



PORTAL WEB INMOBILIARIO

Memoria del Proyecto Final de Carrera
de Ingeniería en Informática
realizado por José Ramón Sáez Zamorano
y dirigido por Joan Sorribes Gomis
Bellaterra, 15 de Junio de 2007

El sotasingnat,

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en

I per tal que consti firma la present.

Signat:

Bellaterra,de.....de 200.....

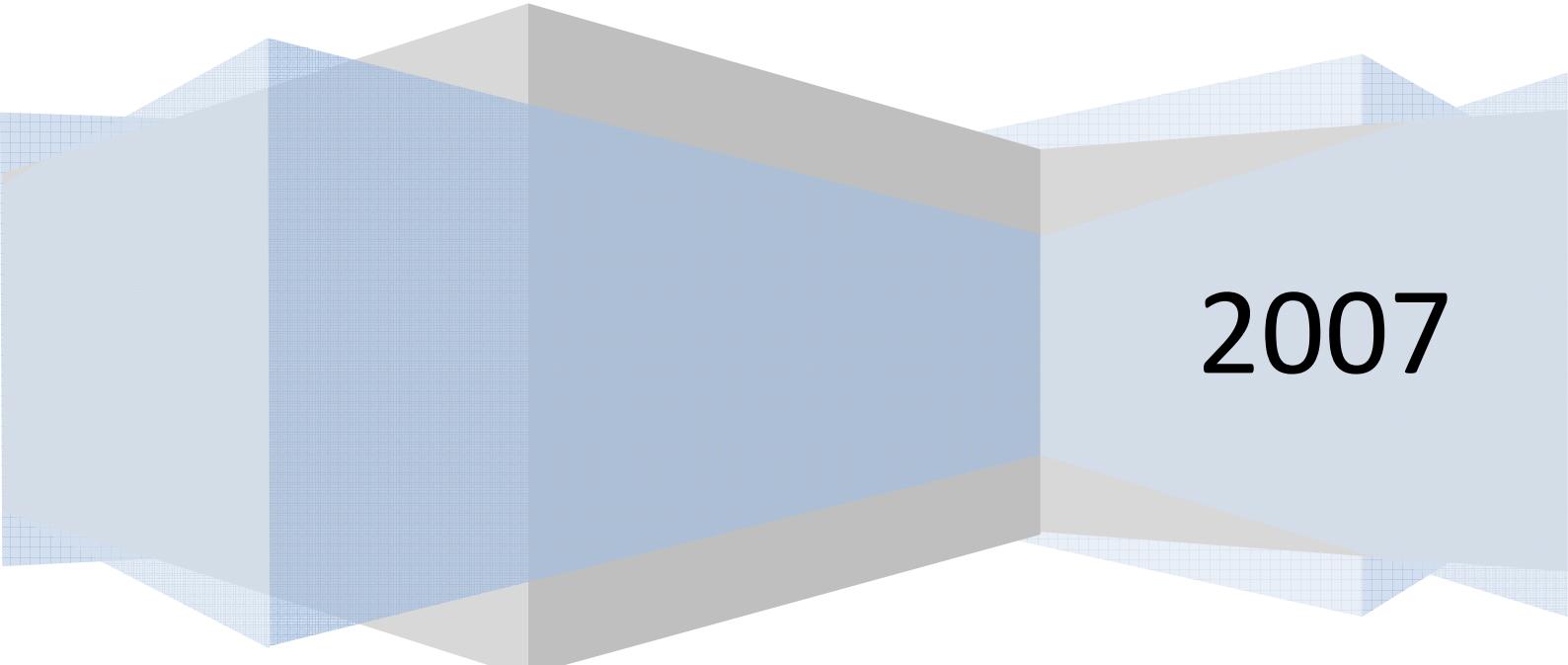
Escola Tècnica Superior d'Enginyeries

PORTAL WEB INMOBILIARIO

Memoria del Proyecto Final de Carrera

Realizado por José Ramón Sáez Zamorano

y dirigido por Joan Sorribes Gomis



2007

Agradezco a todos mis compañeros y a todas aquellas personas que me han apoyado durante la carrera, de manera muy especial a M.C.

ÍNDICE

- 1. Prólogo**
- 2. Introducción**
 - 2.1. Presentación**
 - 2.2. Objetivos**
 - 2.2.1. Objetivos teóricos**
 - 2.2.2. Objetivos en la elaboración del proyecto**
 - 2.2.3. Objetivos del proyectista**
 - 2.3. Estructura de la memoria**
- 3. Análisis**
 - 3.1. Análisis del problema**
 - 3.1.1. Estudio de la situación actual**
 - 3.1.1.1. Estructura y organización de la información**
 - 3.1.1.2. Estilo y diseño de la web**
 - 3.1.1.3. Conclusiones**
 - 3.1.2. Tipos de usuario**
 - 3.1.3. Necesidades**
 - 3.1.3.1. Clientes**
 - 3.1.3.2. Administración**
 - 3.1.3.3. Comerciales**
 - 3.2. Objetivos del sistema**
 - 3.2.1. Objetivos de diseño**
 - 3.2.2. Objetivos generales**
 - 3.3. Requerimientos**
 - 3.3.1. Requerimientos funcionales**
 - 3.3.1.1. Requisitos funcionales del gestor de inmuebles**
 - 3.3.1.2. Requisitos funcionales de la cartera de clientes**
 - 3.3.1.3. Requisitos funcionales del gestor de empleados**
 - 3.3.1.4. Requisitos funcionales del gestor de alertas**
 - 3.3.1.5. Requisitos funcionales del gestor de informes**
 - 3.3.1.6. Requisitos funcionales del gestor de contratos**
 - 3.3.1.7. Requisitos funcionales del módulo de estadísticas**
 - 3.3.1.8. Requisitos funcionales del gestor de visitas**

- 3.3.1.9. Requisitos funcionales del módulo localizador
 - 3.3.1.10. Requisitos funcionales del módulo de cálculos hipotecarios
 - 3.3.1.11. Requisitos funcionales del módulo de comunicación
 - 3.3.1.12. Requisitos funcionales del módulo buscador
 - 3.3.1.13. Requisitos funcionales del gestor de usuarios
 - 3.3.2. Requerimientos no funcionales
 - 3.3.2.1. Requerimientos de rendimiento
 - 3.3.2.1.1. Requerimientos estáticos
 - 3.3.2.1.2. Requerimientos dinámicos
 - 3.3.2.2. Requerimientos de las interfaces externas
 - 3.3.2.3. Requerimientos de calidad
 - 3.4. Restricciones de diseño
 - 3.4.1. Seguridad
 - 3.4.2. Fiabilidad
4. Diseño de la nueva web
- 4.1. Diseño de la base de datos
 - 4.1.1. Descripción de las tablas y relaciones del gestor de clientes
 - 4.1.2. Descripción de las tablas y relaciones del gestor de empleados
 - 4.1.3. Descripción de las tablas y relaciones del gestor de inmuebles
 - 4.1.4. Descripción de las tablas y relaciones del gestor de alertas
 - 4.1.5. Descripción de las tablas y relaciones del gestor de contratos
 - 4.2. Diseño de la interfaz de usuario
 - 4.2.1. Usabilidad
 - 4.2.2. Carga de información
 - 4.2.3. Acciones
 - 4.2.4. Colores en el web
 - 4.2.5. Interfaz de usuario basadas en web
 - 4.2.6. Diseño de la interfaz de usuario
 - 4.2.7. Estructura web

5. Codificación

5.1. Lenguajes

5.1.1. HTML

5.1.2. Javascript

5.1.3. CSS

5.1.4. PHP

5.1.4.1. Alternativa a PHP

5.1.4.2. Elección del lenguaje

5.1.5. Mysql

5.1.6. Flash

5.2. Creación de PDF (FPDF)

5.3. Creación de documentos RTF

5.4. Creación de estadísticas dinámicas (Phpcharts)

5.5. Creación del calendario (Vcalendar)

5.6. Creación de mapas de localización (API de Google Maps)

5.7. Herramientas de diseño de la base de datos(dbDesigner)

5.8. Herramientas de diseño de las imágenes (Adobe Photoshop)

5.9. Servidor

5.9.1. Apache

5.9.2. Servidor de pruebas

5.9.3. Servidor final

6. Pruebas del sistema

6.1. Tipos de prueba

6.2. Pruebas de unidad

6.3. Pruebas de integración

6.4. Pruebas de validación

6.5. Pruebas de sistema

7. Conclusiones

7.1. Objetivos conseguidos

7.2. Objetivos no conseguidos

7.3. Posibles ampliaciones y modificaciones

7.4. Conclusiones finales

8. Apéndices

8.1. Guía de usuario

8.2. Valoración económica

9. Bibliografía

10. Índice de figuras

11. Índice de tablas

1. PROLOGO

El proyecto que aquí se describe nace como la demanda de una empresa del sector inmobiliario de cara a la ampliación y actualización de la web que actualmente dispone.

Tras la correspondiente petición para el desarrollo de dicho proyecto y la aceptación de este como proyecto de final de carrera, se ha desarrollado un proyecto destinado a la implementación de un aplicativo web que permita y simplifique las gestiones inmobiliarias que este tipo de empresas han de llevar a cabo.

Pese a que los objetivos del desarrollo de este proyecto no eran comerciales, sino los de satisfacer una demanda real, finalmente la empresa ha estado interesada en la compra del aplicativo.

2. INTRODUCCIÓN

2.1. PRESENTACIÓN

En los últimos años el sector inmobiliario ha sido uno de los sectores con más crecimiento y más beneficios del país. En la actualidad el estancamiento de este sector lleva a las empresas inmobiliarias a buscar posibles inversores y clientes en diferentes ámbitos, en los cuales antes ni tan siquiera tenían presencia. Así en el afán de llegar a tantos clientes sea posible, muchas pequeñas empresas inmobiliarias han decidido crear o modernizar su imagen en la red. Dentro de este ámbito se enmarca el proyecto que aquí se describe.

Este proyecto surge de la necesidad de modificación y actualización de la página web de una pequeña empresa del sector inmobiliario situada en Sabadell (Barcelona) llamada “Fincas Castillo”. En la actualidad la web que poseen cumple un papel meramente informativo, no cumplen con unos términos de funcionalidad mínimos y comportan unas tareas de mantenimiento costosas para los responsables, puesto que carece de zona de administración.

A sabiendas que estos hechos pueden provocar un uso poco frecuente de la página y en vista de la necesidad de captar nuevos clientes, la empresa decide actualizar su web y ampliar todas las funcionalidades, así como adquirir una nueva interfaz de usuario más corporativa y que ofrezca una buena imagen de esta empresa en la red.

Tras la demanda de estas necesidades por parte de la empresa y tras la aceptación del proyecto como proyecto de final de carrera, se ha procedido a implementar un aplicativo capaz de gestionar inmuebles, clientes, empleados, contratos, estadísticas, etc. cumpliendo con las necesidades y requerimientos estipulados por la empresa e intentando satisfacer en todo momento a la empresa, pese a que el proyecto no estaba pensado con un fin comercial.

2.2. OBJETIVOS

El proyecto que aquí nos ocupa tiene como principal objetivo la realización de una nueva página web para la empresa del sector inmobiliario “Fincas Castillo” teniendo en cuenta las necesidades y requerimientos de la empresa a la hora de desarrollar dicho aplicativo.

2.2.1. OBJETIVOS TEÓRICOS

Para llevar a cabo este proyecto se realizarán dos zonas principales:

- **ZONA DE USUARIOS:** Cualquier usuario podrá visitar la web y, informarse sobre la empresa, su localización, la oferta de inmuebles de ésta, contactar con la empresa etc. todo mediante un área común para todos los usuarios que configurará la parte pública de la web.
- **ZONA DE ADMINISTRACIÓN:** Desde la zona de administración se podrá gestionar todos los datos relacionados con la empresa, ya sean inmuebles, empleados, contratos, etc. Esta zona estará reservada para la administración de la empresa y por tanto no será accesible por cualquier usuario.

Además de la implementación de estas dos áreas, se implementara una interfaz de usuario mejorada que satisfaga las necesidades de la empresa solicitante.

2.2.2. OBJETIVOS EN LA ELABORACIÓN DEL PROYECTO

Como objetivos básicos del desarrollo de este proyecto encontramos:

- Análisis de la situación actual.
- Diseño de las nuevas secciones, nuevas funcionalidades y la interfaz de usuario.
- Implementación del aplicativo.
- Implementación y realización de las pruebas necesarias para asegurar el correcto funcionamiento del sistema
- Elaboración de una documentación del trabajo realizado.

Puesto que el tiempo de realización es limitado y se trata de un solo proyectista para un volumen importante, se dará prioridad a las fases de análisis y diseño, ya que estas permitirán continuar el trabajo de implementación en caso de que éste no fuese finalizado.

2.2.3. OBJETIVOS DEL PROYECTISTA

Más allá de los objetivos expuestos anteriormente, se pretende que el proyectista adquiera experiencia en el ámbito del análisis, diseño y desarrollo de un software que nace de una petición real.

Otro de los objetivos es que el proyectista vive en primera persona los problemas, retrasos y errores que pueden surgir durante un proyecto de desarrollo de software y adquiera las competencias necesarias para solventar dichos problemas y satisfacer a la empresa.

2.3. ESTRUCTURA DE LA MEMORIA

La memoria que aquí nos ocupa contiene toda la información necesaria para el desarrollo de este proyecto, además de unos anexos con la guía de usuario y la valoración económica del trabajo realizado, se incluye un video tutorial para la correcta utilización del aplicativo.

La estructura de la memoria se divide en siete capítulos acompañados por dos anexos y la bibliografía utilizada, los índices de las figuras y tablas que completan esta documentación.

El primer capítulo no es propiamente un capítulo en sí, sino que se trata de un prologo para enmarcar el nacimiento de este proyecto como tal.

El segundo capítulo corresponde a la introducción, en la que se hace una presentación del proyecto que se desarrollará, los objetivos de que se pretenden lograr mediante el mismo divididos según diferentes puntos de vista. En los siguientes capítulos se describen las diferentes etapas en las que se divide el proceso de elaboración del software: análisis, diseño, codificación y pruebas.

El tercer capítulo corresponde al análisis llevado a cabo para determinar la situación actual y las necesidades que de esta situación se derivan. Tras las entrevistas previas con los diferentes tipos de usuario realizadas al comienzo del proyecto, se establecerán todos los requisitos que el aplicativo implementado deberá satisfacer. Estos requerimientos serán clasificados según su naturaleza y documentados explícitamente.

A continuación encontramos un capítulo destinado a documentar la fase de diseño, haciendo especial énfasis en el diseño de la base de datos que hay detrás de la aplicación, así como en el diseño de una nueva interfaz de usuario.

El quinto capítulo está dedicado a la fase de codificación o implementación del aplicativo. En este capítulo se repasan los lenguajes utilizados, comentando la razón de su elección y las características y el uso que le daremos. Además de las funcionalidades logradas gracias a estos lenguajes se comentaran las clases complementarias y herramientas utilizadas durante la implementación.

La última fase de del proyecto, la fase de pruebas, constituye el capítulo seis, en el cual se comentarán los diferentes tipos de pruebas existentes, como y donde se han utilizado, así como los errores surgidos durante esta fase.

Finalmente encontramos un capítulo dedicado a las conclusiones donde encontraremos los objetivos cumplidos y los no cumplidos, las posibles ampliaciones o modificaciones que se podrían implementar en nuestro aplicativo, así como unas conclusiones finales valorando el trabajo realizado.

La memoria se completa con tres apéndices el primero de los cuales constituye una guía de usuario de la aplicación, en el segundo se realiza una valoración económica del trabajo realizado y en el último se muestra las bases de datos de la sección “Visitas”.

3. ANÁLISIS

3.1. ANÁLISIS DEL PROBLEMA

3.1.1. ESTUDIO DE LA SITUACIÓN ACTUAL

Actualmente la inmobiliaria dispone de una web de carácter informativo, en ella se puede encontrar información sobre todos los inmuebles que esta empresa tiene a la venta, así como ver las fotos disponibles de dichos inmuebles. Toda esta información se muestra estáticamente, de forma que cada vez que se tiene que actualizar la información se debe recurrir a la edición de la página correspondiente.

La web no dispone de ningún tipo de buscador, la información acerca de los inmuebles está estructurada en función de la localidad y la zona (barrio) de estos, se puede acceder a ella a través de enlaces situados en la página principal. Todo bajo una interfaz gráfica simplista y poco corporativa.

3.1.1.1. ESTRUCTURA Y ORGANIZACIÓN DE LA INFORMACIÓN

La información contenida en la web actual de *Fincas Castillo* está distribuida bajo una estructura y una organización parcialmente definida, en la cual prácticamente en la totalidad de las páginas se muestra el mismo tipo de letra (letra por defecto del explorador) y el mismo formato para los elementos similares. Pese a ello resulta difícil acceder de forma rápida a la información deseada, puesto que hay gran cantidad de información sobre inmuebles y no se puede realizar una búsqueda eficiente sobre ella.

En la página principal encontramos un conjunto de frames, el superior de carácter informativo, donde encontramos los datos de contacto de la empresa, el izquierdo donde se encuentran

todos los enlaces a las páginas que contienen la información sobre los inmuebles disponibles, y por último el central en el cual se mostrará el contenido de los enlaces del frame izquierdo y que inicialmente nos muestra los inmuebles *“destacados”*

A continuación se muestra un diagrama con la estructura de la web actual.

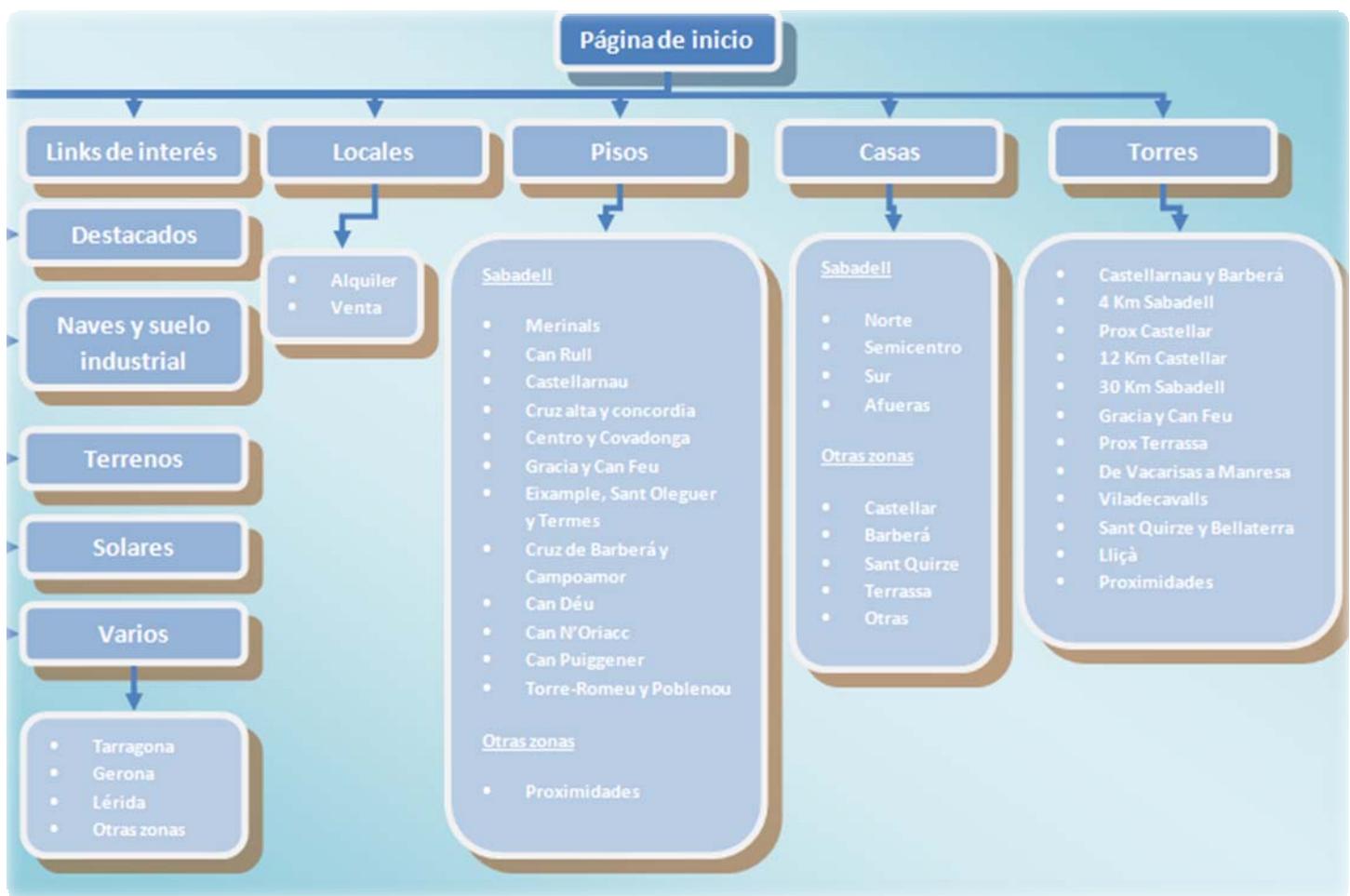


Figura: 1. Estructura de la web actual de Fincas Castillo

Dentro de cada sección encontramos la información de los inmuebles que pertenecen a cada zona ordenados por precio, esta información consta básicamente de una descripción del inmueble y un link a una página donde encontraremos las fotos de dicho inmueble.

Todas las secciones son esencialmente iguales a excepción de la sección de “*links de interés*” donde encontramos una relación de enlaces a otras páginas donde podemos realizar cálculos hipotecarios, ver las notas del registro de propiedad, etc.

Puesto que toda la información es estática, podemos suponer el trabajo que ha de desempeñar el encargado de actualizar la información de la web cada vez que se tiene que insertar un nuevo inmueble: editar a nivel de código HTML la página donde se verá el inmueble, crear una nueva página para colocar las fotos, editar el código de esta nueva página, si se trata de un inmueble destacado editar también la página de destacados, etc. Claramente esto supone una pérdida importante de tiempo debido al volumen de inmuebles que este tipo de empresas puede llegar a tratar.

3.1.1.2. ESTILO Y DISEÑO DE LA WEB

La web actual de la empresa cumple el objetivo principal para el cual fue diseñada: informar a los usuarios o futuros clientes sobre la oferta de inmuebles de la cual dispone. Pese a que este objetivo se logra, no se alcanza eficientemente. Puesto que no se puede realizar búsquedas con criterios diferentes a los pre-establecidos, lo que supone una pérdida de eficiencia y velocidad en la adquisición de información por parte de los usuarios.

La navegación a través de la web resulta bastante sencilla e intuitiva gracias a que el menú está siempre presente en el frame izquierdo, aún y así, algunas de las páginas muestran demasiada información dificultando encontrar información específica en ellas. Por otro lado la falta de una página principal donde encontrar una relación de los contenidos de la web hace que el usuario se desoriente.

El diseño de la web es demasiado monótono, produciendo al usuario una sensación de incomodidad al navegar. La falta de imágenes, así como el estilo utilizado para mostrar los

datos y los enlaces hacen que el usuario caiga en una sensación de monotonía y se canse más rápidamente de navegar por dicha página. Claramente un portal ha de presentar una homogeneidad a lo largo de sus páginas y mantener un estilo en todas ellas, pero sin caer en la monotonía, ya que esto puede ser un factor para la pérdida de visitas, y por lo tanto para la pérdida de potenciales clientes.

Si analizamos el color empleado en la web, notamos una falta de este a lo largo de las páginas, reforzando la sensación de monotonía producida por el diseño. La lectura de texto en una pantalla es más difícil que la de un texto en papel, por ello hay que proporcionar al usuario una buena sensación para disminuir este efecto adverso.

A continuación se muestra el diseño actual de la web de “*Fincas Castillo*”

Figura: 2. Diseño actual de la web de “*Fincas Castillo*”

3.1.1.3. CONCLUSIONES

En primer lugar, el nuevo diseño ha de solucionar los problemas de monotonía que la web actual produce, manteniendo una homogeneidad en las páginas para no desorientar al usuario y sin cambiar los colores ni las señas que identifican a la empresa. La navegación debe seguir siendo sencilla e intuitiva pese a los cambios que se introduzcan.

Obviamente, el trabajo desempeñado por el encargado de actualizar la página se debe de reducir, automatizando los procesos repetitivos de la web y evitando la pérdida de tiempo y el esfuerzo producido por la edición de las páginas actuales. Por otro lado se hace prácticamente imprescindible un buscador para poder presentar la información adecuada a cada usuario, evitando de esta manera, la información que carece de interés para este.

La información proporcionada deberá actualizarse constantemente y de la forma más automática posible, eliminando la información obsoleta y proporcionando al usuario la información más reciente.

Por último parece clara la necesidad de dinamizar los contenidos de la web, así como añadir nuevas funcionalidades para los diferentes actores del sistema. Por este motivo se llevarán a cabo una serie de entrevistas con los diversos tipos de usuario para establecer las necesidades y requerimientos de cada grupo. Todo ello sin perder de vista los objetivos principales de esta web que siempre tienen que estar cubiertos.

3.1.2. TIPOS DE USUARIO

Una web de estas características tiene normalmente dos tipos de usuario: por un lado los usuarios de la web, que representan a los potenciales clientes de la empresa, y por otro lado

los encargados de la administración de ésta. En este caso hay que añadir un subconjunto de usuarios dentro de los encargados de la administración de la web, que son los comerciales de la empresa, éstos son los encargados de realizar las visitas a los diversos inmuebles y por tanto tienen unas necesidades diferentes al resto.

Así pues tenemos tres tipos de usuario en nuestro sistema:

CLIENTES: Representan el conjunto mayor en cuanto a número que visitará el portal. Este tipo de usuario tendrá acceso a una parte de la web donde podrán satisfacer sus necesidades y obtener la información que desean en cada momento.

ADMINISTRADORES: Son los encargados de realizar toda la administración del sitio, añadiendo nuevos contenidos, gestionando los contenidos actuales, etc.

COMERCIALES: Tipo especial de administradores que se encargan de realizar visitas a los inmuebles y que tienen necesidades ligeramente diferentes a la de los administradores generales del sitio.

Una vez definidos los tipos de usuario, se hace necesario establecer tanto las necesidades como los requerimientos de cada uno de ellos, para de esta manera, realizar un diseño y una implementación de un sistema que satisfaga simultáneamente a clientes, administradores y comerciales, llegando a un compromiso entre las posibles necesidades contrapuestas entre los diferentes tipos de usuario.

3.1.3. NECESIDADES

Durante la primera fase del proyecto se llevaron a cabo una serie de entrevistas con los diversos tipos de usuarios, por tal de establecer las necesidades de cada uno de ellos. A continuación se muestra las conclusiones extraídas de dichas entrevistas.

3.1.3.1. CLIENTES

Las entrevistas con los clientes, o usuarios del sitio, estaban destinadas a conocer el uso que éstos daban a la web y a concretar las posibles mejoras bajo su punto de vista.

La totalidad de los clientes encuestados afirmaron utilizar la página como medio de información para obtener los datos sobre los inmuebles disponibles. Pero definieron como insuficiente las funcionalidades que ésta les aportaba, al margen de la funcionalidad de informar. Exceptuando algunos casos aislados, hubo unanimidad en las respuestas proporcionadas. Por un lado todos vieron necesaria la inclusión de un buscador de inmuebles que les permitiese configurar su búsqueda y, de esta manera, obtener unos resultados adecuados a sus necesidades, además la opción de poder dejar una petición de inmueble en caso de no encontrar el deseado también fue unánime.

Por otro lado, los clientes interesados en la venta de su inmueble, veían necesario la inclusión de un módulo que les permitiese transmitir este deseo a la empresa de una forma eficiente y totalmente online.

Prácticamente la totalidad de los encuestados confirmaron la necesidad de poder realizar cálculos hipotecarios, necesarios para saber el presupuesto disponible y las características hipotecarias que les ofrecería un banco de cara a comprar un determinado inmueble. Así pues también se hace fundamental este aspecto bajo el punto de vista de los usuarios de la web.

Finalmente, algunos de ellos requerían información sobre la empresa, para de esta manera, conocer mejor en quien iban a confiar a la hora de realizar sus gestiones inmobiliarias.

3.1.3.2. ADMINISTRADORES

Las entrevistas con los administradores de la página web actual estaban básicamente destinadas a conocer cómo realizan las diversas gestiones que desempeñan y a conocer las posibles funcionalidades que creen necesarias.

Todos los administradores coincidieron en que, actualmente el trabajo de actualización de la página se convierte en una dura y ardua labor que hay que optimizar, estableciendo este requisito por encima de cualquier otro. Por tanto es necesario un mecanismo de gestión de inmuebles que permita insertar, modificar, consultar datos y eliminar de forma eficiente. Este colectivo también determinó que sería interesante poder gestionar la cartera de clientes y los empleados de la empresa, ya que actualmente la empresa carecía de sistemas de gestión informatizados. Y que además se podría informatizar el proceso de generación de fichas para el escaparate.

En vista de las necesidades establecidas por los usuarios, es necesario un sistema de gestión de alertas, este sistema ha de ser capaz de gestionar todas las peticiones de usuario que se producen.

Por otro lado, la dirección de la empresa vio necesario la inclusión de un módulo de generación de estadísticas dinámicas, que informase sobre los inmuebles disponibles según las zonas de acción, las peticiones de los usuarios, la evolución de ventas, etc. Además de un módulo capaz de generar contratos editables (en formato legible por procesadores de texto) y generación de informes, por tal de estar informados sobre todos los aspectos de su empresa.

3.1.3.3. COMERCIALES

Este colectivo también fue unánime en sus decisiones, por un lado, veían totalmente necesario poder gestionar las visitas programadas mediante un calendario. Y por otro lado, la inclusión de algún tipo de mecanismo para visualizar la ruta más corta entre visitas también les pareció una buena propuesta. Alguno de ellos poseía una PDA y propuso poder crear documentos con

los datos del calendario para que, de esta manera, se pudiese descargar a al dispositivo móvil. Esta propuesta fue avalada por el resto, aunque con más discrepancias sobre su funcionalidad que el resto de propuestas.

Finalmente para satisfacer todas estas necesidades se decidió incluir un módulo de localización dentro del site, así como un calendario capaz de generar informes para los comerciales.

3.2 OBJETIVOS DEL SISTEMA

Los objetivos del sistema han sido establecidos en función a las conclusiones del análisis del problema y de las necesidades extraídas de las entrevistas que realizadas en la fase inicial del proyecto. Así pues, se han determinado dos tipos de objetivos. Por un lado los objetivos de diseño derivados del análisis del problema y, por otro lado, los objetivos generales derivados de las necesidades de los diversos tipos de usuario.

3.2.1. OBJETIVOS DE DISEÑO

Uno de los objetivos propuestos en este proyecto es el de actualizar la imagen de la web, mediante un estilo y un diseño que ofrezca una imagen más corporativa, paliando el efecto de monotonía que tiene la actual página web. Sin perder de vista la navegabilidad, que tiene que resultar fluida, cómoda e intuitiva. Para lograr este objetivo, será necesario dividir el contenido del sitio en diferentes secciones, que tendrán que estar claramente identificadas, siendo totalmente accesibles. Por tanto, también será necesaria una remodelación y dinamización de los contenidos actuales, de forma que se relacionen perfectamente con los nuevos contenidos.

Por otro lado, se debe mejorar el uso del color y la carencia de imágenes para que, de esta manera, se le ofrezca al usuario una sensación de comodidad a lo largo de las páginas. Puesto

que se trata de una empresa con unas señas de identidad propias, estos cambios no deben modificar estos aspectos que identifican a la empresa. Por último, el diseño y la organización de las aplicaciones y funcionalidades que se incluirán en la web deben seguir todos los criterios señalados en párrafos anteriores.

3.2.2. OBJETIVOS GENERALES

Tras el estudio de las necesidades de los diferentes tipos de usuario del sistema y del análisis crítico de la actual página web, se han establecido una serie de objetivos funcionales a añadir a la web. Estos objetivos serán descritos a continuación de forma detallada.

El primer objetivo funcional a cumplir, es el de optimizar y automatizar los procesos de actualización de la web, para ello, es necesario crear un *Gestor de inmuebles*, capaz de realizar todas las operaciones que normalmente realiza la administración de la empresa, es decir, insertar nuevos inmuebles, ver datos de los inmuebles disponibles, eliminar inmuebles, registrar operaciones de venta y alquiler, generar fichas para el escaparate, etc.

Siguiendo con las necesidades de los administradores, se creará un *Gestor de clientes*, mediante el cual, se podrá insertar, modificar y eliminar los datos de la cartera de clientes de la empresa. De forma análoga se creará un *Gestor de empleados* mediante el cual se podrá gestionar toda la información referente a la plantilla. Además de un módulo de *Destacados* gracias al cual, se podrá informar a los usuarios de los inmuebles más recientes o de las mejores ofertas.

Para satisfacer las necesidades de la dirección de la empresa se implementará un módulo de generación de estadísticas, que aportará información sobre los inmuebles disponibles en cada momento, la demanda de los clientes, la evolución de las ventas del año actual y de años anteriores, etc. Además también se implementará un módulo de generación de contratos, que permita generar los principales contratos que habitualmente requiere una empresa

inmobiliaria, es decir, el contrato de arras, el contrato de arrendamiento de viviendas y locales, etc. Así como un módulo de generación de informes tanto de clientes como de empleados.

Centrándonos en las necesidades de los comerciales se creará un *Módulo de localización de inmuebles*, que permita determinar la ubicación en el mapa de cualquier inmueble. Para satisfacer la necesidad de programación de visitas, se implementará un *Calendario* mediante el cual se puedan programar visitas a los diversos inmuebles y, además, generar informes y visualizar rutas de éstas.

Por tal de satisfacer las necesidades de los clientes, se implementará un *Buscador de inmuebles* que permitirá personalizar la búsqueda de cada usuario, ofreciendo así, información más ajustada a dicha demanda. Siguiendo con las necesidades de los clientes, se hace necesaria la inclusión de un método de contacto entre la empresa y el cliente, tanto para información general, como para poner a la venta un determinado inmueble. Por otro lado también se hace imprescindible un *Módulo de cálculos hipotecarios* que proporcione toda la información sobre la hipoteca que el cliente requiere. Finalmente se incluirá un *Gestor de alertas* para gestionar las alertas que los usuarios creen cuando no encuentren el inmueble deseado.

3.3. REQUERIMIENTOS

A continuación se procederá a describir cada uno de los requerimientos derivados de las necesidades establecidas anteriormente, clasificados según su naturaleza. Por tanto se analizarán los requerimientos funcionales, los requerimientos no funcionales, requerimientos dinámicos, requerimientos estáticos, requerimientos de las interfaces externas y

3.3.1. REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales son aquellos que vienen establecidos por las funcionalidades del sistema, es decir, definen el comportamiento interno del software y vienen complementados por los requerimientos no funcionales.

3.3.1.1. REQUERIMIENTOS FUNCIONALES DEL GESTOR DE INMUEBLES

INSERTAR NUEVO INMUEBLE: El administrador de la sección será el encargado de insertar nuevos inmuebles en la base de datos, para ello, deberá llenar un formulario con los datos referentes al inmueble y las fotografías que desee mostrar de éste. El sistema deberá comprobar el formato de los datos introducidos en el formulario, insertando dichos datos en caso de que sean correctos.

ELIMINAR INMUEBLE: El encargado de eliminar inmuebles será el administrador de la sección, para realizar esta acción, se tendrá que seleccionar el inmueble a eliminar y confirmar su eliminación. Seguidamente, el sistema archivará la descripción y todas las operaciones que fueron llevadas a cabo en un documento editable (formato RTF, legible por cualquier procesador de textos), además de generar un archivo con todas las fotografías disponibles del inmueble, por si el administrador desea conservar-las de forma local.

VENDER INMUEBLE: Para marcar un inmueble como vendido y, por tanto, registrar una operación de venta, el administrador de la sección deberá llenar un sencillo formulario especificando qué empleado realizó la venta y qué cliente realiza la compra. Tras la verificación de los datos, el sistema modificará el estado del inmueble y registrará una operación de venta sobre él.

ALQUILAR INMUEBLE: De forma exactamente igual al caso anterior, el administrador, tras seleccionar el inmueble deseado, deberá llenar un formulario concretando tanto el empleado como el inquilino involucrados en la operación.

MODIFICAR INMUEBLE: Tras seleccionar el inmueble a modificar, el administrador deberá ser capaz de modificar los datos de dicho inmueble a través de un formulario similar al de

inserción. El sistema tras la verificación del formato de los nuevos datos, actualizara la información del inmueble correspondiente.

VER DATOS DEL INMUEBLE: El administrador podrá visualizar los datos completos del inmueble y de su propietario a través del gestor de inmuebles, para ello, deberá seleccionar el inmueble deseado y consultar dicha información. El sistema además de mostrar toda la información almacenada sobre el inmueble seleccionado, deberá ser capaz de localizar la dirección en el mapa de la localidad correspondiente.

GENERAR FICHA DE INMUEBLE: Para generar una ficha para un inmueble, presumiblemente para una función de promoción de dicho inmueble, el administrador deberá seleccionar el inmueble correspondiente, el sistema, por su parte, generará una ficha con un formato preestablecido y consensuado con la empresa. Una vez se presente la ficha por pantalla el sistema ha de permitir la impresión de dicha ficha mediante un botón, que el administrador pulsará en caso de desear imprimir-la.

AÑADIR FOTOGRAFÍA AL INMUEBLE: Desde el menú de modificación, el administrador deberá ser capaz de añadir fotografías al inmueble, para ello, simplemente se deberá especificar la ruta local donde se encuentra la imagen y el sistema tendrá que encargarse de subir la imagen al directorio remoto adecuado y con el formato y nombre indicados.

ELIMINAR FOTOGRAFÍAS DE UN INMUEBLE: De la misma forma que el requerimiento funcional anterior, también se podrá eliminar fotografías obsoletas. Para tal efecto, el administrador tan sólo tendrá que seleccionar la imagen a eliminar y el sistema se encargará de eliminar la fotografía correspondiente.

3.3.1.2. REQUERIMIENTOS FUNCIONALES DE LA CARTERA DE CLIENTES.

La cartera de clientes se implementará mediante un gestor de clientes diseñado para satisfacer las necesidades de los administradores, dichas necesidades funcionales se resumen en:

AÑADIR CLIENTES: La inserción de clientes se realizará mediante un formulario que el administrador llenará con los datos del cliente. El sistema, por su parte, debe validar el formato de los datos introducidos y, en caso de que éstos sean correctos, serán insertados en la base de datos.

ELIMINAR CLIENTES: Para eliminar un cliente, el administrador tan solo tendrá que seleccionar el cliente que deseé eliminar y el sistema procederá a la eliminación de dicho cliente. El sistema tendrá que tener en cuenta que si un usuario tiene vinculado un inmueble, es decir, es propietario de un inmueble, no se podrá eliminar el cliente hasta que se elimine el inmueble, siendo el sistema el encargado de avisar al administrador de tal evento.

MODIFICAR CLIENTE: La modificación de los clientes se llevará a cabo mediante un formulario similar al de la inserción, que aparecerá tras seleccionar el cliente a modificar. Tras la verificación de los datos introducidos, el sistema procederá a la actualización de los datos de dicho cliente.

VER DATOS DEL CLIENTE: Por tal de visualizar los datos completos de un cliente concreto, el administrador de la sección seleccionará el cliente que deseé y el sistema se encargará de mostrar los datos por pantalla de una forma clara y concisa.

VER INMUEBLES DEL CLIENTE: En caso de que un cliente sea propietario de un determinado inmueble, el sistema permitirá mostrar los datos de los inmuebles que dicho cliente posee. Para ello el administrador tan solo tendrá que seleccionar el propietario y el sistema se encargará de llevar a cabo el proceso.

GENERAR INFORME DE CLIENTES: El sistema permitirá al administrador generar informes con la información principal sobre los clientes, dicho informe ha de estar disponible desde la sección de informes. Para tal efecto, el administrador tan solo tendrá que seleccionar la opción “*Generar PDF*”.

3.3.1.3. REQUERIMIENTOS FUNCIONALES DEL GESTOR DE EMPLEADOS

AÑADIR EMPLEADO: Para agregar un nuevo empleado a la base de datos, el administrador deberá llenar un formulario con los datos de dicho empleado, de forma opcional, también podrá llenar los datos acerca del contrato que vincula al empleado con la empresa. El sistema, validará los datos introducidos y procederá a incorporar éstos datos en la base de datos.

MODIFICAR EMPLEADO: Tras seleccionar el empleado deseado, el sistema permitirá al administrador modificar los datos del empleado mediante un formulario similar al de

inserción. Tras la validación de datos, se actualizará la información del empleado correspondiente.

ELIMINAR EMPLEADO: El sistema mostrará una alerta, puesto que la eliminación es una operación crítica, permitiendo al administrador eliminar la información del empleado seleccionado.

VER DATOS DEL EMPLEADO: Nuestro sistema será el encargado de mostrar, siguiendo el estilo de la web, los datos personales y administrativos (datos del contrato) del empleado seleccionado por el administrador de la sección.

CAMBIAR FOTOGRAFÍA EMPLEADO: El administrador podrá, si lo desea, cambiar la fotografía de un determinado empleado y, de esta manera, actualizar dicha información. Para ello tan solo tendrá que seleccionar el empleado y especificar la ruta local donde se encuentra la nueva imagen. El sistema, tras detectar dicho evento, se encargará de actualizar la fotografía del empleado especificado.

IMPRIMIR FICHA DE EMPLEADO: El sistema ofrecerá al administrador la opción de imprimir una ficha con la información de un determinado empleado. El administrador deberá seleccionar el empleado previamente.

GENERAR INFORME DE EMPLEADOS: El sistema podrá generar, a petición del administrador, un informe con la información de la actual plantilla de la empresa, dicho informe se generará en formato PDF para maximizar la compatibilidad con los diversos sistemas del mercado.

3.3.1.4. REQUERIMIENTOS FUNCIONALES DEL GESTOR DE ALERTAS

AÑADIR ALERTAS: El proceso de inserción de una nueva alerta podrán llevarlo a cabo tanto el administrador de la sección como cualquier otro usuario sin identificar. Para generar una nueva alerta, se deberá llenar un formulario con algunos datos personales del solicitante (especialmente la dirección de correo electrónico), y los datos sobre el inmueble que busca. Tras la validación de dichos datos, el sistema procederá a la generación de la nueva alerta y a su correspondiente inserción en la base de datos.

ELIMINAR ALERTAS: El administrador de la sección será el único encargado de la eliminación de alertas, para ello tan solo tendrá que seleccionar la alerta indicada y el sistema procederá a su eliminación.

COMPROBACIÓN DE ALERTAS: El sistema deberá comprobar las alertas tras añadir cada inmueble, de esta manera, informará al administrador sobre las alertas vinculadas a este inmueble, es decir, sobre clientes que buscan un inmueble de las características introducidas.

ENVÍO DE MAILS DE ALERTA: Tras comprobar las alertas, el sistema permitirá al administrador enviar correos electrónicos a los interesados. Estos correos tendrán que generarse automáticamente y mostrar en el cuerpo los datos de la alerta, así como un enlace al inmueble que cumple éstas características.

3.3.1.5. REQUERIMIENTOS FUNCIONALES DEL GESTOR DE INFORMES

GENERACIÓN DE INFORMES: El sistema llevará a cabo, por deseo del administrador, la creación de diversos informes (clientes, empleados, visitas, etc.). Dichos informes se generarán desde la sección correspondiente pero deben ser gestionables a través del gestor de inmuebles.

ELIMINAR INFORME: El administrador deberá seleccionar el inmueble a eliminar, tras mostrar una alerta y esperar la confirmación del administrador, el sistema eliminará dicho informe y actualizará la lista de informes disponibles.

VISUALIZAR INFORMES: El aplicativo permitirá la visualización online de los informes disponibles, para ello, el administrador seleccionará de una lista de informes disponibles, el informe que desee visualizar.

GENERAR ARCHIVO COMPRIMIDO DE LOS INFORMES DISPONIBLES: El administrador de la sección podrá descargar un fichero en formato ZIP con todos los informes disponibles en ese momento. Esta opción se mostrará desde el menú de informes.

3.3.1.6. REQUERIMIENTOS FUNCIONALES DEL GESTOR DE CONTRATOS

GENERAR CONTRATO DE ARRAS: En caso de que el administrador desee generar un nuevo contrato de arras, deberá llenar un formulario con los datos tanto de los propietarios como de los compradores, además de información propia de la finca, como por ejemplo, los datos del registro de propiedad, el valor otorgado en concepto de arras, etc. Tras la validación de todos los datos introducidos, el sistema se encargará de generar un archivo editable, en formato RTF con el contrato de arras pertinente.

GENERAR CONTRATO DE ARRENDAMIENTO DE VIVIENDA: Para generar un contrato de arrendamiento de vivienda, el administrador deberá seleccionar en primer lugar el inmueble sobre el que se realiza la acción, el inquilino y los datos sobre las cláusulas del contrato, como son el aval, la fianza, etc. Esta acción se realizará mediante un formulario que el sistema validará posteriormente. El contrato generado será un archivo editable, en formato RTF (Rich Text Format), legible por cualquier procesador de textos, con el contrato de arrendamiento correspondiente.

GENERAR CONTRATO DE ARRENDAMIENTO DE LOCAL: De la misma forma que el requerimiento anterior, el administrador será capaz de generar informes de arrendamientos de locales. Para tal efecto, deberá llenar un formulario similar al del caso anterior, donde especificará todos los datos necesarios para el contrato y sus cláusulas. El sistema por su parte, validará dichos datos, generando posteriormente el archivo en formato RTF.

ELIMINAR CONTRATO: El administrador podrá, si lo desea, eliminar un contrato. Para ello, tan solo deberá seleccionarlo de una lista de contratos disponibles. El sistema se encargará de advertir sobre tal evento y de ejecutar la eliminación del contrato especificado.

VISUALIZAR CONTRATO: Tras seleccionar el contrato deseado, el administrador podrá visualizar dicho contrato. Puesto que los contratos se realizan en formato RTF, éstos serán editables y permitirán al administrador modificar los datos y las cláusulas incluidas en el contrato.

GENERAR ARCHIVO COMPRIMIDO DE LOS CONTRATOS DISPONIBLES: Tal y como sucedía en el caso de requerimiento funcional del gestor de informes, el sistema permitirá la descarga de un archivo comprimido en formato ZIP con todos los informes disponibles del momento. De esta manera, el administrador conseguirá una copia local de todos los contratos generados.

3.3.1.7. REQUERIMIENTOS FUNCIONALES DEL MÓDULO DE ESTADÍSTICAS

El sistema será el encargado de generar las estadísticas y las gráficas sobre diferentes datos almacenados en la base de datos. Este proceso será totalmente automático y dinámico, de esta manera, se proporcionará en cada momento unas gráficas totalmente actualizadas.

GENERACIÓN DE GRÁFICAS DE INMUEBLES A LA VENTA: El sistema deberá generar gráficas donde mostrará porcentualmente los inmuebles que la empresa tiene a la venta, de esta manera, ofrecerá al administrador los datos sobre el número de inmuebles a la venta y del tipo de éstos.

GENERACIÓN DE GRÁFICAS DE INMUEBLES VENDIDOS: El administrador podrá consultar de forma gráfica la información sobre los inmuebles vendidos y su tipo. Siendo el sistema el encargado de generar dinámicamente estas gráficas a partir del estado actual de la base de datos.

GENERACIÓN DE GRÁFICAS DE INMUEBLES EN ALQUILER: Se generarán gráficas sobre el tipo de inmuebles en alquiler, así como la cantidad de dichos inmuebles.

GENERACIÓN DE GRÁFICAS DEL TIPO DE INMUEBLES MÁS DEMANDADOS: Nuestro aplicativo permitirá al administrador consultar información gráfica sobre el tipo de inmuebles más demandados por los clientes, de esta manera, se podrán tomar decisiones importantes sobre estrategias de promoción.

GENERACIÓN DE GRÁFICAS DE LAS LOCALIDADES MÁS DEMANDADAS: El sistema proporcionará información visual a través de gráficas de las localidades donde hay más demanda de inmuebles. Esta información será útil tanto la dirección como para la administración de la empresa a la hora de abordar temas como la estrategia comercial en determinadas localidades o la promoción de inmuebles en dichas localidades.

GENERACIÓN DE GRÁFICAS DE LAS ZONAS MÁS DEMANDADAS: Puesto que la zona de acción de la empresa interesada es la localidad de *Sabadell*, el sistema hará especial hincapié en la información proporcionada por los clientes sobre ésta localidad. Generando, de esta forma, gráficas con las zonas o barrios más solicitados por los clientes, de cara a que la empresa lleve a cabo una estrategia comercial y promocional más agresiva en las zonas más demandadas.

GENERACIÓN DE GRÁFICAS DE LA EVOLUCIÓN DE VENTAS: El administrador también podrá consultar información sobre las ventas de los últimos cinco años. El sistema generará de forma dinámica y en forma de gráficas dicha información.

GENERACIÓN DE GRÁFICAS DE LAS LOCALIDADES DE LOS INMUEBLES EN VENTA: Tras la inserción de cada inmueble, el sistema actualizará las gráficas de las localidades con la información actual de la base de datos. Gracias a la comparación entre estas gráficas y las gráficas de las localidades más demandadas se podrán extraer conclusiones cruciales para la empresa.

GENERACIÓN DE GRÁFICAS DE LAS ZONAS DE LOS INMUEBLES EN VENTA: Para completar el módulo de estadísticas, el sistema generará gráficas sobre las zonas de los inmuebles en venta de una determinada localidad. Para ello, el administrador deberá seleccionar el tipo de inmueble y la localidad de dicho inmueble, para que el sistema genere la gráfica correspondiente.

3.3.1.8. REQUERIMIENTOS FUNCIONALES DEL GESTOR DE VISITAS

El gestor de visitas se implementará mediante un calendario, que deberá proporcionar al administrador como mínimo las siguientes opciones:

PROGRAMAR VISITA: El administrador podrá programar tantas visitas como desee mediante el gestor de visitas. Para ello, tan solo deberá seleccionar el día y rellenar un formulario con los datos de la visita. Tras la validación de dichos datos, el sistema procederá a registrar la visita en el día y horario especificados

ELIMINAR VISITA: El sistema debe permitir la eliminación de las visitas canceladas u obsoletas. El administrador que desee eliminar una visita, tendrá que seleccionarla previamente desde el menú de visitas.

MODIFICAR VISITAS: Cualquier administrador podrá modificar las visitas programadas que hayan sido canceladas o se pospongan /avancen en el horario. Simplemente tendrá que seleccionar la visita programada y modificar sus datos mediante un formulario similar al de la programación de visitas.

IMPRIMIR VISITAS: Nuestro sistema permitirá imprimir en cualquier momento la información mostrada por pantalla, siempre y cuando el administrador se encuentre en el gestor de visitas. De esta manera, se podrá obtener ésta información en soporte papel.

VER RUTAS: Este es un requerimiento cruzado con el módulo de localización, El sistema generará enlaces que mostrarán el camino mínimo (el más corto) entre las diferentes visitas. Este camino deberá tener en cuenta el tráfico de las vías por las cuales se establecerá el camino.

GENERAR PDF DE VISITAS: Para poder generar archivos en formato PDF con la información de las visitas y las rutas mínimas entre éllas, el administrador tan solo deberá seleccionar el día y el sistema se encargará de generar el archivo. La elección del formato PDF se debe a la portabilidad de este formato y a la compresión que ofrece, ya que presumiblemente este archivo podrá ser descargado a dispositivos PDA de los comerciales por tal de facilitarles su labor.

3.3.1.9. REQUERIMIENTOS FUNCIONALES DEL MÓDULO LOCALIZADOR

Este módulo se implementará mediante dos API's que proporciona “Google”, estas API's son:

API de Google Maps: Proporciona métodos y funciones para la creación de mapas y localización de direcciones en ellos.

API AJAX de Google Search: Proporciona métodos y funciones en “JavaScript” para la búsqueda local tanto de direcciones en los mapas, como de cualquier tipo de búsqueda.

BÚSQUEDA DE DIRECCIONES Y NEGOCIOS: El sistema permitirá al administrador visualizar la ubicación de cualquier dirección i/o negocio en un mapa de la zona, para tal efecto, el administrador tan sólo tendrá que especificar la dirección o negocio a buscar.

3.3.1.10. REQUERIMIENTOS FUNCIONALES DEL MÓDULO DE CÁLCULOS HIPOTECARIOS

CÁLCULO DE LA MENSUALIDAD QUE SE PUEDE PAGAR AL MES: El sistema permitirá a cualquier usuario efectuar cálculos sobre la mensualidad que dicho usuario podrá emplear para una hipoteca. Para realizar estos cálculos, el usuario deberá especificar los ingresos mensuales y el sistema le proporcionará los cálculos realizados mostrándolos por pantalla.

CÁLCULO DE LA MENSUALIDAD A PAGAR SEGÚN LA HIPOTECA SOLICITADA: Para que el sistema realice los cálculos de la mensualidad a pagar, previamente el usuario deberá especificar el importe de la hipoteca que desea solicitar, así como también el tipo de interés.

COMPROBACIÓN SOBRE LA APROBACIÓN DE LA HIPOTECA: El usuario podrá comprobar si un banco aceptaría su hipoteca o no. Para realizar la comprobación deberá completar los cálculos anteriores y el sistema se encargará de informarle sobre la aprobación o no aprobación de su hipoteca.

3.3.1.11. REQUERIMIENTOS FUNCIONALES DEL MÓDULO DE COMUNICACIÓN

ENVIAR MAILS DE CONTACTO: Cualquier tipo de usuario podrá, desde la sección correspondiente, enviar mails para contactar con la empresa. El usuario tan solo tendrá que llenar un formulario de contacto y el sistema, tras la validación de los datos introducidos, enviará el correo electrónico.

PONER INMUEBLE A LA VENTA: Los usuarios que deseen poner su inmueble a la venta, deberán acudir a dicha sección para llenar un formulario donde deberán proporcionar, además de los datos del inmueble, algunos datos de contacto. El sistema por su parte, llevará a cabo la validación de los datos introducidos por el usuario y enviará un correo electrónico a la empresa con toda la información especificada. De esta manera, la empresa decidirá si conviene dicha gestión bajo las condiciones que el cliente impone.

3.3.1.12. REQUERIMIENTOS FUNCIONALES DEL MÓDULO BUSCADOR

El buscador de inmuebles es el requisito funcional más demandado por los clientes, dicho módulo deberá permitir la búsqueda tanto a usuarios identificados como a usuarios sin identificar. Los criterios de búsqueda serán los siguientes:

- Búsqueda por operación
- Búsqueda por tipo
- Búsqueda por Localidad
- Búsqueda por Zona
- Búsqueda por Superficie
- Búsqueda por Estado
- Búsqueda por habitaciones
- Búsqueda por precio
- Búsqueda por palabra clave
- Combinación de cualquiera de las anteriores

Obviamente esto proporcionará una gran flexibilidad en la búsqueda de cualquier inmueble.

3.3.1.13. REQUERIMIENTOS FUNCIONALES DEL GESTOR DE USUARIOS

Por tal de llevar a cabo la validación de usuarios, se implementará un gestor de usuarios.

Gracias a este módulo los usuarios podrán:

VALIDACIÓN DE USUARIOS: El sistema permitirá a los usuarios identificarse como administradores y, por tanto, concederá permisos para acceder al área profesional o área de administración. Los usuarios que deseen identificarse como administrador deberán llenar un pequeño formulario con su identificador (login) y su contraseña (password). El sistema validará estos datos concediendo los permisos en caso de que la validación sea correcta.

3.3.2. REQUERIMIENTOS NO FUNCIONALES

Los requisitos no funcionales, al contrario que los funcionales, no vienen determinados por el funcionamiento del sistema. Sino que son requerimientos derivados de características que en algún momento u otro pueden afectar a la correcta funcionalidad de dicho sistema.

3.3.2.1. REQUERIMIENTOS DE RENDIMIENTO

Los requerimientos no funcionales de rendimiento se pueden dividir en dos clases, los que afectan al rendimiento estático y los que afectan al rendimiento dinámico.

Por un lado, los estáticos hacen referencia a las limitaciones de rendimiento que el sistema ha de superar sin tener en cuenta el estado actual, mientras que los dinámicos lo único que importa es el rendimiento en dicho momento.

3.3.2.1.1. REQUERIMIENTOS ESTÁTICOS

CONEXIONES SIMULTÁNEAS: Puesto que se trata de una web para una empresa, se debe tener en cuenta el número de visitantes que se espera conseguir. Al ser una empresa inmobiliaria con un radio de acción más o menos local, el número de visitas simultáneas no será excesivo y por tanto, la web y las bases de datos asociadas a ella deberán soportar un número significativo de conexiones simultáneas, pero no excesivo.

BASES DE DATOS: Debido al gran volumen de información que pueden llegar a almacenar, las bases de datos asociadas a la web, deberán estar capacitadas para almacenar un volumen considerable de datos.

3.3.2.1.2. REQUERIMIENTOS DINÁMICOS

TIEMPO DE ESPERA Y CONEXIONES: Como ya hemos visto, el sistema ha de ser capaz de dar servicio a todas las conexiones sin que esto repercuta ni en el rendimiento ni en el tiempo de espera. Algunos estudios sobre el comportamiento de los usuarios de internet revelan que en este tipo de webs se sigue una distribución normal, con algunas desviaciones coincidentes con el tiempo libre de la mayoría de usuarios. Esto nos hace prever que no se producirán grandes picos en cuanto a las conexiones simultáneas y por tanto, con no se tomarán medidas para paliar el posible efecto en el rendimiento que una gran demanda generaría, ya que esta opción es estadísticamente despreciable.

3.3.2.2. REQUERIMIENTOS SOBRE LAS INTERFACES EXTERNAS

ESTILO Y DISEÑO: en este caso se establecen requerimientos sobre el diseño y estilo debido a que se trata de una empresa con siglas de identidad, la empresa quiere llevar a cabo una actualización y un aumento de las funcionalidades de la web, pero sin que esto conlleve un cambio radical en su estilo. Por tanto se establece la necesidad de crear un estilo más corporativo y profesional pero sin cambios bruscos en los colores ni las señas de identidad de la empresa.

3.3.2.3. REQUERIMIENTOS DE CALIDAD

Todo proyecto informático debe tener unos requerimientos de calidad para asegurar que el producto final sea de la mejor calidad posible, dentro del marco de lo acordado con los

clientes. Desde este punto de vista, se ha realizado un gran esfuerzo por potenciar algunos aspectos que inciden sobre la calidad del software implementado.

Por un lado se ha intentado que la navegación sea lo más intuitiva posible a lo largo de todas las páginas, facilitando de esta forma la interacción con los usuarios. Tanto la estructura del web como la de la base de datos, se ha diseñado pensando en posibles ampliaciones y en su mantenimiento. Ya que la vida de un software es de corta duración y requiere muchos pequeños cambios en un corto periodo de tiempo.

Por otro lado el web está dotado de ayuda contextual que facilita la interacción de los usuarios con la interfaz, aportando la información adicional necesaria para llevar a cabo todas las operaciones disponibles.

Se ha seguido un estilo de codificación en el cual se han utilizado funciones modulares lo más generales posible, con diferentes capas y estilos para los objetos, separando dentro de lo posible la implementación del estilo, etc. Todo para facilitar el mantenimiento y las posibles ampliaciones.

Finalmente se ha elaborado una extensa documentación sobre todas las fases del proyecto, además de guías de usuario en formato flash (ejecutable) y en formato PDF que pueden ser consultadas en caso de que sea necesario.

3.4. RESTRICCIONES DE DISEÑO

3.4.1. SEGURIDAD

La seguridad es un tema delicado y a tomarse muy en serio cuando se elaboran proyectos informáticos. En el caso que nos ocupa, la base de datos almacenará información sobre empleados, clientes, etc., esta información es confidencial y no debe ser accesible a cualquier tipo de usuario. Para garantizar la seguridad del sistema se han diseñado, implementado y probado las siguientes medidas de seguridad:

Control de acceso según los perfiles de usuario: El gestor de usuarios se encargará de realizar la validación de usuarios y de otorgar permisos a los administradores para poder acceder a las secciones del área profesional de la empresa.

Control de acceso a información privada: Se denegará el acceso a todos los archivos privados, tales como informes, contratos, etc. A todos los usuarios no identificados.

Vulnerabilidades del lenguaje: Para la realización del proyecto se utilizará: HTML, JavaScript, Flash, ActionScript, PHP, entre otros. De estos, el único que puede suponer una brecha de seguridad dadas sus características es PHP se han tomado algunas medidas en la implementación para minimizar el riesgo de ataques:

- Se ha evitado el uso de variables globales.
- El envío de datos de los formularios se realiza mediante el método POST, evitando así, que se puedan modificar datos desde la dirección URL.
- Se utilizó un sistema modular en las funciones creadas para minimizar riesgos.

Pese a las medidas tomadas, hay que tener en cuenta que no se puede garantizar la seguridad de un sistema al 100%, ya que existen infinidad de ataques posibles. El único sistema seguro es aquel que está aislado de todas las interacciones externas.

3.4.2. FIABILIDAD

La información que será almacenada en la base de datos es mayoritariamente de carácter informativo, por tanto la pérdida de datos no debería resultar demasiado grave, simplemente se puede volver a introducir. Ahora bien, se debe tener en cuenta que gestionaremos gran cantidad de información y, por tanto, la tarea de volver a introducir todos los datos en la base

de datos puede resultar excesiva o incluso inviable. Por este motivo el servidor proporcionará servicios de *backup* o copias de seguridad de los datos solicitados.

4. DISEÑO

4.1 DISEÑO DE LA BASE DE DATOS

La base de datos que dará soporte a la nueva web ha sido diseñada para que el sistema sea fácilmente escalable. Se ha analizado toda la información que la base de datos tiene que almacenar, procurando que ésta ocupe el mínimo espacio posible, optimizando así, el uso del espacio disponible en el servidor.

Antes de llevar a cabo el diseño de la BD, se llevó a cabo una recolecta de información sobre posibles implementaciones de ésta. Finalmente se optó por un diseño que minimizase el uso de tablas y procurando que éstas ocupasen el mínimo espacio en disco.

A continuación se puede observar un diagrama que ofrece una visión global de la base de datos diseñada.

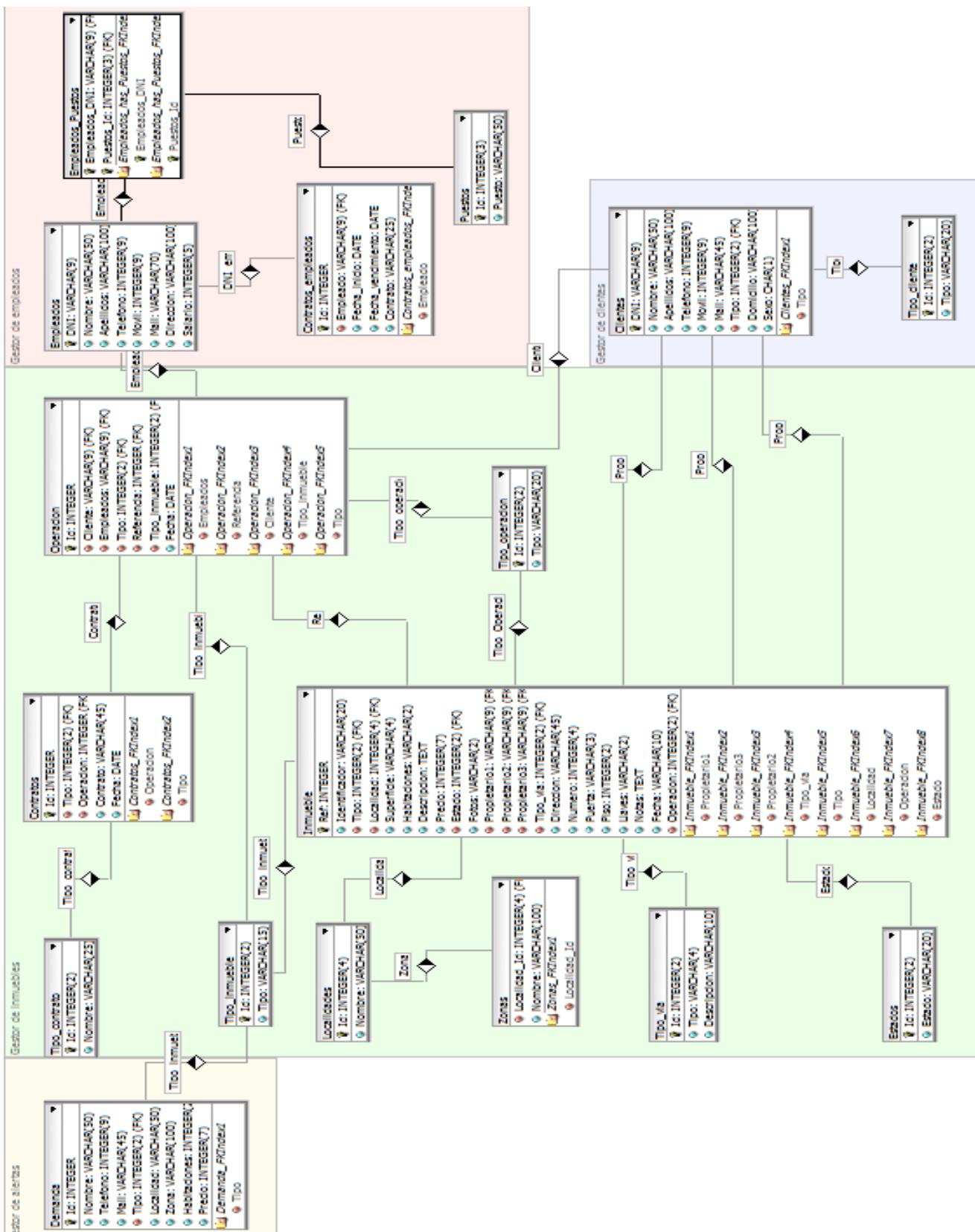


figura: 3. Visión global de la base de datos

4.1.1. DESCRIPCIÓN DE LAS TABLAS Y RELACIONES DEL GESTOR DE CLIENTES

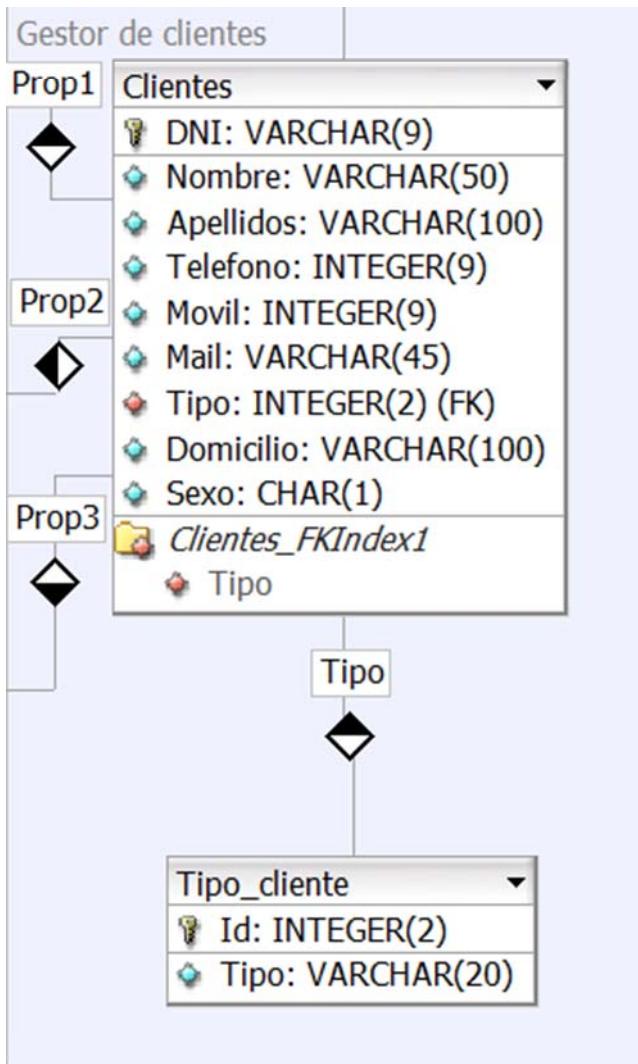


Figura: 4. Diagrama E-R del gestor de clientes.

La tabla *Clientes* contiene toda la información necesaria del cliente, ésta tabla es el artífice principal del gestor de clientes, ya que la totalidad de las funcionalidades que proporciona dicho gestor involucran a esta tabla. A continuación se realizará un análisis detallado tanto de la tabla como de sus relaciones.

Campo	Tipo	Descripción	Comentarios
-------	------	-------------	-------------

DNI	Varchar(9)	Clave primaria, No Nula	DNI con letra del cliente
Nombre	Varchar(20)	No Nulo	Nombre del cliente
Apellidos	Varchar(50)	No Nulo	Apellidos del cliente
Telefono	Integer(9)	No Nulo	Teléfono principal de contacto
Movil	Integer(9)	Nulo	Teléfono móvil (opcional)
Mail	Varchar(50)	Nulo	E-Mail (opcional)
Tipo	Integer(2)	Clave foránea, No Nula	Tipo de cliente
Domicilio	Varchar(100)	Nulo	Dirección del cliente (opcional)
Sexo	Char(1)	No Nulo	Sexo del cliente

Figura: 5. Descripción de la tabla “Clientes”

La tabla “Tipo_clientes” determina el tipo de cliente, es decir, nos indica si el cliente es propietario de algún inmueble, si se trata de un comprador, de un inquilino, etc. A continuación se muestra una descripción de los campos de ésta tabla.

Campo	Tipo	Descripción	Comentarios
Id	Integer(2)	Clave primaria, No Nula, Auto incrementable	Identificador de tipo
Tipo	Varchar(20)	No Nulo	Tipo de cliente

Figura: 6. Descripción de la tabla “Tipo_clientes”

Como se puede observar en la figura xxxx, la tabla “Clientes” mantiene relaciones de 1 a N con la tabla “Inmueble” mediante, es decir, el campo “DNI” de la tabla “Clientes” mantiene tres relaciones de 1 a N con los campos “Propietario1”, “Propietario2” y “Propietario3” de la tabla “Inmueble” . Por tanto, un cliente podrá ser propietario de uno o más inmuebles y, además, podrá ser co-propietario de uno o más inmuebles.

Por otro lado, puesto que un cliente puede estar involucrado en una operación inmobiliaria (venta, alquiler, etc.) la clave primaria de la tabla “Clientes” mantiene una relación de 1 a N

con el campo “Cliente” de la tabla “Operación”, de esta manera, conseguimos que un cliente pueda realizar varias operaciones inmobiliarias.

Finalmente, el campo “Tipo” de la tabla “Clientes”, que determina el tipo de cliente, mantiene una relación de N a 1 con la clave primaria de la tabla “Tipo_clientes”, así se consigue que un mismo tipo de cliente pueda ser asignado a diferentes clientes.

ACLARACIONES: Al insertar un nuevo cliente se le asignará el tipo “Comprador” por defecto, el sistema será el encargado de ir actualizando dicho estado. Por ejemplo al insertar un nuevo inmueble y vincularlo a un cliente, éste pasará a ser “Propietario”.

El tipo de cliente no influye en las operaciones que éste puede realizar, es decir, un “Propietario” podrá realizar una compra de un inmueble sin ningún problema.

4.1.2. DESCRIPCION DE LAS TABLAS Y RELACIONES DEL GESTOR DE EMPLEADOS

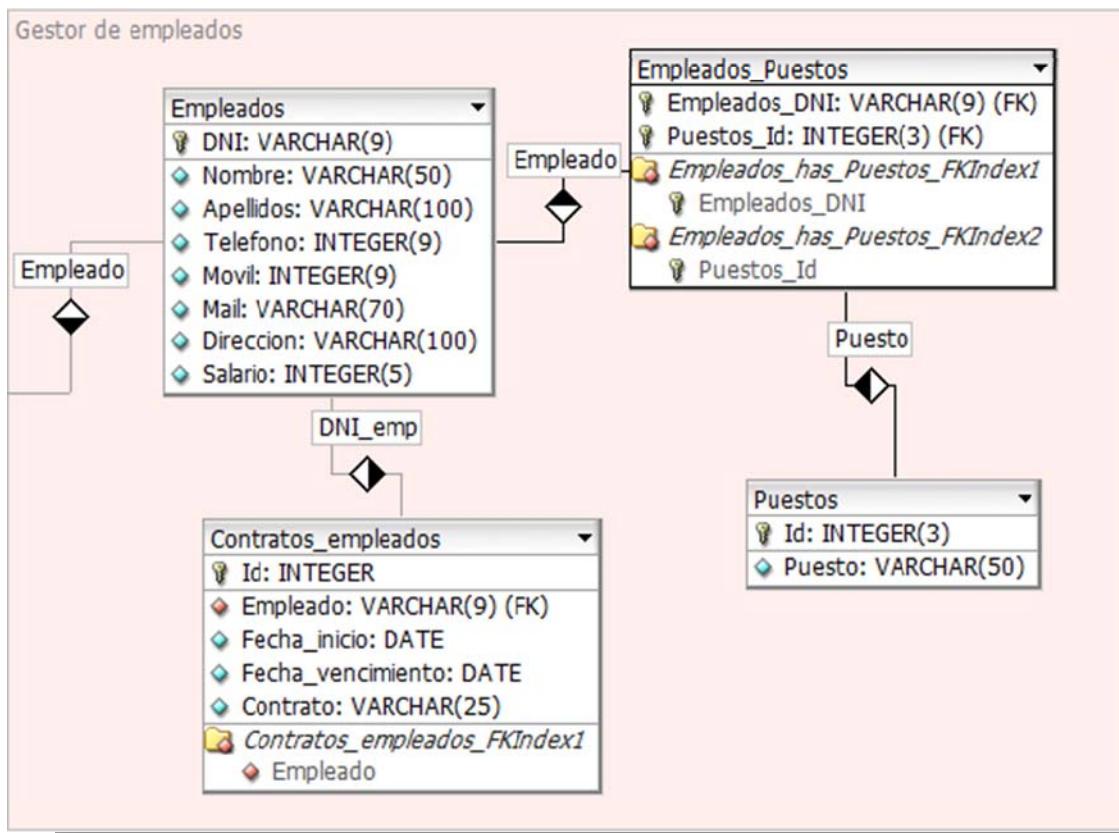


Figura: 7. Diagrama E-R del gestor de empleados

El gestor de empleados está formado por dos tablas, la tabla “Empleados” y la tabla “Contratos_empleados”, siendo esta última compartida por el gestor de contratos. La primera de ellas, almacena toda la información necesaria sobre los empleados de la empresa, mientras que la tabla de “Contratos_empleados” contendrá información sobre el contrato que vincula al empleado con la empresa.

A continuación estudiaremos en detalle cada uno de los atributos que configuran las tablas que forman parte del gestor de empleados, posteriormente se procederá a analizar las relaciones que éstas mantienen con el resto de la base de datos.

Campo	Tipo	Descripción	Comentarios
DNI	Varchar(9)	Clave primaria, No Nula	DNI con letra del empleado

Nombre	Varchar(20)	No Nulo	Nombre del empleado
Apellidos	Varchar(100)	No Nulo	Apellidos del empleado
Telefono	Integer(9)	No Nulo	Teléfono principal de contacto
Movil	Integer(9)	Nulo	Teléfono móvil (opcional)
Mail	Varchar(50)	Nulo	E-Mail (opcional)
Direccion	Varchar(100)	No Nulo	Dirección del empleado
Salario	Integer(7)	No Nulo	Salario actual del empleado
Puesto	Varchar(45)	No Nulo	Puesto actual del empleado

Figura: 8. Descripción de la tabla “Empleados”

Campo	Tipo	Descripción	Comentarios
Id	Integer	Clave primaria, No Nula	Identificador de contrato
Empleado	Varchar(9)	Clave foránea, No Nula	DNI del empleado
Fecha_inicio	Date	No Nulo	Fecha de inicio
Fecha_vencimiento	Date	No Nulo	Fecha de vencimiento
Contrato	Varchar(25)	Nulo	Nombre del documento

Figura: 9. Descripción de la tabla “Contratos_empleados”

Campo	Tipo	Descripción	Comentarios
Id	Integer(2)	Clave primaria, No Nula	Identificador del puesto
Puesto	Varchar(50)	No Nulo	Puesto en la empresa

Figura: 10. Descripción de la tabla “Puestos”

Como se puede observar en el diagrama E-R (véase figura xxxx), ambas tablas están relacionadas mediante los campos “DNI” de la tabla “Empleados” y “Empleado” de la tabla

“Contratos_empleados”, esta relación es de tipo 1 a N, es decir, un empleado podrá tener más de un contrato asociado, mientras que un contrato tan solo podrá tener un empleado vinculado a él.

Por otro lado, existe una relación de 1 a N entre “DNI” de la tabla “Empleados” y “Empleado” de la tabla “Operacion”, de esta manera un empleado podrá tener asociados varias operaciones, pero en cambio, una operación solo tendrá asociado un empleado.

Finalmente existe una relación de N a N entre Empleados y Puestos, por este motivo se crea la tabla “Empleados_puestos” para la implementación de este tipo de relación. Gracias a ella, un empleado podrá tener diversos puestos a su cargo dentro de la empresa, y de la misma forma, un mismo puesto podrá ser llevado a cabo por diferentes empleados.

4.1.3. DESCRIPCIÓN DE LAS TABLAS Y RELACIONES DEL GESTOR DE INMUEBLES

El gestor de inmuebles es la piedra angular de la base de datos, pues es el que capta gran parte de las funcionalidades del sistema, en él, se involucran la mayoría de tablas de la base de datos. A continuación se muestra el diagrama entidad –relación de dicho gestor.

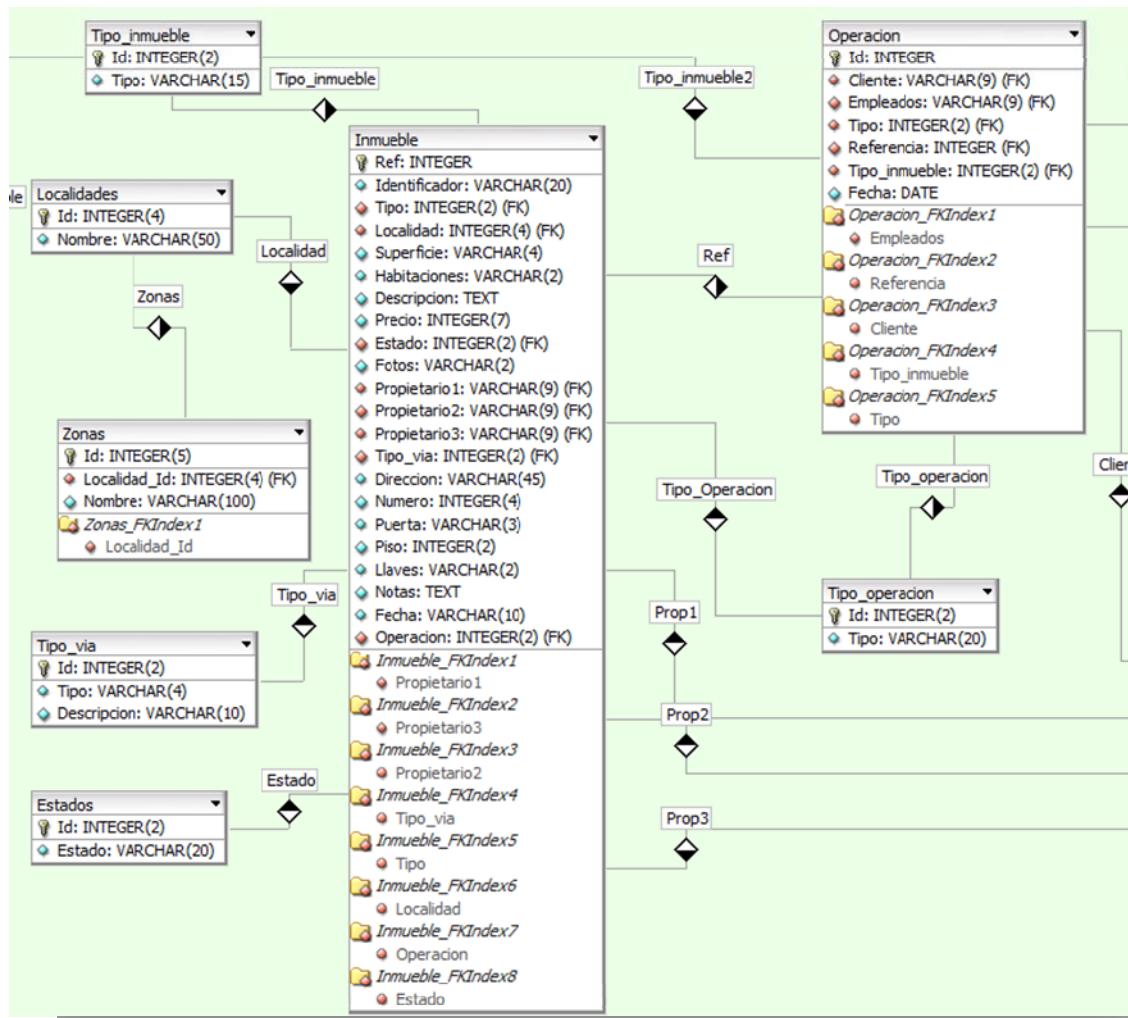


Figura: 11. Diagrama E-R del gestor de inmuebles

Tras comprobar visualmente la estructura del gestor, a continuación se entrará en detalle y se comentará cada tabla, así como las relaciones que estas mantienen con otras tablas. Para empezar analizaremos la tabla “Inmueble”.

Campo	Tipo	Descripción	Comentarios
Ref	Integer(5)	Clave primaria, No Nula	Referencia del inmueble
Identificador	Varchar(15)	No Nulo	Identificador único
Tipo	Integer(2)	Clave foránea, No Nula	Tipo de inmueble
Localidad	Integer(5)	Clave foránea, No Nula	Identificador de la localidad
Superficie	Integer(4)	No Nulo	Superficie en m ² del inmueble
Habitaciones	Integer(2)	Nulo	Número de habitaciones (opcional)
Descripción	Text	No Nulo	Descripción pública
Precio	Integer(10)	No Nulo	Precio en € del inmueble
Estado	Integer(2)	Clave foránea, No Nula	Identificador del estado
Fotos	Integer(2)	No Nulo	Número de fotos del inmueble
Propietario1	Varchar(9)	Clave foránea, No Nula	DNI del propietario 1
Propietario2	Varchar(9)	Clave foránea, No Nula	DNI del propietario 2
Propietario3	Varchar(9)	Clave foránea, No Nula	DNI del propietario 3
Dirección	Varchar(100)	No Nulo	Dirección del inmueble
Numero	Integer(4)	Nulo	Número (opcional)
Piso	Integer(2)	Nulo	Piso (opcional)
Puerta	Integer(2)	Nulo	Puerta (opcional)
Llaves	Bool	No Nulo	Disponibilidad de llaves
Notas	Text	Nulo	Notas importantes (opcional)
Fecha	Date	No Nulo	Fecha de inserción
Tipo_via	Integer(2)	Clave foránea, No Nula	Identificador de la vía
Operacion	Integer(2)	Clave foránea, No Nula	Identificador del tipo de operación

Figura: 12. Descripción de la tabla “Inmueble”

En la tabla “Inmueble” se almacenará la información sobre los inmuebles, y por este motivo mantiene varias relaciones con otras tablas de la base de datos del sistema.

Las relaciones entre la tabla “Inmueble” y la tabla “Clientes” mediante los campos “Propietario1”, “Propietario2” y “Propietario3” ya han sido descritas anteriormente (véase

4.1.1. Descripción de las tablas y relaciones del gestor de clientes). Por tanto, no serán descritas en esta sección.

El campo “Ref” de la tabla mantiene una relación de 1 a N con el campo “Referencia” de la tabla “Operación”, de esta manera, sobre un inmueble se podrán realizar varias operaciones, mientras que una operación solo podrá involucrar a un inmueble.

El campo “Localidades” mantiene una relación de N a 1 con el campo “Id” de la tabla “Localidades”, así un inmueble tan solo podrá tener una Localidad y, en cambio, una localidad podrá ser asignada a más de un inmueble.

Este mismo tipo de relación la mantienen los campos “Tipo” de la tabla “Inmueble” y “Id” de la tabla “Tipos_inmueble”, de tal forma que, un inmueble tan solo podrá ser de un tipo (Casa, Piso, Torre, etc.). Siguiendo con este tipo de relaciones de N a 1, encontramos la relación entre “Estado” de la tabla “Inmueble” y “Id” de la tabla “Estados_inmueble” o la relación entre “Tipo_via” y “Id” de la tabla “Tipo_Via” y la relación entre “Operación” y “Id” de la tabla “Tipo_operacion”.

Todas estas relaciones son para determinar el tipo del atributo con el cual se relacionan y, de esta manera, ahorrar espacio, puesto que tan solo guardamos hemos de guardar un número para identificar el tipo.

Estas son todas las relaciones que las diferentes tablas mantienen con la tabla “Inmueble”, gracias a éstas, podemos optimizar el espacio en disco que ocupará nuestra base de datos y además, facilitamos la labor de mantenimiento y ampliación de la base de datos.

A continuación analizaremos la tabla “Operación”, otra de las tablas más importantes de este gestor, puesto que sobre ella recaen todas las operaciones de venta y alquiler de los inmuebles.

Campo	Tipo	Descripción	Comentarios
-------	------	-------------	-------------

Id	Integer	Clave primaria, No Nula	Identificador de la operación
Cliente	Varchar(9)	Clave foránea, No Nula	DNI del cliente
Empleado	Varchar(9)	Clave foránea, No Nula	DNI del empleado
Referencia	Integer(5)	Clave foránea, No Nula	Referencia del inmueble
Tipo_inmueble	Integer(2)	Clave foránea, No Nula	Tipo del inmueble
Tipo_operacion	Varchar(2)	Clave foránea, No Nula	Tipo de operación
Fecha	Date	No Nulo	Fecha de la operación

Figura: 13. Descripción de la tabla “Operación”

Las relaciones que mantienen los campos “Cliente”, “Empleado” y “Referencia” con campos de otras tablas, ya han sido comentadas previamente y, por tanto, no se volverá a analizar dichas relaciones. En cambio, si comentaremos la relación entre “Tipo_inmueble” y “Id” de la tabla “Tipo_inmueble” que mantienen una relación de N a 1, es decir, una operación tan sólo tendrá un tipo de inmueble asociado, mientras que, un tipo de inmueble podrá ser asignado a varias operaciones. La misma relación mantiene “Tipo_operación” y “Id” de la tabla “Tipo_operación”, así como “Estado” y “Id” de la tabla “Estados”.

Analicemos ahora la tabla “Localidades”.

Campo	Tipo	Descripción	Comentarios
Id	Integer(4)	Clave primaria, No Nula	Identificador de localidad
Nombre	Varchar(50)	No Nulo	Nombre de la localidad

Figura: 14. Descripción de la tabla “Localidades”

Una vez visto cada atributo de la tabla de forma individual, analicemos la relación entre “Id” y “Localidad_Id” de la tabla “Zonas”. Gracias a esta relación, una localidad podrá tener N zonas

diferentes, mientras que, una determinada zona tan solo podrá pertenecer a una localidad específica.

Veamos la tabla Zonas.

Campo	Tipo	Descripción	Comentarios
Id	Integer(5)	Clave primaria, No Nula	Identificador de zona
Localidad_Id	Integer(4)	Clave foránea, No Nula	Identificador de localidad
Nombre	Varchar(50)	No Nulo	Nombre de la Zona

Figura: 15. Descripción de la tabla “Zonas”

Las relaciones que esta tabla mantiene con la tabla “Localidades” ya han sido analizadas, por tanto, no volveremos a recabar en ellas.

Finalmente analizaremos todas las tablas referentes a *tipos*, las relaciones de las cuales, ya han sido analizadas previamente y, por tanto se volverán a comentar.

Campo	Tipo	Descripción	Comentarios
Id	Integer(2)	Clave primaria, No Nula	Identificador de tipo de inmueble
Tipo	Varchar(20)	No Nulo	Tipo de inmueble

Figura: 16. Descripción de la tabla “Tipo_inmueble”

Campo	Tipo	Descripción	Comentarios
Id	Integer(2)	Clave primaria, No Nula	Identificador de tipo de operación
Tipo	Varchar(20)	No Nulo	Tipo de operación

Figura: 17. Descripción de la tabla “Tipo_operación”

Campo	Tipo	Descripción	Comentarios

Id	Integer(2)	Clave primaria, No Nula	Identificador de tipo de vía
Tipo	Varchar(5)	No Nulo	Tipo de vía
Descripción	Varchar(15)	No Nulo	Descripción del tipo de vía

Figura: 18. Descripción de la tabla “Tipo_vía”

Campo	Tipo	Descripción	Comentarios
Id	Integer(2)	Clave primaria, No Nula	Identificador del estado
Estado	Varchar(20)	No Nulo	Nombre del estado

Figura: 19. Descripción de la tabla “Estados”

4.1.4. DESCRIPCIÓN DE LAS TABLAS Y RELACIONES DEL GESTOR DE ALERTAS

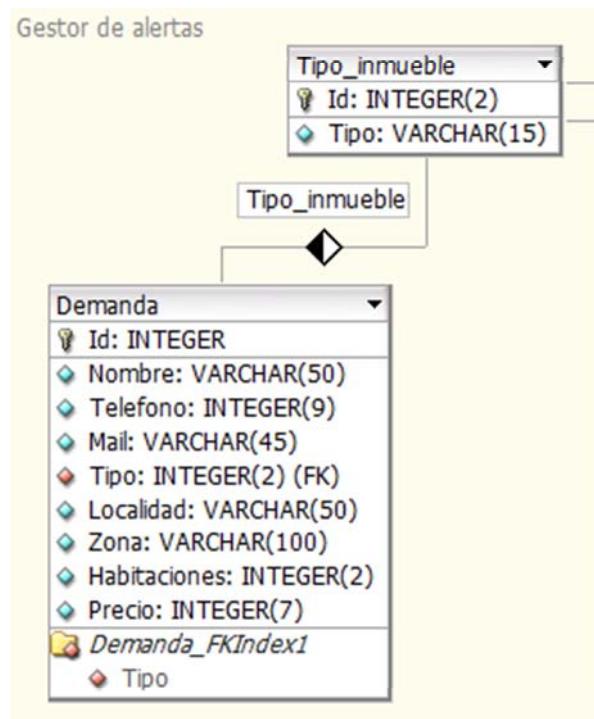


Figura: 20. Diagrama E-R del gestor de alertas

El gestor de alertas está formado por dos tablas, una de ellas, “Tipo_inmueble” ya ha sido analizada con anterioridad. Como se puede observar en la figura xxx, existe una relación de 1 a N entre el campo “Id” de la tabla “Tipo_inmueble” y el campo “Tipo” de la tabla “Demanda”. Esta relación responde a las necesidades de poder asignar un mismo tipo de inmueble a varias demandas, pero que cada demanda tan sólo pueda ser de un tipo de inmueble. Tras analizar las relaciones entre estas dos tablas, veamos en profundidad la tabla “Demanda”.

Campo	Tipo	Descripción	Comentarios
Id	Integer(2)	Clave primaria, No Nula	Identificador de la petición
Nombre	Varchar(50)	No Nulo	Nombre del interesado
Telefono	Integer(9)	No Nulo	Teléfono de contacto
Mail	Varchar(100)	Nulo	Dirección de correo electrónico
Tipo	Integer(2)	No Nulo	Tipo de inmueble que desea
Localidad	Varchar(50)	Nulo	Localidad del inmueble
Zona	Varchar(100)	Nulo	Zona o barrio del inmueble
Habitaciones	Integer(2)	Nulo	Número de habitaciones
Precio	Integer(7)	Nulo	Precio aproximado

Figura: 21. Descripción de la tabla “Demanda”

En esta tabla se almacenarán todas las peticiones que tanto los usuarios, como los administradores en nombre de ellos, inserten. Siendo consultada cada vez que se inserte un nuevo inmueble para determinar si hay algún interesado en la finca que se inserta.

4.1.5. DESCRIPCIÓN DE LAS TABLAS Y RELACIONES DEL GESTOR DE CONTRATOS

El gestor de contratos, encargado de gestionar este tipo de documentos, está formado por dos tablas, la tabla “Contratos” y la tabla “Tipo_contrato”, aunque se podría considerar también de este gestor la tabla “Contratos_empleados” analizada anteriormente (véase “4.1.2. Descripción de las tablas y relaciones del gestor de empleados”).

En cuanto a las relaciones que este gestor proporciona, aparecen básicamente dos, una relación de 1 a N entre “Id” de la tabla “Tipo_contrato” y “Tipo” de la tabla “Contratos” y, por otro lado, una relación de 1 a N entre “Id” de la tabla “Operación” y “Operación” de la tabla “Contratos”. Este tipo de relaciones funcionan de la misma forma que las relaciones 1 a N ya comentadas. A continuación se muestra el diagrama entidad – relación que muestra las tablas y las relaciones del gestor de contratos.

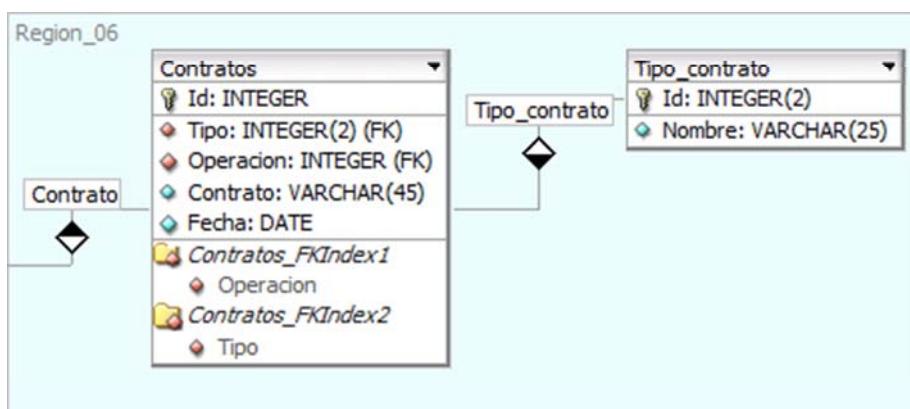


Figura: 22. Diagrama E-R del gestor de contratos.

Una vez vistas las relaciones contenidas en el gestor, se pasará a analizar las dos tablas que conforman dicho gestor y que quedan descritas mediante las siguientes figuras.

Campo	Tipo	Descripción	Comentarios
Id	Integer(2)	Clave primaria, No Nula	Identificador de tipo de contrato
Nombre	Varchar(25)	No Nulo	Tipo de contrato

Figura: 23. Descripción de la tabla “Tipo_contrato”

Campo	Tipo	Descripción	Comentarios
Id	Integer	Clave primaria, No Nula	Identificador del contrato
Tipo	Integer(2)	Clave foránea, No Nula	Identificador del tipo de contrato
Operación	Integer	Clave foránea, Nula	Operación donde se genera el contrato
Contrato	Varchar(45)	No Nulo	Nombre del contrato pertinente
Fecha	Date	No Nulo	Fecha de generación del contrato

Figura: 24. Descripción de la tabla “Contratos”

Con el análisis de estas dos últimas tablas, concluye el análisis de la base de datos propiamente dicha, ya que para el gestor de visitas se ha empleado otra base de datos pre-diseñada que se mostrará en el apartado de anexos de este documento.

Con el diseño implementado se pretende que el aplicativo sea fácilmente escalable y, por tanto, que sea fácilmente ampliable. Además de tener en cuenta el mantenimiento y las posibles ampliaciones, se ha intentado separar al máximo los datos referentes a las diversas entidades y optimizar el espacio que esta base de datos ocupará en el servidor.

4.2. DISEÑO DE LA INTERFAZ DE USUARIO

Hoy en día hay una amplia oferta de sitios web, donde el usuario puede realizar infinidad de acciones. Pero los factores básicos inherentes al éxito o al fracaso de una web son, por una banda, la utilidad de las funcionalidades que el sitio ofrezca y, por otra, la interfaz con la que el usuario interactúa.

En el caso que nos ocupa, los sitios de empresas inmobiliarias, se diferencian básicamente en los servicios y funcionalidades que éstos ofrecen a los administradores, puesto que no existen diferencias significativas entre las funcionalidades ofrecidas a los usuarios. Por tanto en estos casos, la interfaz de usuario cobra más relevancia, ya que la finalidad del web es que lo visiten el máximo número de personas posible.

Pero realmente una interfaz de usuario, ¿Qué es?. Se entiende por interfaz de usuario el punto de unión física y/o funcional entre dos entes diferentes, en este caso, entre hombre y computadora. Una interfaz de usuario es un conjunto de protocolos y técnicas para el intercambio de información entre una aplicación computacional y el usuario de ésta ([Larson, 1992]). Por tanto, es responsable de solicitar las entradas del sistema al usuario y de mostrar las salidas de la aplicación de una manera comprensible. La interfaz de usuario (IU) no es partícipe de los cálculos de la aplicación ni del almacenamiento, recuperación ni transmisión de la información.

En el diseño de este tipo de elementos, influyen no solo aspectos físicos sino también psíquicos, por ello, se ha de tener en cuenta la psicología y fisiología de los usuarios y las acciones del proceso para crear una experiencia efectiva, eficiente y satisfactoria que promueva el uso.

Las interfaces de usuario han de intentar ofrecer una sensación de confort al usuario mientras éste consulta la información y realiza las acciones, evitando la desorientación y la frustración

que puede provocar un diseño equivocado. Si no se logran estas premisas, el sitio caerá en desuso y su implementación no se podrá considerar productiva. Para alcanzar esta meta, las interfaces de usuario se basan en la usabilidad, que comentaremos a continuación, pero antes se han de tener en cuenta algunas cuestiones:

- Es necesario definir tipos de usuario y, para cada tipo de usuario, el nivel de conocimiento, nivel de aprendizaje, requerimientos y necesidades, trasfondo cultural y entorno donde se usará la aplicación, dependencia de las operaciones respecto del sistema o del usuario, etc.
- Profundizando en la interactividad, se debe tener en cuenta la facilidad de los usuarios en realizar las tareas propuestas en un tiempo aceptable, el tiempo dedicado al aprendizaje de la aplicación, posibles errores en la interacción con el sistema, documentación, ayuda, requerimientos especiales, discapacidades, etc.

Todas estas consideraciones son términos difíciles de acotar y, por tanto, el diseño de una buena interfaz de usuario es un proceso muy complejo y con obtención de resultados subóptimos, puesto que el resultado óptimo es una utopía inalcanzable.

Para la implementación de la interfaz de usuario se ha seguido un algoritmo iterativo, el cual *permite un enfoque más efectivo en el desarrollo. El algoritmo es el siguiente:*

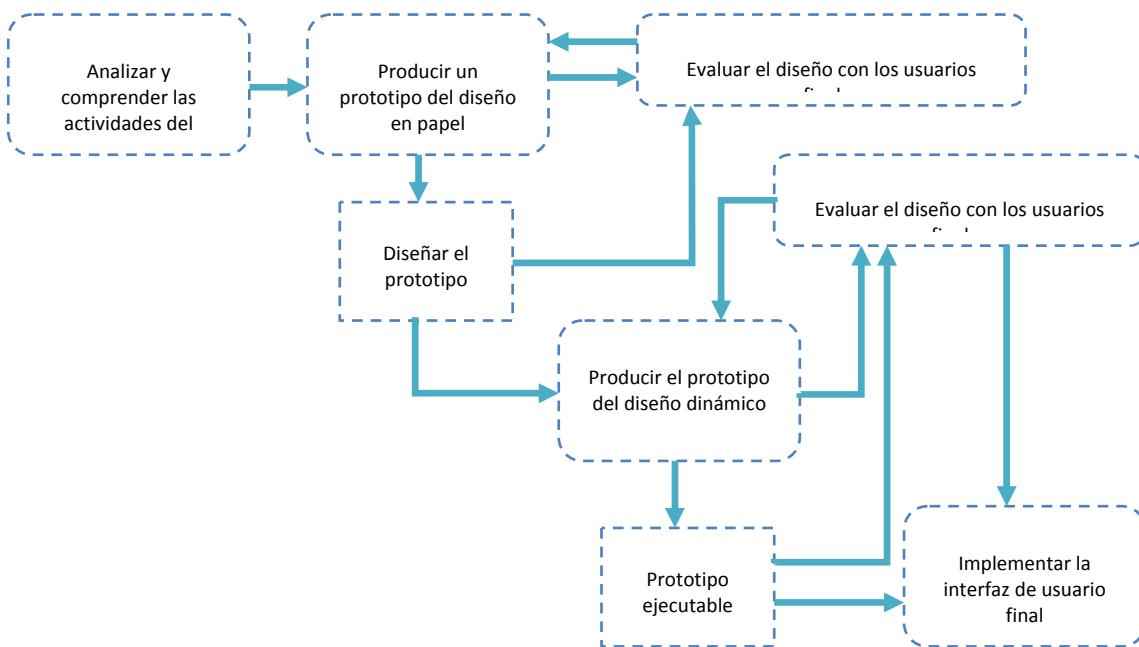


Figura: 25. Algoritmo de diseño de la interfaz de usuario

4.2.1. USABILIDAD

La usabilidad es la premisa básica en la cual se fundamentan los diseños de interfaces de usuario, esta propiedad proporciona una magnitud relativa referente al uso de dicha interfaz. Por tanto, uno de los objetivos de una buena interfaz de usuario es maximizar la usabilidad, para llevar a cabo tal efecto se realizan diversas acciones como son:

- Se intenta que la navegación sea lo más sencilla e intuitiva posible, de esta manera, se reduce el tiempo de aprendizaje que el usuario necesita para usar la aplicación. Normalmente en diseños de interfaces Web, se intenta que cualquier información este disponible con el mínimo número de “Clicks” posibles, estando el umbral en cinco o seis.
- Se proporcionan respuestas visuales a las acciones del usuario, es decir, cada acción del usuario tendrá una respuesta, aunque el resultado de ésta no sea satisfactorio.
- Se generarán controles gráficos para la selección e introducción de la información. La respuesta a un estímulo gráfico siempre es más rápida que la de un estímulo de texto, por este motivo se utilizan botones en lugar de simples links.

- Se dotará a la aplicación con ayuda contextual, de esta manera, aumentamos la comprensión del usuario y, al mismo tiempo, disminuimos la desorientación que éste pudiera tener.
- Se evitará, en la medida de lo posible, las barras de desplazamiento, especialmente la barra de desplazamiento horizontal.

Todas estas consideraciones hacen que la interfaz de usuario sea relativamente fácil de aprender a usar y que usuarios sin experiencia, puedan utilizar la aplicación tras una breve sesión de capacitación o gracias a la ayuda contextual y la documentación disponible.

4.2.2. CARGA DE INFORMACIÓN

La distribución de la información en el web, influye en la facilidad del usuario para acceder a la misma. Una sobrecarga de información en la página genera estrés visual, la cual cosa perjudica el uso de dicha página. Por otro lado, es recomendable generar páginas principales a modo de índice, donde el usuario pueda obtener información sobre el contenido de cada una de las secciones.

4.2.3. ACCIONES

Las acciones básicas a realizar en un aplicativo web son básicamente la consulta y la modificación de los datos almacenados en la base de datos. Dada la posibilidad de introducción de errores por parte de los usuarios, se ha de tener en cuenta uno de los factores importantes del diseño, la recuperación de errores. Puesto que la posibilidad de recuperarse de errores provocados por los usuarios, supone un coste no justificable para un sistema de este tipo. Ante este problema, la alternativa más eficaz es confirmar todas las acciones destructivas del usuario y asegurarse de que se generan de una forma controlada. Además, también es recomendable validar las acciones de modificación y en menor medida las de consulta.

Como medida de seguridad externa, se deben realizar copias de seguridad de los datos críticos o para ahorrarnos la tarea de volver a insertar los datos perdidos, pero este ámbito queda fuera del diseño, ya que depende del grado de seguridad que deseé obtener el usuario.

Dependiendo de la sección en la que se encuentre el usuario habrá diferentes acciones e información accesible, lo que configura diferentes entornos gráficos para la mejor interacción, que serán comentados en la correspondiente sección.

4.2.4. COLORES EN EL WEB

Los colores a utilizar en los sitios web, es otro de los factores a tener en cuenta en el diseño. En la elección de los colores se debe pensar en el mensaje a transmitir al usuario, puesto que cada color se asocia a un determinado estado de ánimo o cualidad. Por tanto hay que adecuar estos al mensaje que se pretende mostrar.

Otro factor a tener en cuenta en la elección de los colores, es el contraste entre los escogidos, es decir, el contraste generado por los colores que diseñan la interfaz ha de ser agradable al ojo humano y reducir, en la medida de lo posible, el estrés del usuario en la navegación.

Los elementos más importantes deben ser fácilmente identificables por el usuario a simple vista. Además, se ha de controlar el uso de color para no generar estrés visual y sensación de agobio, siempre es preferible que el web transmita tranquilidad.

Finalmente, se debe de tener en cuenta que algunos colores no permiten una correcta lectura en pantalla, algunos estudios demuestran que la lectura en la pantalla de un ordenador, no es tan cómoda como la lectura en papel y, por tanto, no hay que producir más impedimentos en la lectura.

Para la interfaz de usuarios se ha escogido básicamente una gama de azules, compuesto por los colores ( ,  ,  ,  , ), y el blanco.

El azul es el color del cielo y del mar, por lo que se suele asociar a estabilidad y profundidad, representa la lealtad, confianza, sabiduría, inteligencia, verdad, que son precisamente los mensajes que la empresa quiere transmitir. Los colores utilizados son azules mezclados con gris, con lo que además adoptan el significado de paz, estabilidad y se asocian al éxito. Finalmente el blanco complementa la imagen que se pretende dar otorgando los significados de bondad y pureza.

Por norma general los contrates entre estos tipos de colores suelen ser una apuesta segura, pues prácticamente cualquier combinación de estos resulta agradable al ojo humano.

4.2.5. INTERFACES DE USUARIO BASADAS EN WEB

Este tipo de interfaces forman parte de las interfaces gráficas de tipo WIMP, pero han de contemplar algunos derivados de su naturaleza que en otras interfaces no se dan o se dan en menor grado, como por ejemplo el hecho de que deben ser transportadas por internet (tamaño de éstas) o que deben interactuar dentro del marco del navegador.

El objetivo primordial de las interfaces web son tanto la promoción como el uso por parte de los usuarios. La principal dificultad de este tipo de interfaces es que no se puede determinar en qué tipo de ordenador la utilizará el usuario, ni la configuración, ni siquiera la velocidad de acceso a la red, etc. La suposición de algunos de estos factores, implica la coerción al usuario que puede afectar directamente en el uso de la aplicación, debido a la producción de malestar durante el uso de la misma.

Estas consideraciones se tendrán en cuenta en el diseño de la interfaz del aplicativo.

4.2.6. DISEÑO DE LA INTERFAZ DE USUARIO

Ya se han analizado los factores de éxito que se han tenido en cuenta a la hora del diseño de nuestro aplicativo. En este punto vamos a mostrar la interfaz implementada y estudiar las partes que la constituyen, obviamente no se analizará la interfaz de cada sección, ya que todas comparten una topología y una fisionomía que determinan el estilo y la estructura que ya hemos comentado.

Para el desarrollo de la interfaz se ha tenido en cuenta el tipo de usuario que la utilizará. Este perfil es el de una persona de 24 a 45 años en su mayoría, aunque esta franja de edad tiene una desviación típica significativa, puesto que existe un número significativo de personas que no entrarán dentro de esta franja de edad y que visita habitualmente sitios inmobiliarios.

En cuanto a la formación también varía notablemente, si bien es cierto que, cada vez más existe una mayor culturalización entorno a la informática y a Internet, aún existe un buen porcentaje de gente sin nociones de navegación ni de utilización de este tipo de herramientas informáticas. Por tanto, aunque la tendencia es claramente a una evolución hacia una sociedad más informatizada, nuestro aplicativo no debe menospreciar las visitas procedentes de usuarios sin experiencia previa, ofreciendo ayuda a dichos usuarios e intentando minimizar el período de aprendizaje que requiere el aplicativo.

A continuación se muestra la interfaz creada:



Figura: 26. Interfaz de usuario general.

- [1] – Cabecera: Se trata de una implementación en “Flash” donde se muestra el nombre y la dirección y datos de contacto de la empresa. Las imágenes y los títulos que van apareciendo, tienen un doble objetivo, por un lado pretenden configurar un entorno agradable para la navegación del usuario, y por otro, promover el uso y estimular al cliente que visite la web.
- [2] – Secciones: Bajo la cabecera encontramos enlaces a las diferentes secciones del web, cada enlace se compone de un botón con un texto identificativo sobre el contenido de la sección.
- [3] – Contenidos: En esta zona central se muestran los contenidos de la página actual, la disposición de los elementos dentro de esta sección, pese a seguir un patrón de homogeneidad, puede variar en función de la página visitada.

- [4] – Destacados: En el lateral izquierdo encontramos la un resumen sobre los inmuebles destacados, el uso del blanco sobre un tono oscuro produce impacto visual. En el caso de que visitemos el buscador, aparecerá el formulario de búsqueda en lugar de los inmuebles destacados.
- [5] – Menú secundario: En este menú situado bajo la zona de contenidos, se encuentran dos enlaces, uno al área profesional (sección de administración) y otro a la sección de enlaces de interés.

Siguiendo con la interfaz de usuario, veamos en esta ocasión la interfaz de usuario de una de las secciones de administración (el buscador de administración).



Figura: 27. Interfaz de usuario de la zona de administración

- [1] – Cabecera: La cabecera se mantiene a lo largo de todas las páginas.
- [2] – Secciones: En este caso, encontramos muchas más secciones, diseñadas para facilitar la labor del personal de la empresa.
- [3] – Operaciones: En la mayoría de páginas de la sección de administración, encontramos operaciones aplicables a los contenidos seleccionados de la zona de contenidos.
- [4] – Zona de contenidos: En este caso concreto, los contenidos son los inmuebles encontrados con las características rellenadas del formulario de búsqueda
- [5] – Lateral izquierdo: Exceptuando el buscador, en esta zona se mostrará información de la sección (título u otra información). En el caso del buscador, se muestra el formulario de búsqueda.
- [6] – Menú secundario: Menú donde aparece la opción de salir de la zona profesional.

Finalmente, destacar que algunos de los formularios se muestran en ventanas emergentes a modo de “Popup”, para no desorientar al usuario y maximizar el área de interacción con la interfaz.

4.2.7. ESTRUCTURA WEB

La estructura web es otro factor clave a tener en cuenta, puesto que una mala estructura dificultará los procesos que el usuario llevará a cabo. Ya hemos visto que la usabilidad es uno de los parámetros en los que se centra cualquier interfaz de usuario, en este caso, al tratarse de un aplicativo web, hemos de tener en cuenta el tiempo que tardará un usuario en realizar las acciones.

La estructura web desarrollada con menos de 4 clicks, siendo la media de 2. Es decir el usuario podrá realizar cualquier acción con una media de dos clics, si a esto le añadimos la disposición de y la estructura de la información en la pantalla, así como la navegación intuitiva que se ha pretendido implementar, obtenemos una interfaz de usuario que interactúa con éste de forma eficiente y que produce una buena sensación al internauta mientras navega por la misma.

A continuación se muestra un diagrama de la estructura web diseñada, en dicha estructura, se dividirán los archivos en carpetas y subcarpetas, algunas de ellas, necesitarán permisos para poder ejecutar o consultar la información que contienen, como es el caso de la carpeta de administración.

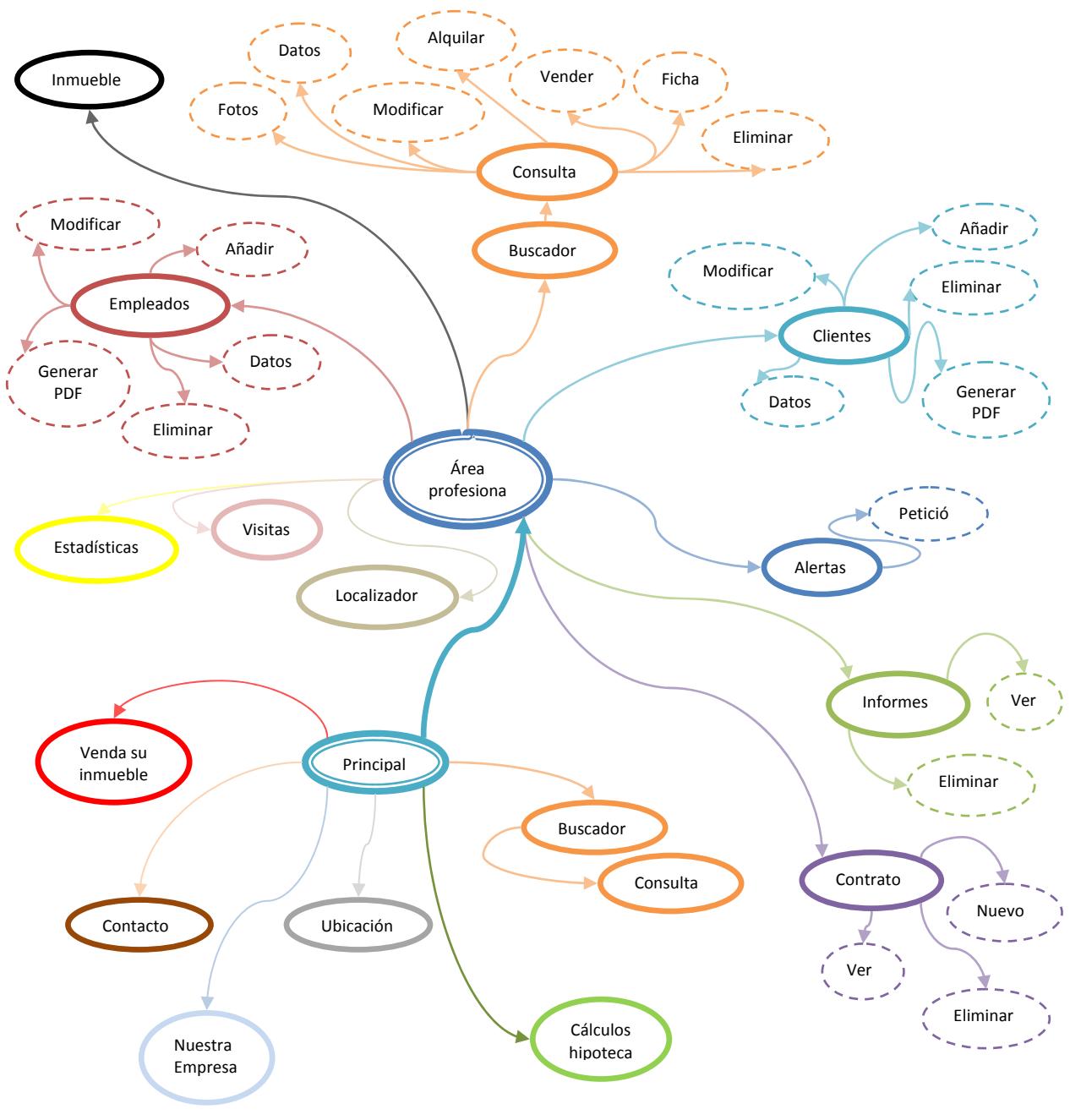


Figura: 28. Estructura web del aplicativo

5. CODIFICACIÓN

5.1. LENGUAJES

En este punto se procederá a analizar los lenguajes utilizados para el desarrollo del aplicativo web, incluyendo un análisis sobre las ventajas y desventajas de cada uno, así como una breve explicación del porqué de esta elección y del uso que se ha dado de éste.

5.1.1. HTML (HYPER TEXT MARKUP LANGUAGE)

HTML son las siglas de “Hyper Text Markup Language” cuya traducción sería “Lenguaje de marcas hipertextuales”. Como su nombre indica se trata de un lenguaje de marcas (“tags”) diseñado para estructurar textos y mostrarlos en forma de hipertexto, que es el formato estándar de las páginas web. HTML es una aplicación de SGML conforme al estándar internacional ISO 8879.

Se trata de un lenguaje sencillo e intuitivo que requiere pocos conocimientos para el desarrollo de páginas web con él. Esta sencillez es uno de los factores que provocan que este lenguaje solo sea útil para la generación de contenidos estáticos, ya que carece de método alguno para tratar información dinámica. Por tanto se requieren otros lenguajes para el tratamiento de dicha información. Además HTML no fue diseñado con el propósito de aplicar estilo o formato a la información contenida en la página, esto supone una limitación a la hora del diseño, sobretodo del diseño de la interfaz de usuario. Por este motivo se ha intentado separar al máximo la información del estilo o formato de estas mediante hojas de estilo (CSS) que se explicarán más adelante.

La codificación de páginas web mediante HTML, puede resultar un tanto caótica si no se siguen unas normas o consideraciones a la hora de programar. En el desarrollo del sistema que aquí nos ocupa, se han intentado seguir las recomendaciones dictaminadas por “W3C”. “W3C” son las siglas de “World Wide Web Consortium”, este organismo es un consorcio internacional que produce estándares para “World Wide Web”. Entre sus consideraciones se incluyen recomendaciones como la indicación del conjunto de caracteres que pueden aparecer en la página, selección de un criterio de nombramiento de objetos, referencias a colores usando notación hexadecimal entre otras muchas.

En vista de los pros y los contras de este lenguaje, su utilización se ha restringido a la codificación del contenido estático de las páginas implementadas, dejando el resto de funcionalidades para otro tipo de lenguajes.

5.1.2. JAVASCRIPT

JavaScript es un lenguaje interpretado orientado a las páginas web, desarrollado por “Sun Microsystems” y “Netscape”. Se trata de un lenguaje interpretado, la cual cosa quiere decir que no necesita ser compilado. Su sintaxis es similar a la del lenguaje Java (también desarrollado por “Sun Microsystems”), pero al contrario que éste, Javascript no es un lenguaje de orientado a objetos propiamente dicho, puesto que no dispone de herencia.

Este lenguaje proporciona dinamismo y funcionalidad a las páginas web, dada su naturaleza, este lenguaje es interpretado por el navegador y, por tanto, es un lenguaje del lado del cliente. Esto limita las funcionalidades del lenguaje a obtener dinamismo local sin posibilidad de obtener información procedente del servidor.

En el sitio web implementado, se ha utilizado JavaScript para diversas funcionalidades, algunas de las cuales son: generar ventanas emergentes, validar el formato de los datos introducidos en los diferentes formularios, detectar la configuración de pantalla del usuario, etc.

5.1.3. CSS (CASCADING STYLE SHEETS)

CSS son las siglas de “Cascading Style Sheets”, cuya traducción sería “Hojas de estilo en cascada”. Se trata de un mecanismo que describe cómo se va a mostrar una página en pantalla, cómo se va a imprimir o incluso como debe ser pronunciada por un dispositivo de lectura. Esta forma de describir el estilo y formato de los datos mostrados, ofrece un gran control sobre dichos aspectos, además de permitir la separación entre la información y el estilo de la misma, facilitando las posibles tareas de modificación posteriores.

Gracias a la maquetación de los estilos mediante un archivo de hoja de estilo, descargamos el código de la página de todo el formato y del estilo de ésta, reduciendo considerablemente el tamaño de la misma y, por tanto, reduciendo el tiempo de carga y de transferencia de la página. Puesto que las hojas de estilo se almacenan en la cache del navegador, ésta solo será enviada una vez, mejorando así el tiempo de carga de las páginas posteriores.

Entre los inconvenientes principales de CSS, encontramos el hecho de la compatibilidad con navegadores antiguos. Si bien es cierto que actualmente la mayoría de usuarios utiliza las últimas versiones de los navegadores más populares, aún existen usuarios que utilizan versiones menos recientes. Estos usuarios no podrán cargar el estilo de la página y, por tanto verán el contenido de la página sin ninguna organización ni formato, salvo que se incluya información de estilo en el código HTML.

Siguiendo con la compatibilidad, el resultado de la aplicación de un estilo puede variar de un navegador a otro, esto dificulta la tarea de codificación de un determinado estilo, ya que hay que tener en cuenta los atributos visibles por cada navegador.

Pese a los inconvenientes que acabamos de analizar, el uso de hojas de estilo es recomendable por los beneficios que estas nos otorgan. En el aplicativo web, se ha intentado separar al máximo el estilo de la información mostrada mediante una hoja de estilo que define el formato de los objetos que constituyen la página.

5.1.4. PHP (HYPERTEXT PREPROCESOR)

“Hypertext PreProcesor” o PHP es un lenguaje abierto, multi-plataforma y totalmente “server – side”, la cual cosa quiere decir que el usuario no necesita ningún programa adicional para interpretar las páginas a las que accede, ya que el script en php es ejecutado por el servidor pasando los resultados mediante contenido HTML.

Este lenguaje posee un bajo consumo de recursos del servidor y permite una carga de la página rápida. Además dispone de una excelente conectividad con un gran tipo de bases de datos diferentes, entre ellas MySQL, que será la utilizada en nuestro aplicativo. Los módulos disponibles de este lenguaje nos ofrecen multitud de funcionalidades, como la seguridad mediante SSL o la prevención de inyección de consultas SQL en los datos.

Gracias a la similitud de este lenguaje con los lenguajes de programación orientados a objeto más comunes (C y Perl) permite que la mayoría de usuarios experimentados puedan realizar aplicaciones complejas con una curva de aprendizaje muy suave.

Pese a que las últimas versiones se encaminan hacia un lenguaje orientado a objeto, es decir, con clases y herencias entre clases, este mecanismo continúa siendo ineficiente de cara a grandes aplicaciones. La filosofía de este lenguaje permite una mayor libertad a la hora de realizar la programación, lo que habitualmente conduce a una realización más rápida del trabajo pero con un mayor desorden que podría llevar a una mayor dificultad en el mantenimiento o ampliaciones a largo plazo.

5.1.4.1. ALTERNATIVAS A PHP

Existen diferentes alternativas a la utilización de PHP, las principales son las siguientes:

- **ASP.NET:** Lenguaje de programación comercializado por “Microsoft”, forma parte de la plataforma .NET de esta compañía y es la tecnología sucesora de ASP, cuyas siglas significan “Active Server Pages” o páginas activas del servidor. Pese a que las características del lenguaje son similares a las de PHP, esta tecnología está estrechamente relacionada con el modelo tecnológico de su fabricante y, no se trata de un lenguaje libre.
- **COLDFUSION:** Lenguaje de programación comercializado por la compañía “Macromedia”. Se trata de un lenguaje de programación web similar a los lenguajes ya comentados, pero con funcionalidades ligeramente más complejas. Su naturaleza hace que no sea lenguaje diseñado para interactuar con bases de datos, aunque proporciona una interacción simple con bases de datos como Oracle, MySQL o Acces. Su sintaxis, similar a la de HTML, permite que la curva de aprendizaje sea suave.
- **JSP/JAVA :** Se trata de un lenguaje de programación libre, aunque su libertad depende de la voluntad de futuro de la empresa “Sun Microsystems”. Al igual que la mayoría de los lenguajes analizados, es multi-plataforma. Esta tecnología está basada en Java y permite mezclar código Java y HTML en el mismo archivo y, puesto que se trata de un lenguaje “server-sided”, las páginas son interpretadas por el servidor antes de ser enviadas al cliente.
- **CGI/PERL:** Se trata de un lenguaje de programación imperativo que combina características de muchos lenguajes de programación como C, AWK o la línea de comandos Shell. Este lenguaje ha sido utilizado para desarrollar webs mediante la creación de scripts o CGI’s, una de las aplicaciones más habituales de este lenguaje es la de nexo de unión entre sistemas o interfaces que no fueron diseñadas específicamente para interactuar entre ellas. Puesto que se trata de un lenguaje que necesita ser compilado la ejecución de scripts escritos en Perl puede resultar más lenta que la de otro tipo de lenguajes.

- **PYTHON:** Es un lenguaje de programación abierto gestionado actualmente por la “Python software foundation”. Al contrario que Perl, considerado su principal rival, se trata de un lenguaje interpretado, cuyo intérprete puede ser usado de forma interactiva. Python permite la creación de módulos que podrán ser ejecutados por otros programas.

5.1.4.2. ELECCIÓN DEL LENGUAJE

Una vez analizados los diferentes lenguajes y las características que estos ofrecen, se escogió como lenguaje de programación PHP. Esta elección se llevó a cabo en base al análisis de los lenguajes y a las conclusiones extraídas de éste.

Los resultados de las diferentes alternativas no distan mucho, por eso, se optó por un lenguaje abierto que no dependiera de ninguna compañía. Además como ya hemos visto, PHP no consume muchos recursos, la carga de páginas es rápida, las nuevas versiones se encaminan hacia un uso más eficiente de la orientación a objeto y existe una gran comunidad dando respaldo.

Estos motivos, junto con el hecho de la formación previa del proyectista en el ámbito de la programación web con PHP y la gran conectividad del lenguaje con las bases de datos, propiciaron la elección de PHP como lenguaje para la implementación algunas de las principales funcionalidades del sistema.

Para paliar algunos inconvenientes del lenguaje, se ha optado por un estilo de programación ordenado y claro, comentando gran número de líneas y funciones.

5.1.5. MYSQL

MySQL es un sistema de gestión de bases de datos multihilo y multiusuario. Desarrollado por “MySQL AB” como software libre bajo dos tipos de licencia, una bajo GNU GPL y otra para empresas que requieran un uso en productos privativos. “MySQL AB” es una empresa privada que posee y patrocina dicho sistema, esta compañía posee la mayor parte del copyright del código.

Se trata de un sistema multiplataforma que permite gran cantidad de soportes, como por ejemplo el soporte para SSL. En su aplicación en el web está muy ligado a PHP, dado que la conjunción de ambos ofrece una gran operatividad, aunque el uso de MySQL no está restringido a este lenguaje, ya que existen API's que permiten a aplicaciones escritas en C, C++, Perl, Python, Lisp, Java, Ruby entre otras, acceder a las bases de datos MySQL. Es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. Puesto que los entornos web hay baja concurrencia en la modificación pero hay un gran número de lectura de datos, MySQL es ideal para este tipo de aplicaciones.

La elección de este sistema para gestionar la base de datos, se basa básicamente en que proporciona mejores resultados que PostgreSQL (principal alternativa de código abierto) en el entorno para el cual se utilizará y, al contrario que Oracle o Microsoft SQL Server, es un software libre (según el tipo de licencia que necesitemos).

5.1.6. FLASH

Adobe Flash permite la creación de películas mediante gráficos vectoriales e imágenes ráster, que pueden ser incluidas como objetos dentro de las páginas HTML.

Gracias a la utilización de un lenguaje de programación basado en JavaScript llamado ActionScript, se pueden dotar a las películas de dinamismo y de algunas funcionalidades propias de JavaScript. Los recursos que se proporcionan a estas animaciones son muy limitados puesto que pueden suponer una brecha de seguridad en el sistema. Actualmente no se le conocen problemas de seguridad, pero sí que han sido declarados algunos problemas de seguridad en versiones anteriores. Por este motivo en nuestro aplicativo se ha limitado el uso de Flash a la creación de animaciones que cumplen un papel meramente ornamental e informativo.

Finalmente hay que tener en cuenta que para visualizar las animaciones creadas, es necesario un plugin (Flash player). Una vez instalado este plugin el navegador será capaz de reproducir cualquier archivo de flash o shockwave.

5.2. CREACIÓN DE PDF (FPDF)

Para la creación de informes se ha escogido el formato PDF, estas son las siglas de “Portable Document Format” de “Adobe Systems”, cuya traducción sería documento de formato portable. Se trata de un estándar abierto, la cual cosa permite que cualquiera pueda escribir aplicaciones que permitan leer o escribir archivos PDF.

La elección de este tipo de formato se debe a que estos documentos son independientes de la aplicación y del sistema operativo utilizado para crearlos. “Adobe Systems” proporciona

soporte multiplataforma para la visualización y lectura de este tipo de archivos bajo una licencia gratuita.

Para llevar a cabo la generación de PDF de forma dinámica des de la web, se ha utilizado una clase llamada FPDF. Esta clase escrita en PHP y de código abierto, permite la generación de este tipo de documentos mediante un sistema de dibujo de texto basado en coordenadas. Entre las funcionalidades que ofrece, cabe destacar la inclusión de imágenes, enlaces o saltos automáticos de página entre otros.

5.3. CREACIÓN DE DOCUMENTOS RTF

Dados los requisitos funcionales de la creación de archivos editables, principalmente para la generación de contratos y archivos de inmuebles, se han desarrollado un conjunto de funciones para tal efecto.

Las funciones implementadas generan dinámicamente archivos RTF (Rich Text Format). En el caso de la generación de contratos, se utiliza una plantilla del contrato para generar el archivo final. Gracias a dichas funciones, podemos obtener archivos de texto con formato que complementan la información que se puede consultar en la web.

La elección de este tipo de archivo está vinculada al formato interno de éste (mucho más fácil de tratar que el de otros formatos) y al hecho de que puede ser leído y editado posteriormente por prácticamente cualquier procesador de textos, independientemente de la plataforma.

5.4. GENERACIÓN DE ESTADÍSTICAS DINÁMICAS (PHPCHARTS)

La generación de estadísticas de forma dinámica se ha realizado mediante una clase llamada PhpCharts, dicha clase ha sido desarrollada por “PhpScripts.SE” bajo una licencia comercial.

PhpCharts proporciona un conjunto de funciones que nos permiten generar diferentes tipos de gráficas y estadísticas entre las que se incluyen los gráficos de barras, los de evolución, histogramas, etc. La sencillez de las funciones que generan la gráfica hacen que sea una solución ideal para el desarrollo de esta funcionalidad.

Para el correcto funcionamiento de esta clase es necesario el modulo de php `php_gd2`, ese modulo está basado en una librería de PHP para la generación de imágenes. De esta manera PhpCharts es capaz de explotar esta librería para la generación de las gráficas a partir de los datos dinámicos obtenidos mediante las consultas a la base de dato.

5.5. CREACIÓN DEL CALENDARIO (VCALENDAR)

Para la creación de un calendario funcional se decidió utilizar una aplicación externa basada en PHP llamada VCalendar. Dicha aplicación es de código abierto, la cual cosa nos permite añadirle algunas funcionalidades como la de generación de PDF de los eventos diarios.

Vcalendar ofrece posibilidades de configuración así como un control de usuarios y un sistema de gestión de eventos o citas que incluye recordatorios de las mismas, la cual cosa hace de esta aplicación un elemento ideal para la programación de visitas, que es uno de los requisitos funcionales establecidos por los clientes.

Pese a que dicha aplicación es multilenguaje y autoconfigurable, se tuvo que desarrollar un paquete de lenguaje para añadirle soporte para español.

5.6. CREACIÓN DE MAPAS DE LOCALIZACIÓN (APIS DE GOOGLE)

La necesidad vigente de crear mapas de localización donde visualizar la ubicación de un determinado inmueble en el mapa o de generar la ruta mínima entre visitas, nos marca uno de los requisitos funcionales del sistema. Para lograr satisfacer esta necesidad, se ha escogido las Apis de “Google” API Google Maps y API AJAX Search, que se comentan a continuación:

- **GOOGLE MAPS:** Se trata de un conjunto de funciones vinculadas a la conocida empresa, que permite la creación de mapas, así como marcar puntos en éste. La API funciona con latitudes y longitudes de los puntos, pero ofrece un conversor de direcciones a puntos del mapa.
- **AJAX SEARCH:** Nos proporciona un conjunto de funciones también vinculadas a “Google”, que nos permiten, no solo la búsqueda de direcciones del mapa de forma local, sino la mayoría de tipos de búsqueda que esta compañía utiliza para dar servicio a sus clientes.

Ambas Apis son gratuitas y están escritas en JavaScript, con lo que tan solo intervienen en la operación el cliente y el servidor de “Google”.

5.7. HERRAMIENTAS DE DISEÑO DE LA BASE DE DATOS (DBDESIGNER)

Para el diseño visual y de los scripts de creación de tablas y triggers de la base de datos se ha utilizado una versión gratuita de la herramienta DB Designer 4. Esta herramienta integra el diseño visual, modelado, creación y mantenimiento de una base de datos.

Se identifica por las siguientes funcionalidades y modos de funcionamiento:

- Modo diseño: utilizado para la creación y mantenimiento visual de las tablas y sus relaciones.
- Modo query: utilizado para trabajar con los datos de las tablas y crear consultas complejas para su posterior utilización externa.
- Ingeniería inversa desde MySQL, Oracle, MS SQL y ODBC.
- Soporte para índices, entidades débiles y creación automática de claves foráneas
- Generación de scripts de creación, modificación y destrucción de la base de datos a partir del modelo visual.

Finalmente se ha utilizado un parser de XML para que la base de datos generada fuese totalmente compatible.

5.8. HERRAMIENTAS DE DISEÑO DE LAS IMÁGENES (ADOBE PHOTOSHOP)

Todas las imágenes, iconos, etc. utilizados en el web han sido creados específicamente para el uso en este aplicativo. Para ello se han diseñado y creado mediante la aplicación Adobe Photoshop CS de “Adobe Systems”.

Photoshop CS es una aplicación comercial de la compañía “Adobe Systems” que proporciona todas las herramientas para el diseño y creación de imágenes en diferentes formatos. En nuestro caso el formato de las imágenes será JPEG (“Joint Photographic Experts Group”) gracias a la compresión de archivo que este algoritmo, obtenemos unas imágenes que ocupan un espacio reducido, ideales para que su transmisión por Internet no lentifique en exceso la carga de la página.

5.9. SERVIDOR

En el desarrollo de este proyecto, se han empleado dos tipos de servidores, por una banda un servidor de prueba sobre “Microsoft Windows Vista Business” y por otro el servidor final (sobre el cual también se han realizado pruebas) bajo la plataforma “Linux”. Ambos utilizan como servidor web “Apache” comentado a continuación.

5.9.1. APACHE

Apache es un software libre de código abierto y multiplataforma, su nombre se debe a que inicialmente consistía en un conjunto de parches para el servidor “NCSA”, sobre el cual fue diseñado, pese a que posteriormente ha sido totalmente reescrito.

Este servidor presenta una organización muy modular, permitiendo una gran configuración, gracias a la posibilidad de activar o desactivar tanto módulos como parámetros de éstos. Pese a que en ediciones más antiguas de este software se han detectado algunas vulnerabilidades, estas solo podían ser explotadas en determinadas circunstancias por usuarios locales.

Actualmente este software se desarrolla bajo el proyecto de HTTP Server de la “Apache Software Foundation” apoyado por una gran comunidad de usuarios y desarrolladores.

La elección de este software como servidor se debe básicamente al soporte multiplataforma que ofrece y al hecho de tratarse de un software de código abierto.

5.9.2. SERVIDOR DE PRUEBAS

En la fase de implementación y durante una parte de las pruebas llevadas a cabo sobre el sistema, se utilizó software llamado “WAMP” como servidor sobre la plataforma “Microsoft Windows Vista Business”.

“WAMP” es el acrónimo de “Windows Apache MySQL PHP” que son las principales tecnologías necesarias para el desarrollo del proyecto. Se trata de un software de libre y, por tanto, gratuito que nos proporciona el soporte para las principales tecnologías esmentadas.

La elección de este software se basa principalmente en la imposibilidad de desarrollo en la plataforma “Linux” debido al equipo utilizado en éste.

5.9.3. SERVIDOR FINAL

El servidor final de nuestro aplicativo es un servidor “Linux” con “Apache” contratado como servicio de hosting a través de Internet.

Finalmente la empresa decidió la contratación de los servicios de hosting de una compañía que opera a través de internet frente a la alternativa de utilizar un servidor propio para tal efecto. El motivo principal de esta elección fue la falta de infraestructuras para tal efecto y el precio relativamente elevado de un servidor que cumpliese con las características necesarias para satisfacer las necesidades establecidas.

6. PRUEBAS

6.1. TIPOS DE PRUEBA

Tras la implementación del aplicativo, se ha llevado a cabo una fase de pruebas de éste para comprobar el correcto funcionamiento del mismo.

La fase de pruebas habitualmente se infravalora dando por supuesto la correcta implementación por parte de los desarrolladores, acortando así los plazos de entrega del proyecto. Pero nada más lejos de la realidad, puesto que generar un software libre de fallos obviando la fase de pruebas es una utopía, habitualmente sucede que en esta fase se detectan diversos tipos de errores y se incide en un proceso recursivo de implementación – prueba hasta no detectar fallos.

Resulta prácticamente imposible asegurar el correcto funcionamiento de un software bajo cualquier condición (aunque los desarrolladores sean expertos muy cualificados), si bien es cierto que mientras más pruebas se realicen y más variadas sean éstas, se podrán detectar y corregir un mayor número de errores, es muy común que tras la realización del proyecto se reporten al equipo de desarrolladores algunos errores que no fueron detectados en su momento.

Para minimizar los posibles errores y el tiempo que conllevará solucionar los mismos, se debe de validar cada fase y contar con un análisis y diseño robusto. Un mal diseño puede comportar errores tras la implementación, éstos errores de diseño conllevan la modificación de dicho diseño y por tanto la re-implementación de algunas partes del código, con la consiguiente pérdida de tiempo y esfuerzo que ello conlleva. Por tanto, es importante realizar un buen análisis y, posteriormente, un buen diseño para evitar errores de este tipo.

Típicamente debido a retrasos en los plazos de entrega de fases del proyecto anteriores se reduce el número de pruebas a realizar sobre el sistema, pese a que no es una práctica muy recomendable, puede ser necesaria por exigencias en los plazos de entrega al cliente. Pese a ello, se debe ofrecer una garantía de calidad del producto que satisfaga al cliente y por ello se debe tener muy en cuenta el tipo de pruebas a realizar sobre el software.

Sobre el aplicativo implementado se han realizado pruebas de unidad, integración, validación y sistema, divididas en dos fases, la primera fase se llevó a cabo mientras se producía el desarrollo del web y la segunda tras su implementación.

Las pruebas pasadas al sistema se dividen en dos tipos básicos, según la metodología de funcionamiento de estas.

- **PRUEBAS DE CAJA BLANCA:** también llamadas pruebas de cobertura, se basan principalmente en la idea realizar un conjunto de casos en los que se ejecuten el mayor número de líneas de código posible, de esta manera “cubrimos” la mayoría de código, asegurando que los diferentes casos funcionan correctamente. Gracias a este tipo de pruebas se puede detectar el denominado “código muerto”, que no es más que código que nunca se llega a ejecutar bajo ninguna condición.
- **PRUEBAS DE CAJA NEGRA:** Las pruebas de caja negra se basan en proporcionar las entradas del sistema o módulo a analizar y comprobar que las salidas de dicho sistema son las esperadas, sin entrar en detalles del código que conforma dicho sistema.

A continuación se exponen las pruebas realizadas sobre el aplicativo.

6.2. PRUEBAS DE UNIDAD

Este tipo de pruebas consiste en aislar un módulo para comprobar el correcto funcionamiento de este de forma individual, de esta manera nos aseguramos de que el funcionamiento de dicho módulo es correcto. Posteriormente, mediante las pruebas de integración se comprobara que el funcionamiento en conjunción con el resto de módulos es correcto.

Las pruebas de unidad se han implementado mediante pruebas de caja blanca. Estas pruebas se han ido realizando a medida que avanzaba la fase de implementación, verificando el correcto funcionamiento de cada uno de los componentes que configuran las páginas implementadas.

Cada uno de los módulos que realizan las funciones principales de las diferentes secciones, han sido verificados en diversas ocasiones y se han corregido los errores surgidos, intentando eliminar partes de código innecesario.

En nuestro aplicativo han sido testadas la mayoría de funciones implementadas, así como los módulos en PHP, formularios y hojas de estilo CSS, etc. Mediante este tipo de pruebas.

6.3. PRUEBAS DE INTEGRACIÓN

Las pruebas de integración están diseñadas para comprobar el funcionamiento de un conjunto de módulos de forma global, es decir, que los módulos comprobados mediante las pruebas de unidad o pruebas unitarias interactúan correctamente y proporcionan resultados correctos.

Estas pruebas también se han ido realizando durante la fase de codificación o implementación, a medida que se iban interconectando los diferentes módulos se realizaban pruebas de unidad para comprobar la correcta interacción entre estos.

Este tipo de pruebas han sido implementados mediante pruebas de caja negra, ya que los módulos internos habían sido probados previamente gracias a las pruebas de unidad anteriormente comentadas.

6.4. PRUEBAS DE VALIDACIÓN

El objetivo básico de las pruebas de validación es, como su nombre indica, validar la implementación del algoritmo o sistema implementado. Una vez pasadas las pruebas unitarias y de integración, podemos creer que el sistema está libre de errores en tiempo de ejecución (la veracidad de esta afirmación depende en gran medida de la calidad de las pruebas pasadas), la cual cosa no quiere decir que el sistema esté libre de errores lógicos.

Se han utilizado pruebas de caja negra para la validación de cada una de las páginas que conforman el aplicativo, en la mayoría de pruebas realizadas sobre ellas, se han realizado para testar los casos límite de los rangos permitidos, ya que en esta zona suelen aparecer errores.

6.5. PRUEBAS DEL SISTEMA

Este tipo de pruebas, como su nombre indica, están destinadas a la comprobación del correcto funcionamiento de todo el sistema implementado.

Finalmente y de cara a comprobar el funcionamiento del sistema en general, se han realizado algunas pruebas del sistema, probando el funcionamiento de las diferentes secciones que configuran el aplicativo, tanto del área pública, como del área profesional.

7. CONCLUSIONES

7.1. OBJETIVOS CONSEGUIDOS

Como se ha ido explicando a lo largo de esta documentación, se ha seguido un modelo de desarrollo evolutivo, basado en las fases de análisis, diseño, implementación y pruebas. Tanto es así que resulta difícil determinar el número de iteraciones del ciclo de este algoritmo que se han utilizado para dicho desarrollo.

Se han completado todas las fases del proyecto satisfactoriamente, y por tanto, se han alcanzado todos los objetivos marcados inicialmente, satisfaciendo a la empresa solicitante, la cual tras interesarse por el aplicativo implementado ha solicitado su compra.

Por otro lado, se ha elaborado una extensa documentación sobre todas las partes del proyecto, incluyendo información adicional como diagramas, guía de usuario, video tutorial, valoración económica del trabajo realizado, índices de tablas, etc. Satisfaciendo así los objetivos marcados por el proyectista.

7.2. OBJETIVOS NO CONSEGUIDOS

Pese a la repercusión sobre la planificación temporal que ha tenido algunos aspectos como la falta del hardware adecuado, la falta de un espacio acondicionado convenientemente, etc. Se han logrado los objetivos marcados inicialmente.

Durante el desarrollo del aplicativo se detectaron posibles ampliaciones de funcionalidad que podría ser interesante de incluir en el aplicativo, pese a que no se recibió ninguna petición

expresa de la empresa para la inclusión de estas funcionalidades. Éstas están detalladas a continuación (véase 7.3. Posibles ampliaciones y modificaciones).

7.3. POSIBLES AMPLIACIONES Y MODIFICACIONES

Debido a que no existía una petición explícita de la empresa y a la falta de tiempo para desarrollar una aplicación con las estás más funcionalidades de las previstas, no se incluyeron el aplicativo algunas funcionalidades, que se comentarán a continuación y que configuran algunas de las posibles ampliaciones y modificaciones sobre el sistema implementado.

- **FIRMA DIGITAL:** El aplicativo podría permitir la firma digital de contratos que no precisan la presencia de un notario, para corroborar su validez. Este moduló llegó a ser considerado, pero la falta de certificados digitales, infraestructura como lectores de tarjetas, etc. hizo que se incluyera entre las posibles ampliaciones y no entre las funcionalidades implementadas.
- **TÚNEL SSL/TLS:** SSL son las siglas de “Secure Sockers Layers” y nos ofrece la transmisión segura de datos a través de una conexión a internet. Al contrario que otras aplicaciones como IPSec que proporciona seguridad a nivel de red (bajo nivel), SSL nos ofrece seguridad a nivel de transporte, lo que nos permitiría el envío de información cifrada a través de un canal inseguro como es Internet. Así SSL sería una posible ampliación para aumentar la seguridad principalmente útil para el área profesional.

Pese a que solo se han comentado un par de posibles ampliaciones, el sistema se ha desarrollado modularmente y con un diseño en capas, que permite añadir secciones con diversas funcionalidades prácticamente sin tener que modificar el esqueleto de la aplicación.

7.4. CONCLUSIONES FINALES

Este proyecto ha permitido al proyectista que lo ha llevado a cabo realizar un aplicativo surgido de una demanda real, es decir, de una petición de software por parte de una empresa. Por tanto se puede considerar como una “simulación” (aunque finalmente el producto haya sido tangible e incluso se haya vendido a la empresa) del desarrollo de un proyecto de una empresa de desarrollo de software.

Gracias a este proyecto se han adquirido competencias y experiencia en el desarrollo de proyectos y en las diferentes fases en que estos se componen, ya que pese a que la carrera de ingeniería informática está repleta de prácticas en diferentes ámbitos, éstas son de corta duración y no enmarcan todo el trabajo que conlleva el desarrollo de un proyecto real. Al utilizar técnicas como el “brainstorming”, captura de requerimientos, entrevistas, etc. se ha logrado cierta experiencia en el uso de éstas de cara al futuro profesional del proyectista.

Finalmente y, a modo de valoración personal, considero muy satisfactoria la experiencia del desarrollo del proyecto y, sobretodo, las competencias que aporta de cara al futuro profesional