasta los mas 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 distintivo del requisito. Deberá resumir su funcionalidad en pocas palabras.
* **Descripción:** Texto explicando la funcionalidad del requisito de manera extendida. Deberá explicar de forma clara y concisa qué condiciones se están imponiendo.

#### Requisitos Funcionales

|  |  |  |
| --- | --- | --- |
| **GESTION DE PERFILES** | | |
| **Indice** | **Nombre** | **Descripcion** |
| 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 otra del mismo tipo. |
| 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.1.2.4 | Datos | Se deberá introducir una contraseña. |
| 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. |
| RF.1.3.1 | Restricciones | Una cuenta de tipo usuario solo podrá borrarse a si misma. |

#### Requisitos No Funcionales