

ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA INFORMÁTICA

OPENRS
(OPEN REAL ESTATE SOFTWARE)

AUTOR: ÁNGEL LUIS BERASUAIN RUIZ

7 de septiembre de 2017

ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA INFORMÁTICA

OPENRS
(OPEN REAL ESTATE SOFTWARE)

DEPARTAMENTO: INGENIERÍA INFORMÁTICA

DIRECTOR: DANIEL MOLINA CABRERA

AUTOR: ÁNGEL LUIS BERASUAIN RUIZ

Cádiz, 7 de septiembre de 2017

FDO: ÁNGEL LUIS BERASUAIN RUIZ

Agradecimientos

Muchas personas han contribuido en mayor o menor medida en la realización del proyecto. Ana, gracias por tu impulso, contigo empezó todo. A mi tutor, Daniel, por su paciencia y buenos consejos. A mi mujer, por su apoyo constante, a mi hermana, por su orientación, y a la empresa TIPYCOS, por su soporte técnico. Gracias de todo corazón.

Pero especialmente, este proyecto se lo dedico a mis padres. Papá y Mamá, al final, lo conseguí (o eso espero). Espero que estéis orgullosos de mi.

Resumen

Este proyecto tiene como objetivo general poner en práctica todos los conocimientos adquiridos a lo largo de la titulación de Ingeniería Técnica en Informática de Gestión e Ingeniería Informática, intentando mejorar las experiencias previas con otros proyectos de desarrollo web aprovechando para implantar una metodología de desarrollo así como un núcleo de software que pueda ser utilizado en otros proyectos futuros.

En 2013, un primer proyecto fue desarrollado para satisfacer las necesidades de una pequeña empresa del sector inmobiliario, GESTICADIZ.

Actualmente, las tecnologías empleadas para aquel proyecto han quedado obsoletas además de ser un software demasiado rígido para las necesidades de las inmobiliarias actuales. Por tanto, se hace necesario retomar con perspectiva y altura de miras aquella idea intentando ofrecer menos para abarcar más, es decir, realizar un sistema con un núcleo básico de funcionalidades pero con la suficiente potencia y flexibilidad para poder expandirse en el futuro y que tenga validez para cualquier inmobiliaria del territorio español.

Hoy día, las inmobiliarias disponen de multitud de herramientas para poder llevar a cabo su actividad diaria así como la promoción de su negocio. El principal problema que radica en las mismas es que, o bien son privativas, y por consiguiente, conllevan un desembolso por parte de la empresa, o aquellas que no lo son, dependen de un dominio u empresa externa para su implantación y mantenimiento.

OPENRS viene a dar un giro a estos enfoques y proporcionar un soplo de aire fresco en el sector, siendo el primer software libre de uso exclusivo para PYMES inmobiliarias que abarca tanto el aspecto de la gestión del negocio como la promoción del mismo, y lo más importante, la propia empresa tiene absoluta libertad para adaptar y modificar su funcionamiento. La inmobiliaria que implante OPENRS en su negocio contará con:

- Un sitio web público, en el que podrá publicar los inmuebles que desee en diferentes categorías y visualizar una amplia gama de detalles, así como disponer de algunas secciones predefinidas útiles para la promoción de la empresa, como noticias, contacto

o el buscador de inmuebles, pero tendrá la libertad de agregar cualquier otra que considere apropiada.

- Un software de gestión privado para poder gestionar los datos más importantes de su negocio, como son, las propiedades que oferte, su cartera de clientes (tanto demandantes como ofertantes) y un módulo demandas para poder cruzar estos datos automática o manualmente. También dispondrá de herramientas para configurar el aspecto del sitio web y las preferencias del sistema.

. En definitiva, el software viene a satisfacer las principales necesidades de gestión y publicación de información de una PYME inmobiliaria bajo un único software sin tener que delegar su actividad en otra empresa, con una interfaz sencilla e intuitiva y totalmente responsive.

Palabras clave: Inmobiliario, CMS, OPEN, Real Estate, Gestión.

Licencia

Copyright (c) 2017 Ángel Luis Berasuain Ruiz.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Índice general

| | |
|--|-----------|
| I Prolegómeno | 1 |
| 1. Introducción | 3 |
| 1.1. Motivación | 3 |
| 1.2. Alcance | 4 |
| 1.3. Glosario de Términos | 5 |
| 1.4. Organización del documento | 6 |
| 2. Estudio Viabilidad | 11 |
| 2.1. Situación actual | 11 |
| 2.1.1. Procesos de Negocio | 13 |
| 2.1.2. Entorno Tecnológico | 14 |
| 2.1.3. Sistemas actuales | 16 |
| Software orientado al cliente (Portales inmobiliarios) | 16 |
| Software orientado a inmobiliarias | 16 |
| 2.1.4. Estructura organizativa | 21 |
| 2.1.5. Fortalezas y Debilidades | 21 |
| 2.2. Necesidades de Negocio | 25 |
| 2.2.1. Objetivos de Negocio | 25 |
| 2.2.2. Procesos de Negocio | 31 |
| 2.3. Objetivos del Sistema | 32 |
| 2.4. Catálogo de Requisitos | 33 |
| 2.4.1. Requisitos funcionales | 33 |
| 2.4.2. Requisitos no funcionales | 36 |
| 2.4.3. Reglas de negocio | 36 |
| 2.4.4. Requisitos de información | 37 |

| | | |
|---------------------------|--|-----------|
| 2.5. | Alternativas de Solución | 41 |
| 2.6. | Solución Propuesta | 42 |
| 3. | Planificación | 45 |
| 3.1. | Metodología de desarrollo | 45 |
| 3.2. | Planificación del proyecto | 51 |
| 3.3. | Organización | 56 |
| 3.3.1. | Recursos Hardware | 57 |
| 3.3.2. | Recursos Software | 58 |
| Dia v0.97.2 | | 58 |
| Latex | | 59 |
| Trello | | 60 |
| Gimp | | 60 |
| AAA Logo | | 61 |
| GanttProject | | 61 |
| Balsamiq Mockups | | 62 |
| 3.4. | Costes | 63 |
| 3.5. | Riesgos | 65 |
| 3.6. | Aseguramiento de calidad | 66 |
| 3.7. | Seguimiento | 70 |
| II | Desarrollo | 73 |
| 4. | Análisis del Sistema | 75 |
| 4.1. | Modelo de datos | 75 |
| 4.1.1. | Modelo Conceptual | 75 |
| 4.1.2. | Modelo lógico | 82 |
| 4.2. | Catálogo de requisitos detallado y Subsistemas de análisis | 86 |
| 4.3. | Modelo de Casos de Uso | 91 |
| 4.3.1. | Actores | 91 |
| 4.3.2. | Diagramas | 92 |
| 4.3.3. | Descripción detallada | 95 |
| Consideraciones generales | | 96 |
| Gestión clientes | | 96 |

| | |
|--|------------|
| Gestión inmuebles | 106 |
| Gestión demandas | 111 |
| Generales | 112 |
| 4.4. Modelo de Comportamiento | 115 |
| 4.5. Modelo de Interfaz de Usuario | 121 |
| 5. Diseño del Sistema | 131 |
| 5.1. Arquitectura del Sistema | 131 |
| 5.1.1. Arquitectura Física | 131 |
| 5.1.2. Arquitectura Lógica | 133 |
| Introducción | 133 |
| Patrón Modelo-Vista-Controlador | 135 |
| Diagramas | 136 |
| Interfaces con sistemas externos | 142 |
| 5.2. Parametrización del software base | 142 |
| 5.3. Diseño físico de Datos | 144 |
| 5.3.1. Motor de almacenamiento MySQL | 144 |
| 5.3.2. Creación de las tablas | 145 |
| 5.3.3. Interacción de módulos con el diseño físico | 148 |
| 5.4. Diseño detallado de Componentes | 150 |
| 5.5. Diseño detallado de la Interfaz de Usuario | 156 |
| 6. Construcción del Sistema | 161 |
| 6.1. Entorno de Construcción | 161 |
| 6.1.1. Frameworks | 161 |
| Bootstrap 3 | 161 |
| Materialize 0.97 | 162 |
| Codeigniter 3.1 | 163 |
| jQuery | 164 |
| 6.1.2. Lenguajes de programación | 165 |
| HTML 5 | 165 |
| PHP 5.5 | 166 |
| CSS 3 | 166 |
| Javascript | 167 |

| | |
|--|------------|
| MySQL v.5.6 | 167 |
| 6.1.3. Herramientas de desarrollo | 169 |
| MySQL Workbench v6.3.6 | 169 |
| PHPMyAdmin v4.1.14 | 169 |
| Git v1.9.15 y Github | 170 |
| SourceTree v1.6.18 | 172 |
| WampServer v2.5 | 172 |
| Netbeans IDE 8.1 | 173 |
| Doxygen v1.8.13 | 174 |
| Composer v1.3.2 | 175 |
| Filezilla v3.27.1 | 176 |
| 6.2. Código Fuente | 176 |
| 6.3. Scripts de Base de datos | 179 |
| 6.4. Buenas prácticas | 180 |
| 6.4.1. Codeigniter | 180 |
| 6.4.2. Bases de datos | 181 |
| 6.4.3. Otras prácticas | 181 |
| 7. Pruebas del Sistema | 183 |
| 7.1. Estrategia | 183 |
| 7.2. Entorno de Pruebas | 184 |
| 7.3. Roles | 184 |
| 7.4. Niveles de Pruebas | 185 |
| 7.4.1. Pruebas Unitarias | 185 |
| 7.4.2. Pruebas de Integración | 185 |
| 7.4.3. Pruebas de Sistema | 186 |
| Pruebas Funcionales | 186 |
| Pruebas No Funcionales | 186 |
| Pruebas de Regresión | 188 |
| 7.4.4. Pruebas de Aceptación | 188 |
| 7.5. Ejemplo práctico de integración de componentes: Librería de Google Maps . | 188 |

| | |
|--|------------|
| III Epílogo | 191 |
| 8. Manual de implantación y explotación | 193 |
| 8.1. Introducción | 193 |
| 8.2. Requisitos previos | 193 |
| 8.3. Inventario de componentes | 194 |
| 8.3.1. Relación de componentes de backend | 194 |
| 8.3.2. Relación de componentes de frontend | 195 |
| 8.4. Procedimientos de instalación | 196 |
| 8.4.1. Inicial | 196 |
| 8.4.2. Migración desde otro servidor con diferente dominio | 196 |
| 8.5. Pruebas de implantación | 197 |
| 8.6. Procedimientos de operación y nivel de servicio | 197 |
| 8.7. Limitaciones conocidas | 198 |
| 9. Manual de usuario | 201 |
| 9.1. Introducción | 201 |
| 9.2. Zona privada | 203 |
| 9.2.1. Acceso al sistema | 203 |
| 9.2.2. Estructura general | 205 |
| 9.2.3. Perfil de Agente Inmobiliario | 207 |
| Estadísticas | 207 |
| Inmuebles | 211 |
| Clientes | 237 |
| Demandas | 239 |
| 9.2.4. Perfil de Administrador | 245 |
| Estadísticas | 245 |
| Usuarios | 245 |
| Copias de seguridad | 247 |
| Configuración | 248 |
| Configuración de zona pública | 252 |
| 9.3. Web Corporativa | 257 |
| 9.3.1. Cabecera | 257 |
| 9.3.2. Inicio | 258 |

| | |
|---|------------|
| 9.3.3. Detalle del inmueble | 260 |
| 9.3.4. Nosotros | 264 |
| 9.3.5. Servicios | 264 |
| 9.3.6. Noticias | 265 |
| 9.3.7. Contacto | 267 |
| 9.3.8. Pie de página | 267 |
| 10. Conclusiones | 269 |
| 10.1. Dimensiones del proyecto | 269 |
| 10.2. Objetivos alcanzados | 270 |
| 10.3. Dificultades | 271 |
| 10.4. Lecciones aprendidas | 274 |
| 10.5. Trabajo futuro | 276 |
| 10.6. Valoración personal | 277 |
| Bibliografía y referencias en internet | 277 |
| Información sobre Licencia | 283 |
| 1. APPLICABILITY AND DEFINITIONS | 283 |
| 2. VERBATIM COPYING | 285 |
| 3. COPYING IN QUANTITY | 286 |
| 4. MODIFICATIONS | 286 |
| 5. COMBINING DOCUMENTS | 289 |
| 6. COLLECTIONS OF DOCUMENTS | 289 |
| 7. AGGREGATION WITH INDEPENDENT WORKS | 289 |
| 8. TRANSLATION | 290 |
| 9. TERMINATION | 290 |
| 10. FUTURE REVISIONS OF THIS LICENSE | 291 |
| 11. RELICENSING | 291 |

Índice de figuras

| | |
|---|----|
| 2.1. Comparativa sectorial de la adopción de la tecnología en microempresas (año 2013) | 12 |
| 2.2. Comparativa sectorial de la adopción de la tecnología en microempresas (año 2014) | 13 |
| 2.3. Proceso de negocio actual del demandante | 13 |
| 2.4. Proceso de negocio actual del ofertante | 14 |
| 2.5. Proceso de negocio en el sistema de información del demandante | 31 |
| 2.6. Proceso de negocio en el sistema de información del ofertante | 31 |
| | |
| 3.1. Modelo en espiral de Boehm [2] | 49 |
| 3.2. Ciclo de vida con reutilización de componentes | 50 |
| 3.3. Planificación temporal estimada (Parte 1) | 53 |
| 3.4. Planificación temporal estimada (Parte 2) | 54 |
| 3.5. Planificación temporal estimada (Parte 3) | 55 |
| 3.6. Logo DIA | 58 |
| 3.7. Logo LateX | 59 |
| 3.8. Logo Trello | 60 |
| 3.9. Logo Gimp | 60 |
| 3.10. Logo AAA Logo | 61 |
| 3.11. Logo Gantt Project | 61 |
| 3.12. Logo Balsamiq Mockups | 62 |
| 3.13. Modelo de verificación en W | 66 |
| 3.14. Diagrama de Flujo GIT | 67 |
| 3.15. Evolución de estrategia Develop-Test-Deploy | 69 |
| 3.16. Evolución real de horas diarias empleadas | 72 |

| | | |
|-------|--|-----|
| 4.1. | Diagrama E/R inicial | 76 |
| 4.2. | Diagrama de clases del Subsistema de Inmuebles | 78 |
| 4.3. | Diagrama de clases del Subsistema de Clientes | 79 |
| 4.4. | Diagrama de clases del Subsistema de Demandas | 80 |
| 4.5. | Diagrama de clases de la Documentación | 81 |
| 4.6. | Diagrama Casos de Uso: Sitio web (Zona pública) | 92 |
| 4.7. | Diagrama Casos de Uso: Software de gestión (Zona privada) | 92 |
| 4.8. | Diagrama Casos de Uso: Gestión de clientes | 93 |
| 4.9. | Diagrama Casos de Uso: Gestión de inmuebles | 94 |
| 4.10. | Diagrama Casos de Uso: Gestión de demandas | 95 |
| 4.11. | Diagrama secuencia: Insertar cliente - Escenario normal | 115 |
| 4.12. | Diagrama secuencia: Insertar cliente - Escenario alternativo | 116 |
| 4.13. | Diagrama secuencia: Editar cliente | 118 |
| 4.14. | Diagrama secuencia: Eliminar cliente | 119 |
| 4.15. | Diagrama secuencia: Buscar cliente | 120 |
| 4.16. | Mockup inicial de zona privada | 121 |
| 4.17. | Mockup inicial del cliente | 122 |
| 4.18. | Mockup inicial del inmueble | 123 |
| 4.19. | Mockup inicial de la demanda | 124 |
| 4.20. | Mockup final del inmueble | 125 |
| 4.21. | Mockup responsive del inmueble | 126 |
| 4.22. | Plantilla de diseño pública (Parte 1) | 127 |
| 4.23. | Plantilla de diseño pública (Parte 2) | 128 |
| 4.24. | Mapa de navegación del cliente | 129 |
| 5.1. | Diagrama de despliegue | 132 |
| 5.2. | Patrón Layers (Capas) | 133 |
| 5.3. | Flujo de ejecución en Codeigniter 3 | 135 |
| 5.4. | Diseño de controladores | 137 |
| 5.5. | Diseño de modelos | 138 |
| 5.6. | Diseño de software de gestión | 139 |
| 5.7. | Diseño de sitio web | 140 |
| 5.8. | Nivel de acceso de KCFINDER a ficheros adjuntos | 141 |
| 5.9. | Interacción de módulos y composición de tablas | 149 |

| | |
|--|-----|
| 5.10. Diagrama de acceso al sistema | 150 |
| 5.11. Diagrama de secuencia - Insertar Cliente (Flujo normal) | 151 |
| 5.12. Diagrama de secuencia - Insertar Cliente (Error) | 152 |
| 5.13. Diagrama de secuencia - Mostrar mapa de inmuebles de buscador | 154 |
| 5.14. Diagrama de clases sobre el proceso de generación de Documentación | 155 |
| 5.15. Buscador de clientes - Cargar Poblaciones | 156 |
| 5.16. Buscador de clientes - Mostrar Calendario | 156 |
| 5.17. Clientes - Exportar CSV | 157 |
| 5.18. Clientes - Importar CSV | 157 |
| 5.19. Clientes - Interfaz Insertar | 158 |
| 5.20. Clientes - Visualizar dirección en Google Maps | 158 |
| 5.21. Clientes - Insertar con éxito | 159 |
| 5.22. Clientes - Insertar con errores | 159 |
| 6.1. Logo Bootstrap | 161 |
| 6.2. Logo Materialize | 162 |
| 6.3. Logo CodeIgniter | 163 |
| 6.4. Logo Jquery | 164 |
| 6.5. Logo HTML 5 | 165 |
| 6.6. Logo PHP 5.5 | 166 |
| 6.7. Logo CSS 3 | 166 |
| 6.8. Logo JavaScript | 167 |
| 6.9. Logo MYSQL | 167 |
| 6.10. Logo MYSQL Workbench | 169 |
| 6.11. Logo GIT y GITHUB | 170 |
| 6.12. Logo SourceTree | 172 |
| 6.13. Logo WAMP Server | 172 |
| 6.14. Logo NetBeans | 173 |
| 6.15. Logo Doxygen | 174 |
| 6.16. Logo Composer | 175 |
| 6.17. Logo Filezilla | 176 |
| 6.18. Implementación de constructor de clientes | 177 |
| 6.19. Implementación de funcionalidad de Insertar Cliente | 178 |
| 6.20. Creación de tabla de fichas de los clientes | 179 |

| | |
|--|-----|
| 6.21. Creación de vista de inmuebles | 179 |
| 9.1. Interfaz de acceso al sistema | 203 |
| 9.2. Interfaz de recuperación de contraseña olvidada | 204 |
| 9.3. Correo de recuperación de contraseña olvidada | 204 |
| 9.4. Interfaz de recuperación de cambiar contraseña | 205 |
| 9.5. Estadísticas: Ejemplo de configuración de vista interfaces | 206 |
| 9.6. Estadísticas Personales: Inmuebles | 208 |
| 9.7. Estadísticas Personales: Clientes | 209 |
| 9.8. Estadísticas Personales: Demandas | 210 |
| 9.9. Inmuebles: Buscador | 212 |
| 9.10. Inmuebles: Búsqueda Avanzada | 213 |
| 9.11. Inmuebles: Vista de publicación | 216 |
| 9.12. Inmuebles: Vista pública en Google Maps | 217 |
| 9.13. Inmuebles: Vista real en Google Maps | 217 |
| 9.14. Inmuebles: Vista de coincidencias de marcas en Google Maps | 218 |
| 9.15. Inmuebles: Exportar CSV | 219 |
| 9.16. Inmuebles: Formulario de alta de inmueble (Parte 1) | 220 |
| 9.17. Inmuebles: Formulario de alta de inmueble (Parte 2) | 221 |
| 9.18. Inmuebles: Subir CSV para importación | 221 |
| 9.19. Inmuebles: Resultado validación de CSV | 223 |
| 9.20. Inmuebles: Resultado importación de CSV | 224 |
| 9.21. Inmuebles: Editar (Parte 1) | 226 |
| 9.22. Inmuebles: Asignar demandas | 226 |
| 9.23. Inmuebles: Crear demanda | 227 |
| 9.24. Inmuebles: Editar (Parte 2) | 227 |
| 9.25. Inmuebles: Publicación (Parte 1) | 228 |
| 9.26. Inmuebles: Publicación (Parte 2) | 229 |
| 9.27. Inmuebles: Listado de imágenes | 230 |
| 9.28. Inmuebles: Listado responsive de imágenes | 231 |
| 9.29. Inmuebles: Slider de imágenes | 232 |
| 9.30. Inmuebles: Subir imágenes | 232 |
| 9.31. Inmuebles: Progreso al subir imágenes | 233 |
| 9.32. Inmuebles: Listado de enlaces | 233 |

| | |
|--|-----|
| 9.33. Inmuebles: Insertar enlace | 234 |
| 9.34. Inmuebles: Generar cartel | 234 |
| 9.35. Inmuebles: Editar cartel | 235 |
| 9.36. Inmuebles: Listado de ficheros adjuntos | 236 |
| 9.37. Inmuebles: Adjuntar fichero | 236 |
| 9.38. Clientes: Búsqueda avanzada | 237 |
| 9.39. Clientes: Insertar | 238 |
| 9.40. Demandas: Búsqueda avanzada | 239 |
| 9.41. Demandas: Buscador | 240 |
| 9.42. Demandas: Sin filtros de búsqueda | 241 |
| 9.43. Demandas: Proponer inmuebles | 241 |
| 9.44. Demandas: Inmuebles propuestos | 242 |
| 9.45. Demandas: Modificar inmueble propuesto | 242 |
| 9.46. Demandas: Con filtros de búsqueda | 244 |
| 9.47. Estadísticas Generales: Inmuebles | 245 |
| 9.48. Usuarios: Listado | 246 |
| 9.49. Usuarios: Editar | 246 |
| 9.50. Copias de Seguridad: Listado | 247 |
| 9.51. Copias de Seguridad: Realizar | 247 |
| 9.52. Configuración: Estados | 249 |
| 9.53. Configuración: Plantilla de documentación | 250 |
| 9.54. Configuración: Zonas de municipios | 251 |
| 9.55. Gestión de Zona Pública: Idiomas | 253 |
| 9.56. Gestión de Zona Pública: Aspecto | 253 |
| 9.57. Gestión de Zona Pública: Secciones | 254 |
| 9.58. Gestión de Zona Pública: Detalle de sección | 254 |
| 9.59. Gestión de Zona Pública: Tipo de bloque Slider | 255 |
| 9.60. Gestión de Zona Pública: Noticias | 255 |
| 9.61. Gestión de Zona Pública: Detalle Noticia | 256 |
| 9.62. Web Corporativa: Cabecera | 257 |
| 9.63. Web Corporativa: Slider | 258 |
| 9.64. Web Corporativa: Buscador de inmuebles | 258 |
| 9.65. Web Corporativa: Oportunidades | 258 |

| | |
|---|-----|
| 9.66. Web Corporativa: Resultados de búsqueda | 259 |
| 9.67. Web Corporativa: Mapa interactivo | 259 |
| 9.68. Web Corporativa: Mapa | 260 |
| 9.69. Web Corporativa: Vídeo | 260 |
| 9.70. Web Corporativa: Galería de imágenes | 261 |
| 9.71. Web Corporativa: Enlaces | 261 |
| 9.72. Web Corporativa: Contacto del inmueble | 262 |
| 9.73. Web Corporativa: Datos del Inmueble (Parte 1) | 263 |
| 9.74. Web Corporativa: Datos del Inmueble (Parte 2) | 263 |
| 9.75. Web Corporativa: Nosotros | 264 |
| 9.76. Web Corporativa: Servicios | 264 |
| 9.77. Web Corporativa: Listado de noticias | 265 |
| 9.78. Web Corporativa: Detalle de noticia | 266 |
| 9.79. Web Corporativa: Contacto | 267 |
| 9.80. Web Corporativa: Pie de página | 267 |

Índice de tablas

| | |
|---|-----|
| 3.1. Planificación inicial de actividades | 52 |
| 3.2. Iteraciones para la construcción del software | 52 |
| 3.3. Estimación de Costes del proyecto (estimación segun CPIIA) | 64 |
| 3.4. Estimación de Costes del proyecto (estimación segun tarifa autónomo) | 64 |
| 3.5. Riesgos | 65 |
| 3.6. Costes reales de recursos empleados en el proyecto | 70 |
| 3.7. Coste temporal real de actividades | 71 |
| 3.8. Coste temporal real de Iteraciones para la construcción del software | 71 |
| 8.1. Comparativa de componentes de backend | 195 |
| 8.2. Comparativa de componentes de frontend | 195 |
| 10.1. Comparativa de componentes | 274 |
| 10.2. Comparativa de tecnologías | 275 |
| 10.3. Comparativa de procesos y herramientas | 275 |

Parte I

Prolegómeno

Capítulo 1

Introducción

1.1. Motivación

Todo comenzó en marzo de 2013 cuando mi hermana empezó a colaborar con la inmobiliaria GESTICADIZ como tasadora y certificadora energética. Me comentó que necesitaban una pequeña web para ofertar sus inmuebles y sus servicios y darse un poco a conocer.

Al principio, empezó solo con un buscador simple de inmuebles en una plantilla bastante antigua que ya usaba en otros proyectos anteriores. Además contenía algunas secciones especificadas explícitamente por la inmobiliaria como en qué consistía la certificación energética (en aquel entonces todavía no existía decreto que lo regulara), sección de noticias, buscador de inmuebles, etc.

Rápidamente, en los siguientes meses, la gerente de GESTICADIZ, Gloria Chamorro, viendo el potencial que ofrecía poder organizar la información de su propio negocio en Internet y tenerla publicada al instante, este software se fue ampliando pero con una clara orientación a satisfacer las necesidades particulares de su inmobiliaria.

En cambio, yo, por otro lado, viendo el potencial de desarrollo que dentro del ámbito del mercado inmobiliario podía tener este software, me comprometí a terminarlo pero a cambio de que este quedaría como software libre ya que serviría como base de estudio y desarrollo para el proyecto actual, el cual, pretende adaptar y generalizar lo que en su día se construyó como una aplicación personalizada sólo para GESTICADIZ.

Actualmente las inmobiliarias disponen de multitud de herramientas para poder llevar a cabo su actividad diaria así como la promoción de su negocio. En resumen, podríamos decir que el principal problema que radica en las mismas es que, o bien son privativas, y por consiguiente, conllevan un desembolso por parte de la empresa. Aquellas que no lo

son, dependen de un dominio u empresa externa para su implantación y mantenimiento. Por otro lado, otras, como la que se elaboró a medida para GESTICADIZ, no son lo suficientemente flexibles.

Por tanto, OPENRS viene a dar un giro a estos enfoques y proporcionar un soplo de aire fresco en el sector, siendo el primer software libre de uso exclusivo para PYMES inmobiliarias españolas que abarca tanto la gestión de los datos más relevantes del negocio como la promoción pública del mismo, y lo más importante, la propia empresa tiene absoluta libertad para adaptar y modificar su funcionamiento.

Por otro lado, también me motivó el hacer este proyecto el poder renovar mi entorno tecnológico usado hasta la fecha con el objetivo de dar un salto evolutivo en el ámbito de la ingeniería del software web que me permitiera ser más competitivo como analista y programador, así como practicar metodologías y técnicas que permitan desarrollar con mayor calidad y fiabilidad el software.

1.2. Alcance

El sistema que se desea construir será conocido como OPENRS (OPEN Real Estate Software). Es un software pensado para satisfacer las necesidades de una inmobiliaria pequeña o mediana que dispongan tanto de un hosting propio o usen un software comercial externo. Dentro de la organización estándar de una inmobiliaria, este software se dirige principalmente a los agentes inmobiliarios.

Los clientes interactuarán con la zona pública pero tendrán un rol pasivo. Podrán visualizar los datos definidos desde la zona privada como públicos y ponerse en contacto con la inmobiliaria, pero no dispondrán de una zona privada como ocurre con los agentes inmobiliarios.

Los dos grandes objetivos que pretende alcanzar el software son:

- Gestionar los datos más relevantes del negocio así como su documentación. Entre ellos cabe destacar los datos de las propiedades o inmuebles que oferta, su cartera de clientes (tanto demandantes como ofertantes) y las demandas tanto automáticas como manuales. Además, contará con una zona de administración en las que, a parte de la configuración de los datos que se van a utilizar en el sistema y sus preferencias, dispondrá de un CMS para la configuración del aspecto y los contenidos de la zona pública.
- Dar a conocer la empresa así como sus inmuebles y servicios. Para ello la inmobiliaria

deberá contar con un sitio web conectado con la zona privada anterior, en el que podrá publicar los inmuebles que deseé en diferentes categorías y visualizar una amplia gama de detalles, así como disponer de algunas secciones predefinidas, útiles para la promoción de la empresa como noticias, contacto o el buscador de inmuebles, pero tendrá la libertad de agregar cualquier otra que considere apropiada.

Posteriormente, se desarrollará con más detalle los requisitos particulares que se desglosan a partir de los especificados anteriormente.

1.3. Glosario de Términos

Ofertante: Persona que ofrece un inmueble a la inmobiliaria en cualquier tipo de modalidad, alquiler o venta.

Demandante: Persona que demanda un inmueble a la inmobiliaria ya sea porque desea alquilarlo o comprarlo.

SEO: Search Engine Optimization (Optimización para Motores de Búsqueda).

CRM: Customer Relationship Management (Gestión de relaciones con los clientes).

ERP: Enterprise Resource Planning (Planeamiento de Recursos Empresariales).

CMS: Content Management System (Sistema de gestión de contenidos).

DSS: Diagrama de secuencia del sistema.

DS: Diagrama de secuencia.

ER: Referido principalmente al modelo Entidad-Relación.

IDE: Integrated Development Environment (Entorno de desarrollo integrado).

HTTP: Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto).

SGBD: Sistema de gestión de base de datos.

PHP: Hypertext Preprocessor (Preprocesador de Hipertexto). Lenguaje de programación usado en el Back-End.

HTML: HyperText Markup Language (Lenguaje de marcas de hipertexto).

CSS: Cascade Style Sheet (Hoja de estilo en cascada).

JS: JavaScript (lenguaje de programación interpretado, dialecto del estándar ECMAScript usado en el Front-End).

Responsive: Técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos.

API: Application Programming Interface (Interfaz de programación de aplicaciones).

Frontend: Parte del software que interactúa con los usuarios (comúnmente conocido como lado del cliente).

Backend: Parte que procesa la entrada desde el front-end (comúnmente conocido como lado del servidor).

GIT: Sistema de control de versiones distribuido.

PCS: Propuesta de cambio de software.

Módulo: Componente software compuesto, al menos, por un controlador, un modelo y una vista con el objetivo de desarrollar un conjunto de funcionalidades específicas.

MVC: Referido al patrón de diseño modelo-vista-controlador.

CRUD: Referido a módulos o funcionalidades que realizar operaciones de comunes como actualizar, insertar, eliminar y leer datos del sistema.

WYSIWYG: es el acrónimo de What You See Is What You Get (en español, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

Web corporativa - Zona pública: Son sinónimos y se utilizaran indistintamente refiriéndose al sitio web de la inmobiliaria de carácter público.

Software Gestión - Zona privada: Son sinónimos y se utilizaran indistintamente refiriéndose a aquella zona que administra todos los contenidos de la zona pública y los datos de gestión de la inmobiliaria y necesita un acceso privado.

SSL: Acrónimo de Secure Socket Layer, es la tecnología de seguridad estándar para el establecimiento de un enlace encriptado entre un servidor web y un navegador. Este enlace seguro garantiza que todos los datos transferidos se mantienen privados.

PSR: PHP Standard Recommendation: referido a determinadas guías o sugerencias sobre el estilo de codificación en PHP.

1.4. Organización del documento

La primera parte (**Prolegómeno**) de la memoria contendrá:

- **Introducción:** que describa brevemente en qué consiste y por qué se ha realizado el proyecto.

- **Estudio Viabilidad:** Para ello primero se evaluará la situación actual del sector inmobiliario y sus necesidades. Como caso de estudio en particular, utilizaremos el de la empresa GESTICADIZ. Estableceremos los objetivos del sistema y el conjunto de requisitos que los satisfacen. Finalmente, evaluaremos varias propuestas de solución que se han tenido en cuenta para llevar el proyecto a cabo y cuál ha sido la solución elegida. Aunque esta sección se titule Estudio de viabilidad, también define los requisitos del sistema sobre los cuales se realizará posteriormente el análisis.
- **Planificación:** Todas las métricas existentes incluyen la planificación como una etapa anterior al estudio de viabilidad. No obstante, debido a la complejidad para definir el alcance de la aplicación así como la solución adoptada, y sobre todo, para que fuera más fácil entender y planificar con exactitud las iteraciones del proceso de construcción del software, hemos decidido tanto mi tutor como yo, que esta era la estructura más adecuada a seguir en la memoria, sobre todo porque aclara la compresión del sistema por parte del lector.

En la segunda parte (**Desarrollo**), se describirá el desarrollo del proyecto siguiendo una determinada metodología de software. Esta sección vendrá marcada por las diferentes etapas que se dan en un proceso de ingeniería del software:

- **Análisis del sistema:** A partir del catálogo de requisitos, realizaremos un análisis acorde las tres perspectivas en las que, tradicionalmente, la ingeniería del software ha utilizado para describir el funcionamiento del software. Información (modelo conceptual), Funcionalidad (modelo de casos de uso) y Temporal (Modelo de comportamiento). Se incluirá también un modelado de interfaces de usuario para que nos sirva como prototipado inicial.
- **Diseño del sistema:** Se describirá la arquitectura del software realizando un estudio de los componentes software utilizados para implementar el proyecto, las diferentes interacciones entre ellos, qué parametrizaciones se han realizado sobre el software de base utilizado así como un diseño detallado de la interfaz de usuario. Finalmente, como conclusión del modelo de datos se realiza un diseño físico a partir del modelo conceptual.
- **Construcción del sistema:** Se describirá los detalles sobre el entorno de construcción empleado, las mejores prácticas utilizadas y descripción detallada de librerías y componentes de software empleados.

- **Pruebas del sistema:** En este capítulo se presenta el plan de pruebas del sistema de información, incluyendo los diferentes tipos de pruebas que se han llevado a cabo, tanto manuales como automatizadas así como los entornos empleados.

En la tercera parte (**Epílogo**), se incluirán todos aquellos anexos que sirvan tanto para el despliegue del sistema como manuales con otras consideraciones como:

- **Manual de usuario.**
- **Manual de instalación y explotación.**
- **Conclusiones.**
- **Licencia de la documentación del software.**

Finalmente se incluirá un apartado con la **Bibliografía**.

El software entregado en soporte informático vendrá dividido en las siguiente partes:

- **Código:** Última versión del código del sistema descargado desde su repositorio. La cual, además del código fuente incluye:
 - **PCS:** Documentación de los cambios realizados en las diferentes iteraciones del software que además de ofrecernos cómo y cuando se ha ido realizando la construcción del sistema, se propone un listado detallado de futuras mejoras.
 - **Matrices de pruebas:** Conocidas también como 'check lists', son una lista de comprobaciones, principalmente funcionales, que se realizarán para verificar que el sistema cumple con lo especificado y forman parte de las pruebas de sistema 7.4.3.
 - **Base de datos:** Script de bases de datos y versiones de modelado realizadas en MYSQL Workbench.
 - **Documentación online:** Realizada con Doxygen en HTML.
- **Memoria:** Código fuente en latex de la memoria así como el PDF final.
- **Resumen:** Resumen del proyecto acorde al modelo suministrado por la UCA.
- **Anexos:**
 - Incluye el software de gestión utilizado por GESTICADIZ (la base de lo que iba a ser OpenInmoCMS).

- Documentación de metodología de Métrica v.3.
- **Retos:** También podríamos llamarlo diario de a bordo, viene a recoger lo que se ha ido realizando cada día.
- **Librerías:** Código de las librerías utilizadas en el sistema así como de las evaluadas pero que por distintos motivos luego no se han empleado.
- **APIs:** Documentación de API de Idealista para búsqueda de inmuebles (Propuesto para futuras mejoras como la captación de clientes).

Capítulo 2

Estudio Viabilidad

2.1. Situación actual

Actualmente, las PYMES, en particular las microempresas, representan la gran mayoría del tejido empresarial español. En el ámbito inmobiliario siguen la misma dinámica, y la mayoría de inmobiliarias, generalmente, poseen sólo una sede y disponen de 1-5 empleados. Además desde el comienzo de la crisis económica, allá por el año 2008, cuando el mercado inmobiliario se derrumbó, ha tenido una paulatina pérdida de relevancia en el sector empresarial.

Cómo se observan en las siguientes gráficas, el sector inmobiliario ha perdido influencia en el uso de herramientas tecnológicas, principalmente, debido al coste de algunas de ellas de carácter eminentemente comercial.

Dentro del conjunto de estas herramientas de carácter comercial vamos a centrarnos en analizar aquellas que ayudan a la gestión de los datos y la promoción de la inmobiliaria ya que es el ámbito en el que va a actuar el software.

Por tanto, se hace necesario desarrollar, o al menos contribuir, en la construcción de herramientas de software libre que abaraten los costes de producción de la inmobiliaria así como que ayuden a organizar los datos de los diferentes procesos de negocio.

Por otro lado, decir inmobiliaria es sinónimo de publicidad, ya que su principal fuente de ingresos es la intermediación entre ofertantes y demandantes. Además, como comentaba anteriormente, la crisis económica ha hecho que diversos profesionales, con el objetivo de abaratrar también sus costes de producción, se hayan asociado con otras inmobiliarias ofreciendo diferentes servicios:

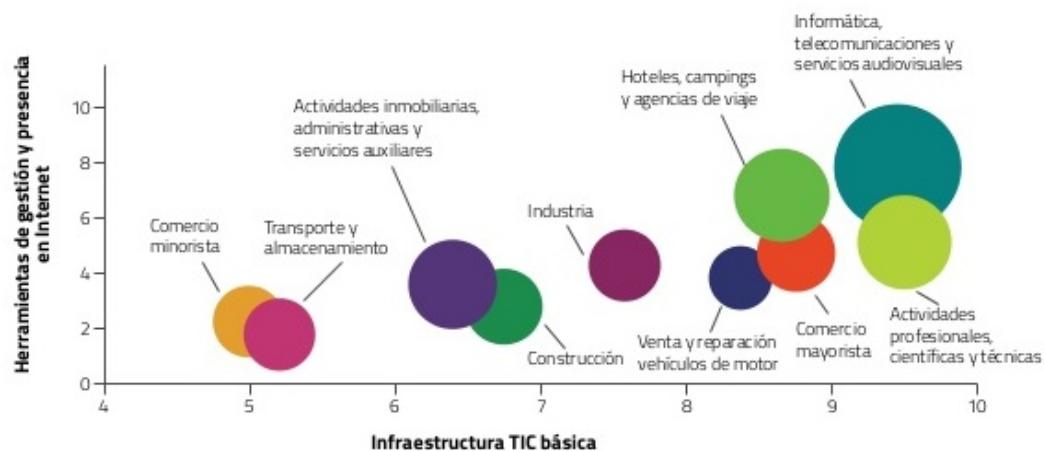
- Certificadores energéticos.

- Decoradores e interioristas.
- Arquitectos, peritos y albañiles.
- Otros agentes inmobiliarios.

En definitiva, tanto la gestión de los datos principales de los procesos de negocio así como la promoción de bienes y servicios se hace indispensable en el día a día del sector.

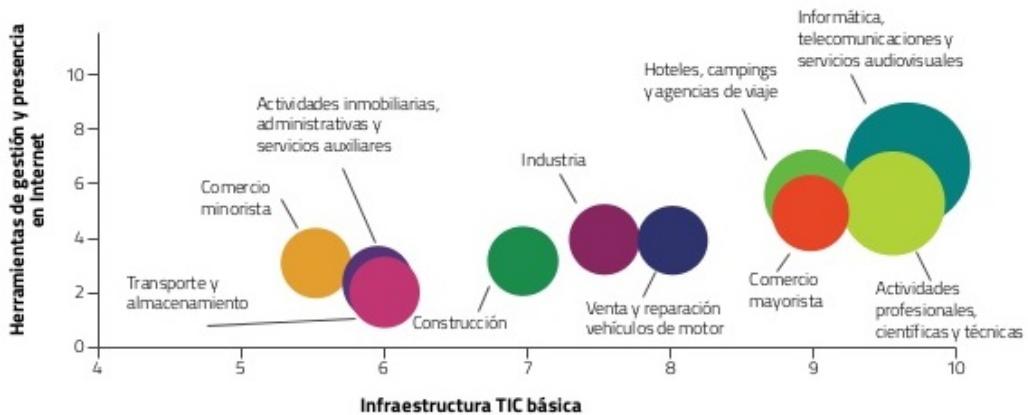
A partir de ahora, aunque se haga un análisis generalizado sobre el sector inmobiliario, se tomará como caso de estudio la posible implantación en la empresa GESTICADIZ.

La principal razón por la cual se ha cogido esta empresa, además de por mi vinculación personal con la misma, es porque por estadio o entorno tecnológico actual y estructura organizativa, encaja con uno de los posibles entornos a los cuales va dirigido la aplicación.



Fuente: CNTSI a partir de datos INE 2013. Base: total de empresas.

Figura 2.1: Comparativa sectorial de la adopción de la tecnología en microempresas (año 2013)



Fuente: CNTSI a partir de datos INE 2014. Base: total de empresas.

Figura 2.2: Comparativa sectorial de la adopción de la tecnología en microempresas (año 2014)

2.1.1. Procesos de Negocio

Como comentaba anteriormente, el negocio inmobiliario se basa, principalmente en poner en contacto a un ofertante con un demandante, es decir, en la intermediación. Para entender este modelo, se van a ofrecer dos diagramas desde los puntos de vista de ambos tipos de clientes así como la participación del agente inmobiliario en el proceso.

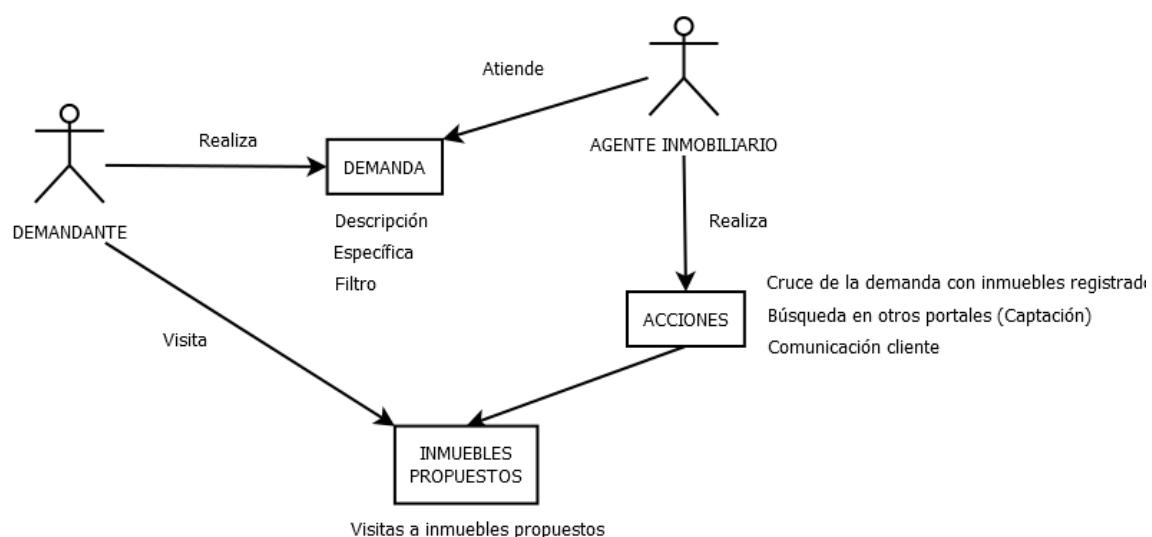


Figura 2.3: Proceso de negocio actual del demandante

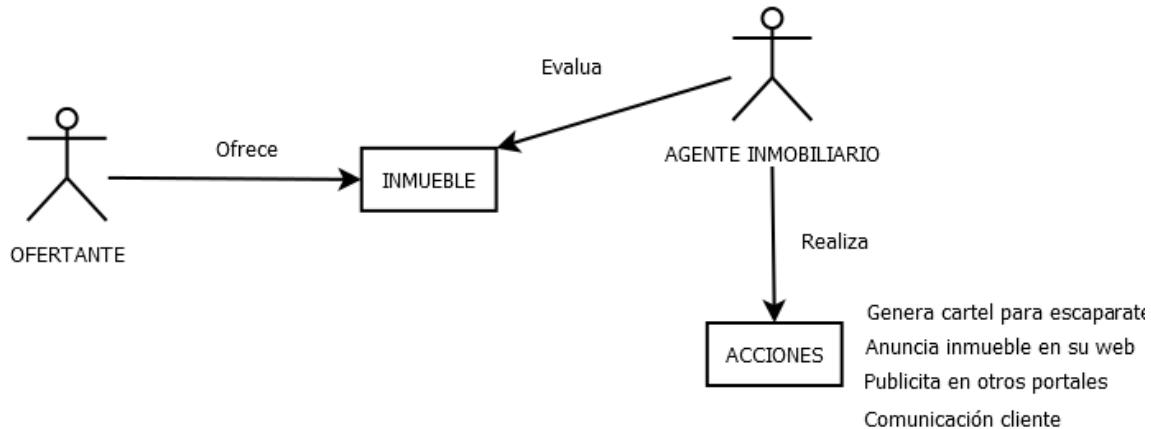


Figura 2.4: Proceso de negocio actual del ofertante

2.1.2. Entorno Tecnológico

Algunas preguntas que se han realizado para poder llevar a cabo el estudio son:

- ¿Tiene sitio web que promociona la inmobiliaria?
- ¿Tiene software de gestión?
- ¿Usa algún CRM? ¿ERP? ¿Conocen lo que son?
- ¿El nivel de flexibilidad en cuanto a configuración cual es?
- ¿Disponen de hosting propio o de software en la nube?
- ¿El mantenimiento del software lo realizan ustedes o una empresa externa?
- ¿Usan algún software de escritorio? ¿Herramientas de bases de datos tipo Access u otros software como Excel?
- ¿Publican su contenido en algún portal inmobiliario?
- ¿Cuántos inmuebles les gustaría registrar?

Algunas inmobiliarias que acaban de comenzar su andadura en el negocio o aquellas que tienen un enfoque tradicional suelen tener un entorno tecnológico básico, compuesto generalmente por:

- Documentos en papel.
- Documentos digitales (escaneados o de software ofimático).

- Bases de datos en access o relación de datos en excel.
- Software de gestión de escritorio instalado en algún equipo compartido en su intranet.
Sobre este punto se analizarán más detalles posteriormente.

Adicionalmente a estos sistemas, las inmobiliarias suelen contar con otros software, los cuales, se podrían agrupar en dos grandes grupos:

1. El software de gestión y promoción de la inmobiliaria está todo o en parte en un hosting de la empresa. Normalmente, suelen contar sólo con un sitio web que promociona el negocio. Esta suele ser implementado a través de algún CMS (usualmente WordPress) o una página web estática convencional. Además pueden tener implementado un pequeño software de gestión privado desarrollado por terceros pero realizado, generalmente, a medida. Este puede estar integrado o no con el software descrito anteriormente.
2. El software de gestión y/o promoción de la inmobiliaria está subcontratado todo o en parte en un hosting en la nube: Esta es la tendencia más actual, la cual consiste en contratar la implantación de todos los servicios que necesita la inmobiliaria en un software todo en uno (all-in-one) en la nube (in cloud). Estos pueden ir desde CMS que sólo gestionan la parte pública hasta sistemas complejos que suelen incluir tanto la web pública como un CRM conectado con la misma con diferentes funcionalidades (dependiendo del software contratado).

En ambos tipos de modelos de negocio descritos anteriormente, el entorno tecnológico clásico suele convivir con cualquiera de las dos opciones elegidas. Se verá un análisis más profundo en la sección de Sistemas actuales.

Nuestro caso de estudio en concreto, GESTICADIZ, se sitúa en un estadio tecnológico de nivel 1, ya que actualmente posee un hosting propio en 1AND1 y además:

- Un software de gestión, desarrollado por el autor de este proyecto en 2013 (con diferentes funcionalidades que se comentarán posteriormente).
- Posee una pequeña zona pública realizada por una empresa externa que publica la información de los inmuebles registrados en el software anterior.

2.1.3. Sistemas actuales

El análisis realizado en esta parte es, posiblemente, uno de los que más tiempo y más ha costado comprender debido a la saturación y variedad de alternativas que existen. Me gustaría exponer algunos software que se han analizado, ya sea para tomar ideas o para ver la perspectiva de negocio.

Software orientado al cliente (Portales inmobiliarios)

Su principal objetivo es poner en comunicación a ofertantes y demandantes. Son buenos portales para anunciar el inmuebles (publicidad), pero como veremos posteriormente, eso sólo es una parte del negocio. Entre estos portales podríamos destacar como los más importantes:

- Fotocasa.es.
- Idealista.com.
- Milanuncios.com.
- Pisos.com.
- Tucasa.com
- Habitat24.com

Software orientado a inmobiliarias

Franquicias

Aunque no forme como parte principal de lo que se quiere analizar, existen un conjunto de franquicias a las que se pueden adherir la inmobiliaria y que pueden proporcionar software para la empresa, principalmente con módulos para la publicidad del inmueble.

Normalmente todas cuentan con las siguientes características:

- Para utilizar su software tienes que ser franquicia.
- Orientado a la captación de clientes e interconexión de profesionales.
- La gestión de los datos de la inmobiliaria queda en segundo plano o simplemente no existe.

Entre estos portales podríamos destacar como los más importantes:

- Tecnocasa.
- Don Piso.
- RH Properties.
- Inmo-Cion.
- Alfa Inmobiliarias.
- Sólo alquileres.
- Etc.

En la web oficial de franquiciador [9] podrá obtener más información.

Soluciones para servidores de la empresa

Podríamos decir que existen tres grandes grupos:

- Aquellas que usan un software de base, sea comercial o no, que promociona el negocio a través de gestión de contenidos. Estos pueden tener plug-in o módulos adicionales que complementan lo anterior aportando algo de gestión siendo unas soluciones perfectamente válidas en algunas casos. Por ejemplo Wordpress con instalación de algún módulo adicional.
- Aquellas que tienen una web estática, de las cuales sólo se hace una promoción del negocio.
- Aquellas que usan software hecho a medida, el cual puede estar conectado con una web estática o ser un producto integrado.

Soluciones CLOUD

Entre ellas se puede realizar las siguiente agrupaciones:

- CMS.
- ERP.
- CRM.

Posteriormente, se pasó a realizar una búsqueda atendiendo a su coste (principalmente porque todos los anteriores son comerciales) y el tipo de licencia de software. Se agrupan en las siguientes categorías, aunque estos realmente son software del tipo anterior:

- gratuitos.
- Software libre.

CMS

CMS son las siglas de Content Management System, o lo que viene a ser un sistema de gestión de contenidos. Un CMS es un programa desarrollado para que cualquier usuario pueda administrar y gestionar contenidos de una web con facilidad y sin conocimientos de programación Web. En este sector, básicamente, este tipo de software sirve para implementar la promoción del negocio o web corporativa. Existe una amplia variedad de este tipo, pero aquellos con los que he tenido más relación, y posiblemente se encuentren entre los más conocidos son:

- Wix.
- Arsys.
- 1AND1.

Aunque los dos últimos son realmente proveedores de hosting web, entre sus servicios ofertan plantillas personalizables que pueden ser adaptadas a este sector.

ERP

Se llama ERP a los sistemas de planificación empresariales que administran aspectos de producción, distribución entre otros aspectos. ERP es una sigla que significa 'Enterprise Resource Planning', o bien, 'Planeamiento de Recursos Empresariales'.

A grandes rasgos, suelen centrarse más en la gestión de las actividades del negocio y el estado de los mismos y la relación con los proveedores. La relación con los clientes entra en un segundo plano aunque muchos suelen incluirlo como paquete adicional o se integra con alguna herramienta externa. Entre los más conocidos están:

- Prinex: Según su portal, líder número 1 en el sector (aunque veremos que esto es una constante en todos los productos que se ofertan a partir de ahora), más del 80 % de inmobiliarias lo usan, y resaltan casos de éxito con grandes empresas como Aliseda (Banco Popular) o Junta de Andalucía.

- Quonex: Aunque tenga una menor cuota de mercado, posee cierta flexibilidad y posibilidad de publicación de inmuebles en base a determinadas plantillas.

CRM

CRM es una estrategia de negocios dirigida a entender, anticipar y responder a las necesidades de los clientes actuales y potenciales de una empresa para poder hacer crecer el valor de la relación.

El software basado en esta estrategia, suelen contar, a grandes rasgos, con diferentes funcionalidades que sirven como software de gestión del negocio pero el cual suele estar enfocado al cruce de demandas así como a la comunicación entre agentes inmobiliarios de la empresa, o el mantenimiento de una propia agenda de acciones entre otras utilidades. Todos constan, o suelen incorporar, almacenamiento de los inmuebles y clientes, además de otras muchos datos con multitud de funcionalidades adicionales.

Por otro lado, cabe comentar que suelen ofrecer, de forma opción, ya sea a través de plantillas o de un diseño a medida, la posibilidad de realizar la web corporativa, o lo que nosotros hemos denominado como la promoción del negocio o sitio web.

Estos software suelen contar con diferentes módulos que, dependiendo del software, puedes contratar de forma modular o por diferentes niveles como paquetes premium o paquetes básicos, pero en todos tienes que acabar pagando una cuota mensual u anual.

De entre todos los sistemas analizados, estos suelen ser los más populares y con mejor relación servicios/precio. Podríamos destacar como los más importantes:

- Inmoenter.
- Inmofactory (Software oficial de FOTOCASA).
- Witei.
- Mobiliagestion.
- Inmovilla.
- InmoPC.
- Inmoweb.
- CRMInmobiliario.
- Pipedrive.

Aunque se han consultado muchas más herramientas, esto es sólo un resumen de las que he encontrado o de las que se han aconsejado entre diversas inmobiliarias. De todas ellas me gustaría destacar y hacer una comparativa de las dos primeras, Inmoenter e Inmofactory, de las cuales he podido tener acceso online a una versión demo para valorar sus características. Para ello me he tenido que hacer pasar por inmobiliaria y tener varias entrevistas con agentes comerciales, incluso, en Inmofactory me llegaron a hacer un tour guiado de más de hora y media de duración.

Gratis

El único que he podido encontrar es Mapaprop, el cual ofrece una solución online con algunas características básicas pero puede ser una buena herramienta para PYMES. Como característica principal que veo, es que no dispone de posibilidad de generar carteles de inmuebles con código QR.

Software libre

- Pligg: Red social adaptable y más orientada a la comunicación entre los agentes inmobiliarios.
- Open Innova (Odoo): Aunque lo venden como software inmobiliario, realmente es una herramienta modular que se puede ir adaptando a cualquier sector.

Soluciones Escritorio

- Habitsoft: El cual permite la publicación en habitat24.com
- Inmotool: Aplicación de escritorio escrita en JAVA para CRM externo.

Resumen

Después de echar un vistazo a diferentes herramientas se sacan en claro cuáles deberían ser las coordenadas a tener en cuenta para valorar un software:

- Promoción del negocio: ¿Es posible tener un sitio web o web corporativa con la herramienta? ¿Está sincronizado con otros sistemas de los que toman sus datos?
- Gestión del negocio: ¿Registra inmuebles? ¿Demandas? ¿Clientes? ¿Estadísticas? ¿Tiene más capacidades? ¿Orientado a CRM? ¿Orientado a ERP?
- Coste: ¿Cuánto cuesta implantarlo? ¿Existen varios niveles?
- Tipo de código: ¿Software libre o privativo?

- Mantenimiento: ¿La solución la implanta otra empresa? ¿Solución Cloud, Escritorio, Servidor de empresa?
- Usabilidad: ¿Es fácil de manejar e intuitivo? ¿Es fácil introducir datos? ¿Se navega bien entre secciones y conecta bien la información?
- Flexibilidad: ¿Tiene diferentes usos? ¿Adaptación a cambios?
- Integración portales inmobiliarios: ¿Puede sincronizarse con portales inmobiliarios?

Se han empleado multitud de técnicas para realizar este análisis como brainstorming, búsqueda en internet, aprendizaje en el puesto de trabajo, entrevistas (con comerciales de los software del mercado y con agentes inmobiliarios), etc. En el siguiente apartado de Fortalezas y debilidades, haremos un análisis en función de estos parámetros.

2.1.4. Estructura organizativa

Esta empresa cuenta con los siguientes participantes:

- Una gerente: La cual se encarga de la dirección del negocio.
- Un agente inmobiliario: El cual se encarga de la actividad diaria del negocio así como de la relación principal con los clientes.
- Un colaborador externo: En este caso concreto, una arquitecta, la cual realiza diferentes servicios asociados (tasación, reformas, certificación energética, etc.) de los diferentes inmuebles que la empresa va captando en el tiempo, así como de otros inmuebles de otras empresas colaboradoras.

2.1.5. Fortalezas y Debilidades

Los posibles valores podrán ser muy malo, malo, regular, bueno, muy bueno. En el caso de tipo de código libre o privativo. En algunas ocasiones, si existe duda o varias soluciones posibles, podrá venir especificado DEPENDE o varios valores.

Soluciones para servidores de la empresa

- Promoción del negocio: DEPENDE
- Gestión del negocio: DEPENDE

- Coste: DEPENDE
- Licencia: Generalmente privativo
- Mantenimiento: REGULAR
- Usabilidad: MUY BUENA
- Flexibilidad: MALA
- Integración portales inmobiliarios: MUY MALO

Esta rama depende del tipo de software implantado. Por ejemplo si se usa WORDPRESS con algunos plug-in podría ser:

- Promoción del negocio: MUY BUENA
- Gestión del negocio: REGULAR
- Coste: MUY BUENO
- Licencia: LIBRE
- Mantenimiento: REGULAR-MALO
- Usabilidad: REGULAR
- Flexibilidad: BUENA
- Integración portales inmobiliarios: MUY MALO

Soluciones CLOUD

CMS

- Promoción del negocio: MUY BUENA
- Gestión del negocio: MUY MALA
- Coste: MUY BUENO
- Licencia: PRIVATIVA
- Mantenimiento: MUY BUENO
- Usabilidad: BUENO

- Flexibilidad: MUY BUENO
- Integración portales inmobiliarios: MUY MALO

ERP

- Promoción del negocio: DEPENDE
- Gestión del negocio: MUY BUENA
- Coste: MUY MALO
- Licencia: PRIVATIVO
- Mantenimiento: MUY BUENO
- Usabilidad: REGULAR
- Flexibilidad: REGULAR
- Integración portales inmobiliarios: DEPENDE

CRM

- Promoción del negocio: BUENA-MUY BUENA
- Gestión del negocio: BUENA
- Coste: REGULAR-MALO
- Licencia: PRIVATIVO
- Mantenimiento: MUY BUENO
- Usabilidad: REGULAR-BUENA (Muchos módulos y detalles innecesarios)
- Flexibilidad: REGULAR-BUENA (Los datos suelen estar predefinidos)
- Integración portales inmobiliarios: MUY BUENO

Como principales ventajas podríamos destacar:

- Tener en todo momento tu actividad comercial controlada, así como la de tus trabajadores.

- Publicar nuevos servicios en tu página web y actualizar anuncios constantemente.
- Gestionar bases de datos de clientes e inmuebles.
- Adaptar tus servicios a las legislaciones inmobiliarias vigentes.
- Realizar presupuestos.
- Enviar emails automáticamente.
- Ahorran mucho tiempo en la publicación en portales inmobiliarios.

No obstante, he podido comprobar que en ninguna de las herramientas, por ejemplo, tiene una gestión documental flexible ni dispone de generación de carteles con código QR. En mi opinión, el software con el que rivalizar en un futuro y base para la implantación de posibles modelos de negocios.

Gratis

- Promoción del negocio: MALA
- Gestión del negocio: BUENA
- Coste: MUY BUENO
- Licencia: PRIVATIVO
- Mantenimiento: BUENO
- Usabilidad: BUENA
- Flexibilidad: REGULAR-BUENA
- Integración portales inmobiliarios: MALO

Software libre

- Promoción del negocio: MALA
- Gestión del negocio: BUENA
- Coste: MUY BUENO
- Licencia: LIBRE

- Mantenimiento: BUENO
- Usabilidad: BUENA
- Flexibilidad: REGULAR-BUENA
- Integración portales inmobiliarios: MUY MALO

Soluciones Escritorio

- Promoción del negocio: MUY MALA
- Gestión del negocio: MUY BUENA
- Coste: REGULAR
- Licencia: PRIVATIVA
- Mantenimiento: BUENO
- Usabilidad: BUENA
- Flexibilidad: REGULAR
- Integración portales inmobiliarios: MALO

2.2. Necesidades de Negocio

2.2.1. Objetivos de Negocio

A continuación se ofrece un análisis catalogado en diferentes ámbitos, que, según la información que he podido obtener sobre los diferentes sistemas existentes (comerciales y no comerciales) así como de la información ofrecida por los asesores inmobiliarios se divide en los siguientes:

- FACTURACIÓN
 - Factura de contratos (compra-venta y alquiler).
 - Factura por servicios prestados.
- PUBLICIDAD
 - Envío de correos a demandantes con información de diversos inmuebles.

- Envío de SMS a clientes.
- Subida de inmuebles a portales inmobiliarios (FOTOCASA, IDEALISTA, TEC-NOCASA, etc).
- Cartel.

■ PROMOCIÓN DE LA EMPRESA

- Publicación de servicios ofertados.
- Publicación de noticias.
- Venta y alquiler.
- Quienes somos, dónde estamos, contacto, etc.
- Promoción de servicios y profesionales.

■ DOCUMENTACIÓN

- Documentos adjuntos:
 - Generales.
 - Inmuebles (Imágenes, planos, otros, etc).
 - Clientes (DNI, CIF, NIE, nómina, otros, etc).
 - Demandas (Ficha visita, etc).
- Documentos automatizados:
 - Ficha de encargo.
 - Ficha de visita.
 - Cartel publicitario.
 - Fichas de inmuebles y clientes.
 - Responsabilidad de certificación energética.
 - Presupuestos.
 - Contratos (compra-venta y alquiler).
 - Ficha de renuncia.
 - Factura de contratos (compra-venta y alquiler).
 - Factura de servicios prestados.

■ CAPTACIÓN

- Buscar en otros portales inmuebles que puedan coincidir con las demandas.

■ DEMANDAS

- Registrar los intereses de los demandantes:
 - Inmuebles específicos.
 - Cruce de inmuebles acordes a varios parámetros.
 - Descripción general.
- Recomendaciones de inmuebles.

■ DATOS

- Inmuebles (ubicación, datos principales, características, bajadas y subidas de precio, visualización en Google Maps).
- Clientes.
- Promociones (complejos inmobiliarios).

■ PERSONALIZACIÓN Y MANTENIMIENTO DEL SISTEMA

- Configuración de datos del sistema (idiomas, datos de la inmobiliaria, tipos de inmueble, opciones extras, etc.).
- Opciones de funcionamiento.
- Copias de seguridad (datos de la base de datos y documentación).
- Plantilla de documentación.
- Elección de varias plantillas de diseño del sitio web público.
- Configuración de plantilla de diseño del sitio web público seleccionada.

■ ACTIVIDADES

- Agenda de acciones con los clientes (llamar, entrevistas, petición de documentación, visitas, etc.).
- Comunicación entre los agentes inmobiliarios.

■ ESTADÍSTICAS

- Resumen estadístico de datos, actividades, etc en el dashboard principal.

- Realización de informes estadísticos sobre inmuebles y clientes acordes a diversos criterios.
- Registro de actividad de la web en Google Analytics.

- COMUNICACIÓN DEL CLIENTE

- Contactar a través del envío de un correo electrónico.
- Envío de SMS a clientes.
- Ofertantes puedan insertar un inmueble a través de la zona pública.
- Gestionar sus fotos y/o videos.
- Obtener sus facturas.
- Generar presupuestos automatizados de inmuebles.
- Comunicarse directamente con la inmobiliaria a través de chat.

- COLABORACIONES

- Perfiles para distintos tipos de colaboradores:
 - Tasador.
 - Certificador.
 - Interiorista.
 - Otros.

- APIS PÚBLICAS

- Conexión segura y pública.
- Operaciones necesarias para consultar la información más relevante de los inmuebles.
- Información adicional.

- APLICACIONES MÓVILES

- Usar las APIs públicas definidas en el apartado anterior.
- Enfocadas tanto en los clientes y agentes para implementar la comunicación con el cliente.
- Agenda del agente inmobiliario.

- Gestión de la documentación del inmueble.

Estos ámbitos y objetivos son sólo un catálogo inicial que podrá ir variando con el tiempo y que sólo pretende ofrecer una visión general y resumida de las diferentes necesidades del negocio.

Particularmente, en GESTICADIZ se detectaron las siguientes necesidades en relación al software realizado en 2013:

- La zona pública era inservible:
 - Necesitaba mejorarse el buscador ya que era muy básico.
 - El diseño debía actualizarse ya que parecía muy anticuado y no se adaptaba a diferentes dispositivos.
 - Era necesario modificar los colores corporativos.
 - Modificar las secciones y textos, para, por ejemplo, promocionar los servicios de los colaboradores externos u otros secciones de interés.
 - Los usuarios podían suscribirse a un RSS de noticias pero no podían visualizarlas en la zona pública.
 - No se podían publicar los enlaces a redes sociales como Twitter o Facebook.
- La introducción de datos iniciales en el sistema era muy laboriosa.
- No existía la posibilidad de configuración de datos del sistema, por lo que no era flexible.
- El contenido del sistema en otros idioma se traducía ON LIVE, y a parte de que algunas traducciones eran prácticamente incomprensibles tardaba muchísimo.
- La documentación generada tenía un único modelo implementado que no se podía modificar ni ampliar.
- Las copias de seguridad se enviaban por correo y sólo se realizaban de la base de datos. Además eran automáticas cuando alguien iniciaba sesión por primera vez en un día en el sistema no había posibilidad de elegir cuando realizarlas.
- Existían multitud de documentos a generar, muchos de ellos ni se usaban.

- Finalmente, al tener las restricciones de los proceso de negocios implementadas era poco flexible con la realidad diaria de la empresa.

Particularmente, en GESTICADIZ el software aportaría las siguientes mejoras:

- Respecto a la zona pública adquirirían las siguientes capacidades:
 - Dispondrían de buscador de inmuebles en diferentes secciones y además se podría visualizar conjuntamente los inmuebles encontrados en un mapa de Google maps.
 - La visualización del sitio web se adaptaría a diferentes dispositivos y el diseño sería mucho más atractivo.
 - Modificar los colores corporativos, el logo u otras aspectos del sitio web.
 - Configurar las secciones y textos, para, por ejemplo, promocionar los servicios de los colaboradores externos u otros secciones de interés.
 - Publicar noticias de interés.
 - La web podría estar traducida a otros idiomas de forma flexible y eficiente.
- Respecto a la zona privada mejoraría los siguientes aspectos:
 - La introducción de datos iniciales en el sistema estaría muy automatizada al disponer de varios tipos de datos por defecto y de importaciones por CSV.
 - La generación de informes, al hacerse en CSV y directamente desde un buscador, sería más precisa y podrían representarla luego de cualquier forma. Para obtener estadísticas también serían muy sencillo al tener un documento con un formato estandarizado.
 - Al tener diferentes preferencias y formas de configuración el software sería más flexible tanto a la hora de usarlo como a la hora de generar documentación.
 - Podrían valorar cómo queda el formato de los inmuebles antes de publicarse.
 - Existiría una independencia entre el formato de la documentación generada y su contenido.
 - Las copias de seguridad serían flexibles y abarcarían toda la información del sistema.
 - El sistema ganaría en seguridad en el acceso y almacenamiento de los datos.

- El software tendría pocas restricciones del proceso de negocio implementadas por lo que el trabajo diario de la empresa con el software sería más ágil y flexible.

2.2.2. Procesos de Negocio

Retomando los diagramas de la anterior sección, vamos a señalar en rojo aquellas partes del proceso de negocio que son vitales en el día a día del funcionamiento de la empresa y que implantaremos a través del sistema de información.

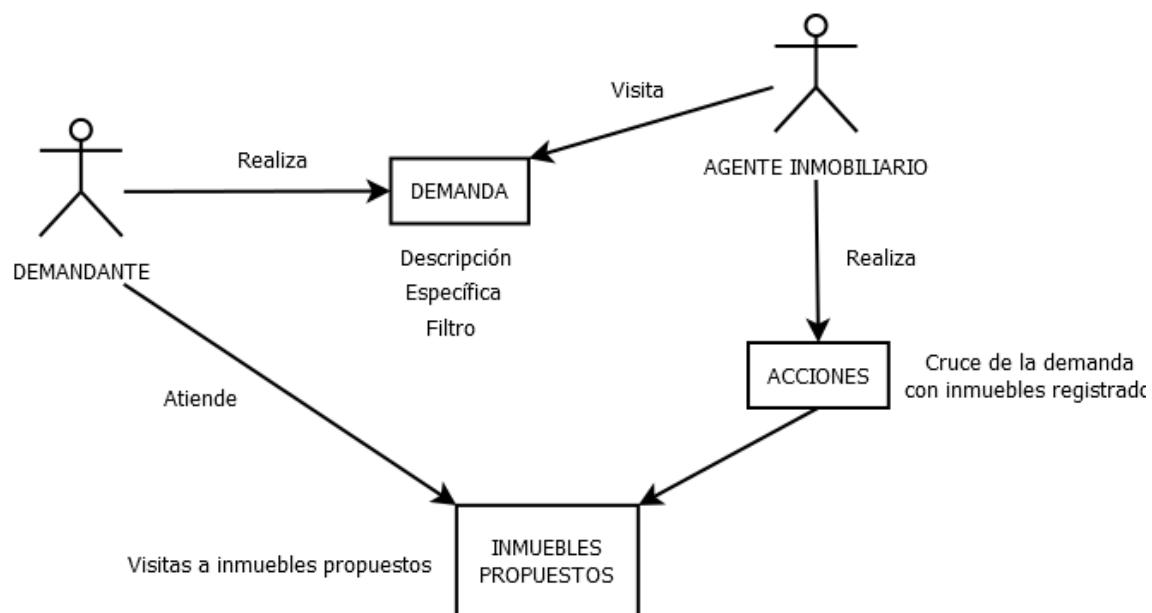


Figura 2.5: Proceso de negocio en el sistema de información del demandante

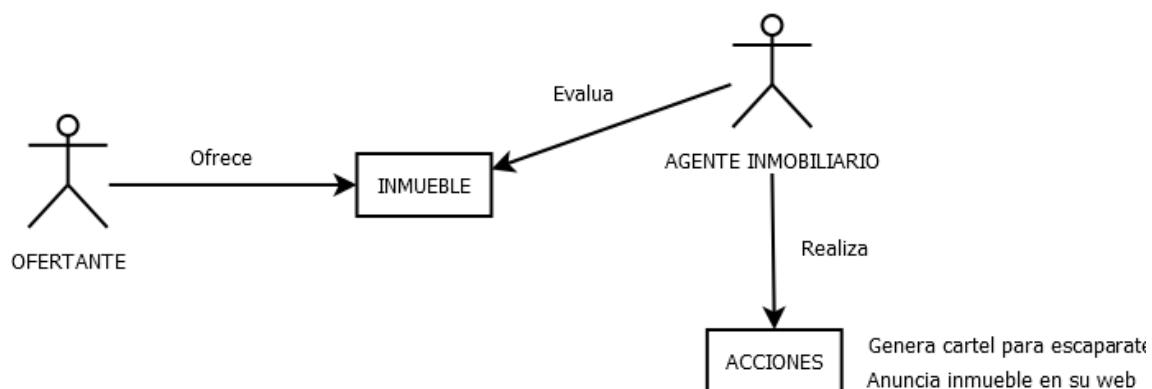


Figura 2.6: Proceso de negocio en el sistema de información del ofertante

2.3. Objetivos del Sistema

Mostramos los diferentes objetivos que se pretenden cumplir el software una vez esté concluido, teniendo en cuenta sólo aquellos procesos y necesidades más importantes especificado en los apartados anteriores. Han sido orientadas a satisfacer las necesidades detectadas para GESTICADIZ pero teniendo una visión de conjunto del mercado inmobiliario:

- Crear un software válido, no sólo para una inmobiliaria específica, si no para cualquiera dentro del territorio español.
- Reducir los costes de implantación del software de gestión y promoción inmobiliario.
- Fusionar en una única herramienta todas las características esenciales para que un negocio inmobiliario pueda llevar su actividad diaria de forma clara y eficiente (catálogo de inmuebles, clientes y demandas).
- Disponer de un sitio web propio que de a conocer a la inmobiliaria en el mercado, ofreciendo sus inmuebles de forma directa y otros servicios relacionados.
- Tener un sistema que pueda ser visualizado y manipulado fácilmente en todos los dispositivos.
- Facilitar la introducción de datos en el sistema y su navegabilidad.
- Tener suficiente flexibilidad en la configuración de todos los datos del sistema así como de la plantilla de diseño web.
- Disponer de un sistema intuitivo, directo, rápido y fácil de usar. Que tenga una curva de aprendizaje baja.
- Proponer funcionalidades básicas y prácticas que sirvan luego para una posterior ampliación del sistema.
- Ofrecer canales de comunicación para que el cliente se comunique con la inmobiliaria directamente.
- Generar documentación de gestión e informes de manera rápida y flexible y en diversos formatos.
- Poder realizar copias de seguridad tanto de los datos adjuntos como de los datos introducidos en el sistema en su base de datos.

- Visualizar estadísticas básicas sobre el funcionamiento del negocio.
- Aplicar seguridad a la hora de manipular la información y el acceso a los datos sensibles.

En definitiva, a través del sistema se debe poder saber qué es lo que desean los clientes, qué es lo que ofrecen, cuáles son los principales cruces entre los mismos (tanto automáticos como manuales) centralizando en un único sistema toda la documentación generada y adjunta aplicando los procesos de negocio de la inmobiliaria digitalmente. Por otro lado, se debe publicar la oferta de inmuebles pertinente así como otros servicios en un sitio web atractivo y responsive, que permita al demandante encontrar lo que desee con facilidad. Por otro lado, y no menos importantes, existen otros objetivos de carácter personal que me gustaría remarcar:

- Crear un núcleo de funcionalidades que sirvan para otros sistemas de temáticas completamente diferentes.
- Adquirir buenas prácticas tanto en la metodología de desarrollo como en la construcción del sistema.
- Adquirir nuevos conocimientos técnicos y usar nuevas herramientas que me permitan ser más competitivo como profesional.
- Aplicar los conocimientos adquiridos tanto en la Ingeniería Técnica como en la Ingeniería en Informática.
- Obtener el título de Ingeniería informática.

2.4. Catálogo de Requisitos

2.4.1. Requisitos funcionales

El sistema debe satisfacer los siguientes puntos. Por claridad, se van a dividir atendiendo a los dos subsistemas que se van a analizar:

SITIO WEB DE LA INMOBILIARIA (ZONA PÚBLICA)

- Ofrecer información pública sobre los inmuebles ofertados para venta o alquiler.
- Poder visualizar un detalle de los inmuebles que amplíe la información inicial.

- Enviar una petición de información de un inmueble en particular.
- Buscar los inmuebles atendiendo a diversos criterios de filtrado.
- Publicar noticias u ofertas que la inmobiliaria considere interesantes.
- Disponer de un formulario de contacto en el describir un pequeño fragmento de texto.
- Poder crear cualquier tipo de sección pública en la que se puedan introducir textos e imágenes (como por ejemplo una sección que explique quienes son dentro del panorama inmobiliario, etc.) así como sub-secciones y páginas con contenidos simples.
- Disponer de una sección en la que muestre donde se encuentra ubicada la empresa en google maps y diferentes formas de contactar, horarios, etc.
- Ofrecer toda la información anteriormente descrita en varios idiomas.

SOFTWARE DE GESTIÓN (ZONA PRIVADA)

- Gestionar la cartera de clientes de la inmobiliaria.
- Gestionar los inmuebles que se ofrecen y cuales han sido vendidos o alquilados además de diversos datos de información como las subidas y bajadas de precio.
- Subir imágenes de los inmuebles y publicarlas.
- Añadir enlaces externos a los inmuebles y publicarlos.
- Adjuntar ficheros a los inmuebles, clientes y demandas registradas.
- Generar las fichas de los clientes, los inmuebles y las demandas y poder exportarlas a PDF.
- Importar la información inicial de clientes e inmuebles desde un CSV.
- Registrar las demandas de los clientes tanto de uno o varios inmuebles como una descripción general.
- Especificar una serie de criterios básicos en las demandas para que se puedan cruzar con la cartera de inmuebles registrados.
- Gestionar los diferentes usuarios del sistema, asignar sus permisos así como el idioma que utilizará en la plataforma.

- Gestionar las opciones extras que puedan asignarse a los inmuebles.
- Gestionar los lugares de interés que puedan asignarse a los inmuebles.
- Gestionar los tipos de documentos adjuntos.
- Gestionar los estados en los que se pueden encontrar las demandas, clientes e inmuebles.
- Definir las preferencias del sistema.
- Gestionar los municipios de cada provincia.
- Indicar los municipios y provincias de los cuales se podrán registrar inmuebles.
- Gestionar las diferentes zonas que componen los municipios.
- Realizar copias de seguridad de la base de datos y de todos los ficheros adjuntos.
- Definir las diferentes plantillas de documentos automatizados.
- Adjuntar el logo de la organización.
- Adjuntar ficheros de interés general.
- Gestionar el contenido y diseño de la cabecera y pie de página de la zona pública.
- Modificar los colores de la plantilla web de la zona pública.
- Gestionar los diferentes marcos y enlaces que aparecen en el pie de la zona pública.
- Gestionar las secciones y sub-secciones así como las páginas que contienen.
- Habilitar los diferentes idiomas que están activos en la plataforma.
- Mostrar un resumen estadístico de la actividad realizada en el sistema y de los datos registrados.

Uno de los objetivos anteriores, 'Disponer de suficiente flexibilidad en la configuración de todos los datos del sistema así como de la plantilla de diseño web', se satisface con muchas funcionalidades indicadas en la zona privada, y viene a establecer, por ejemplo, que se puedan configurar los estados de los inmuebles, de las demandas o los clientes o que sobre la plantilla de diseño web se pueda modificar el logo o los colores de la cabecera y el pie.

2.4.2. Requisitos no funcionales

- Adaptación de la interfaz a múltiples dispositivos, navegadores y plataformas: El sistema debe poder visualizarse y manipularse en los principales navegadores y poder usarse en las versiones más modernas de tablets (Android y IOS), smartphones (Android y IOS), ordenadores portátiles y ordenadores personales con las resoluciones más comunes.
- Facilidad a la hora de introducir los datos iniciales en el sistema: El sistema debe tener datos definidos por defecto para llenar el menor número de campos posibles así como procedimientos para poder introducir datos a través de CSV con todos los campos posibles. Compatibilidad con Mac, Windows y Linux.
- Ofrecer una imagen moderna de la empresa a través de la web: El funcionamiento responsivo de la zona pública así como la organización de plantilla inicial deben tener los elementos más modernos y con el diseño más atractivo posible.
- Disponer de un sistema intuitivo, rápido, fácil de usar y con funcionalidades básicas y prácticas: Uno de los pecados que se cometieron con el anterior sistema fue la de implementar muchas restricciones durante el proceso de vida de un inmueble. Por tanto, deben implementarse pocas restricciones en ese sentido, deb estar claramente definidas y accesibles y sólo el número de funcionalidades necesarias para que la inmobiliaria pueda publicar la información en la zona pública y almacenar los datos más relevantes de inmuebles, demandas y clientes. Por otro lado, todas las interfaces deben desplegarse, en condiciones normales, en menos de un segundo, salvo por situaciones excepcionales (como la carga de un CSV con muchos registros o el despliegue de muchos inmuebles por ejemplo).

2.4.3. Reglas de negocio

No existirán demasiadas restricciones, ya que una de las características de este proyecto es precisamente que no sea estático en ese aspecto. Aún así, el sistema deberá verificar las siguientes restricciones:

- No se podrá publicar un inmueble que no tenga, al menos, una foto de portada. Esta restricción es para evitar que publiquen por error inmuebles en la zona pública que no tengan una imagen inicial que mostrar en los listados en los que aparezca.

- No se puede despublicar la portada ni tampoco poner una imagen no publicada como portada.
- Sólo se podrá publicar un enlace a video de Youtube por inmueble. Esto se debe a que la plantilla pública correspondiente al detalle del inmueble sólo admite un enlace al reproductor de vídeo.
- Si sólo existe un administrador en el sistema, este no podrá ser borrado, para evitar que se queden sin acceso a la zona de administración fortuitamente.
- En general, los datos que puedan ser configurados (como los tipos de inmuebles, los estados, las plantillas de documentación, etc.), no podrán ser eliminados si tienen algún elemento vinculado en el resto del sistema. En la sección de Modelo Conceptual 4.1.1 podrá visualizar más información.
- Posteriormente, cuando se realice el análisis y se definan las diferentes claves candidatas, se comprobará su unicidad.

2.4.4. Requisitos de información

Necesitará registrarse la siguiente información:

INMUEBLES

- Referencia.
- Metros totales.
- Metros útiles.
- Habitaciones.
- Baños.
- Año Construcción.
- Dirección.
- Observaciones.
- Municipio.
- Zona.

- Tipo de inmueble.
- Certificación energética.
- Consumo de Kilowatios por metro cuadrado al año.
- Estado en el que se encuentra.
- Captador.
- Precio compra.
- Precio alquiler.
- Subida y bajadas de precio.
- Fecha de alta
- Datos de publicación (en varios idiomas):
 - Título.
 - Descripción.
 - Url SEO.
 - Descripción SEO.
 - Palabras clave.
- Si está publicado.
- Secciones de publicación (oportunidad y destacado).
- Dirección pública.

IMÁGENES

- Nombre.
- Si es una portada.
- Si está publicada.

ENLACES

- Título.

- Si es un video de youtube.
- Si está publicada.
- URL.

CLIENTES

- NIF/NIE/CIF.
- Nombre o razón social.
- Apellidos o tipo entidad.
- Fecha de nacimiento.
- Dirección.
- Correo electrónico.
- Observaciones.
- Números de teléfono de contacto.
- País de residencia.
- Provincia.
- Población.
- Agente asignado.
- Estado en el que se encuentra.
- Fecha de alta.

DEMANDAS

- Referencia.
- Cliente.
- Agente asignado.
- Observaciones.
- Estado en el que se encuentra.

- Fecha de alta.
- Oferta (venta o alquiler)
- Tipo de demanda (con o sin filtros)
- Filtros de búsqueda:
 - Metros totales (min. y máx).
 - Habitaciones (min. y máx).
 - Baños (min. y máx).
 - Año Construcción (min. y máx).
 - Provincia
 - Municipio.
 - Listado de zonas.
 - Listado de tipos de inmuebles.
 - Certificación energética mínima.
 - Precio compra (min. y máx).
 - Precio alquiler (min. y máx).

PLANTILLAS DOCUMENTACIÓN

- Nombre.
- Tipo.
- Contenido.
- Descripción.

FICHEROS ADJUNTOS

- Texto.
- Tipo.

DOCUMENTACIÓN GENERADA

- Plantilla asignada.

- Fecha.
- Contenido.

Aclaraciones

De los documentos generados por los inmuebles habrá que almacenar el idioma, y del cartel, que es un tipo de documentación en especial, habrá que almacenar si está impreso.

De los lugares de interés y las opciones extras, se almacenará una descripción general en castellano y varios textos públicos en varios idiomas.

Los campos para tipos varios o datos base del sistema contendrán un nombre y una descripción en castellano (Tipos de fichero, estados y zonas). Si el dato es público, se almacenará una descripción general en castellano y varios textos públicos en varios idiomas (Tipos de inmuebles, lugares de interés, y características). Los tipos de las plantillas deberán contener las marcas que se pueden asignar.

La información de las provincias, municipios y comunidades autónomas vendrán determinadas por el formato impuesto por el INE [7].

Los datos de los usuarios, permisos, copias de seguridad así como todos los campos definibles en la zona pública, al pertenecer a módulos integrados vendrán definidos por estos módulos y no es objetivo de análisis de esta memoria.

2.5. Alternativas de Solución

El análisis de las alternativas de solución fue uno de los puntos que más tiempo tomó, de hecho fue motivo de paralización del proyecto durante muchos meses. Al final se barajaron dos opciones:

PRIMERA OPCIÓN

Usar un CMS conocido como base para el sitio web (como Drupal, Joomla o Wordpress, principalmente este último) e implementar o reutilizar algunos módulos relacionados con inmobiliarias (completa o parcialmente). En función de lo que se implementará añadido al CMS, también se debería realizar una aplicación de gestión que sólo se centrara en el ámbito del CRM+ERP y conectara con la misma.

Esta opción se desechó por los siguientes motivos:

- La solución hacia que el software no fuera completamente de mi propiedad y dependiera de otro externo, aunque es cierto que tendría mucha mas potencia y versatilidad.

- La implementación requería de un nivel de conocimiento y de una curva de aprendizaje demasiado elevada y se iba a tardar demasiado en realizar.
- Finalmente, los usuarios iban a tener que aprender Wordpress cuya curva de aprendizaje es mayor que la de este software.

SEGUNDA OPCIÓN

Reutilizar el software realizado anteriormente para GESTICADIZ. Este software, como hemos visto en las necesidades, tenía una base tecnológica obsoleta, además de un conjunto de funcionalidades y restricciones implementadas que ya no eran de utilidad y por tanto, iba a suponer un coste elevado en tiempo, a parte, de no aportarme nada nuevo, en el ámbito tecnológico.

2.6. Solución Propuesta

Finalmente se optó por implementar un nuevo proyecto desde cero sobre un framework de backend (finalmente me decidí por Codeigniter, aunque se valoraron muchos otros) y utilizar otro de frontend (finalmente me decidí por Bootstrap aunque se valoró e incluso se implementó inicialmente al principio con Materialize, basado en material Design en lugar de Flat Desing). En el apartado del Construcción [6.1](#) podrán encontrar más información.

Por tanto, la solución constaría de:

- Un CMS que proporcione las funcionalidades públicas para la promoción de la inmobiliaria.
- Un software de gestión que permita gestionar la información y actividades del negocio y desde la cual extraerá la información el CMS.

Su utilizarán módulos libres para el framework seleccionado como software de base pero de los cuales tendríamos que parametrizar las siguientes partes:

- Del CMS implementaríamos el formulario de contacto y donde estamos, el buscador de inmuebles y el detalle así como la página de inicio.
- Del Módulo de backup, se debe añadir la posibilidad de realizar copias completas de seguridad.
- Del Módulo de autenticación, configuraríamos su funcionamiento para que se adapte a las necesidades del negocio y se integre con el resto de los componentes (por

ejemplo, definir la posibilidad de poner un idioma para los usuarios de forma que sea el idioma utilizado en la plataforma cuando inicie sesión).

Además se deberá cambiar completamente el frontend empleado en todas ellas, ya que algunas disponían sólo de ejemplos básicos. En la sección se Diseño 5.2 se dan más detalles al respecto.

Capítulo 3

Planificación

3.1. Metodología de desarrollo

El desarrollo de sistemas software no es una tarea sencilla, sobre todo cuando estos comienzan a tener un cierto grado de complejidad, y por ello es necesario la utilización de alguna metodología que permita llevar un orden a lo largo del proceso de desarrollo del sistema. La principal metodología de desarrollo utilizada ha sido Métrica v.3 aunque también me he inspirado, en menor medida, en MADEJA.

MÉTRICA es una metodología de planificación, desarrollo y mantenimiento de sistemas de información, promovida por el Ministerio de Hacienda y Función Pública (antiguo Ministerio de Administraciones Públicas) del Gobierno de España para la sistematización de actividades del ciclo de vida de los proyectos software en el ámbito de las administraciones públicas. Esta metodología propia está basada en el modelo de procesos del ciclo de vida de desarrollo ISO/IEC 12207 (Information Technology - Software Life Cycle Processes) así como en la norma ISO/IEC 15504 SPICE (Software Process Improvement And Assurance Standards Capability Determination).

Al igual que ISO/IEC 12207, MÉTRICA está orientada al proceso y, en su versión 3, estos procesos son:

1. Planificación de Sistemas de Información (PSI)
2. Estudio de Viabilidad del Sistema (EVS)
3. Análisis del Sistema de Información (ASI)
4. Diseño del Sistema de Información (DSI)

5. Construcción del Sistema de Información (CSI)
6. Implantación y Aceptación del Sistema (IAS)
7. Mantenimiento de Sistemas de Información (MSI)

MADEJA que es el Marco de Desarrollo de la Junta de Andalucía. Su misión es proporcionar un entorno que permita a todos los implicados en el desarrollo y en la explotación del software tener una referencia clara de cuáles son las directrices que han de guiar esta actividad, así como dar a conocer los recursos y herramientas que están a su disposición. MADEJA se aplica sobre los productos de software de la Junta de Andalucía, pero no se limita a intervenir en el proceso de desarrollo. Su objetivo es estar presente desde que surge la necesidad de poner en marcha un Sistema de Información hasta que este se encuentra en explotación, incluyendo la fase de mantenimiento. Aportará indicadores que permitirán realizar mejoras en la distintas fases del ciclo de vida de un Sistema de Información. A la hora de plantear MADEJA y durante su proceso de construcción, se han seguido y se seguirán los siguientes principios básicos:

- Tiene un planteamiento independiente y abierto. Su primera versión es fundamentalmente fruto de la dedicación de recursos internos y de la aportación de información de todas las Consejerías y Organismos, además de la aportación de expertos independientes. Posteriormente se han ido incorporando contenidos fruto de la aportación de empresas especializadas sobre áreas concretas.
- Los contenidos tendrán un carácter práctico. Por ello, en MADEJA podemos encontrar guías de uso y manuales acerca de las tecnologías recomendadas, definiciones de normativas y procedimientos de aplicación interna o exigibles a los que desarrollen para la Junta de Andalucía. Además, se propondrá el uso de herramientas, preferentemente de software libre, existentes o desarrolladas a medida para cubrir en la medida de lo posible las necesidades de todos.

Hay que destacar que MADEJA no tiene intención de ser un flujo unidireccional de directrices y normas. Al contrario, se ha planteado como un producto que necesita ser realimentado por los distintos actores implicados para que su validez y cobertura sea la óptima, ajustándose a las necesidades reales de los desarrollos de la Junta de Andalucía. Además, se proporcionará un soporte en la medida en que sea necesario y cuya cobertura se irá ajustando en función de la demanda y las necesidades concretas de todos.

Por otro lado, existen diversos modelos de desarrollo de software. En cada uno de ellos, se especifican los procesos, actividades y tareas involucradas en el desarrollo, explotación y mantenimiento de un producto software. A pesar de toda esta diversidad en cuanto a procesos de software, todos deben incluir cuatro actividades que son fundamentales para la ingeniería de software [3]:

1. Especificación del software: Tienen que definirse tanto la funcionalidad del software como las restricciones de su operación.
2. Diseño e implementación del software: Debe desarrollarse el software para cumplir con las especificaciones.
3. Validación del software: Hay que validar el software para asegurarse de que cumple lo que el cliente quiere.
4. Evolución del software: El software tiene que evolucionar para satisfacer las necesidades cambiantes del cliente.

El desarrollo de este proyecto, está basado en el **Modelo en espiral**. Este modelo fue propuesto en primer lugar por *Barry Boehm*. Es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados del modelo de cascada.

El Modelo en espiral permite hacer un desarrollo rápido de versiones cada vez más completas. Con el empleo de este modelo, el software se desarrolla en una serie de *entregas evolutivas*. En las primeras fases de desarrollo el software puede ser un prototipo pero, mediante iteraciones, se obtienen versiones cada vez más completas del producto. No se desarrolla por adelantado una solución completa del problema, si no que se avanza en una serie de pasos hacia una solución y se retrocede cuando se detectan errores. Al desarrollar el software de manera incremental, resulta más barato y fácil realizar cambios en el software conforme éste se diseña.

Cada versión del software introduce algunas de las funciones que necesita el cliente. Las primeras versiones incluyen las funciones principales o más importantes. De esta manera, el cliente puede ir evaluando y modificando el desarrollo del software para así ver si lo que se ofrece es lo deseado. Esto tiene una gran ventaja, y es que de esta manera, siempre se modificará la versión actual y los cambios se introducirán para versiones posteriores en lugar de tener que redefinir los requisitos iniciales, siendo esto engorroso y retrasando todo el proceso cada vez que el cliente quiera introducir un cambio o mejora.

Debido a la naturaleza del problema en el que es necesario obtener una rápida evaluación del entorno y evaluar el feedback, este modelo se recomienda como el más adecuado. Para el desarrollo del software propuesto, a modo general, se han realizado las siguientes actividades iterativas:

1. **Refinamiento y Recolección de requerimientos:** Fase inicial donde se establece comunicación con el cliente para definir los requisitos y objetivos de la aplicación a desarrollar.
2. **Diseño rápido:** Una vez definidos los requisitos y objetivos a cumplir, se analiza y estudian las posibles soluciones, se evalúan y resuelven los posibles riesgos y se realiza una planificación y diseño rápido del software para valorar si su desarrollo es viable con los riesgos encontrados.
3. **Construcción del Prototipo:** Desarrollo del proyecto creando entregas iterativas para que el cliente pueda evaluarlo.
4. **Evaluación del Prototipo por el Cliente:** El cliente, previa formación por parte del desarrollador si fuese necesario, comprueba que el prototipo entregado funciona correctamente y cumple los objetivos esperados.
5. **Refinamiento del Prototipo:** Corregir errores detectados por el cliente si los hubiera, o realizar cambios sugeridos y continuar refinando el proyecto un poco más ya que sólo era un prototipo.
6. **Producto de Ingeniería:** Implantación del producto software una vez completado.

Este método de desarrollo divide el problema inicial en sub-problemas más pequeños. En cada iteración de la espiral, se van resolviendo estos sub-problemas de manera que si al final de la iteración el sistema aún no estuviera terminado, se volvería a la fase de diseño rápido para seguir refinándolo, repitiendo de la fase 2 a la 5 tantas veces como sea necesario hasta que el producto final estuviese completo. En cada iteración se realizan las fases descritas, creando de manera evolutiva el software propuesto como se muestra en la siguiente figura.

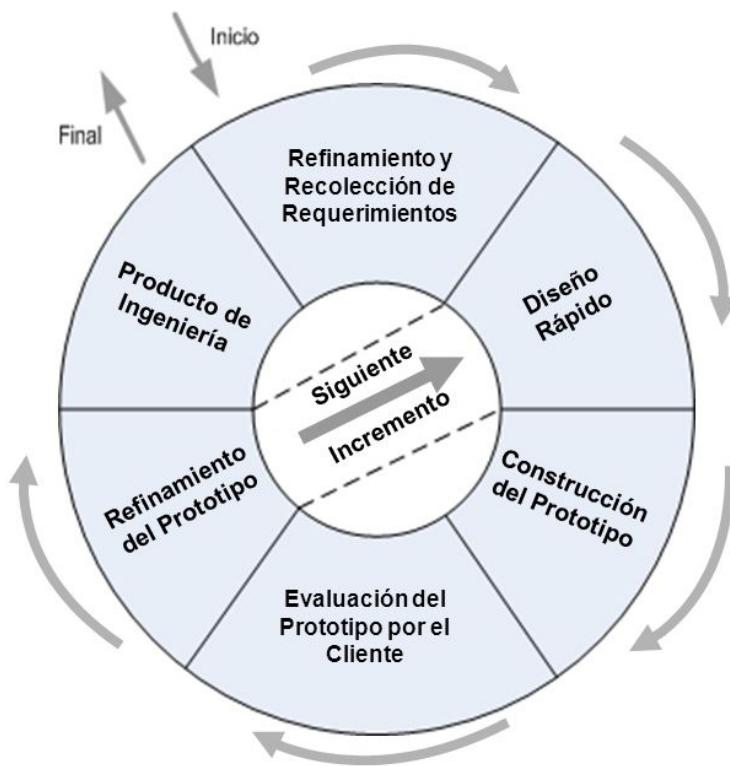


Figura 3.1: Modelo en espiral de Boehm [2]

A continuación, en el apartado de planificación de proyecto podremos ver un calendario detallado tanto de las iteraciones o hitos que se han ido realizando así como del desglose de las actividades realizadas durante la realización del proyecto.

Finalmente, me gustaría añadir un diagrama de decisión de cómo he adaptado todo lo explicado anteriormente para tener una idea gráfica del ciclo de vida del sistema, teniendo en cuenta, sobre todo la búsqueda e integración de componentes de software, los cuales han sido muchos.

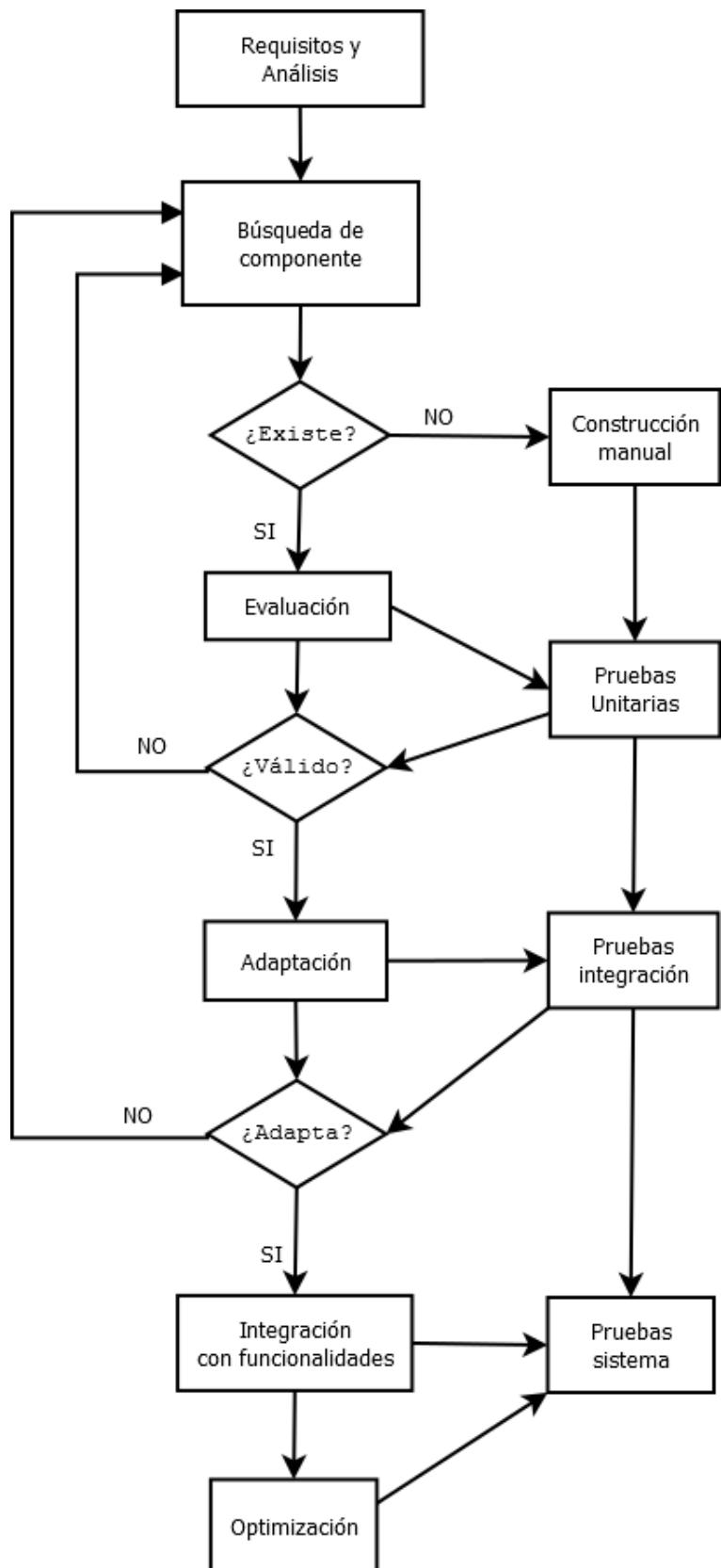


Figura 3.2: Ciclo de vida con reutilización de componentes

En cada iteración del software, en base a los requisitos definidos y el análisis realizado se ha ejecutando diferentes pasos para determinar si un componente pasa a formar parte del sistema:

1. **Evaluación:** ¿Puedo tecnológicamente usar el componente? ¿Funciona correctamente? ¿Hace cosas útiles para mi sistema? ¿Todo lo que necesito o en parte?. Están son las preguntas que deben responderse en este proceso y puede consistir en multitud de tipos de prueba, dependiendo del componente, pero cuyo fin último es evaluarlo en solitario.
2. **Adaptación:** Algunos de los componentes ya están adaptados al framework y hacen todo lo que necesitamos, por lo que esta etapa es opcional y puede realizarse conjuntamente con la anterior. En caso contrario, habrá que intentar adaptar el componente a lo que necesitamos o usarlo parcialmente, depende del caso.
3. **Integración con funcionalidades:** Una vez ya tenemos un componente que funciona en el framework seleccionado y hace lo que debe, hay que pasar a ver qué acciones hay que realizar y cómo utilizarlo dentro de una funcionalidad en concreto. Posteriormente deberemos realizar una evaluación de la funcionalidad usando el componente.
4. **Optimización:** Finalmente hay que comprobar que el componente funciona acorde a las restricciones no funcionales.

Los niveles de pruebas de cada paso no tienen por qué ser exactamente así, ya que hay algunos pasos que se fusionan y por tanto, hay que realizar varios niveles de prueba al mismo tiempo, se ha dejado un esquema genérico de qué pruebas pueden predominar.

3.2. Planificación del proyecto

En la siguiente tabla se desglosan las actividades realizadas durante el desarrollo de este proyecto. Durante todas ellas se han ido realizando labores de documentación (tanto de memoria, anexos diversos, etc.) cuyo tiempo está contabilizado en la duración de las mismas.

| TAREA | COMIENZO | FIN | DURACION |
|--|----------|-------|----------|
| Estudio del dominio del problema y sistemas actuales | 09/01 | 22/01 | 14 |
| Identificación de los objetivos del sistema | 23/01 | 24/01 | 2 |
| Identificación de los requisitos funcionales | 25/01 | 31/01 | 7 |
| Análisis de las diferentes alternativas tecnológicas | 01/02 | 17/03 | 45 |
| Estudio de componentes de la alternativa seleccionada | 18/03 | 07/04 | 21 |
| Análisis | 08/04 | 16/04 | 7 |
| Diseño | 17/04 | 23/04 | 7 |
| Implementación inicial de la base de datos y framework de desarrollo | 24/04 | 30/04 | 7 |

Tabla 3.1: Planificación inicial de actividades

Una vez que ya tenía un conocimiento más detallado de los componentes que inicialmente iba a emplear, de lo que iba a hacer y cómo lo iba a hacer, se realizó una primera implementación de la base datos así como del framework de desarrollo.

Esta primera implementación fue de gran utilidad, ya que me sirvió para sentar las bases de trabajo diario, y sobre todo, poder afinar en cuántas iteraciones debía realizar y cuánto iba a tardar en cada una de ellas. Por tanto, se planifica, una división en procesos más pequeños para los cuales se realizan diferentes sprints (o iteraciones) en los que se vuelven a repetir, o al menos repasar, todas las etapas de desarrollo del software (Requisitos, Viabilidad, Análisis, Diseño, Construcción, etc.) procediendo a reestructurar la planificación si procede.

A continuación se describe este calendario de iteraciones:

| TAREA | COMIENZO | FIN | DURACION |
|---|----------|-------|----------|
| Integrar los principales componentes y obtener un framework básico de trabajo funcional | 02/05 | 22/05 | 21 |
| Personalización y mantenimiento del sistema | 23/05 | 05/06 | 14 |
| Módulo de Clientes | 06/06 | 19/06 | 14 |
| Módulo de Inmuebles | 20/06 | 10/07 | 21 |
| Zona pública | 11/06 | 17/07 | 7 |
| Refinamiento de funcionalidades y conclusión de integración de componentes | 18/07 | 24/07 | 7 |
| Módulo Demandas | 25/07 | 31/07 | 7 |
| APIs | 01/08 | 03/08 | 3 |
| Pruebas de aceptación | 04/08 | 08/08 | 5 |
| Documentación detallada proyecto, refinamientos finales e imprevistos | 09/08 | 23/09 | 45 |

Tabla 3.2: Iteraciones para la construcción del software

A continuación, en las siguientes páginas, se muestran los diagramas de Gantt de las actividades anteriormente descritas.

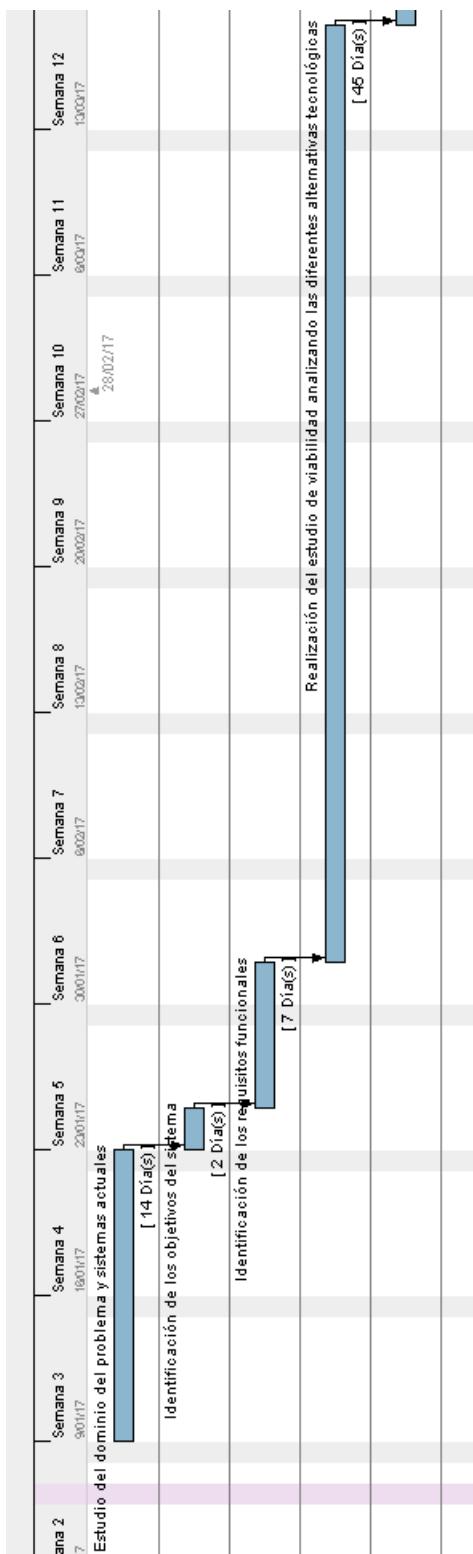


Figura 3.3: Planificación temporal estimada (Parte 1)

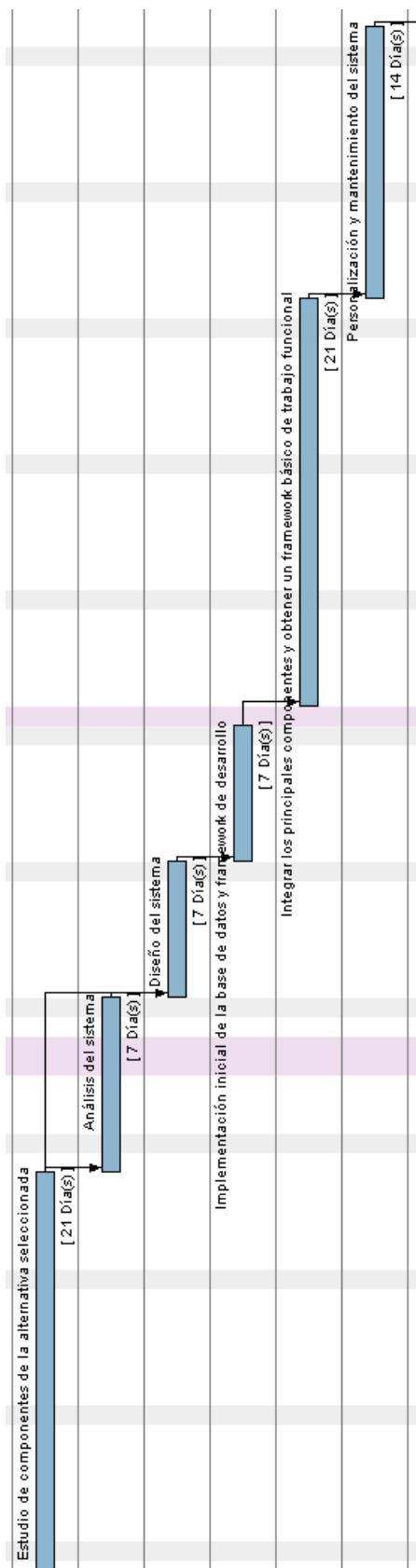


Figura 3.4: Planificación temporal estimada (Parte 2)

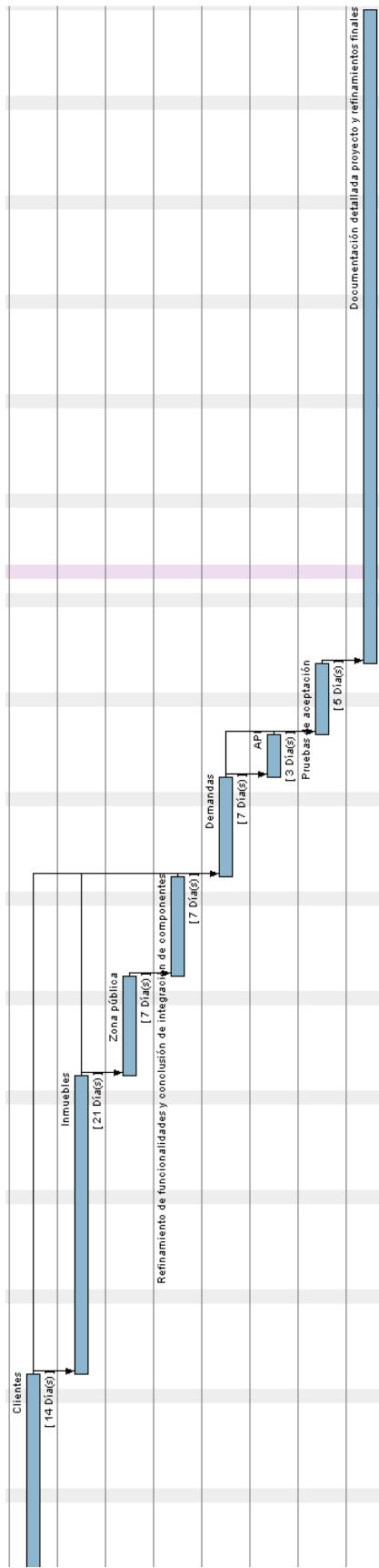


Figura 3.5: Planificación temporal estimada (Parte 3)

3.3. Organización

Las personas involucradas en el proyecto son:

- Asesores de CMS: Álvaro Léon e Iván Rubiales han contribuido a aclarar cómo integrar y adaptar el CMS desarrollado por ellos (TYPICOS).
- Agentes Inmobiliarios: Gloria Chamarro como gerente de la empresa de GESTICADIZ así como Silvía García como agente inmobiliaria han sugerido mejoras en el software y han ayudado a realizar las pruebas de aceptación del mismo.
- Ingeniero Técnico Informático: Ángel Luis Berasuain Ruiz, además de implementar el proyecto, ha tenido que realizar labores de coordinación entre los miembros anteriormente citados para poder ajustarme al calendario así como satisfacer los objetivos propuestos.

Por otro lado, los diferentes servidores que se han utilizado han sido:

- Servidor de desarrollo: El principal cometido de este servidor es realizar una implementación inicial que detecte los primeros errores encontrados y permita pasar una primera batería de pruebas (principalmente, unitarias, integración y funcionales). También servirá para testear el funcionamiento en entorno Windows, ya que el despliegue (en la inmensa mayoría de los casos) tanto en staging como en producción será bajo un entorno Linux.
- Servidor de staging: El servidor de staging, como indica su nombre, es una "puesta en escena", es decir, es un servidor que, según las mejoras prácticas, deberá ser lo más parecido, tecnológicamente hablando, al servidor de producción aunque lo ideal es que sea una imagen del mismo. Sobre este servidor ejecutaremos todas las pruebas y en especial aquellas que no hayamos podido realizar con anterioridad porque tengan dependencia de componentes externos (como por ejemplo google maps). En el apartado sobre verificación se aportan más detalles.
- Servidor de producción: Es el servidor donde se desplegará finalmente el sistema y que usará la empresa cliente. En este proyecto la implantación definitiva en la empresa cliente queda supeditada a la aprobación por parte de la gerente y se hará a posteriori de la presentación del proyecto.

3.3.1. Recursos Hardware

En esta sección se analiza el hardware necesario para poder utilizar o probar el sistema, además de los componentes hardware que se requieren. Al tratarse de un software de gestión, para poder ejecutarse no es necesario un equipo de gama alta o con características especiales.

El software se ha desarrollado utilizando el siguiente portátil:

- **Modelo:** MSI CX61 2PC
- **Procesador:** Intel(R) Core(TM) i7-2410M 2.30 GHz
- **Memoria RAM:** 16 GB (DDR4)
- **Tipo de sistema:** Sistema operativo de 64 bits
- **Disco duro:** 240 GB SSD para el S.O. y 1 TB de 5400 para datos
- **S.O:** Windows 7 Professional Ultimate (Service Pack 1)
- **Tarjeta gráfica:** NVIDIA GeForce GT 820M, 2GB DDR3

Este ordenador ha sido modificado para contener un disco duro externo para almacenar datos y luego un SSD que aumenta enormemente la velocidad del manejo de los diferentes programas, aunque se puede programar en entornos con muchas menos prestaciones, ya que la aplicación no necesita demasiada potencia.

También se han utilizado para realizar las pruebas el siguiente smartphone:

- **Modelo:** Alcatel Pixi 4
- **Procesador:** Quad Core 1.3 GHz
- **Memoria RAM:** 1 GB (DDR3)
- **Batería:** Iones de litio de 2000 mAh
- **Memoria:** 8 GB
- **S.O:** Android 6.0
- **Dimensiones:** 5"

Y la siguiente tablet:

- **Modelo:** BQ Edison 3
- **Procesador:** Quad Core 1.3 GHz
- **Memoria RAM:** 2 GB (DDR3)
- **Batería:** 7.000 mAh con más de 9 horas.
- **Memoria:** 12 GB
- **S.O:** Android 4.4
- **Dimensiones:** 10.1"(1200 x 800 píxeles)

Además, algunos de los colaboradores del proyecto han ayudado a testearlo en dispositivos con sistema operativo IOS. Todos estos dispositivos son importantes, ya que uno de los principales objetivos es que toda la aplicación sea RESPONSIVE y se adapte a multitud de dispositivos.

3.3.2. Recursos Software

A continuación, se presentará algunos programas que se han utilizado para la elaboración del proyecto. A estos, además hay que sumar los que se especifican en la sección Entorno de construcción 6.1.

Dia v0.97.2

Dia es un software informático, parte del proyecto GNOME, que sirve para realizar diagramas de diferentes tipos, desde diagramas de flujos y redes hasta circuitos.

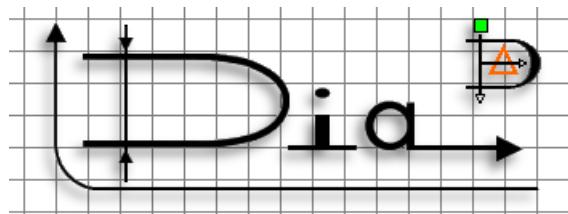


Figura 3.6: Logo DIA

Esta aplicación se ha utilizado para generar los diagramas del modelo de comportamiento en la fase de análisis, y los diagramas de entidad relación (E/R) y de estructura de cuadros (DEC) en la fase de diseño.

Latex

Este documento ha sido realizado en LaTeX empleando una distribución para Windows.

Para la escritura del mismo se ha utilizado el editor **WinEdt v5.5** con el paquete **MiKTeX v2.9**.



Figura 3.7: Logo LaTeX

LaTeX es un sistema de composición de textos, orientado a la creación de documentos escritos con una alta calidad tipográfica. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, debido a que la calidad tipográfica de los documentos realizados con LaTeX es comparable a la de una editorial científica de primera línea. Es por este motivo por el que se ha decidido utilizar LaTeX para escribir la documentación de este proyecto.

MiKTeX es la distribución LaTeX para Windows. El paquete MiKTeX no hace nada por sí solo, son únicamente programas de LaTeX que interpretan un archivo de texto con comandos de Tex y LaTeX y que producen un documento. Los documentos podrían editarse con cualquier editor de texto como WordPad o el Bloc de notas, pero existen editores de texto para Windows como WinEdt, que han sido específicamente diseñados y configurados para integrarse perfectamente con un sistema TeX como lo es MiKTeX. Desde este entorno integrado se pueden crear, editar y compilar archivos para producir el documento final en LaTeX.

Trello



Figura 3.8: Logo Trello

Trello es un software colaborativo que permite organizar cualquier tipo de proyecto con tan solo echar un vistazo al tablero, y todo se actualiza en tiempo real. No hay nada que configurar y todo el mundo lo obtiene al instante.

Trello es simple en la superficie, pero las tarjetas tienen todo lo necesario para hacer el trabajo. Publicar comentarios para obtener respuesta inmediata. Cargar archivos desde un ordenador, Google Drive, Dropbox, Box y OneDrive. Añadir checklists, etiquetas, fechas de vencimiento y otros elementos. Además, con las notificaciones se puede asegurar de saber siempre cuándo sucede algo importante.

Gimp



Figura 3.9: Logo Gimp

Gimp es un programa libre y gratuito de edición de imágenes digitales en forma de mapa de bits (dibujos y fotografías). Es el programa de manipulación de gráficos disponible en más sistemas operativos. Este programa se ha utilizado para retocar imágenes (recortar, reajustar el tamaño, etc.) de todas las capturas de pantalla presentes a lo largo del manual de usuario y de instalación.

AAA Logo



Figura 3.10: Logo AAA Logo

AAA Logo es una herramienta para crear logos con una interfaz fácil, intuitiva y muy atractiva. No es necesario poseer conocimientos de diseño para utilizar este programa ya que tiene varias herramientas para crear logos desde cero o para basarse en algunas ideas ya definidas. Una vez creado el logo, se puede exportar en distintos formatos y tamaños. Esta herramienta, en su versión de prueba de 30 días, se ha utilizado para crear el logo principal de la aplicación.

GanttProject



Figura 3.11: Logo Gantt Project

Esta aplicación es libre y de código abierto. GanttProject es una aplicación de escritorio multiplataforma hecha en Java, que permite organizar y planificar proyectos a través de diagramas Gantt. Cada proyecto puede dividirse en sub-tareas, con fecha de comienzo, duración, dependencias, progreso y anotaciones. Almacena la información en formato XML y puede exportar los diagramas a imágenes PNG/JPG, páginas web o documentos PDF. Este software se ha utilizado para realizar el diagrama Gantt del presente proyecto.

Balsamiq Mockups



Figura 3.12: Logo Balsamiq Mockups

Cuando queremos crear un proyecto nuevo, es normal que hagamos sketches, bocetos o dibujos rápidos para tener una imagen visual de cómo será nuestra interfaz. Esto nos ayuda a más adelante poder crear el proyecto lo más semejante posible a nuestro boceto en papel. Mientras lo realizamos a lápiz, hacemos cambios, tachamos, tenemos mejor ideas y estos sketches se convierten en el prototipo del proyecto final, en definitiva en Mockups. A este proceso se le denomina Wireframing que es la creación de la estructura de una interfaz. Una de las mejores herramientas para crear prototipos, bocetos o wireframes es Balsamiq Mockups. Este programa tiene varias ventajas:

- Es un programa de escritorio, es decir, solo tienes que registrarte para poder empezar a utilizarlo sin ningún tipo de descarga.
- Programado en Flex y AIR por lo que crea muy rápido Wireframes.
- Interfaz fácil de usar Como se ha creado con AIR es instalable tanto en Windows como Linux y Mac OS X.
- Balsamiq te permite escoger entre un montón de objetos prediseñados como: barras de estado, menús, barras de progreso, etc. Además, te permite exportar el diseño que realizes en PNG, PDF e incluso al portapapeles. Permite incrustar tus diseños en tu página web o en informes de errores.
- Puedes compartir tus diseños a través de un sistema muy parecido a Dropbox.
- Balsamiq funciona como cualquier aplicación normal, te permite arrastrar, soltar, guardar mockups en un archivo, copiar y pegar, deshacer, etc.

Cabe pensar que hacer esto en lápiz y papel es más rápido que realizar wireframes, pero con Balsamiq Mockups no es así. Además nos permite tener un imagen más atractiva para visualizar lo que sería en un futuro nuestro proyecto y de esta manera difundirla a nuestro equipo de trabajo o los clientes.

3.4. Costes

En este apartado se van a describir los costes relacionados con los recursos humanos y materiales (hardware y software) necesarios para la realización del proyecto.

Para la estimación de los recursos hardware se tendrá en cuenta todos los dispositivos enumerados en el apartado de organización. Además se realizará la contratación de un pequeño hosting con PLESK y un dominio propio (Openrs.es) para utilizarlo como servidor de STAGING.

Respecto a los recursos software y herramientas al ser software libre y gratuito no ha habido inversión alguna. Aquellos privativos, con su versión de prueba ha sido suficiente como el caso de Balsamiq Mockup. Por otro lado la plantilla de front-end empleada si tenía un pequeño coste fijo por adquirirla.

En cuanto a los recursos humanos, se realizarán los cálculos teniendo en cuenta el coste hora que indica el Colegio Profesional de Ingenieros en Informática de Andalucía (CPIIA) [8], que determina que son 23 €. Se prevee que el número de horas de media diaria que se realicen sea de 3 todos los días de la semana (tal como está estipulado en la planificación).

Además el autor, como es autónomo y con más de 3 años cotizados como tal, tiene unos costes en seguros sociales que también hay que reflejar. Aunque el Colegio Profesional de Ingenieros de Andalucía indica ese precio está bastante alejado de la realidad empresarial del sector, por lo que elaboraré otro presupuesto con el precio que cobraría el autor a cualquier empresa por realizar el mismo servicio.

El resto de costes correspondiente a los impuestos que exige el estado por realizar estos trabajos (IRPF e IVA) no están contabilizados, ya que doy por hecho, que son deducibles por la empresa cliente.

Por último, se han añadido una estimación de días correspondiente a los riesgos, que se detallen en el apartado de Riesgos, que afectan directamente al presupuesto omitiendo aquellos que pueden retrasar el proyecto por otras causas pero no cuentan como desarrollo.

En la siguiente tabla se muestra el desglose de los costes descritos anteriormente siguiendo

los criterios establecidos según el CPIIA:

| CONCEPTO | CANTIDAD | COSTE UNITARIO | COSTE TOTAL |
|----------------------|---------------|----------------|-------------|
| Desarrollo | 255 d (765 h) | 23 € | 17595 € |
| Plantilla Responsive | 1 | 5 € | 5 € |
| Dominio Openrs.es | 1 | 4,99 € | 4,99 € |
| Servidor STAGING | 5 meses | 8,99 € | 44,95 € |
| Laptop | 1 | 672 € | 672 € |
| Tablet | 1 | 181 € | 181 € |
| Smarthphone | 1 | 50 € | 50 € |
| Riesgos previstos | 15 d (45 h) | 23 € | 1035 € |
| Seguros sociales | 9 meses | 267,03 € | 2403,27 € |

Tabla 3.3: Estimación de Costes del proyecto (estimación segun CPIIA)

Obtenemos un coste neto total de **21991,21 €**.

En la siguiente tabla se muestra el desglose de los costes descritos anteriormente siguiendo los criterios establecidos según la tarifa de autónomo:

| CONCEPTO | CANTIDAD | COSTE UNITARIO | COSTE TOTAL |
|----------------------|---------------|----------------|-------------|
| Desarrollo | 255 d (765 h) | 13,75 € | 10518,75 € |
| Plantilla Responsive | 1 | 5 € | 5 € |
| Dominio Openrs.es | 1 | 4,99 € | 4,99 € |
| Servidor STAGING | 5 meses | 8,99 € | 44,95 € |
| Laptop | 1 | 672 € | 672 € |
| Tablet | 1 | 181 € | 181 € |
| Smarthphone | 1 | 50 € | 50 € |
| Riesgos previstos | 15 d (45 h) | 13,75 € | 618,75 € |
| Seguros sociales | 9 meses | 267,03 € | 2403,27 € |

Tabla 3.4: Estimación de Costes del proyecto (estimación segun tarifa autónomo)

Obtenemos un coste neto total de **14498,71 €**, por lo que habría una diferencia de **7492,5 €**, es decir, un ahorro del 34,07 % del coste total del presupuesto inicial estimado.

3.5. Riesgos

A continuación se va realizar una enumeración de los riesgos del proyecto, indicando su posible impacto, al que llamaremos impacto directo (efecto que la ocurrencia del citado riesgo tendría en el desarrollo del proyecto) y la probabilidad de ocurrencia. El resultado de multiplicar los dos datos anteriores nos dará el impacto estimado.

Una vez los riesgos han sido identificados y priorizados, se han definido los planes necesarios para reducir los efectos del riesgo una vez se haya materializado o la disminución para que estos ocurran.

| RIESGO | PROB. | I. DIRECTO | I. ESTIMADO |
|--|-------|------------|-------------|
| Otras causas imprevisibles | 90 | 5 d | 4.5 d |
| Problemas en despliegue | 90 | 3 d | 2.7 d |
| Problemas con integración de componentes | 75 | 10 d | 7.5 d |
| Peticiones de clientes externos | 50 | 20 d | 10 d |
| Falta flexibilidad | 50 | 5 d | 2.5 d |
| Falta concreción áreas clave | 25 | 5 d | 2.5 d |
| Enfermedad | 5 | 5 d | 0.25 d |
| Cambio en la normativa vigente | 1 | 1 d | 0.01 d |

Tabla 3.5: Riesgos

Obtenemos una estimación total de **30** días, es decir, tenemos que tener un mes completo de margen para poder cumplir con la fecha límite del proyecto.

Para paliar los efectos de los riesgos sobre el proyecto, en la planificación del calendario del proyecto se ha tenido en consideración, en la medida de lo posible, el impacto de los riesgos, aumentando un poco la estimación de la duración de las actividades con más peso para intentar contrarrestar la pérdida de tiempo que se daría si ocurriesen.

Por otro lado, y como se indica en el diagrama PERT, se han dejado determinadas actividades como opcionales, de forma que si hubiera qué elegir qué hacer en un determinado momento de presión, poder descartar alguna de ellas.

Además, aplicaremos los criterios de aseguramiento de la calidad que se describen posteriormente y las técnicas de seguimiento especificadas anteriormente para poder tener un control efectivo del proyecto a cada paso de su desarrollo.

3.6. Aseguramiento de calidad

Además de los explicado en el apartado Metodología de desarrollo, existen dos procesos fundamentales para asegurar la calidad del producto así como la aplicación de las mejores prácticas para su construcción.

Primero, se establece la ejecución de pruebas de calidad desde el inicio de los proyectos hasta su finalización, definiendo un conjunto de procesos que aplican de forma coordinada con los procesos de desarrollo, con la finalidad de incrementar la calidad de los proyectos, y establecer un conjunto objetivo de criterios de validación y verificación de las entregas a lo largo del ciclo de vida de las aplicaciones. Para conseguirlo se ha tenido en cuenta el enfoque de MADEJA acerca del ciclo de vida de desarrollo, prestando especial atención a los productos esperados en cada etapa (documentación y software). Como resultado se va a adoptar un modelo en W para la revisión de estos productos tal como describe la siguiente figura:

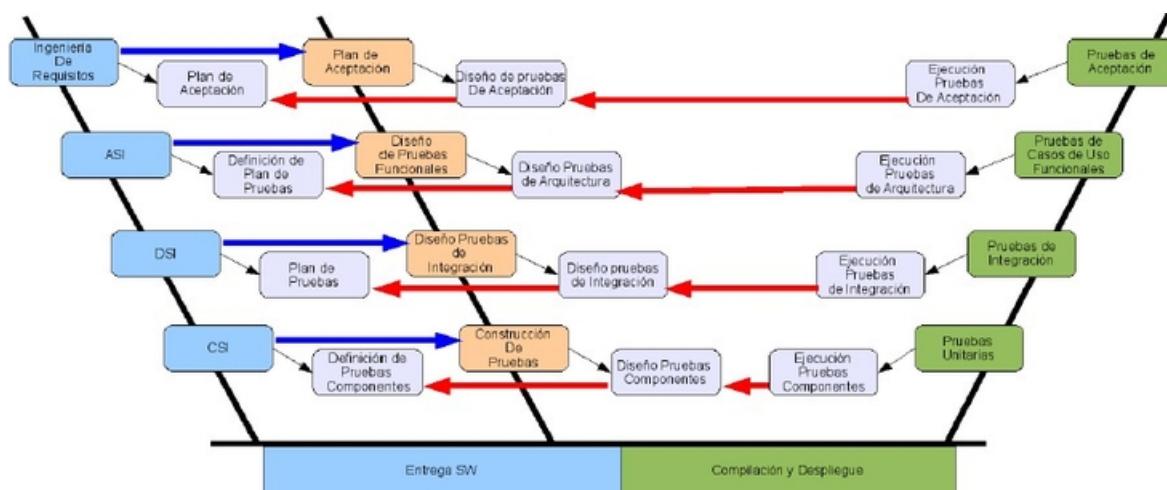


Figura 3.13: Modelo de verificación en W

Este modelo plantea que a la vez que se vayan realizando los diferentes procesos de ingeniería se vayan definiendo los planes de pruebas que posteriormente se irán validando a medida que se vaya construyendo el sistema en las sucesivas iteraciones.

Por otro lado, emplearemos el uso de lo que denomino como PCS (Propuesta de cambio de software) junto con GIT (Sistema de control de versiones distribuido) siguiendo el proceso que define la siguiente figura:

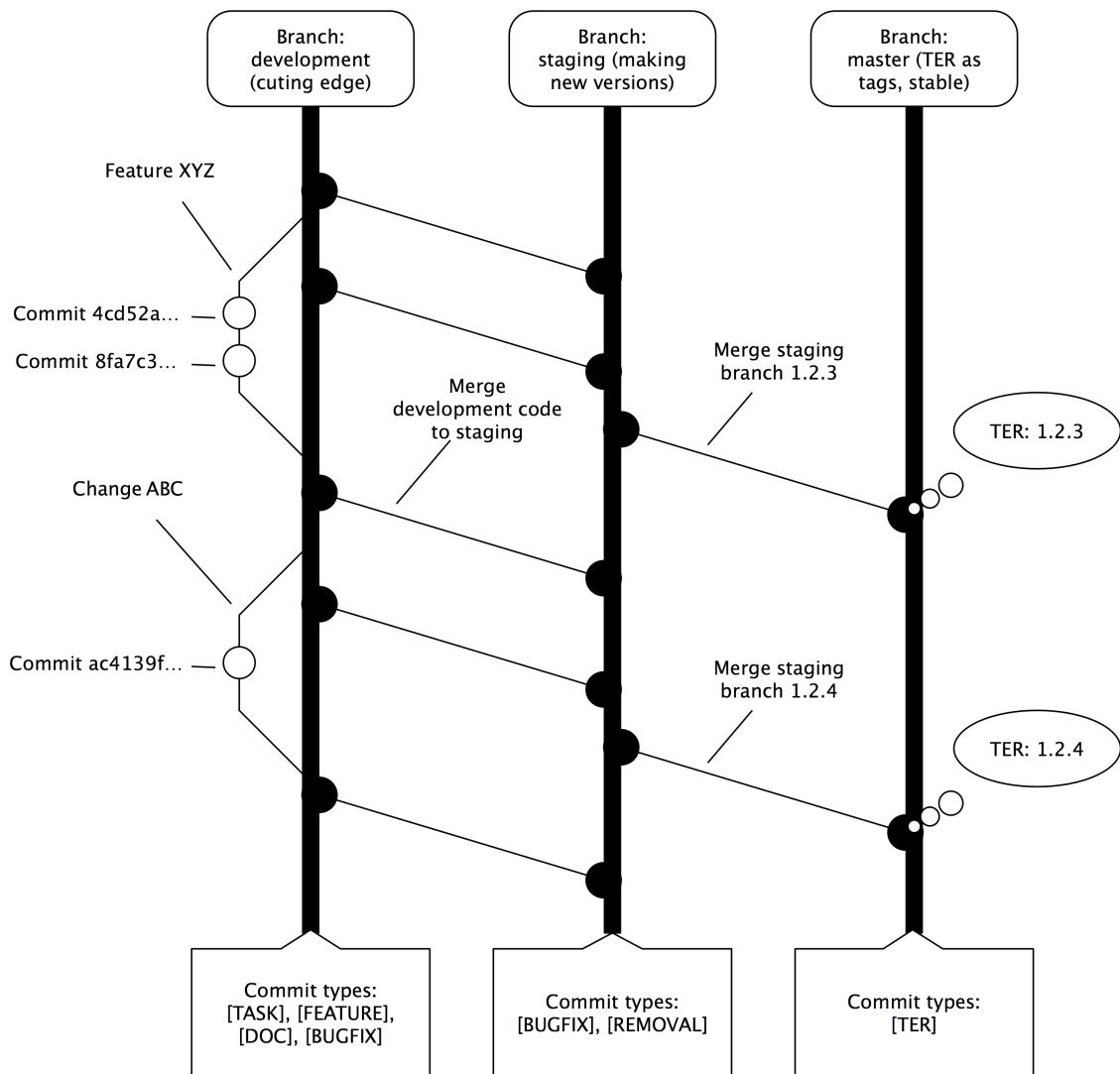


Figura 3.14: Diagrama de Flujo GIT

Tal como muestra el diagrama de flujo, seguiremos la siguiente secuencia iterativa de pasos para elaborar los distintos prototipos a la vez que aseguramos la integridad de los cambios y la validez de los mismos:

1. Cada vez que vayamos a realizar un cambio en el software (ya sea debido a la creación de una funcionalidad, la corrección de un error, etc), generaremos un documento denominado PCS (con una numeración única), en el que describiremos los cambios que hemos ido realizando así como las soluciones propuestas a las diferentes dificultades.

Este documento estará ligado exclusivamente a una sólo rama (branch) de desarrollo de GIT. Aunque nos sirve como CHANGE LOG, va dirigido exclusivamente a los desarrolladores, y permite tener un control de lo que se ha hecho con exactitud en esa rama en concreto. Dependiendo de la complejidad de los cambios implementados, un PCS podrá contener la información de un sólo commit o de varios.

Adjuntos a estos documentos, dispondremos de un proceso de migración que define como actualizar la actual versión del software hacia la siguiente versión. El framework escogido para su desarrollo (Codeigniter 3) permite automatizar estos procesos a través de código PHP usando determinadas librerías que permiten realizar tanto los rollbacks como las migraciones.

Por agilidad y simplicidad, así como que, por norma general, los prototipos al ser incrementales no necesitarán rollbacks, se crearán ficheros SQL para aplicar directamente en el servidor correspondiente.

2. Adjunto a cada PCS se elabora una matriz de prueba que se ejecuta en el entorno de desarrollo local.
3. Una vez los dos anteriores pasos han sido realizados, se pasa a mergear la rama en la que estamos trabajando sobre la rama de STAGING.
4. Utilizamos una herramienta de despliegue, en este caso Filezilla, para subir los cambios así como toda la documentación anteriormente descrita al servidor de STAGING.
5. Luego pasamos a ejecutar los script de migración SQL (si los hubiere) en el SGBD de STAGING. Para nuestro caso, será PHPMYADMIN, que es la herramienta que nos proporciona el PLESK (Panel de control del entorno de STAGING). También puede hacerse configurando la conexión directamente a través de MYSQL WORKBENCH y aplicando los cambios a través de su consola de comandos.
6. Sobre este entorno volveremos a pasar todas las pruebas matrices de prueba, pero sobre todo haremos énfasis en comprobar aquellas funcionalidades o requisitos donde el entorno de implantación juegue un papel importante o simplemente, sobre aquellas en las que en el entorno de desarrollo no se han podido verificar. Por ejemplo, es imposible de verificar en un entorno de desarrollo, que el sistema conecta correctamente con Google Analytics y registra los datos en la cuenta correspondiente o que

la conexión con Google Maps funciona correctamente utilizando la API-KEY correspondiente. Por otro lado, hay funcionalidades como aquellas en las que interactúe con el sistema de ficheros, que debido a la diferencia de sistemas operativos de los entornos hay que revisar cuidadosamente.

Adicionalmente, y si se considera necesario, también podrán ejecutarse las pruebas de integración y unitarias que se consideren oportunas, aunque, en principio, la mayoría de estas pruebas se pasarán en el entorno de desarrollo. No obstante, si una determinada funcionalidad falla, estas sirven de apoyo en este entorno para depurar los errores obtenidos.

7. Si todas las pruebas se ejecutan correctamente en el entorno de STAGING, entonces, la rama correspondiente ha sido comprobada con éxito y pasa a mergearse con la rama de MASTER (que representará al entorno de producción) y creará una nueva versión estable del sistema.
8. En caso contrario, deberemos anotar los problemas ocurridos, buscar una solución, implementar y volver a realizar el proceso de verificación desde el primer paso.

En definitiva, lo que se pretende con todo esto es abandonar definitivamente desarrollos realizados con la estrategia (develop-test-deploy) y pasar a usar una estrategia que incluya a un entorno de STAGING como paso previo al despliegue.

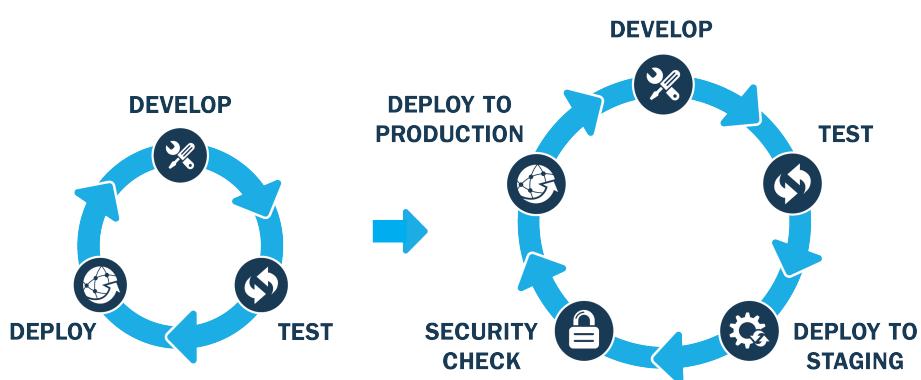


Figura 3.15: Evolución de estrategia Develop-Test-Deploy

A los lectores de esta memoria les recomiendo encarecidamente que lean los PCS, las matrices de prueba así como el diario de retos (en la medida de lo posible), ya que la vida y evolución real del software está descrita con detalle en estos ficheros.

3.7. Seguimiento

En esta sección se pretende diferenciar la planificación estimada frente a la invertida realmente tanto en coste temporal como de recursos así como determinar qué técnicas de seguimiento del proyecto se han empleado.

| CONCEPTO | CANTIDAD | COSTE UNITARIO | COSTE TOTAL |
|-------------------------|---------------|----------------|-------------|
| Desarrollo Actividades | 110 d (330 h) | 13,75 € | 4537,5 € |
| Desarrollo Iteraciones* | 122 d (705 h) | 13,75 € | 9693,75 € |
| Plantilla Responsive | 1 | 5 € | 5 € |
| Dominio Openrs.es | 1 | 4,99 € | 4,99 € |
| Servidor STAGING | 5 meses | GRATIS | - |
| Laptop | 1 | 672 € | 672 € |
| Tablet | 1 | 181 € | 181 € |
| Smarthphone | 1 | 50 € | 50 € |
| Riesgos previstos | 12 d (36 h) | 13,75 € | 495 € |
| Seguros sociales | 9 meses | 267,03 € | 2403,27 € |

Tabla 3.6: Costes reales de recursos empleados en el proyecto

*El número de horas tiene en cuenta la disminución de horas durante el mes de Julio tal y como se comenta posteriormente.

El coste final ascendió a **18042,51 €**, por lo que habría un incremento del coste del proyecto en **3543,8 €**, el cual radica en los siguientes motivos:

1. La implementación, redacción y ejecución de los planes de pruebas han sido mucho más costosas de lo supuesto inicialmente debido a mi inexperiencia en la aplicación de buenas prácticas en este ámbito.
2. La integración de componentes y el aprendizaje de estos ha sido más costoso durante la construcción. Además se han tenido que buscar algunos adicionales y su aprendizaje sobre la marcha.
3. Se han añadido funcionalidades que inicialmente no se habían tenido en cuenta, como por ejemplo el mapeado de Google de inmuebles público y privado, las cuales han resultado ser costosas.

4. Los ingenieros somos personas, y el cansancio, sobre todo en las fases finales del proyecto, hace más mella que en las fases iniciales por lo que la productividad es peor.

Por otro lado, el calendario temporal, quedó definitivamente así:

| TAREA | COMIENZO | FIN | DURACION |
|---|----------|-------|----------|
| Estudio del dominio del problema y sistemas actuales | 09/01 | 29/01 | 21 |
| Identificación de los objetivos del sistema | 30/01 | 31/01 | 2 |
| Identificación de los requisitos funcionales | 01/02 | 06/01 | 5 |
| Realización del estudio de viabilidad analizando las diferentes alternativas tecnológicas | 07/02 | 17/03 | 40 |
| Estudio de componentes de la alternativa seleccionada | 18/03 | 07/04 | 21 |
| Análisis | 08/04 | 11/04 | 3 |
| Diseño | 12/04 | 17/04 | 5 |
| Documentación inicial del proyecto | 18/04 | 12/05 | 10 |
| Implementación inicial de la base de datos y framework de desarrollo | 29/04 | 02/05 | 4 |

Tabla 3.7: Coste temporal real de actividades

| TAREA | COMIENZO | FIN | DURACION |
|---|----------|-------|----------|
| Integrar los principales componentes y obtener un framework básico de trabajo funcional | 03/05 | 12/05 | 10 |
| Personalización y mantenimiento del sistema | 13/05 | 22/05 | 10 |
| Módulo de Clientes | 22/05 | 05/06 | 14 |
| Módulo de Inmuebles | 06/06 | 19/06 | 14 |
| Módulo de Demandas | 20/06 | 24/06 | 5 |
| Refinamiento de funcionalidades | 24/06 | 09/06 | 8 |
| Generación de informes | 29/06 | 04/07 | 7 |
| Documentación del proyecto | 10/07 | 05/08 | 20 |
| Estadísticas | 06/08 | 11/08 | 5 |
| Zona pública | 12/08 | 26/08 | 14 |
| Conclusión de integración de componentes y refinamientos finales | 27/08 | 14/09 | 7 |
| Pruebas de aceptación | 01/09 | 05/09 | 3 |
| Conclusión documentación proyecto | 28/08 | 07/09 | 7 |

Tabla 3.8: Coste temporal real de Iteraciones para la construcción del software

Se observa que a pesar de que se han trabajado incluso menos días (ya que la fecha final del proyecto fue incluso menor por desconocimiento del límite de plazo para entregar la documentación), se ha compensado aumentando la jornada de trabajo en el tramo de construcción del sistema y pruebas. Para ello ha sido irremediable disminuir la cantidad de tiempo que dedicaba a trabajos que tenía con mi cliente actual a partir de mediados de Mayo, ya que la compatibilidad total de ambos proyectos era inviable debido a los motivos comentados anteriormente. En el siguiente gráfico se puede apreciar la evolución de horas empleadas de media al día:



Figura 3.16: Evolución real de horas diarias empleadas

Además se puede apreciar que durante el mes de Julio tuve unos problemas de salud que duraron cerca de dos semanas por lo el número de horas de media trabajadas disminuyó. Para realizar el seguimiento del proyecto, se ha utilizado el diario de retos así como los PCS que se explican más adelante. La herramienta de Gantt Project y esta propia memoria también ha sido de gran utilidad para poder visualizar gráficamente el calendario así como decidir de qué funcionalidades se podían prescindir. También se ha ido anotando posibles mejoras y cambios pendientes, de forma que se podía saber qué cosas faltaban por realizar y poder darles una prioridad. También se ha hecho uso de Trello para coordinar actividades con los asesores de la empresa TIPYCOS, aunque en menor medida, ya que fue una herramienta que descubrí en etapas muy tardías del proyecto, pero de la cual, me gustaría indagar más en el futuro.

Parte II

Desarrollo

Capítulo 4

Análisis del Sistema

4.1. Modelo de datos

4.1.1. Modelo Conceptual

Tradicionalmente, a partir de los requisitos de información se ha desarrollado un diagrama E/R (Entidad Relación), cuya misión era ofrecer, a un alto nivel de abstracción, la información qué iba a registrar el sistema y cómo se iba a relacionar.

En este se puede apreciar una organización inicial del sistema, algo desordenada, en demandas, inmuebles, clientes, los agentes inmobiliarios y la documentación generada, así como los ficheros que se iban a adjuntar. El diagrama inicial que pensé se representa a través de la siguiente figura 4.1.

No obstante, y tal como especifica Métrica v3 [1] para aquellos desarrollos Orientados a Objetos, se hace necesario ofrecer un modelado más preciso que desarrolle más en profundidad las ideas base anteriormente descritas. Para ello, utilizaremos como técnica el diagrama de clases UML, identificando las clases, atributos, relaciones y restricciones adicionales de nuestro sistema.

Marcas de campos según el tipo de documento
%descripción_vivienda para un inmuble

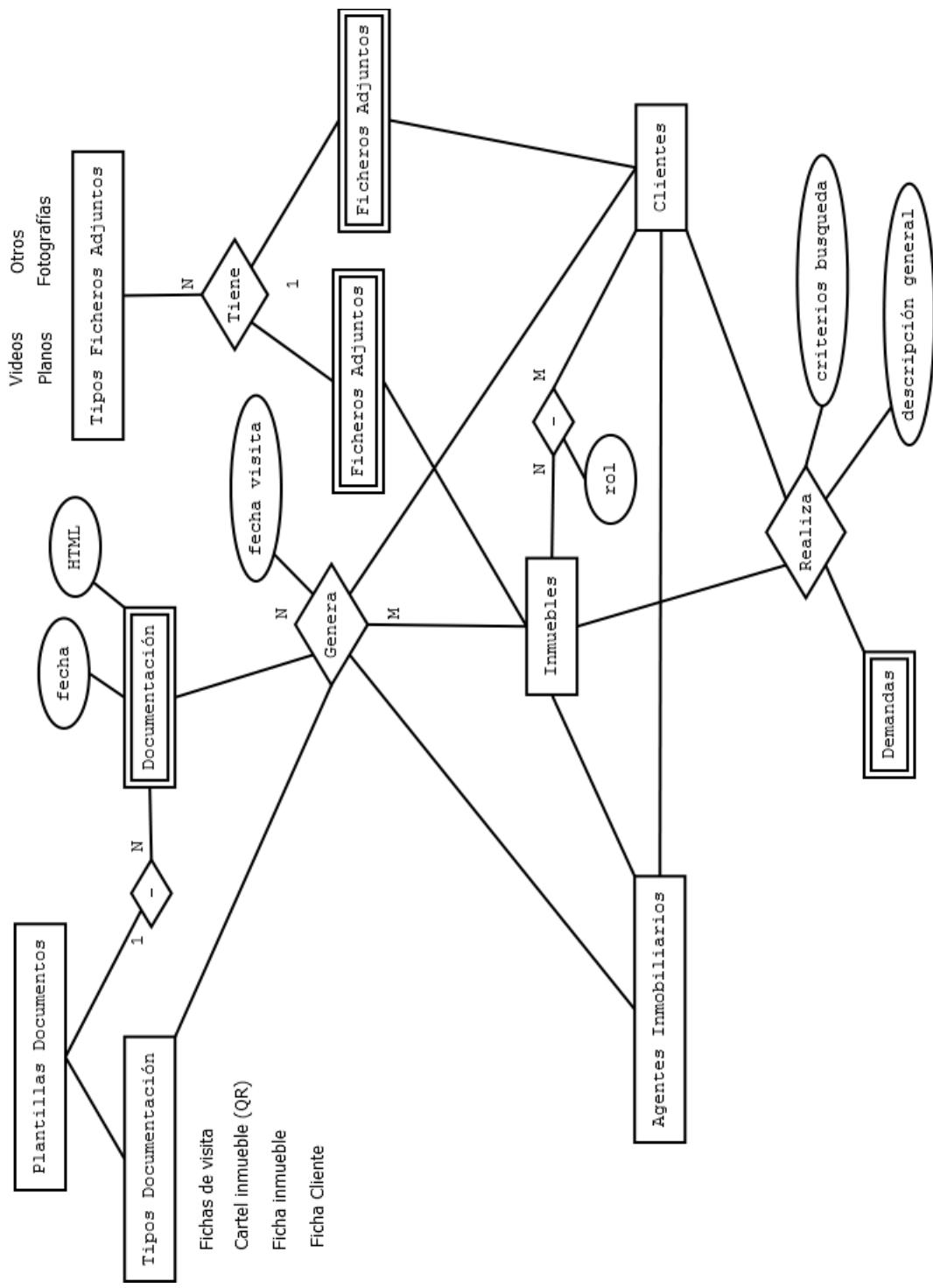


Figura 4.1: Diagrama E/R inicial

Debido a la dimensión del sistema se van a ofrecer diferentes diagramas atendiendo a las principales subsistemas de los que va a constar el software. No obstante, el lector debe tener en cuenta que estos módulos no son inconexos por lo que deberá visualizarlos desde una perspectiva global.

Toda la información procedente del software de base no ha sido especificada en esta documentación, ya que, son los creadores de estos subsistemas los que les toca la responsabilidad de documentar su software.

No obstante, es posible que para aclarar ciertas parametrizaciones realizadas para adaptarlos al sistema, se expliquen algunos detalles (como por ejemplo, la inclusión de las clases agente o idioma, pertenecientes a los subsistemas de gestión de usuario y gestión de sitio web).

Posteriormente, en la sección de Modelo Lógico de datos 4.1.2, se aplicarán una serie de reglas de derivación para obtener el esquema relacional y se detallarán el resto de campos que, por motivos de espacio en el diagrama, no aparecen en algunas clases como las del cliente o el inmueble.

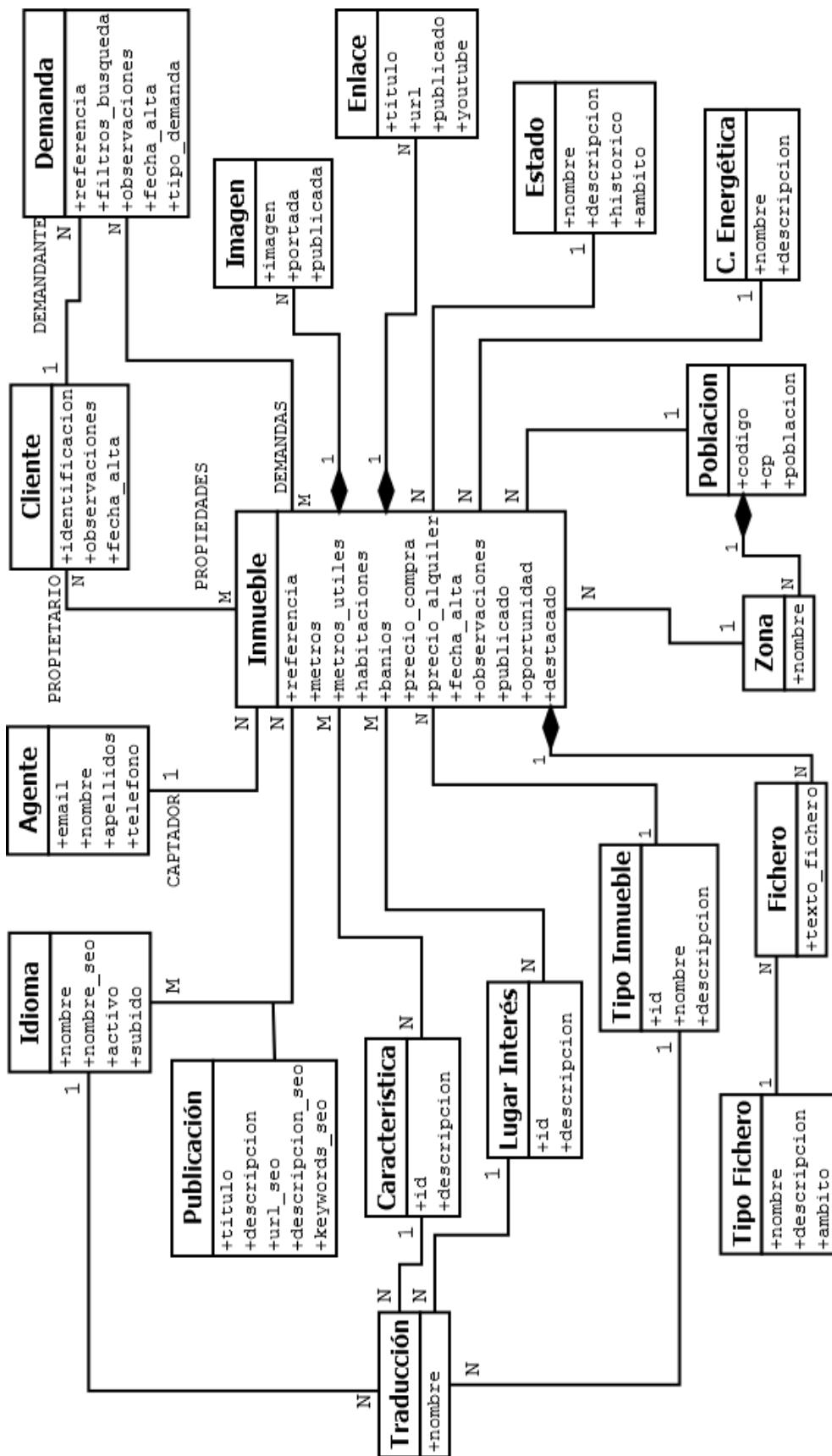


Figura 4.2: Diagrama de clases del Subsistema de Inmuebles

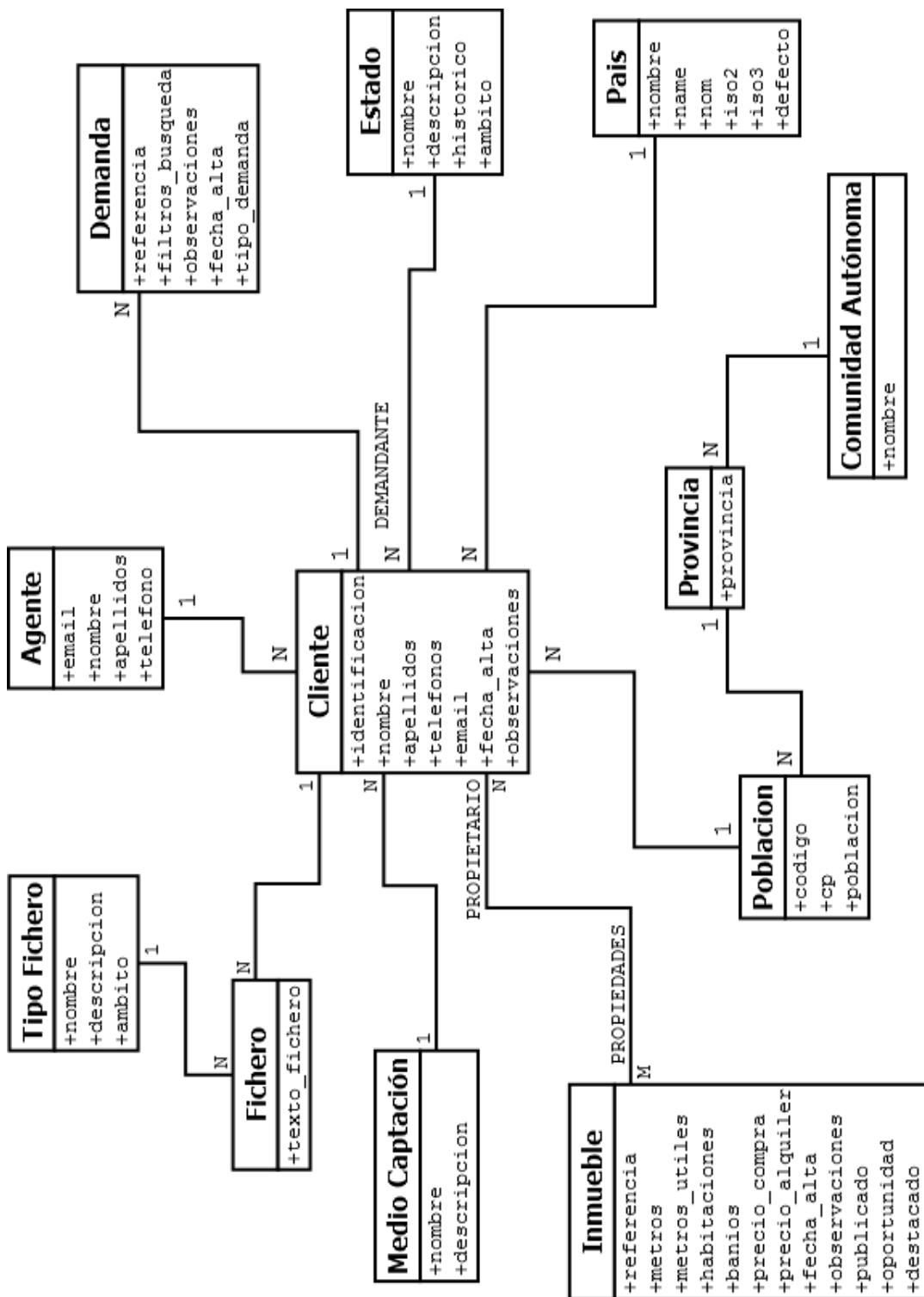


Figura 4.3: Diagrama de clases del Subsistema de Clientes

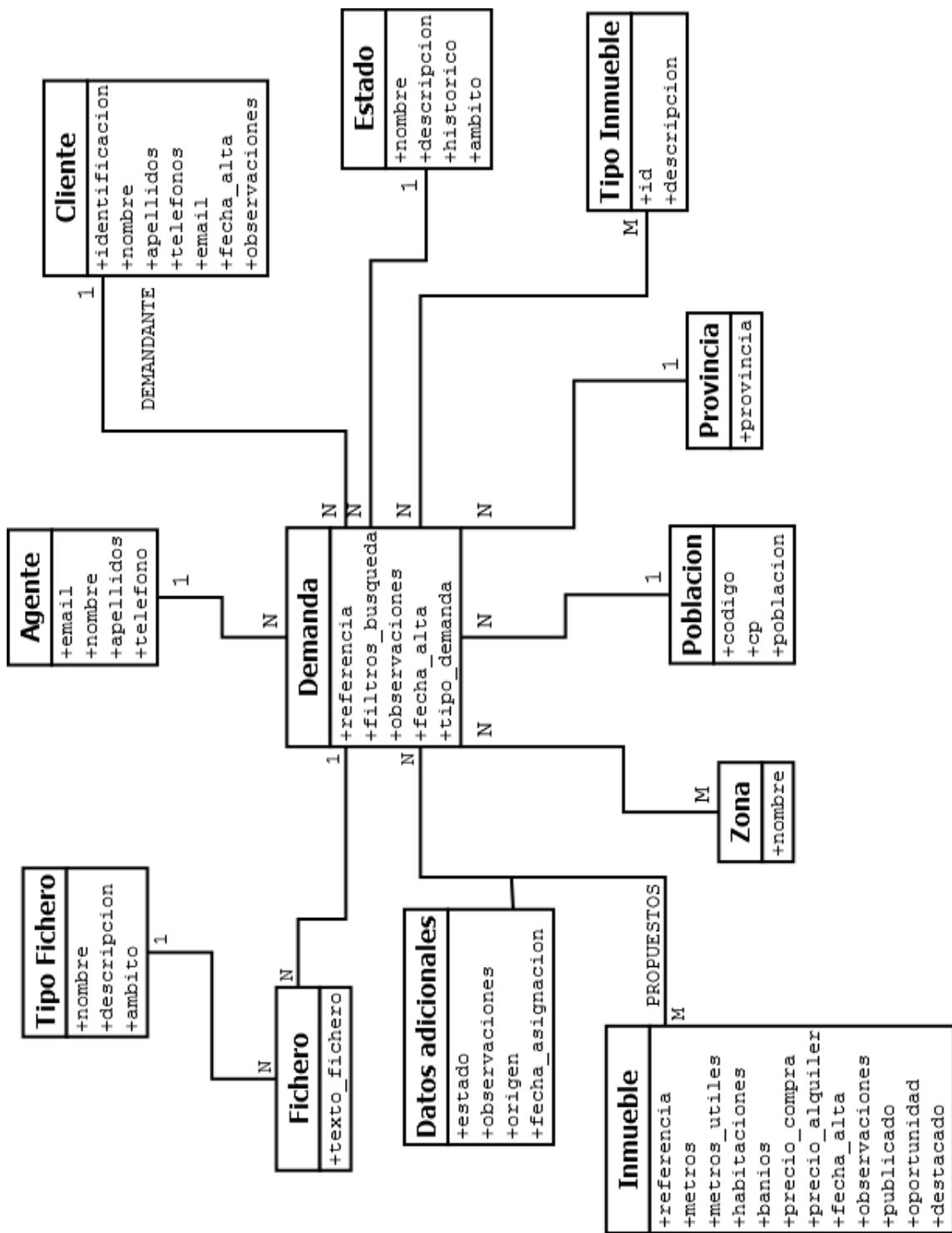


Figura 4.4: Diagrama de clases del Subsistema de Demandas

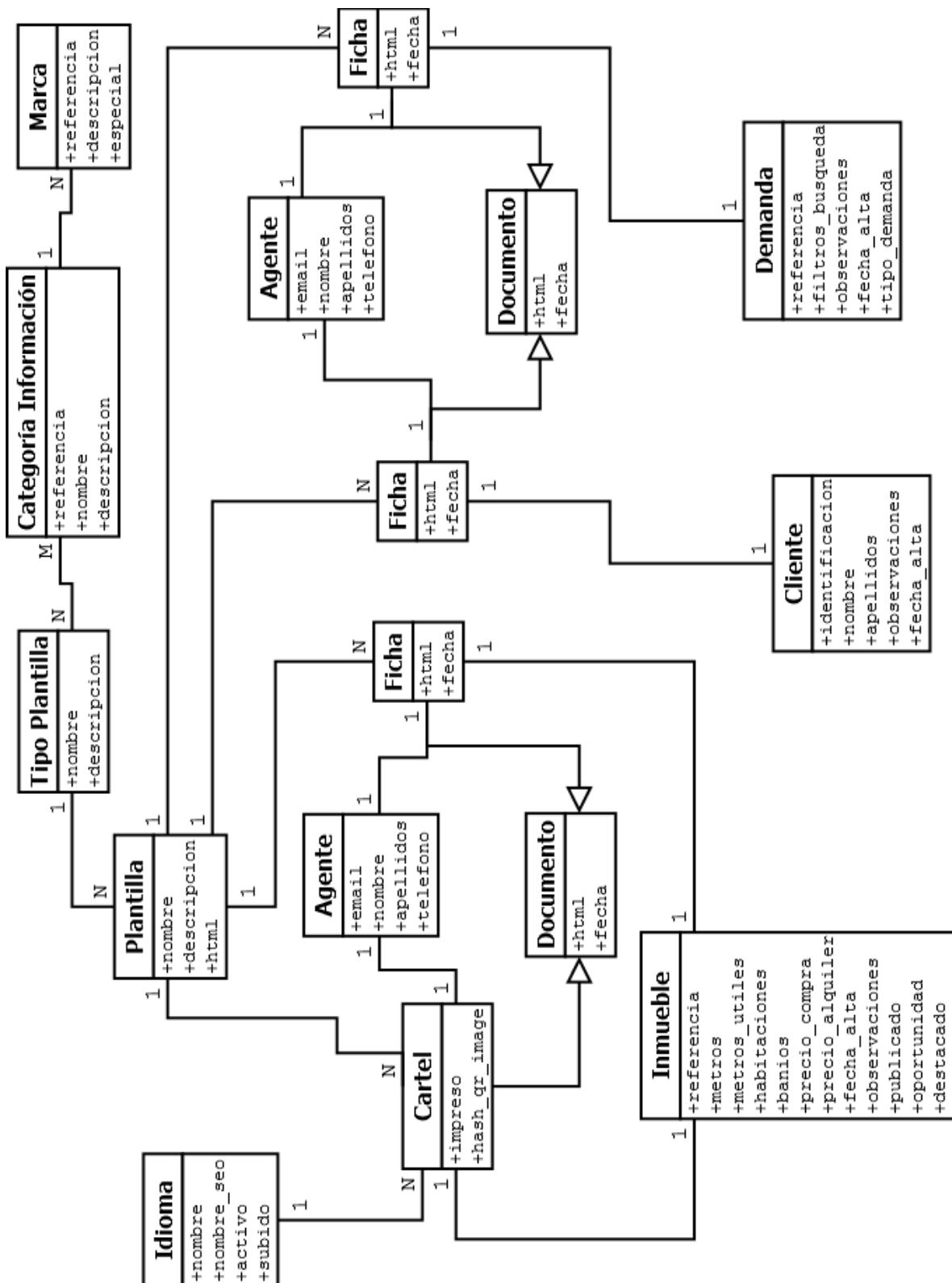


Figura 4.5: Diagrama de clases de la Documentación

4.1.2. Modelo lógico

Para obtener el modelo lógico del sistema utilizaremos el modelo relacional el cual nos sirve para modelar la base de datos obteniendo los siguientes beneficios:

- *Independencia física*: modificación del almacenamiento físico sin modificación de aplicaciones.
- *Independencia lógica*: manipulación lógica de la base de datos sin modificación de las aplicaciones.
- *Flexibilidad*: ofrecer los datos a las aplicaciones como ellas lo requieran.
- *Uniformidad*: presentación de la información en tablas.
- *Sencillez*: lenguajes sencillos de manejar.

Los principios fundamentales que se seguirán para diseñar una base de datos relacional a partir del diagrama de clases son los siguientes:

- Partiendo de los diagramas de clase para los distintos módulos diseñados en la sección anterior, las clases pasarán a ser **tablas** manteniendo la independencia de esta estructura lógica con las características físicas (independencia de ordenación, indexación y caminos de acceso).
- Los atributos pasarán a ser **columnas** (a excepción de los atributos multivaluados). Los atributos compuestos se transformarán en todos los atributos simples que los componen.
- A las **filas** se les llaman **tuplas** (ocurrencias de la tabla o relación).
- El conjunto no vacío de atributos que identifica únicamente y mínimamente cada tupla se denomina **Clave candidata**. El usuario debe de elegir una y ésta será la **Clave primaria**.
- Se denomina **Clave Ajena o Foránea** a un atributo o conjunto de atributos de una tabla o relación que coinciden con los de la clave primaria de otra tabla o relación. Ambas claves, primaria y ajena, toman valores del mismo dominio. La clave ajena sirve para establecer relaciones entre tablas.

- Las relaciones 1:N o 1:1 que aparecen en los diagramas se transforman propagando el atributo identificador principal de la clase que tiene cardinalidad máxima 1 a la que tiene cardinalidad máxima N. Si la relación fuese 1:1, la propagación de la clave se podría realizar en cualquier sentido.
- Las relaciones N:M se transforman en una tabla cuya clave primaria será la concatenación de los atributos principales de las clases que asocia. Estos atributos serán **Claves Ajena**s que referencian a las respectivas tablas donde son claves primarias. Los atributos de la relación (si los tuviera) serán columnas de la tabla.

Debido a la dimensión del diagrama el lector que deseé visualizar el modelo relacional de manera gráfica puede hacerlo realizando una importación del mismo (situado en los anexos) a través de la herramienta MYSQL WorkBench [11].

Primer forma normal: 1NF

Dependiendo del autor, existen diferentes condiciones mas o menos restrictivas sobre como considerar que una tabla está en 1FN. Nosotros vamos a considerar aquellas que cumplan estas tres:

- No contienen tuplas repetidas.
- No contiene atributos multivaluados.
- Todo valor depende de la clave primaria.

Todas las relaciones cumplen estas condiciones excepto las siguientes, que dependiendo de lo restrictiva de la aplicación de las anteriores normas podrían no cumplir:

- Demandas: Todos los filtros de búsqueda no son obligatorios, por lo que podría insertarse muchos registros iguales si no se definen ninguno.
- Clientes: Los teléfonos del cliente se almacenan como una secuencia separada por comas de los diferentes teléfonos.

Segunda forma normal: 2NF

En todas las relaciones hay una dependencia completa de sus claves por lo que está en Segunda Forma Normal.

Tercera forma normal: 3NF

No existen dependencias transitivas, es decir, de un campo que no es clave sobre otras, excepto en las demandas, cuyo filtro de búsqueda depende del tipo de demanda, ya que

si este no está definido, todos los valores se almacenan vacíos. No obstante, se ha dejado esta forma ya que al principio, la implementación obligaba a establecer filtros de búsqueda, pero por sugerencia de GESTICADIZ, es cierto, que las demandas suelen ser con frecuencia sobre inmuebles individuales, por lo que, estos campos no se tienen en cuenta.

Relaciones resultantes

- categorias_informacion_documentacion(referencia, nombre, descripcion)
- clientes(identificacion, nombre, apellidos, fecha_nac, direccion, pais_id, poblacion_id, telefonos, observaciones, fecha_alta, correo, estado_id, agente_asignado_id)
- clientes_fichas(cliente_identificacion, plantilla_id, agente_id, html, fecha): Fichas de los clientes.
- clientes_ficheros(id, cliente_identificacion, texto_fichero, tipo_fichero_id): Ficheros adjuntos de los clientes.
- clientes_inmuebles(cliente_identificacion, inmueble_referencia): Propietarios de los inmuebles.
- comunidades_autonomas: Comunidades autónomas.
- demandas: Demandas de los clientes.
- demandas_fichas: Fichas de las demandas.
- clientes_ficheros: Ficheros adjuntos de las demandas.
- demandas_poblaciones_zonas: Zonas de los municipios correspondientes a los filtros de búsqueda de una demanda.
- demandas_tipos_inmueble: Tipos de inmuebles correspondientes a los filtros de búsqueda de una demanda.
- estados: Estados que se pueden asignar los inmuebles, clientes, y demandas.
- grupos: Grupos posibles de usuarios registrados en el sistema (agentes o administradores).
- idiomas: Idiomas registrados en el sistema.
- inmuebles: Inmuebles

- inmuebles_carteles: Carteles de los inmuebles.
- inmuebles_demandas: Demandas de los inmuebles.
- inmuebles_enlaces: Enlaces de los inmuebles.
- inmuebles_fichas: Fichas de los inmuebles.
- inmuebles_ficheros: Ficheros adjuntos de las demandas.
- inmuebles_idiomas: Datos de publicación en cada idioma de los inmuebles.
- inmuebles_imagenes: Imágenes de los inmuebles.
- inmuebles_lugares_interes: Lugares de interés (o sitios cercanos) asignadas a los inmuebles.
- inmuebles_opciones_extras: Opciones extras (características) asignadas a los inmuebles.
- lugares_interes: Información base sobre los Lugares de interés.
- lugares_interes_idiomas: Datos dependientes del idioma de los lugares de interés.
- marcas_documentacion
- opciones_extras: Información base sobre las Opciones extras.
- opciones_extras_idiomas: Datos dependientes del idioma de las opciones extras.
- paises: Paises (Lugares de residencia).
- plantillas_documentacion: Plantillas disponibles para generar documentación en el sistema (fichas y carteles).
- poblaciones: Municipios registrados a 1 de Enero según [7].
- poblaciones_zonas: Zonas de los municipios.
- provincias: Provincias registradas.
- tipos_certificacion_energetica: Niveles de certificados energéticas y estados posibles.
- tipos_ficheros: Tipos de ficheros en los que pueden ser catalogados los ficheros adjuntos.

- tipos_inmueble: Información base sobre los diferentes tipos de inmuebles.
- tipos_inmueble_idiomas: Información en diferentes idiomas de los diferentes tipos de inmuebles.
- tipos_plantilla_documentacion: Tipos de plantillas de documentación que se pueden utilizar.
- tipos_plantilla_documentacion_categorias_asignadas: Categorías de información asignadas a los tipos de plantillas anteriores.
- usuarios: Usuarios registrados.
- usuarios_grupos: Permisos de los usuarios en el sistema.

En la sección de Diseño Físico de Datos 5.3 se añadirán otras estructuras adicionales para dar soporte a otras funcionalidades o que completan las funcionalidades de los componentes externos.

4.2. Catálogo de requisitos detallado y Subsistemas de análisis

Debido a la cantidad de requisitos especificados se hace necesario descomponerlos en otros subsistemas que se encarguen de organizar y agrupar funcionalidades bajo un mismo objetivo. Estos objetivos se especificarán en la siguiente sección y luego, haremos una visión particular de uno de los subsistemas definidos.

En el Sitio web (zona pública), no existe tanta complejidad en cuanto análisis, por lo que se pueden especificar directamente los casos de uso. No obstante, en el Software de gestión (zona privada), podríamos definir los siguientes subsistemas:

- **Configuración del sitio web:** Encargado principalmente de definir el aspecto y el contenido (excepto los inmuebles publicados) de las diferentes secciones de la zona pública. Se subdivide a su vez en:
 - Gestión de los idiomas.
 - Configurar el aspecto (colores corporativos y logo).
 - Establecer datos del pie.

- Gestión de las noticias.
 - Gestión de la secciones.
- **Configuración de datos de la aplicación:** Así como sus preferencias, define los diferentes datos que se pueden establecer en el sistema. Se subdivide a su vez en:
- Gestión de estados.
 - Gestión de plantillas de documentación.
 - Gestión de tipos de ficheros.
 - Gestión de tipos de inmuebles.
 - Gestión de lugares de interés (sitios cercanos).
 - Gestión de opciones extras (características).
 - Gestión de poblaciones y zonas.
 - Establecer preferencias.
- **Gestión de usuarios:** Su principal objetivo es registrar los usuarios que tienen acceso al sistema y establecer sus permisos.
- **Gestión de copias de seguridad:** Su principal objetivo es realizar las copias de seguridad del sistema.
- **Gestión de inmuebles:** En el que podemos destacar los siguientes casos de uso:
- Buscar
 - Insertar
 - Eliminar
 - Duplicar
 - Editar:
 - Publicar.
 - Visualizar en google maps.
 - Gestión características:
 - ◊ Listar
 - ◊ Asignar
 - ◊ Quitar

- Gestión sitios cercanos:
 - ◊ Listar
 - ◊ Asignar
 - ◊ Quitar
- Gestión demandas asociadas:
 - ◊ Listar
 - ◊ Asociar
 - ◊ Crear
 - ◊ Editar
 - ◊ Desasociar
- Gestión propietarios:
 - ◊ Listar
 - ◊ Asociar
 - ◊ Crear
 - ◊ Editar
 - ◊ Desasociar
- Gestión de cartel:
 - ◊ Generar
 - ◊ Editar
 - ◊ Eliminar
- Gestión de ficha:
 - ◊ Generar
 - ◊ Editar
 - ◊ Eliminar
- Gestión de ficheros adjuntos:
 - ◊ Listar
 - ◊ Adjuntar
 - ◊ Eliminar
- Gestión de imágenes:
 - ◊ Listar
 - ◊ Publicar

- Establecer como portada
- Descargar
- Eliminar
- Subir múltiples imágenes
- Gestión de enlaces:
 - Listar
 - Insertar
 - Editar
 - Eliminar
- Importar CSV
- Exportar CSV
- Visualizar inmuebles en Google maps (vista pública y privada)
- **Gestión de clientes:** En el que podemos destacar los siguientes casos de uso:
 - Buscar
 - Insertar
 - Eliminar
 - Duplicar
 - Editar:
 - Visualizar en google maps.
 - Gestión inmuebles demandados:
 - Listar
 - Crear
 - Desasociar
 - Gestión demandas asociadas:
 - Listar
 - Crear
 - Editar
 - Gestión propiedades asociadas:
 - Listar

- Asociar
 - Crear
 - Desasociar
 - Gestión de ficha:
 - Generar
 - Editar
 - Eliminar
 - Gestión de ficheros adjuntos:
 - Listar
 - Adjuntar
 - Eliminar
 - Importar CSV
 - Exportar CSV
 - Visualizar inmuebles en Google maps (vista pública y privada)
- **Gestión de demandas:** En el que podemos destacar los siguientes casos de uso:
- Buscar
 - Insertar
 - Eliminar
 - Duplicar
 - Editar:
 - Gestión inmuebles propuestos:
 - Listar
 - Proponer
 - Quitar
 - Establecer datos adicionales
 - Visualizar inmuebles en Google maps (vista privada)
 - Gestión de ficha:
 - Generar
 - Editar

- ◊ Eliminar
- Gestión de ficheros adjuntos:
 - ◊ Listar
 - ◊ Adjuntar
 - ◊ Eliminar
- Importar CSV
- Exportar CSV

4.3. Modelo de Casos de Uso

A partir de los requisitos funcionales descritos anteriormente, se emplearán los casos de uso como técnica para representar las interacciones entre los actores y el sistema bajo estudio. Se expresarán en imperativo los casos de uso (Por ejemplo: Insertar cliente), mientras que la agrupación de casos de uso se expresará en descriptivo (Gestión de clientes).

4.3.1. Actores

En este apartado se describirán los diferentes roles que juegan los usuarios que interactúan con el sistema. El sistema cuenta con tres actores bien diferenciados:

- **Usuario anónimo o visitante web:** El cual interactúa única y exclusivamente con la parte pública del sistema. Principalmente serán demandantes que buscan un determinado inmueble.
- **Usuario registrado:** El cual interactúa única y exclusivamente con el software de gestión (zona privada). Se distinguen los siguientes roles:
 - **Agente inmobiliario:** Su objetivo se centrará en manejar la actividad diaria de la empresa, para ello sólo dispondrá de acceso a los módulos de demandas, inmuebles y clientes. También visualizará las estadísticas principales del sistema en su dashboard.
 - **Administrador:** Su objetivo es configurar y mantener los tipos de datos y las preferencias tanto del funcionamiento de la zona privada como el de la zona pública. Por tanto, interactuará con todos los casos de uso que no aborda el agente inmobiliario.

4.3.2. Diagramas

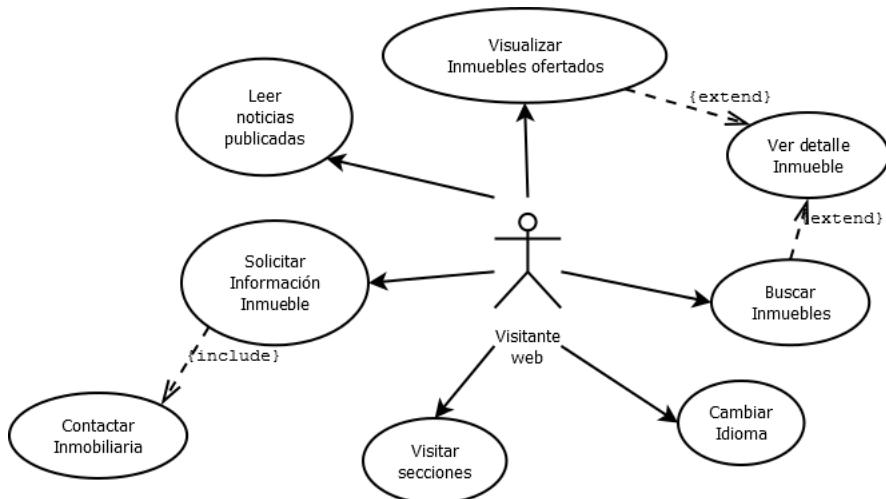


Figura 4.6: Diagrama Casos de Uso: Sitio web (Zona pública)

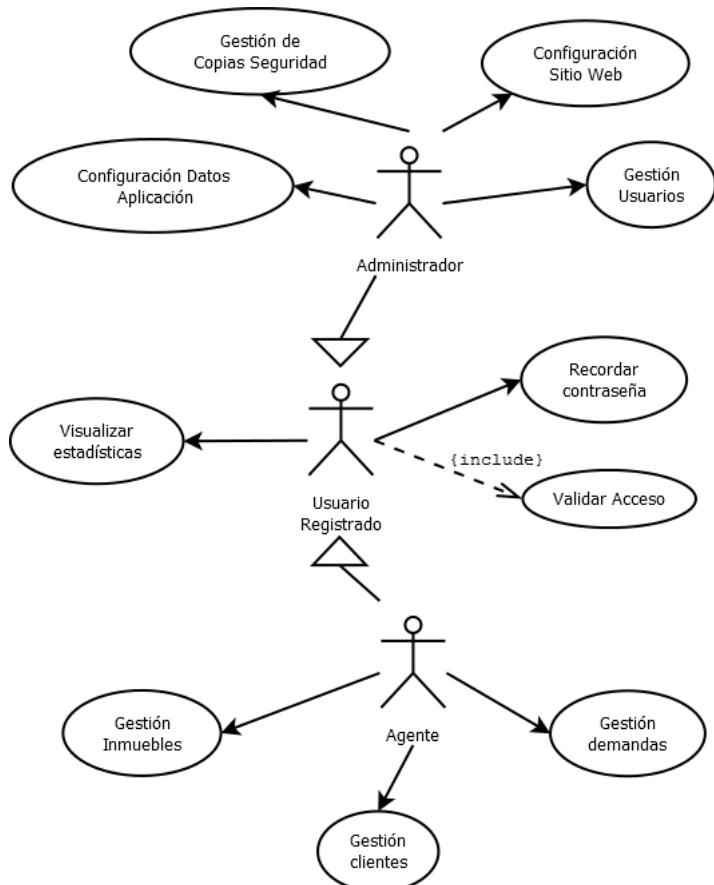


Figura 4.7: Diagrama Casos de Uso: Software de gestión (Zona privada)

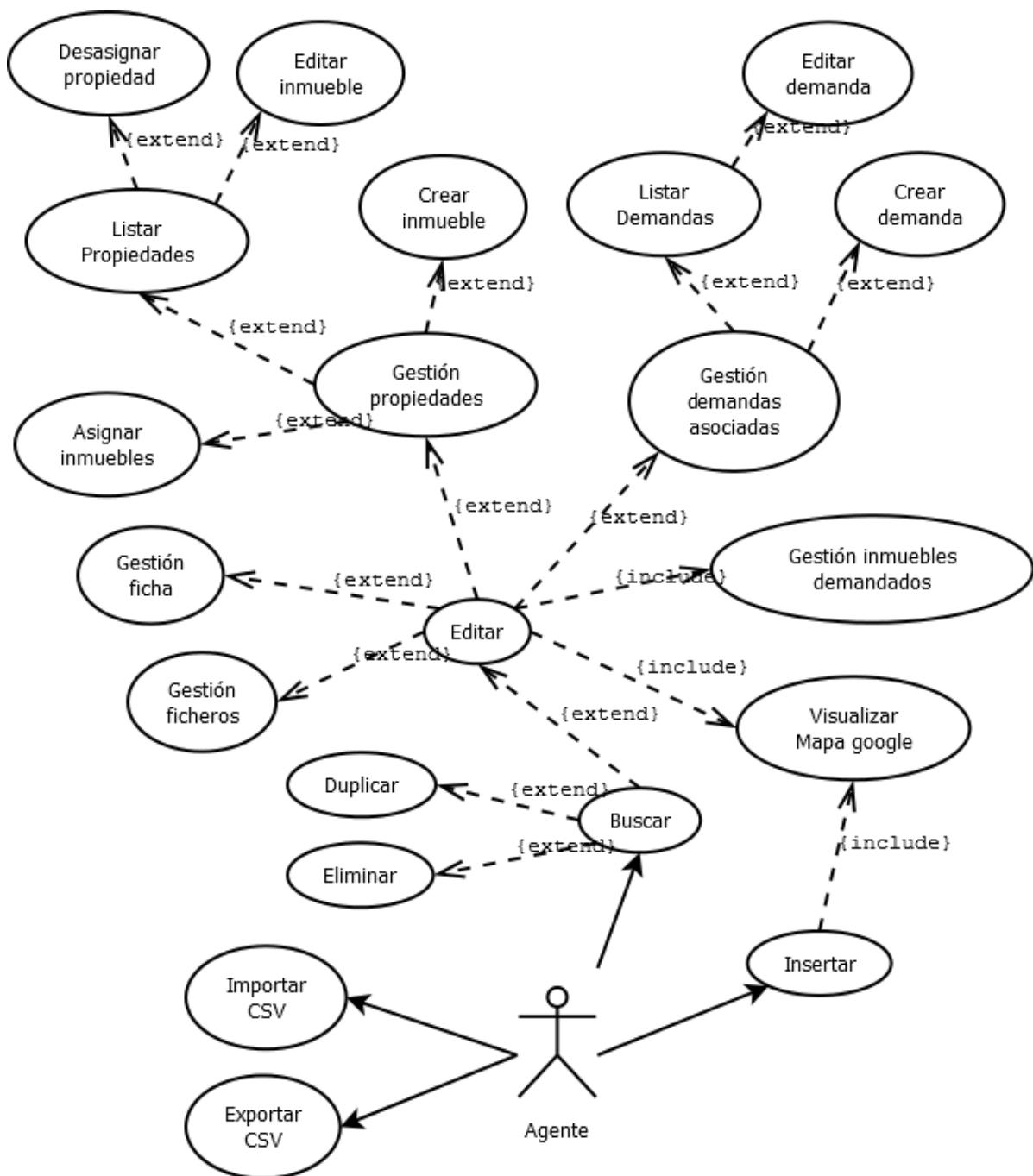


Figura 4.8: Diagrama Casos de Uso: Gestión de clientes

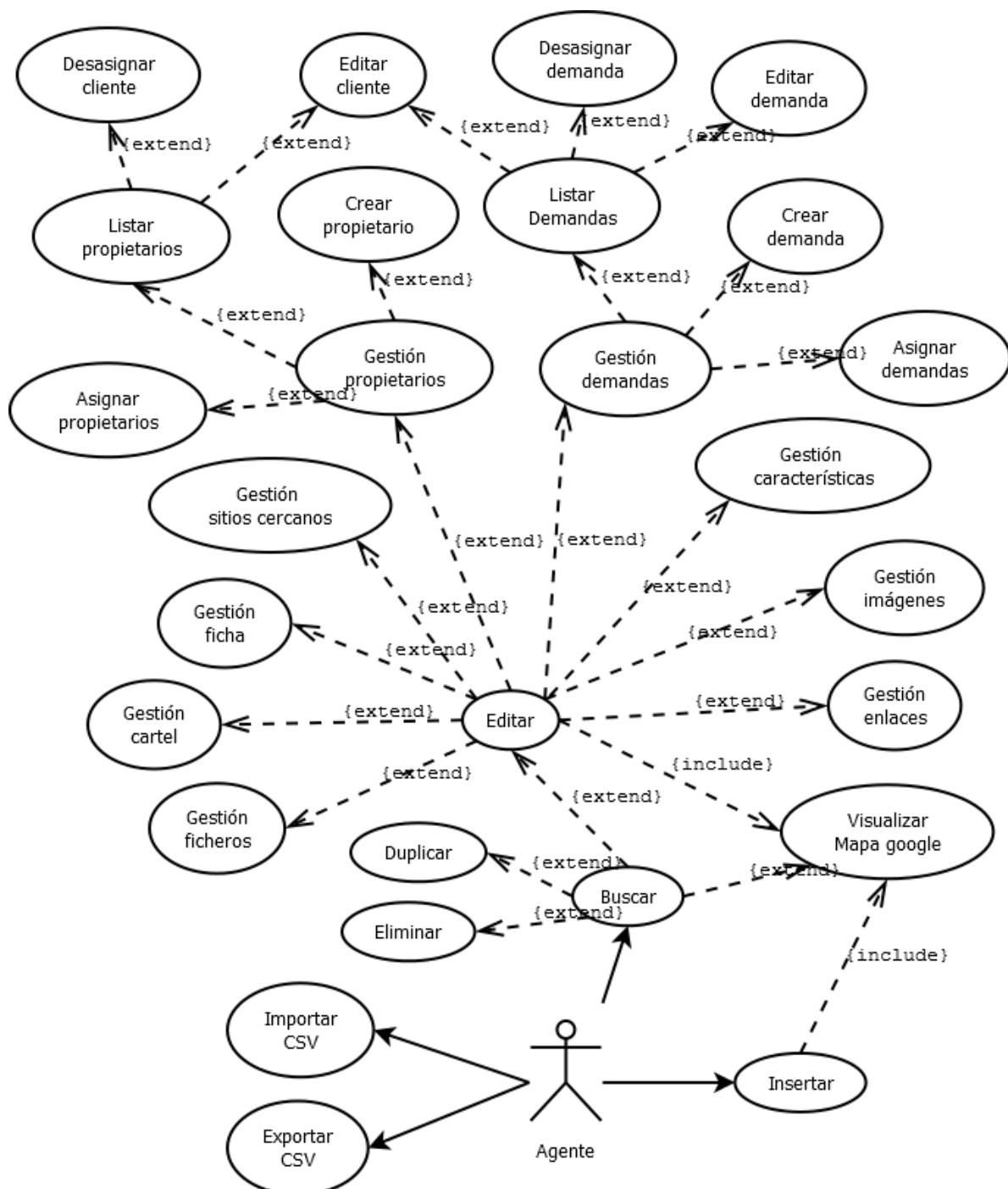


Figura 4.9: Diagrama Casos de Uso: Gestión de inmuebles

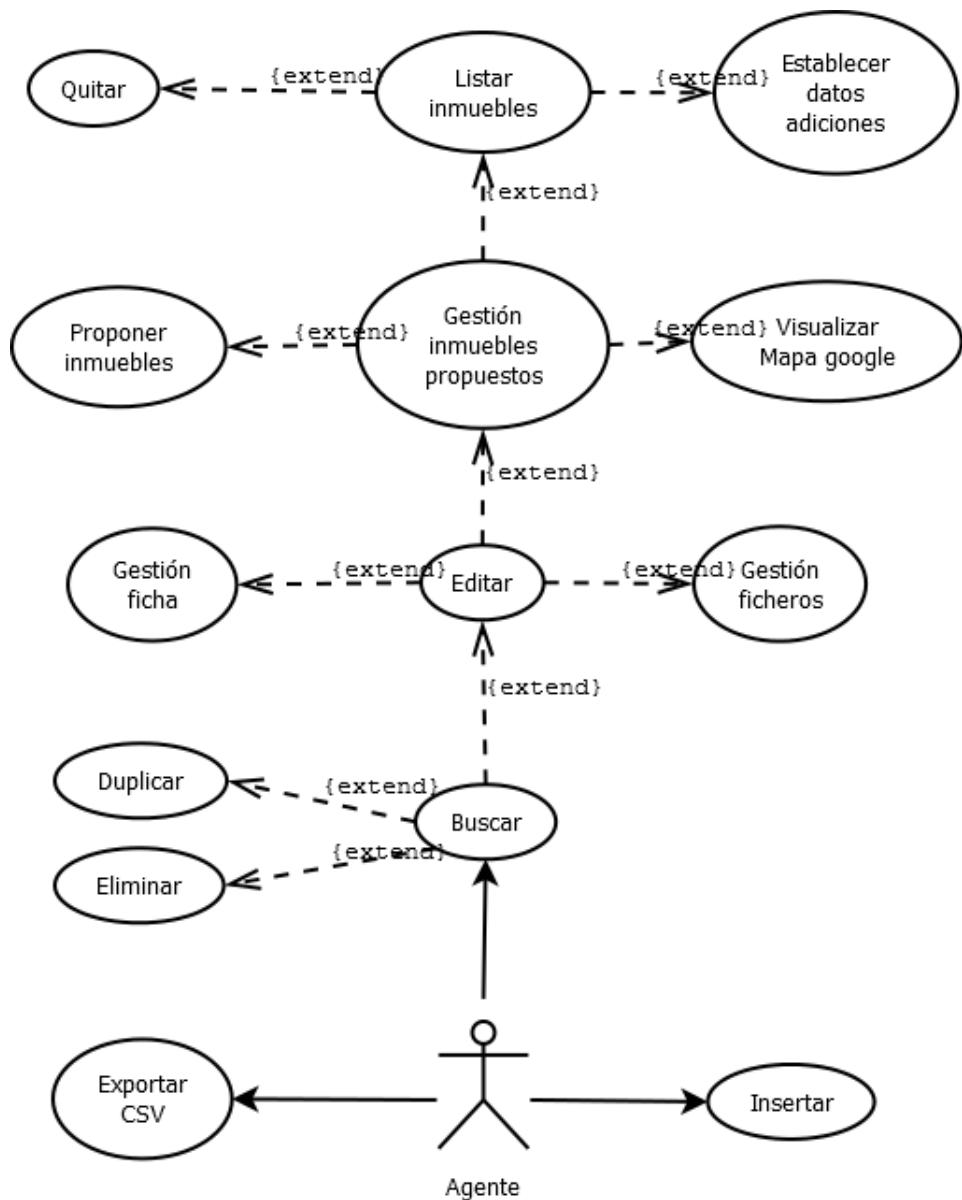


Figura 4.10: Diagrama Casos de Uso: Gestión de demandas

En este último diagrama se puede apreciar mucho mejor el nivel de profundidad que se alcanza al ir navegando por los diferentes casos de uso y que luego puede servir como base para el mapa de navegación web del sistema.

4.3.3. Descripción detallada

Debido a la cantidad de casos de uso del sistema (superá ampliamente los 100), no se pueden describir todos y cada uno de los mismos. Este apartado tendrá como objeto de estudio aquellos casos de uso más relevantes y cuya descripción no haya sido descrita por

los subsistemas externos que lo componen.

Consideraciones generales

Esta parte tiene como objetivo explicar la dinámica de muchos casos de uso que agrupan funcionalidades CRUD. La mayoría de casos de uso titulados como Gestión de X, incluye los casos de uso:

- Listar o Buscar (La búsqueda es simplemente aplicar unos criterios de búsqueda y luego listar el resultado)
- Insertar
- Editar
- Eliminar

Para no repetir la descripción de estos casos de uso, se puede tomar, a modo de ejemplo, la gestión de cliente, eliminando los escenarios secundarios.

Los escenarios de error vendrán dados por las comprobaciones de las restricciones indicadas en el modelo de datos y los requisitos de información. Es decir, por ejemplo, en la gestión de tipos de inmuebles, no se podrá eliminar un tipo de inmueble si está asignado a algún inmueble. Tampoco se podrá registrar un tipo de inmueble que tenga el mismo nombre en el mismo idioma.

Por otro lado, todos los casos de uso que cuenten con validaciones de datos (la mayoría), realizan dos tipos de validaciones:

- Sintácticas: Aquellas que tienen por objetivo comprobar que la validez del dato en sí respecto a su tipo, tamaño, etc. Estas no serán objeto de estudio de los casos de uso por norma general salvo que existe algún tipo de interés en especial.
- Semánticas: Aquellas que se encargan de comprobar las reglas del negocio y restricciones del modelo de datos conceptual. Estas sí se especificarán en los casos de uso.

Gestión clientes

Insertar cliente

Descripción: Registrar la información introducida del cliente en el sistema.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: Ninguna.

Postcondiciones: El sistema registra los datos introducidos del cliente.

Escenario Principal:

1. El agente inmobiliario desea insertar un cliente desde el buscador.
2. El sistema muestra la interfaz 'Insertar Cliente'.
3. El agente inmobiliario introduce todos los datos del cliente.
4. Al terminar de introducir los datos de la dirección el sistema ejecuta el caso de uso 'Visualizar Dirección en Mapa de Google Map'.
5. El sistema valida los datos introducidos.
6. El sistema almacena los datos introducidos del cliente.
7. Muestra un mensaje de confirmación de la operación realizada.

Escenarios Error:

5.a) Existe un cliente con el mismo e-mail o identificación registrado en el sistema:

1. El sistema muestra un mensaje indicando el error.

Escenario Alternativo: Crear propietario

Descripción: Describe el modo de actuar el sistema cuando se registra un propietario de un determinado inmueble.

Precondiciones: Se ha ejecutado el caso de uso 'Editar inmueble'.

Postcondiciones: El sistema registra los datos introducidos del cliente y asocia el inmueble seleccionado como propiedad.

1. El agente inmobiliario desea insertar un cliente desde la interfaz de 'Editar Inmueble'.
2. El sistema muestra la interfaz 'Insertar Cliente' con los datos del inmueble seleccionado.
3. Ejecuta los pasos 3-6 del escenario principal.
4. Asocia el inmueble como propiedad del cliente introducido.

5. Muestra un mensaje de confirmación de la operación realizada.

Editar cliente

Descripción: Modifica la información introducida del cliente en el sistema.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El cliente seleccionado ha sido registrado a través del caso de uso 'Insertar Cliente'.

Postcondiciones: El sistema modifica los datos introducidos del cliente.

Escenario Principal:

1. El agente inmobiliario desea editar un cliente de cualquier listado del sistema.
2. El sistema lee los datos del cliente y muestra la interfaz 'Editar Cliente'.
3. El agente inmobiliario modifica los datos que deseé del cliente.
4. Al terminar de introducir los datos de la dirección el sistema ejecuta el caso de uso 'Visualizar Dirección en Mapa de Google Map'.
5. El sistema valida los datos introducidos.
6. El sistema modifica los datos introducidos del cliente.
7. Muestra un mensaje de confirmación de la operación realizada.

Escenarios Error:

5.a) Existe un cliente con el mismo e-mail o identificación registrado en el sistema:

1. El sistema muestra un mensaje indicando el error.

Eliminar cliente

Descripción: Elimina un cliente del sistema.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El cliente seleccionado ha sido registrado a través del caso de uso 'Insertar Cliente'.

Postcondiciones: El sistema elimina al cliente del sistema así como toda su información relacionada.

Escenario Principal:

1. El agente inmobiliario desea eliminar un cliente desde el buscador.
2. El sistema elimina al cliente del sistema.
3. Muestra un mensaje de confirmación de la operación realizada.

Buscar cliente

Descripción: Buscar clientes a través de determinados filtros de búsqueda .

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: Ninguna.

Postcondiciones: Muestra los clientes que coincidan con los criterios de búsqueda introducidos.

Escenario Principal:

1. El agente inmobiliario desea realizar una búsqueda de clientes.
2. El sistema muestra la interfaz 'Buscador de Clientes'.
3. El agente inmobiliario introduce los criterios de búsqueda.
4. El sistema busca los clientes que coincidan con los criterios de búsqueda introducidos.
5. Muestra un listado de los clientes encontrados.

Escenarios Alternativo:

5.a) No existen clientes con los criterios de búsqueda introducidos:

1. El sistema muestra un mensaje informando de la situación.

Duplicar cliente

Descripción: Duplicar un cliente del sistema.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El cliente seleccionado ha sido registrado a través del caso de uso 'Insertar Cliente'.

Postcondiciones: El sistema crea una copia del cliente seleccionado generándole un nuevo e-mail e identificación pero con la fecha de alta actual.

Escenario Principal:

1. El agente inmobiliario desea duplicar un cliente desde el buscador.
2. El sistema crea una copia del cliente seleccionado generándole un nuevo e-mail e identificación pero con la fecha de alta actual.
3. Muestra un mensaje de confirmación de la operación realizada.

Importar CSV cliente

Descripción: Importar los clientes especificados en el CSV introducido.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: Ninguna.

Postcondiciones: Registra a los clientes en el sistema.

Escenario Principal:

1. El agente inmobiliario desea importar un CSV desde el buscador.
2. El sistema comprueba el tipo y tamaño del fichero y sube el fichero al servidor.
3. El sistema comprueba que cada registro del CSV es correcto.
4. El sistema muestra un resumen de los registros que se van a importar y los errores encontrados.
5. El agente inmobiliario confirma la importación de los registros válidos.
6. El sistema registra los clientes que son válidos como si ejecutara el caso de uso 'insertar cliente'.
7. El sistema muestra un resumen de los registros importados y los no importados.

Escenarios Error:

- 2.a) El tamaño del fichero excede el límite permitido:

1. El sistema muestra un mensaje indicando el error.
- 2.b) El tipo del fichero no es CSV:
1. El sistema muestra un mensaje indicando el error.
- 3) Para cada registro del CSV se muestran los siguientes errores en el resumen:
- 3.a) El email está repetido en otro registro del CSV.
 - 3.b) La identificación del cliente está repetida en otro registro del CSV.
 - 3.c) El estado introducido no existe.
 - 3.d) El país de residencia introducido no existe.
 - 3.e) La provincia introducida no existe.
 - 3.f) El municipio introducido no existe.
 - 3.g) Cualquier otro error sintáctico o semántico especificado en el caso de uso 'Insertar Cliente'.

Exportar CSV clientes

Descripción: Exportar los CSV de los clientes filtrados en el buscador.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El cliente ha ejecutado el caso de uso 'Buscar Cliente'.

Postcondiciones: El sistema devuelve un CSV descargable con los datos de los clientes procesados en la última búsqueda realizada.

Escenario Principal:

1. El agente inmobiliario desea exportar una búsqueda realizada.
2. El sistema lee los datos de la última búsqueda realizada.
3. Muestra un CSV para descargar con los datos de los clientes.

Listar ficheros adjuntos del cliente

Descripción: Mostrar los ficheros adjuntos del cliente seleccionado.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El cliente seleccionado existe en el sistema.

Postcondiciones: Ninguna.

Escenario Principal:

1. El agente inmobiliario desea listar los ficheros adjuntos del cliente.
2. El sistema muestra un listado con los ficheros adjuntos del cliente.

Escenarios Alternativo:

2.a) El cliente seleccionado no tiene ficheros adjuntos:

1. El sistema muestra un mensaje informando de la situación.

Adjuntar fichero a cliente

Descripción: Subir el fichero al sistema y registrar el nombre introducido por el agente.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El cliente seleccionado existe en el sistema.

Postcondiciones: El sistema registra el nombre del fichero adjunto y lo sube físicamente al servidor.

Escenario Principal:

1. El agente inmobiliario desea adjuntar un fichero a un cliente.
2. El sistema muestra la interfaz 'Adjuntar fichero de Cliente'.
3. El agente inmobiliario selecciona el fichero a adjuntar e introduce su nombre y selecciona el tipo de fichero.
4. El sistema valida los datos introducidos.
5. El sistema registra el nombre del fichero adjunto y lo sube físicamente al servidor.
6. Muestra un mensaje de confirmación de la operación realizada.

Escenarios Error:

4.a) El tamaño del fichero excede el límite permitido:

1. El sistema muestra un mensaje indicando el error.

4.b) Existe un fichero adjunto para el cliente seleccionado con el mismo nombre registrado en el sistema:

1. El sistema muestra un mensaje indicando el error.

Eliminar fichero adjunto de cliente

Descripción: Eliminar un fichero adjunto del cliente.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El fichero seleccionado ha sido registrado a través del caso de uso 'Adjuntar fichero de Cliente'.

Postcondiciones: El sistema elimina los datos del fichero registrados así como el fichero físico.

Escenario Principal:

1. El agente inmobiliario desea eliminar un fichero adjunto del cliente.
2. El sistema elimina el fichero adjunto físicamente del sistema y los datos registrados.
3. Muestra un mensaje de confirmación de la operación realizada.

Generar ficha de cliente

Descripción: Generar la ficha de un cliente en formato HTML.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El cliente seleccionado existe en el sistema.

Postcondiciones: El sistema genera la ficha.

Escenario Principal:

1. El agente inmobiliario desea generar una ficha de un cliente.
2. El sistema muestra la interfaz 'Generar ficha de cliente'.
3. El agente inmobiliario selecciona la plantilla a aplicar.
4. El sistema aplica los datos del cliente a la plantilla seleccionada y la almacena en el sistema.
5. Se ejecuta el caso de uso 'Editar ficha de cliente' con un mensaje de confirmación de la operación realizada.

Escenarios Error:

3.a) No ha seleccionado plantilla:

1. El sistema muestra un mensaje indicando el error.

Editar ficha de cliente

Descripción: Modifica la información almacenada de la ficha del cliente en el sistema.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: La ficha seleccionada ha sido registrada a través del caso de uso 'Generar ficha de Cliente'.

Postcondiciones: El sistema modifica los datos almacenados de la ficha del cliente.

Escenario Principal:

1. El agente inmobiliario desea editar la ficha del cliente.
2. El sistema lee los datos de la ficha del cliente y muestra la interfaz 'Editar ficha de Cliente'.
3. El agente inmobiliario modifica el contenido de la ficha del cliente.
4. El sistema modifica la ficha del cliente con el contenido introducido.
5. Muestra un mensaje de confirmación de la operación realizada.

Eliminar ficha de cliente

Descripción: Eliminar una ficha del cliente.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: La ficha seleccionada ha sido generada a través del caso de uso 'Generar ficha de Cliente'.

Postcondiciones: El sistema elimina los datos almacenados de la ficha del cliente.

Escenario Principal:

1. El agente inmobiliario desea eliminar la ficha del cliente.
2. El sistema elimina la ficha del cliente del sistema.

3. Muestra un mensaje de confirmación de la operación realizada.

Listar propiedades del cliente

Descripción: Mostrar las propiedades del cliente seleccionado.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El cliente seleccionado existe en el sistema.

Postcondiciones: Ninguna.

Escenario Principal:

1. El agente inmobiliario desea listar las propiedades del cliente.
2. El sistema muestra un listado con las propiedades del cliente.

Escenarios Alternativo:

- 2.a) El cliente seleccionado no tiene propiedades asociadas:

1. El sistema muestra un mensaje informando de la situación.

Asignar inmuebles a cliente

Descripción: Designar varios inmuebles como propiedades del cliente seleccionado.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El cliente seleccionado existe en el sistema. Los inmuebles a asociar no son propiedades del cliente.

Postcondiciones: Los inmuebles seleccionados pasan a ser propiedades del cliente.

Escenario Principal:

1. El agente inmobiliario desea asociar inmuebles a un cliente.
2. El sistema muestra la interfaz 'Asociar inmuebles a cliente' con un listado de inmuebles que no son de su propiedad.
3. El agente inmobiliario selecciona los inmuebles que desea asociar al cliente.
4. El sistema vincula los inmuebles seleccionados como propiedades del cliente.
5. Muestra un mensaje de confirmación de la operación realizada.

Escenarios Error:

3.a) No ha seleccionado ningún inmueble:

1. El sistema muestra un mensaje indicando el error.

Desasignar propiedad

Descripción: Quitar la propiedad del inmueble del cliente seleccionado.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El inmueble ha sido asignado a través del caso de uso 'Asignar inmuebles a cliente'.

Postcondiciones: El inmueble seleccionado ha dejado de ser propiedad del cliente.

Escenario Principal:

1. El agente inmobiliario desea quitar la propiedad del cliente.
2. El sistema elimina la asociación entre el inmueble y el cliente del sistema.
3. Muestra un mensaje de confirmación de la operación realizada.

Gestión inmuebles

Se describen sólo aquellos casos de uso cuya funcionalidad es diferente a los explicados anteriormente. Muchos de los casos detallados para el cliente son aplicables para el inmueble pero desde el punto de vista de este, es decir, para importar un CSV de inmuebles hay que aplicar las mismas validaciones pero las especificadas para el inmueble.

Generar ficha de inmueble

Descripción: Generar la ficha de un inmueble en formato HTML.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El inmueble seleccionado existe en el sistema.

Postcondiciones: El sistema genera la ficha.

Escenario Principal:

1. El agente inmobiliario desea generar una ficha de un inmueble.

2. El sistema muestra la interfaz 'Generar ficha de inmueble'.
3. El agente inmobiliario selecciona la plantilla a aplicar y el idioma.
4. El sistema aplica los datos del inmueble a la plantilla seleccionada y la almacena en el sistema.
5. Se ejecuta el caso de uso 'Editar ficha de inmueble' con un mensaje de confirmación de la operación realizada.

Escenarios Error:

3.a) No ha seleccionado plantilla o idioma:

1. El sistema muestra un mensaje indicando el error.

Editar ficha de inmueble

Descripción: Modifica la información almacenada de la ficha del inmueble en el sistema.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: La ficha seleccionada ha sido registrada a través del caso de uso 'Generar ficha de inmueble'.

Postcondiciones: El sistema modifica los datos almacenados de la ficha del inmueble.

Escenario Principal:

1. El agente inmobiliario desea editar la ficha del inmueble.
2. El sistema lee los datos de la ficha del inmueble y muestra la interfaz 'Editar ficha de inmueble'.
3. El agente inmobiliario modifica el contenido de la ficha del inmueble.
4. El sistema modifica la ficha del inmueble con el contenido introducido.
5. Muestra un mensaje de confirmación de la operación realizada.

Eliminar ficha de inmueble

Descripción: Eliminar una ficha del inmueble.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: La ficha seleccionada ha sido generada a través del caso de uso 'Generar ficha de inmueble'.

Postcondiciones: El sistema elimina los datos almacenados de la ficha del inmueble.

Escenario Principal:

1. El agente inmobiliario desea eliminar la ficha del inmueble.
2. El sistema elimina la ficha del inmueble del sistema.
3. Muestra un mensaje de confirmación de la operación realizada.

Gestión del cartel del inmueble

Descripción: Los casos de uso son idénticos a los redactados para la generación de la ficha del inmueble excepto que para el cartel del inmueble se puede generar un código QR siempre y cuando se haya realizado una publicación previa del inmueble ya que el cartel necesita de la URL-SEO para encriptarla en el QR.

Listar imágenes subidas del inmueble

Descripción: Mostrar las imágenes subidas al inmueble seleccionado.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El inmueble seleccionado existe en el sistema.

Postcondiciones: Ninguna.

Escenario Principal:

1. El agente inmobiliario desea listar las imágenes subidas al inmueble.
2. El sistema muestra un listado con las imágenes del inmueble.

Escenarios Alternativo:

2.a) El inmueble seleccionado no tiene imágenes asociadas:

1. El sistema muestra un mensaje informando de la situación.

Subir imágenes a inmueble

Descripción: Subir imágenes al sistema y asociarlas al inmueble seleccionado por el agente.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: El inmueble seleccionado existe en el sistema.

Postcondiciones: El sistema registra el nombre de cada imagen y la sube físicamente al servidor.

Escenario Principal:

1. El agente inmobiliario desea subir imágenes a un inmueble.
2. El sistema muestra la interfaz 'Subir imágenes a inmueble'.
3. El agente inmobiliario selecciona las imágenes que desea subir (a través del explorador de ficheros o arrastrando una carpeta a la interfaz).
4. El sistema comprueba que cada imagen puede subirse individualmente.
5. El sistema registra el nombre de cada imagen y la sube físicamente al servidor.
6. Muestra un listado de todas las imágenes subidas.

Escenarios Error:

4.a) El fichero adjuntado no es una imagen:

1. Muestra el error en la interfaz del paso 6.

4.b) El tamaño de la imagen excede el límite permitido:

1. Muestra el error en la interfaz del paso 6.

Escenarios Error:

5.a) Existe un fichero adjunto para el inmueble seleccionado con el mismo nombre registrado en el sistema:

1. El sistema registra el nombre de la imagen pero le añade un número contador del número de copias que hay al nombre y la sube físicamente al servidor.

Eliminar imagen de inmueble

Descripción: Eliminar una imagen del inmueble.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: La imagen ha sido subida a través del caso de uso 'Subir imágenes a inmueble'.

Postcondiciones: El sistema elimina los datos de la imagen así como el fichero físico.

Escenario Principal:

1. El agente inmobiliario desea eliminar una imagen del inmueble.
2. El sistema elimina la imagen físicamente del sistema y los datos registrados.
3. Muestra un mensaje de confirmación de la operación realizada.

Publicar imagen de inmueble

Descripción: Publicar una imagen del inmueble.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: La imagen ha sido subida a través del caso de uso 'Subir imágenes a inmueble'. La imagen seleccionada no está publicada.

Postcondiciones: El sistema habilita la imagen como pública y puede ser visualizada desde el caso de uso 'Ver detalle de inmueble' de la zona pública.

Escenario Principal:

1. El agente inmobiliario selecciona una imagen para hacerla pública del inmueble.
2. El sistema habilita la imagen como pública.
3. Muestra un mensaje de confirmación de la operación realizada.

Establecer imagen de inmueble como portada

Descripción: Hacer que una imagen del inmueble sea su portada.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: La imagen ha sido subida a través del caso de uso 'Subir imágenes a inmueble'. La imagen seleccionada está publicada y no es portada.

Postcondiciones: El sistema habilita la imagen como portada y será la imagen principal que se visualice del inmueble tanto en la zona privada como en la pública.

Escenario Principal:

1. El agente inmobiliario selecciona una imagen para establecerla como portada del inmueble.
2. El sistema habilita la imagen como portada.
3. Muestra un mensaje de confirmación de la operación realizada.

Descargar imagen de inmueble

Descripción: Descarga la imagen del inmueble físicamente al dispositivo.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: La imagen ha sido subida a través del caso de uso 'Subir imágenes a inmueble'.

Postcondiciones: La imagen se descarga físicamente al dispositivo.

Escenario Principal:

1. El agente inmobiliario selecciona una imagen del inmueble para descargar.
2. El sistema muestra una ventana indicando la descarga de la imagen.
3. El agente inmobiliario acepta y se descarga la imagen a su dispositivo.

Gestión demandas

Proponer inmuebles

Descripción: Asignar varios inmuebles como propiedades en las que el cliente de la demanda podría estar interesado.

Actores:

- Principal: Agente Inmobiliario.

Precondiciones: La demanda seleccionada existe en el sistema. Los inmuebles a asociar no son propiedades del cliente.

Postcondiciones: Los inmuebles seleccionados pasan a estar propuestos para la demanda.

Escenario Principal:

1. El agente inmobiliario desea asociar inmuebles a una demanda.
2. El sistema muestra la interfaz 'Proponer inmuebles'.

3. El agente inmobiliario selecciona los inmuebles que desea proponer para la demanda.
4. El sistema vincula los inmuebles seleccionados como propiedades de la demanda asignadas por el agente. Se establece como estado por defecto 'Propuesto para visita'.
5. Muestra un mensaje de confirmación de la operación realizada.

Escenarios Error:

3.a) No ha seleccionado ningún inmueble:

1. El sistema muestra un mensaje indicando el error.

Generales

A continuación se describen algunos casos de uso común en varias funcionalidades.

Ver detalle de inmueble

Descripción: Visualizar toda la información publicada por el agente del inmueble.

Actores:

- Principal: Visitante web.

Precondiciones: El inmueble seleccionado existe en el sistema y está publicado.

Postcondiciones: Ninguna.

Escenario Principal:

1. El visitante web desea visualizar el detalle de un inmueble.
2. El sistema muestra la interfaz 'Detalle de un inmueble' en el idioma que está activo en ese momento.

Ver detalle de inmueble

Descripción: Visualizar toda la información publicada por el agente del inmueble.

Actores:

- Principal: Visitante web.

Precondiciones: El inmueble seleccionado existe en el sistema y está publicado.

Postcondiciones: Ninguna.

Escenario Principal:

1. El visitante web desea visualizar el detalle de un inmueble.

2. El sistema muestra la interfaz 'Detalle de un inmueble' en el idioma que está activo en ese momento.

Visualizar mapa de Google

Descripción: Mostrar un mapa de google con la ubicación de una dirección o un conjunto de direcciones.

Actores:

- Principal: Visitante Web o Agente Inmobiliario.

Precondiciones: El API-Key de Google Maps ha sido registrado en la configuración del sistema. Se debe dar acceso al dispositivo para que lea la posición actual.

Postcondiciones: Ninguna.

Escenario Principal: Visualizar una dirección

1. El usuario desea visualizar una única dirección.
2. El sistema muestra la dirección con una marca en un Mapa de google centrado en ella.

Escenarios Error:

2.a) La dirección no es encontrada por la API de Google:

1. El sistema muestra un mapa centrado en el océano.

2.b) No es posible conectar con la API de Google:

1. El sistema muestra un mensaje indicando el error en el propio mapa.

Escenario Alternativo: Visualizar inmuebles propuestos

Precondiciones: Hay algún inmueble propuesto.

1. El usuario desea visualizar los inmuebles propuestos por un agente en una demanda.
2. El sistema muestra cada inmueble con una marca diferente en un Mapa de Google.
El mapa estará centrado en la provincia de los criterios de búsqueda de la demanda si se seleccionó una, en caso contrario en la posición física actual del usuario.

Escenario Alternativo: Visualizar inmuebles propuestos con direcciones coincidentes

Precondiciones: Hay algún inmueble propuesto.

2.b) Alguna de las direcciones está ubicada en la misma posición:

1. El sistema muestra en el punto coincidente un listado de los inmuebles con las mismas posiciones.

Escenario Alternativo: Visualizar inmuebles del buscador

Precondiciones: Se ha encontrado algún inmueble al realizar la búsqueda.

1. El usuario desea visualizar los inmuebles resultado de realizar una búsqueda.
2. El sistema muestra cada inmueble con una marca diferente en un Mapa de Google.
El mapa estará centrado en la provincia de los criterios de búsqueda del buscador si se seleccionó una, en caso contrario en la posición física actual del usuario.

Escenario Alternativo: Visualizar inmuebles del buscador con direcciones coincidentes

2.b) Alguna de las direcciones está ubicada en la misma posición:

1. El sistema muestra en el punto coincidente un listado de los inmuebles con las mismas posiciones.

El resto de las especificaciones y escenarios posibles de los casos de uso se describen en las matrices de prueba o simplemente son lo suficientemente parecidas a los descritos anteriormente como para poner una descripción detallada.

4.4. Modelo de Comportamiento

A partir de los casos de uso anteriores, se crea el modelo de comportamiento. Para ello, se realizarán los diagramas de secuencia del sistema, donde se identificarán las operaciones o servicios del sistema. Luego, se detallará el contrato de las operaciones identificadas.

La estrategia que se seguirá para identificar las operaciones del sistema es identificar los eventos que provocan respuestas en el sistema y a raíz de ellos hacer el contrato de operaciones de cada uno de ellos. El objetivo de este estudio es que los diagramas sirvan de enlace entre el análisis y el diseño, para que en el diseño detallado sólo se amplíe y especifique cómo van a interactuar los artefactos de software para cumplir el contrato especificado. En el hilo de lo descrito en el anterior apartado, sólo describiremos algunos representativos.

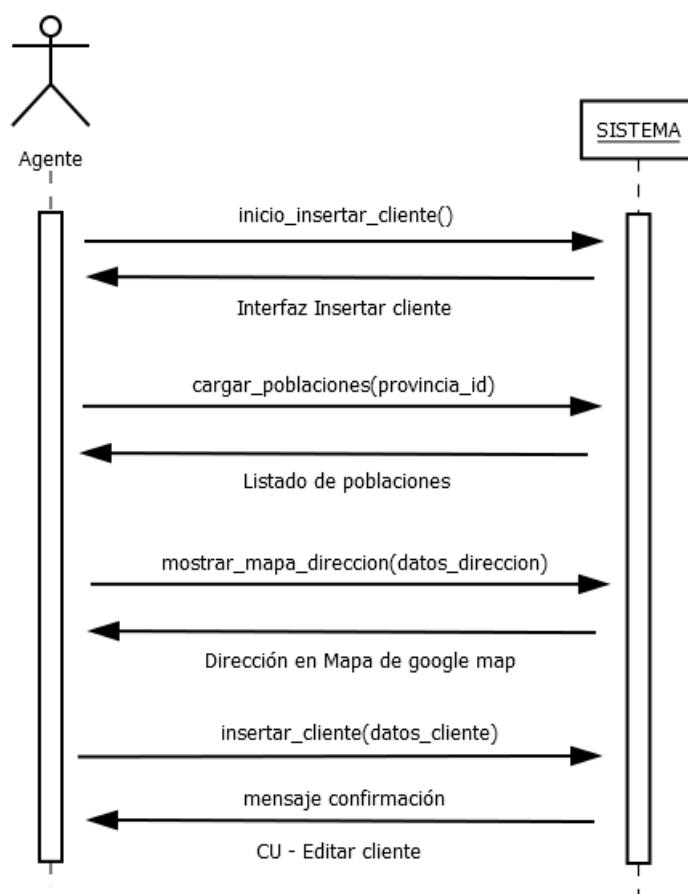


Figura 4.11: Diagrama secuencia: Insertar cliente - Escenario normal

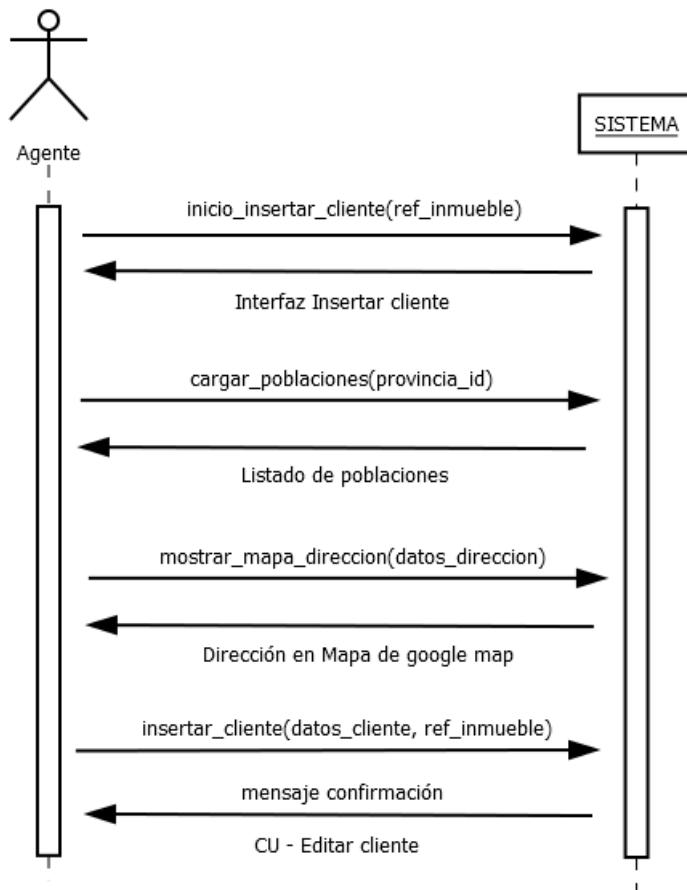


Figura 4.12: Diagrama secuencia: Insertar cliente - Escenario alternativo

Operación: inicio_insertar_cliente(ref_inmueble)

- Responsabilidades: Mostrar la interfaz 'Insertar cliente'.
- Referencias cruzadas: CU - 'Insertar cliente'.
- Precondiciones: La referencia del inmueble, si está especificada, debe existir en el sistema.
- Postcondiciones: Carga la interfaz de 'Insertar cliente'. Si la referencia del inmueble está especificada, muestra también los datos del inmueble que coincidan con la referencia introducida.

Operación: insertar_cliente(datos_cliente,ref_inmueble)

- Responsabilidades: Registrar los datos del cliente en el sistema. Si está especificada la referencia del inmueble, indicar que el inmueble con esa referencia es propiedad del cliente introducido.

- Referencias cruzadas: CU - 'Insertar cliente', 'Duplicar cliente' , 'Importar CSV de clientes'.
- Precondiciones: La referencia del inmueble, si está especificada, debe existir en el sistema.
- Postcondiciones: Los datos introducidos del cliente quedan registrados en el sistema. Si está especificada la referencia del inmueble, crea una asociación entre el inmueble que tenga esa referencia y el cliente.

Aunque las dos siguientes operaciones se detectan por primera vez en este caso de uso, tendrán muchos más usos en el sistema.

Operación: cargar_poblaciones(provincia_id)

- Responsabilidades: Mostrar los municipios de la provincia que haya seleccionado el usuario.
- Referencias cruzadas: CU - 'Insertar cliente', 'Editar cliente', 'Buscador de clientes', 'Insertar inmueble', 'Editar inmueble', 'Buscador de inmuebles', 'Insertar demanda', 'Editar demanda', 'Buscador de demandas', 'Visualizar estadísticas'.
- Precondiciones: El usuario ha seleccionado una provincia.
- Postcondiciones: Los municipios pertenecientes a la provincia indicada quedan cargados en el formulario.

Operación: mostrar_mapa_direccion(datos_direccion)

- Responsabilidades: Mostrar la dirección en un mapa de Google Maps.
- Referencias cruzadas: CU - 'Insertar cliente', 'Editar cliente', 'Insertar inmueble', 'Editar inmueble', 'Ver detalle de inmueble'.
- Precondiciones: El introducido los datos de una dirección.
- Postcondiciones: Se carga un mapa de Google Maps con una marca en el lugar exacto donde se encuentra la dirección indicada.

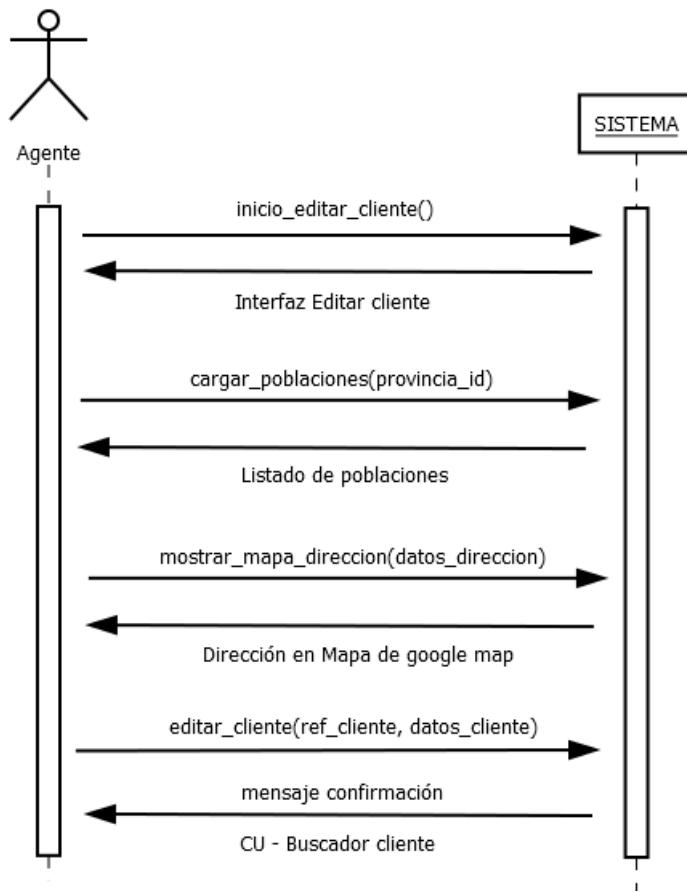


Figura 4.13: Diagrama secuencia: Editar cliente

Operación: inicio_editar_cliente(ref_cliente))

- Responsabilidades: Mostrar la interfaz 'Editar cliente'.
- Referencias cruzadas: CU - 'Editar cliente'.
- Precondiciones: La referencia del cliente debe existir en el sistema.
- Postcondiciones: Carga la interfaz de 'Editar cliente' con los datos del inmueble que coincidan con la referencia introducida.

Operación: editar_cliente(ref_cliente,datos_cliente)

- Responsabilidades: Modificar los datos del cliente en el sistema.
- Referencias cruzadas: CU - 'Editar cliente'.
- Precondiciones: La referencia del cliente debe existir en el sistema.

- Postcondiciones: El cliente con la referencia indicada queda modificado con los datos introducidos.

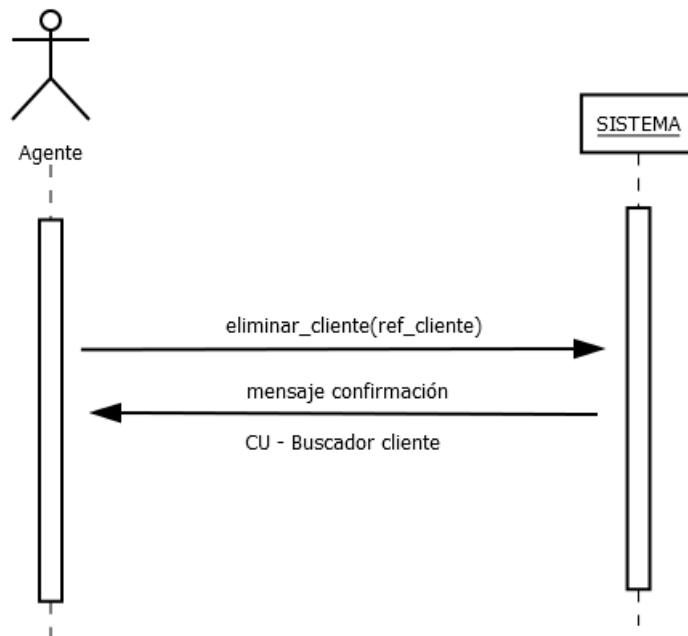


Figura 4.14: Diagrama secuencia: Eliminar cliente

Operación: eliminar_cliente(ref_cliente)

- Responsabilidades: Elimina los datos del cliente en el sistema.
- Referencias cruzadas: CU - 'Eliminar cliente'.
- Precondiciones: La referencia del cliente debe existir en el sistema.
- Postcondiciones: El cliente que coincide con la referencia indicada queda eliminado del sistema.

Operación: inicio_buscar_clientes()

- Responsabilidades: Mostrar la interfaz 'Buscador de clientes'.
- Referencias cruzadas: CU - 'Buscar clientes'.
- Precondiciones: Ninguna.
- Postcondiciones: Carga la interfaz de 'Buscador de clientes'. Si se realizó una búsqueda con anterioridad se realiza la operación de buscar_clientes() antes de cargar la interfaz.

Operación: buscar_clientes(filtros_busqueda)

- Responsabilidades: Buscar a clientes en el sistema.
- Referencias cruzadas: CU - 'Buscar clientes' y OP - inicio_buscar_clientes().
- Precondiciones: Ninguna.
- Postcondiciones: Los clientes que coincidan con los filtros de búsqueda introducidos son cargados en la interfaz de 'Buscador de clientes'.

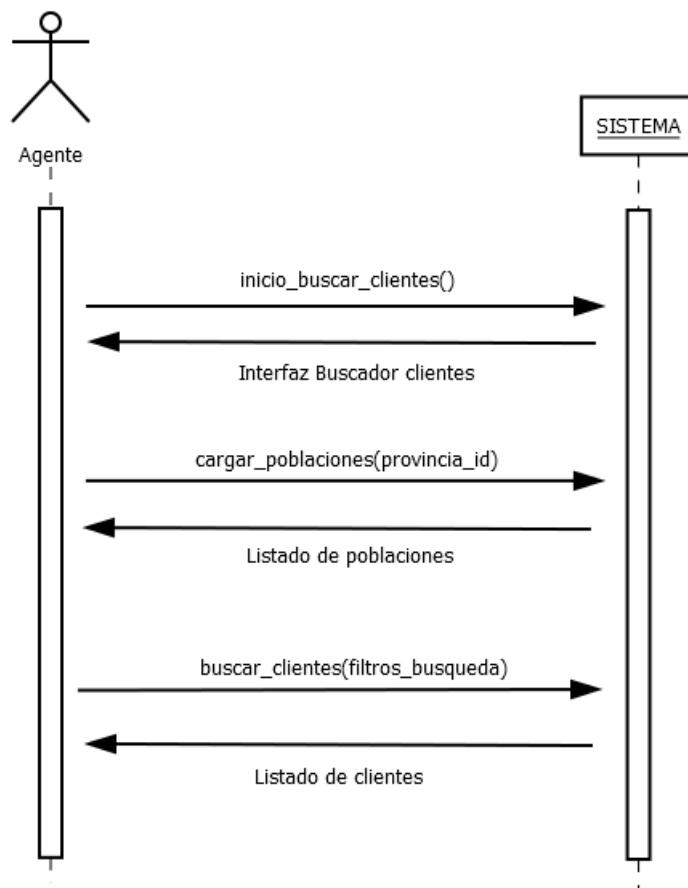


Figura 4.15: Diagrama secuencia: Buscar cliente

4.5. Modelo de Interfaz de Usuario

Tal como se ha especificado en la sección de requisitos, el sistema debe ser claro, preciso e intuitivo, por lo que se propone el siguiente mockup como plantilla de diseño de la interfaz privada.

Este mockup ilustra la estructura de la interfaz de usuario para la gestión de inmuebles. Se divide en secciones principales:

- Barra superior:** Muestra "OPENRS" en la parte izquierda y "BIENVENIDO USUARIO (PERFIL | SALIR)" en la parte derecha.
- Menú lateral izquierdo:** Una lista vertical con las siguientes opciones: INICIO, ● INMUEBLES, CLIENTES, DEMANDAS, USUARIOS, COPIAS, CONFIGURACIÓN → y ZONA PÚBLICA →.
- Barra de navegación central:** Una barra horizontal con enlaces: > DATOS INMUEBLE, IMAGENES, ENLACES, ADJUNTOS, FICHA y CARTEL. El primer ítem, "DATOS INMUEBLE", es el activo y muestra el sub-título "DATOS INMUEBLE del inmueble REF0005 >> Editar".
- Contenido central:** Un formulario para editar datos de un inmueble. Incluye pestañas para DEMANDAS, PROPIETARIOS, CARACTERÍSTICAS y SITIOS CERCANOS. Botones para "Asociar demanda" y "Crear demanda". Una tabla con los siguientes datos:

| Ref. | Cliente | Tipos inm. | Lugar | Precios | Metros | Hab. | Opc. |
|---------|-------------|------------|-------|----------------|--------|------|------------|
| DEM0002 | López Jesús | Duplex | Cádiz | 100000-150000€ | 80-100 | 3-4 | Borrar (X) |
- Sección UBICACIÓN:** Un cuadro azul que contiene campos para Provincia (Cádiz), Población (San Fernando), Zona (La cardila) y Dirección (Calle de la huerta, 2, 3º C). Abajo de estos campos, se muestra una miniatura de mapa de Google Maps que indica la ubicación del inmueble con un marcador amarillo y una "X" roja en la parte inferior derecha.
- Pie de página:** Muestra "OPENRS 2017" en la parte inferior.

Figura 4.16: Mockup inicial de zona privada

En esta podemos diferenciar las diferentes zonas que componen la plantilla, como la cabecera (con la zona de acceso y perfil el margen superior derecho), el menú de secciones en el lateral izquierdo, el pie de página así como la zona central (o sección de interfaz), compuesta por un menú de sub-secciones del inmueble, y luego una serie de pestañas para creación y acceso rápido de información. Finalmente, se describe brevemente como quedaría parte del formulario del inmueble. También observamos una rueda dentada a la derecha que permitirá visualizar la interfaz en diferentes formatos.

Posteriormente, se pasó a realizar una plantilla de diseño preliminar de las principales secciones.

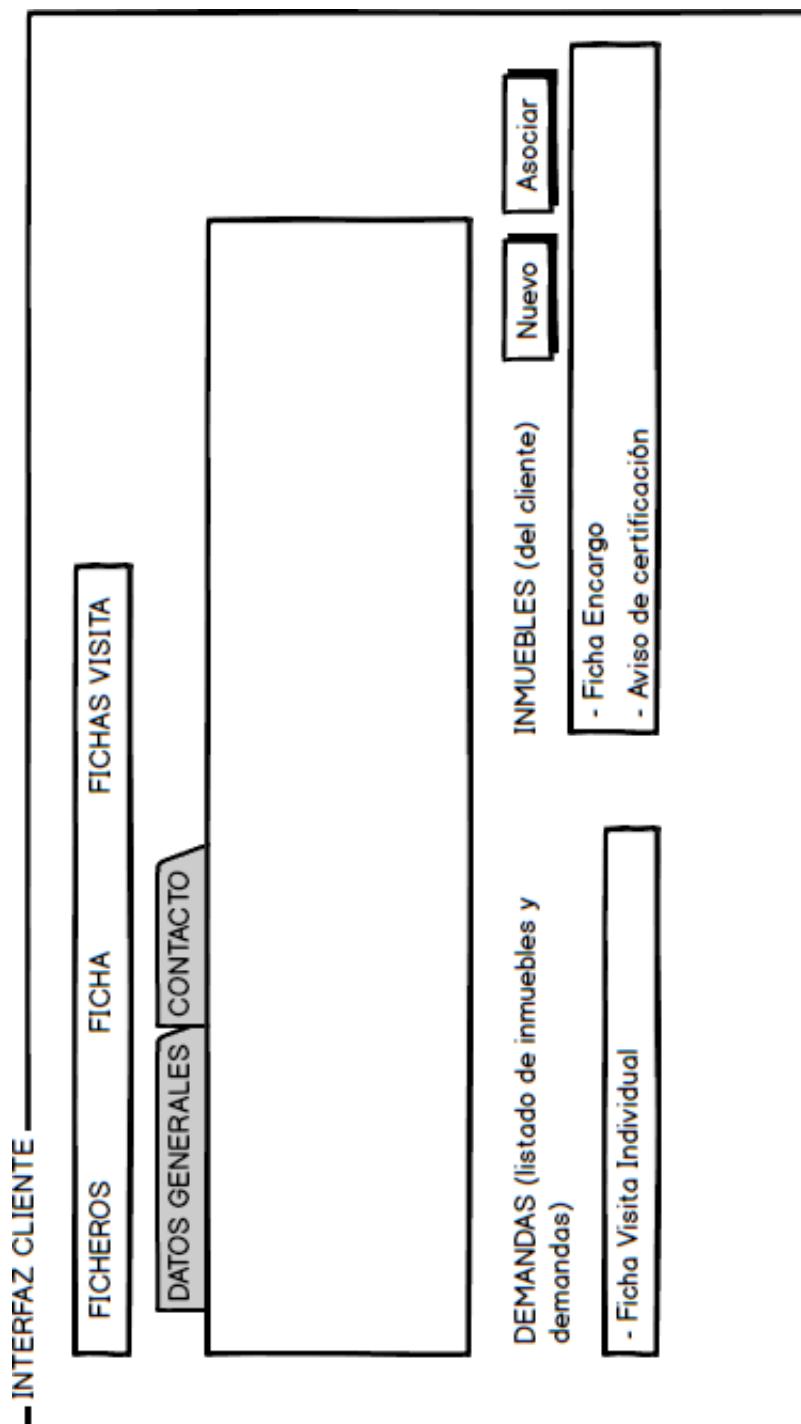


Figura 4.17: Mockup inicial del cliente

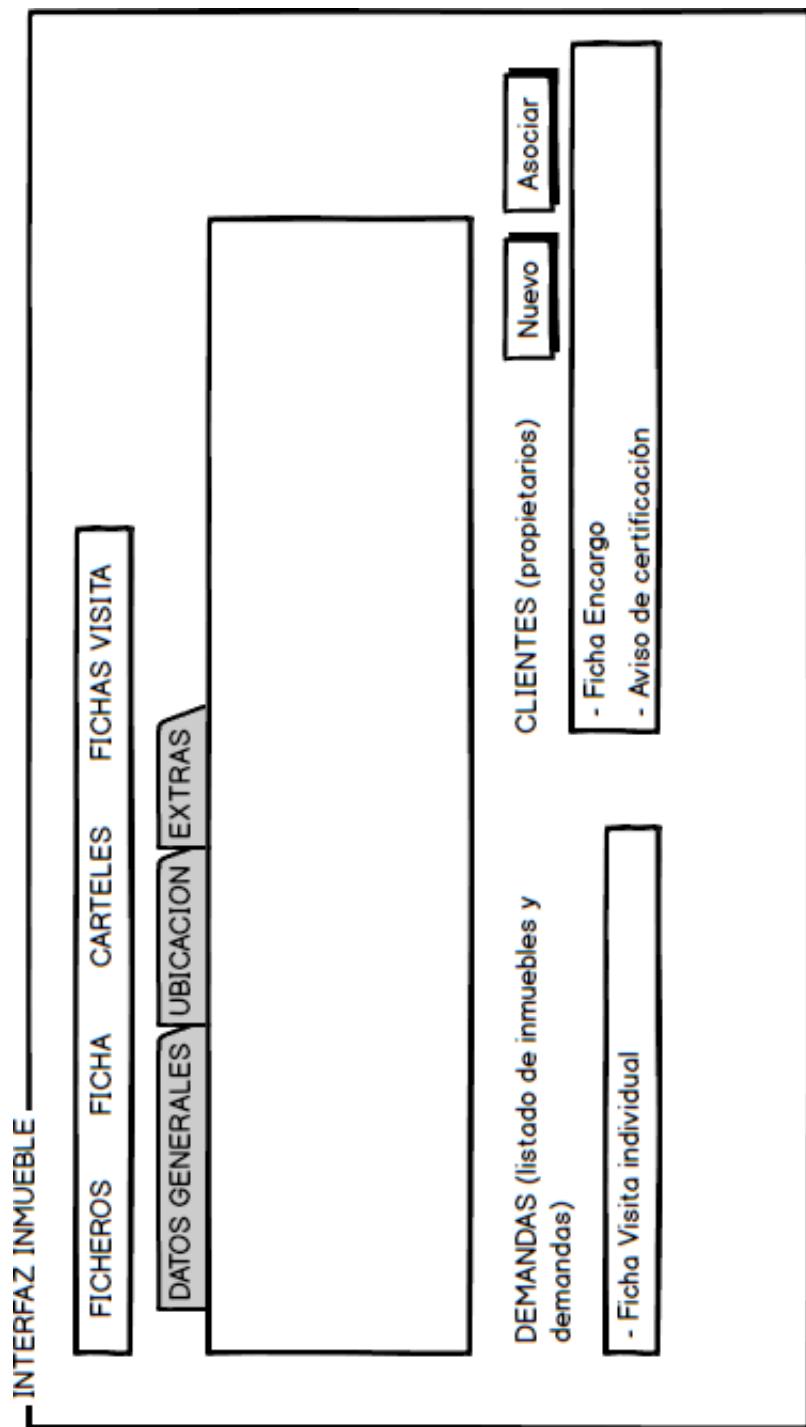


Figura 4.18: Mockup inicial del inmueble

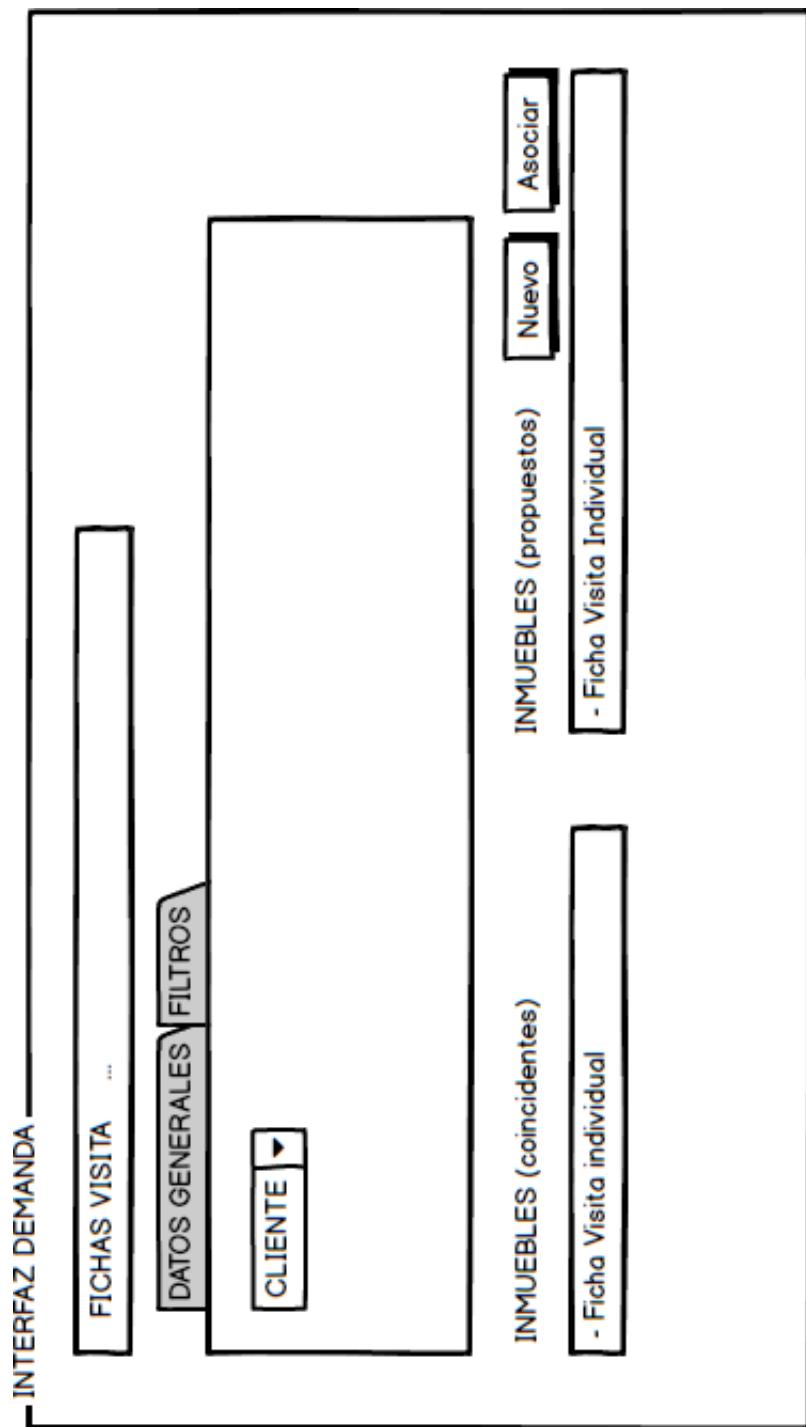


Figura 4.19: Mockup inicial de la demanda

Estos diseños preliminares no convencieron mucho a GESTICADIZ, y teniendo en cuenta sus indicaciones, se pasó a realizar un segundo diseño, en el que toma más relevancia la interconexión entre las distintas secciones de la web y clasifica mejor la información:

| | | | | | |
|--|--------------|-----------------|-----------------|---------------------------------|-------------------------------|
| DATOS INMUEBLE | IMAGENES | ENLACES | ADJUNTOS | FICHA | CARTEL |
| <u>DATOS INMUEBLE del inmueble REF0005 >> Editor</u> | | | | | |
| DEMANDAS | PROPIETARIOS | CARACTERISTICAS | SITIOS CERCANOS | | |
| | | | | Asociar demanda | Crear demanda |
| Ref. | Cliente | Tipos inm. | Lugar | Precios | Metros |
| DEM0002 | López Jesús | Duplex | Cádiz | 100000-150000€ | 80-100 |
| Hab. | Opc. | | | | |
| 3-4 | Borrar (X) | | | | |

| | |
|-----------------------|--|
| UBICACIÓN | |
| Provincia | <input type="text" value="Cádiz"/> |
| Población | <input type="text" value="San Fernando"/> |
| Zona | <input type="text" value="La ardila"/> |
| Dirección | <input type="text" value="Calle de la huerta, 2, 3º C"/> |
| Ubicación Google Maps | |

Figura 4.20: Mockup final del inmueble

La adaptación de la visualización de la web a múltiples dispositivos es muy importante, y debe realizarse un diseño lo más "user-friendly" posible. Se propone esta elegante interfaz:

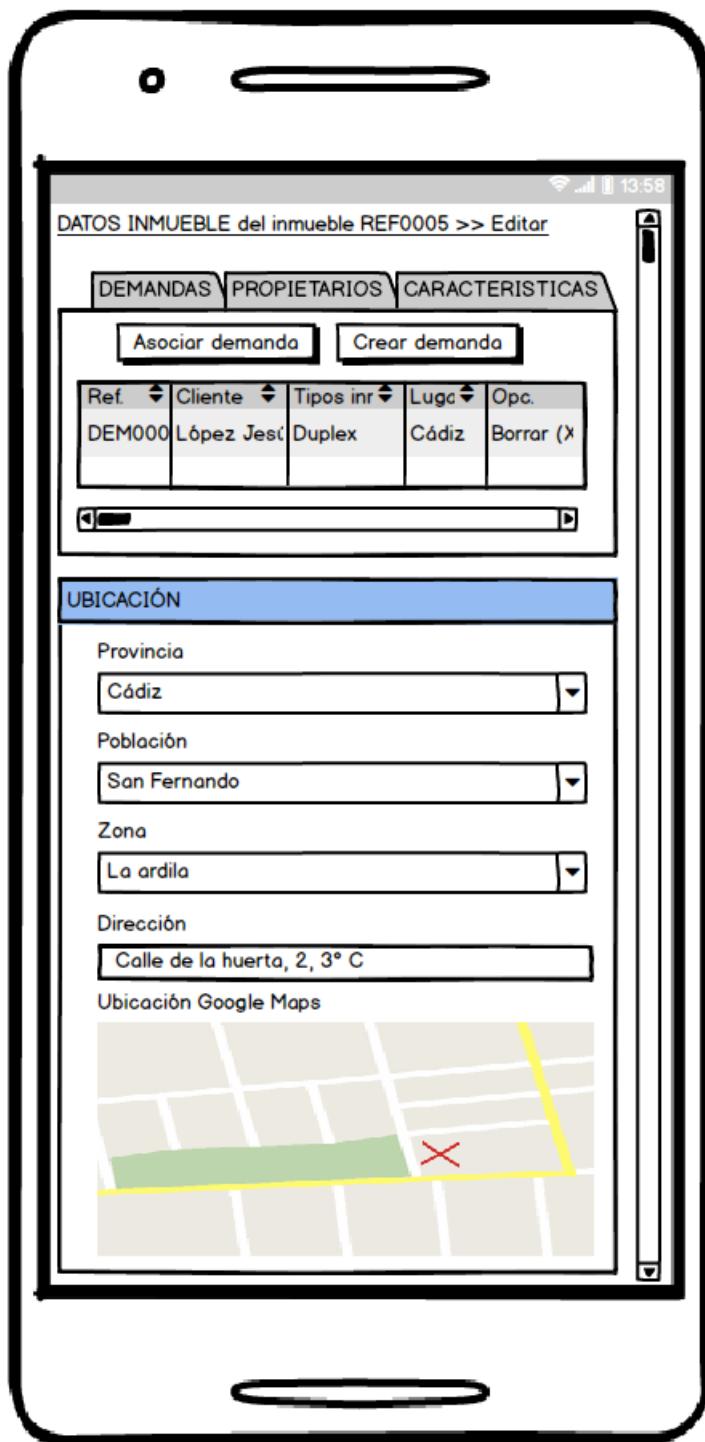


Figura 4.21: Mockup responsive del inmueble

Cómo plantilla para la sección pública, como depende de la organización y configuración que defina el usuario, no se puede incluir un mockup específico, no obstante, se propone una muy similar a la plantilla New York [21] de Inmoenter [20]. Se expone aquí la página

de inicio a modo de ejemplo:

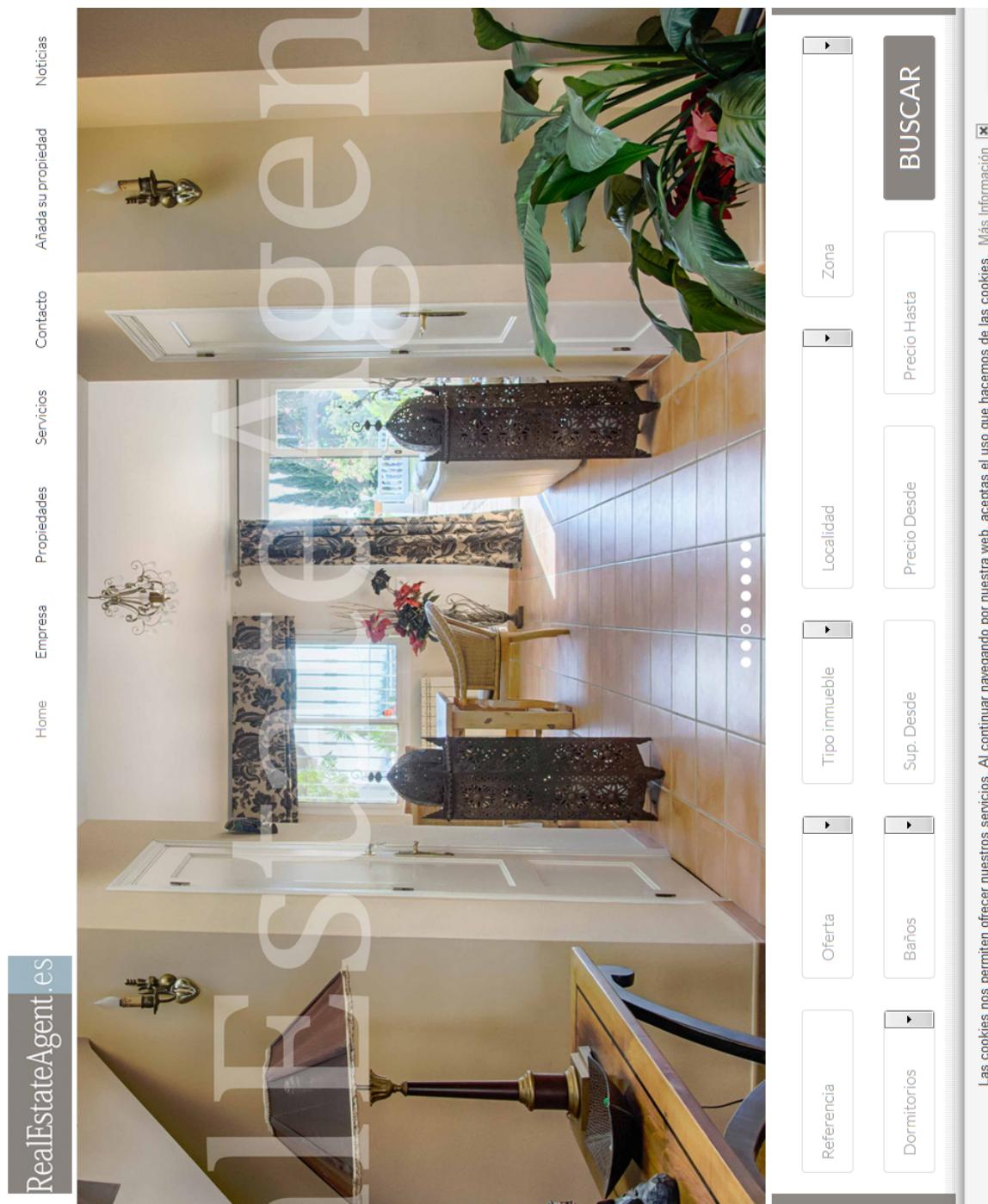


Figura 4.22: Plantilla de diseño público (Parte 1)

ÚLTIMAS PROPIEDADES

Villa en venta en Cortijos de San Rafael (Frigiliana)



Gran Villa en el exclusivo sector de Cortijos de San Rafael, a solo 10 minutos del...

900.000 € [VER](#)

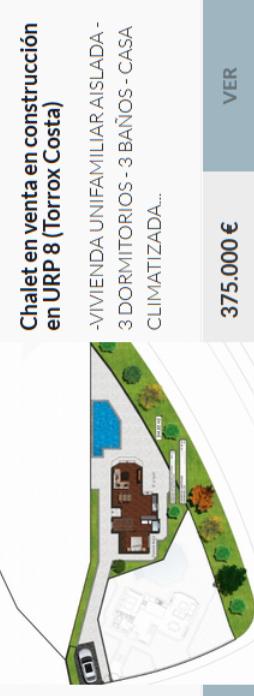
Chalet de Lujo en venta en Torre del Mar



Diseño, calidad, modernidad y exclusividad: estas son sólo algunas de las palabras...

387.000 € [VER](#)

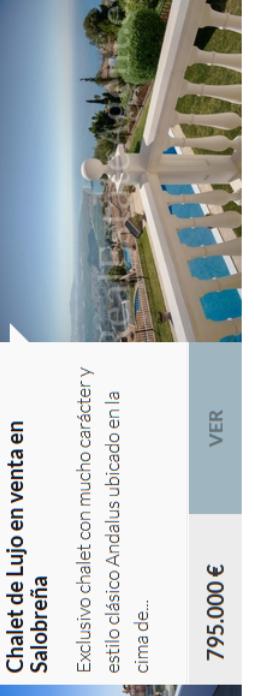
Chalet en venta en URP 8 (Torrox Costa)



-VIVIENDA UNIFAMILIAR AISLADA - 3 DORMITORIOS - 3 BAÑOS - CASA CLIMATIZADA...

375.000 € [VER](#)

Chalet de Lujo en venta en Salobreña



Exclusivo chalet con mucho carácter y estilo clásico Andaluz ubicado en la cima de...

795.000 € [VER](#)

< >

< >

OPORTUNIDADES

Oportunidad

< >

< >

< >

< >

Casa en venta en Capistrano area (Nerja)

Magnífica propiedad con vistas panorámicas sobre el mar, las montañas y todo el pueblo, ubicada en una de las zonas más

Figura 4.23: Plantilla de diseño público (Parte 2)

El diagrama de navegación del sistema puede desprenderse, en su mayor parte, de la división de apartados especificada en los casos de uso. No obstante se ofrece un pequeño diagrama en el que se explica la lógica seguida para elaborar el mapa de navegación del cliente, el cual sirve de modelo para los demás:

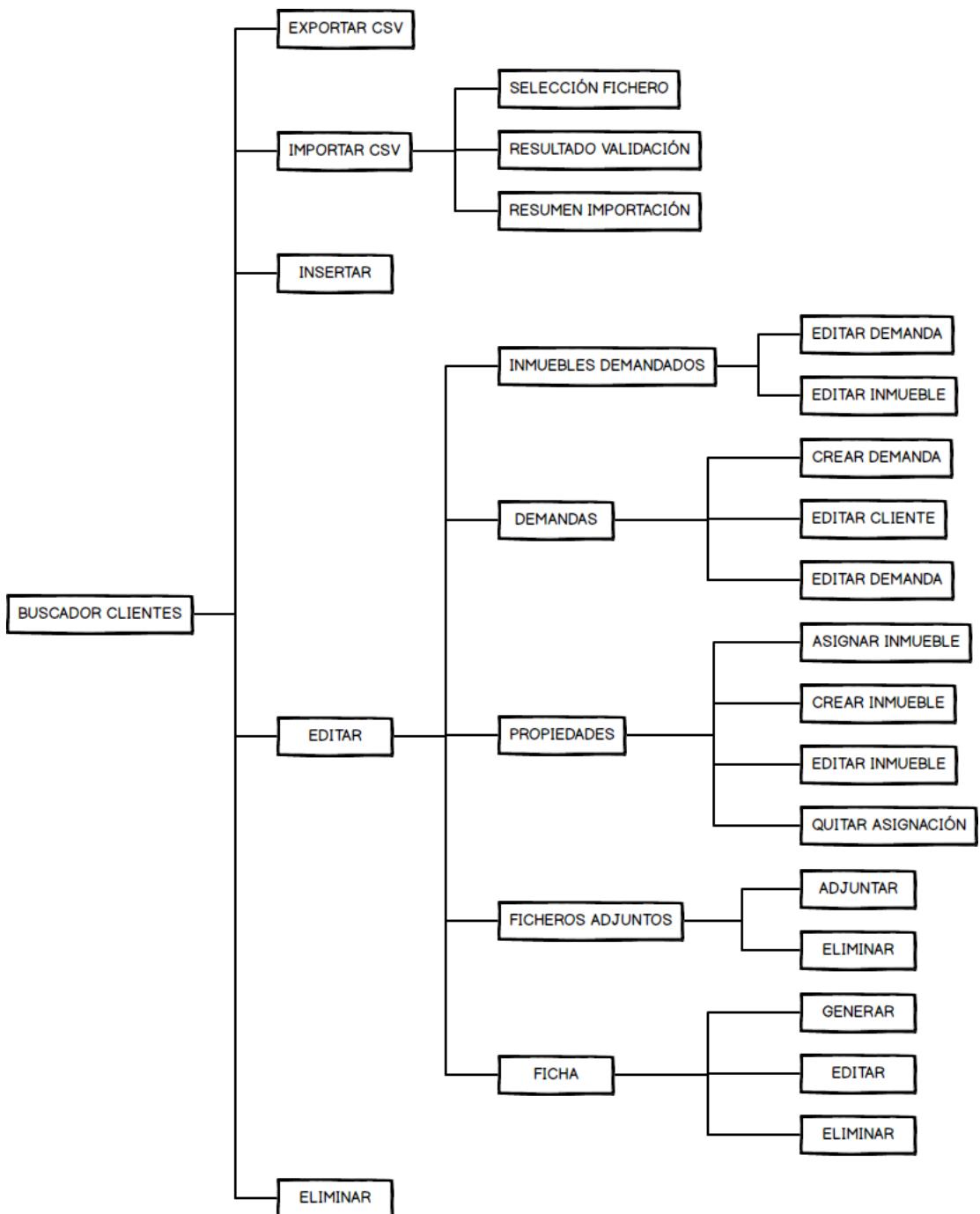


Figura 4.24: Mapa de navegación del cliente

Capítulo 5

Diseño del Sistema

5.1. Arquitectura del Sistema

En esta sección se define la arquitectura general del sistema de información, especificando la infraestructura tecnológica necesaria para dar soporte al software y la estructura de los componentes que lo forman.

5.1.1. Arquitectura Física

En este apartado, describimos los principales elementos hardware que forman la arquitectura física de nuestro sistema, recogiendo por un lado los componentes del entorno donde se va a desplegar la aplicación (en nuestro caso STAGING), los del entorno origen (en nuestro caso DEVELOPMENT) así como un ejemplo de cómo se realiza la petición y se resuelve una petición desde el cliente (en nuestro caso el navegador web).

Con el diagrama 5.5 se quiere dar una visión global no solo de cómo se despliega la aplicación desde un servidor local a cualquier otro servidor, si no relacionar todos estos conceptos con el proceso de Aseguramiento de la Calidad 3.6.

Al utilizar tecnologías multiplataformas no hace falta ningún requerimiento especial en los elementos de hardware, ya que la potencia de los dispositivos actuales supera con creces los requisitos mínimos para poder hacer funcionar la aplicación. Con cualquier instalación predeterminada que poseen los típicos paquetes WAMP o LAMP es suficiente. No obstante si me gustaría comentar, al menos, las versiones mínimas que deberían disponer el servidor a nivel de entorno tecnológico:

- PHP 5.5

- MYSQL 5.6
- APACHE 2.4
- Unos 100 MB de espacio (aunque si se quita la documentación adjunta del proyecto puede ser menor).

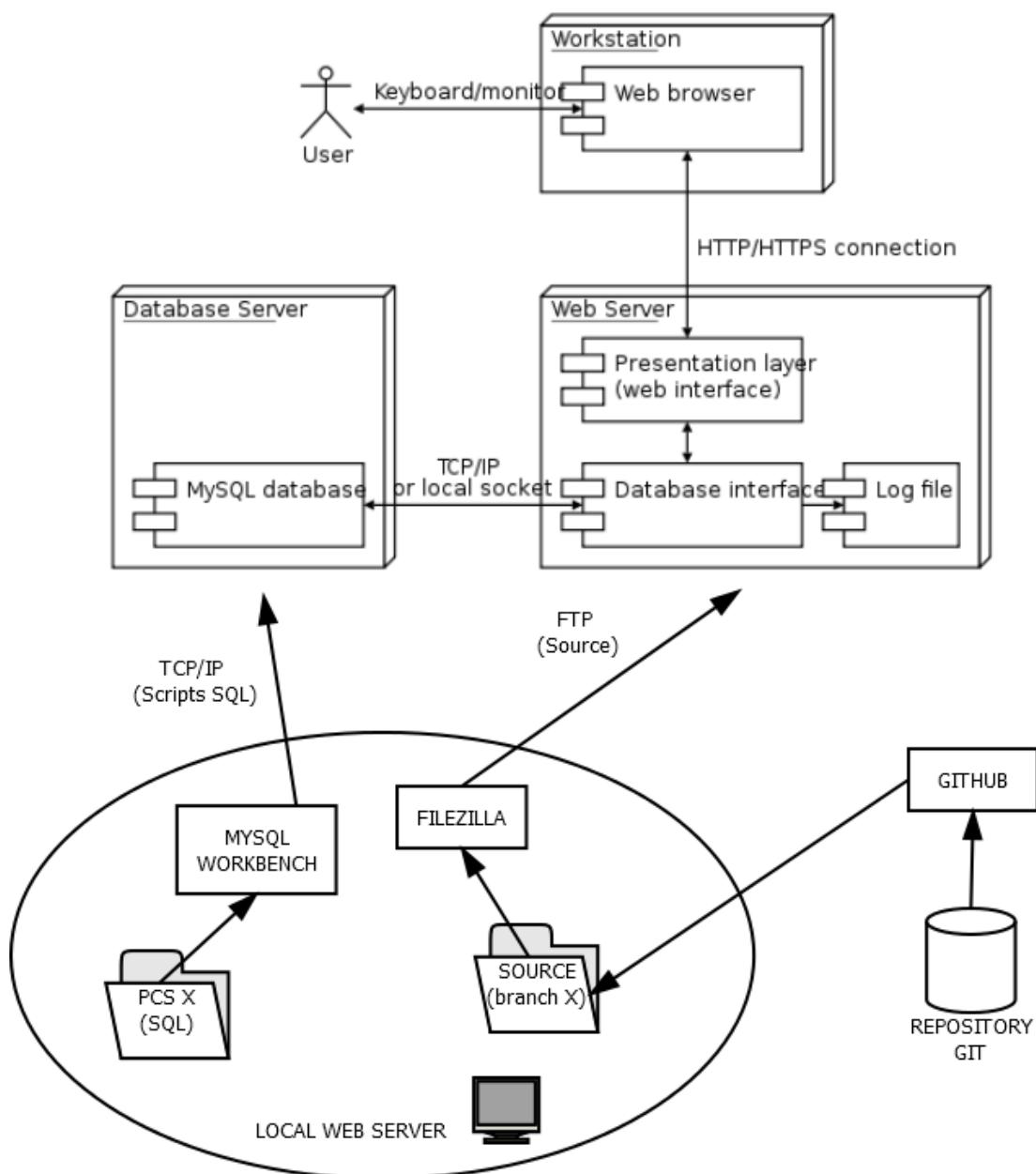


Figura 5.1: Diagrama de despliegue

En la práctica, como hemos explicado anteriormente en la Planificación, el servidor de STAGING, para que sea lo más parecido a un entorno real, será un HOSTING con PLESK

proporcionado por TIPYCOS que apuntará al dominio OPENRS.es y correrá bajo Linux. En el entorno de desarrollo se utilizará el LAPTOP especificado corriendo sobre Windows. En cuanto a los navegadores que debería disponer el cliente, se recomiendan las últimas versiones de Mozilla, Safari, Chrome e Internet Explorer (al menos la 11). Los elementos del entorno tecnológico con el que se va a implementar y testear el sistema se han descrito en la planificación 3.3.1.

5.1.2. Arquitectura Lógica

Introducción

La arquitectura de diseño especifica la forma en que los artefactos software interactúan entre sí para lograr el comportamiento deseado en el sistema. En esta sección se muestra la comunicación entre el software base seleccionado, los componentes reutilizados y los componentes desarrollados para cumplir los requisitos de la aplicación. También, se recogen los servicios de sistemas externos con los que interactúa nuestro sistema.

En los sistemas web de información es común la utilización del patrón Layers (Capas), con el cual estructuramos el sistema en un número apropiado de capas, de forma que todos los componentes de una misma capa trabajan en el mismo nivel de abstracción y los servicios proporcionados por la capa superior utilizan internamente los servicios proporcionados por la capa inmediatamente inferior. Habitualmente se tienen las siguientes capas:

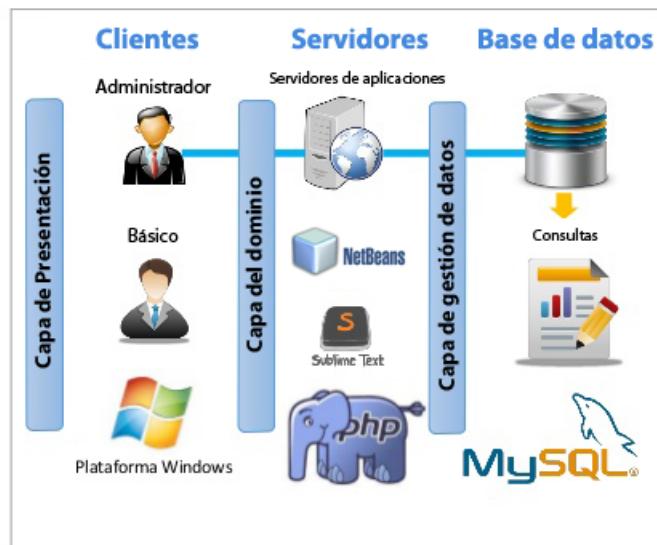


Figura 5.2: Patrón Layers (Capas)

Capa de presentación Este grupo de artefactos software conforman la capa de presentación del sistema, incluyendo tanto los componentes de la vista como los elementos de control de la misma.

Capa de negocio Este grupo de artefactos software conforman la capa de negocio del sistema, incluyendo los elementos del modelo de dominio y los servicios (operaciones del sistema).

Capa de persistencia Este grupo de artefactos software conforman la capa de integración del sistema, incluyendo las clases de abstracción para el acceso a datos (BD o sistema de ficheros) o a sistemas heredados.

No obstante, aunque esta explicación anterior es más o menos válida para el sistema a construir, es demasiado genérica y se hace necesario profundizar un poco más en la arquitectura real del framework a utilizar, Codeigniter 3:

1. El index.php sirve como controlador frontal, inicializando los recursos de base necesarios para ejecutar CodeIgniter.
2. El router examina la solicitud HTTP para determinar qué se debe hacer con ella.
3. Si existe un archivo de caché, se envía directamente al navegador, evitando la ejecución normal del sistema.
4. Antes de cargar el controlador de aplicación, la solicitud HTTP y los datos enviados por el usuario se filtran para garantizar su seguridad.
5. El controlador carga el modelo, librerías, helpers y cualquier otro recurso necesario para procesar la solicitud específica.
6. La vista finalizada se procesa y luego se envía al navegador web. Si se habilita el almacenamiento en caché, la vista se almacena en caché primero de modo que en las solicitudes posteriores esta pueda ser servida directamente.

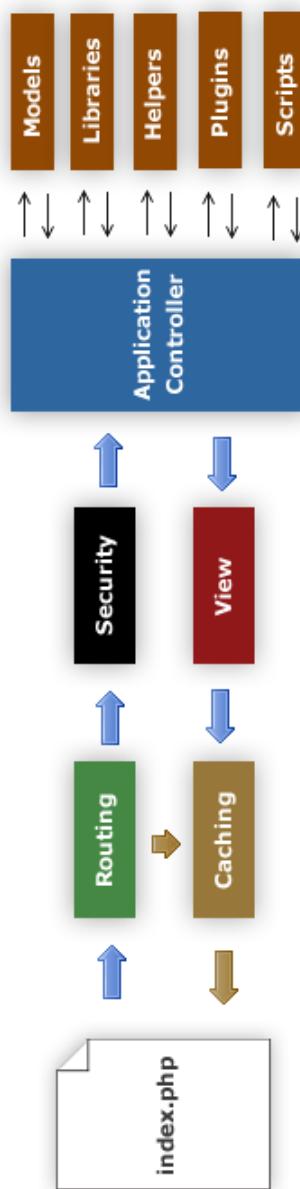


Figura 5.3: Flujo de ejecución en Codeigniter 3

Patrón Modelo-Vista-Controlador

MVC es un enfoque de software que separa la lógica de la aplicación de la presentación. En la práctica, permite que las páginas web contengan scripts mínimos ya que la presentación está separada de los scripts PHP. CodeIgniter implementa este patrón siguiendo los siguientes criterios:

- El **modelo** representa a las estructuras de datos. Normalmente, las clases de modelo contendrán funciones que le ayudarán a recuperar, insertar y actualizar informa-

ción en su base de datos, aunque en este sistema, se le añadirán comportamientos adicionales.

- La **vista** es la información que se presenta al usuario. Una Vista normalmente será una página web, pero en CodeIgniter, una vista también puede ser un fragmento de página como un encabezado o pie de página. También puede ser una página RSS, o cualquier otro tipo de "página".
- El **controlador** sirve como intermediario entre el modelo, la vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web o cualquier otro tipo de respuesta (como por ejemplo un JSON).

Realizando una comparación con la explicación inicial del patrón layers, los modelos encapsularían las funcionalidades de las capas de negocio y de persistencia, mientras que la vista representaría la capa de presentación. Luego tenemos a los controladores, que al ser intermediarios y manejar el flujo de control del sistema, van a proporcionar servicios tanto a la capa de negocio como a la de presentación.

Otro de los aspectos más importantes del sistema son el resto de artefactos software que pueden ser usados por distintos elementos de cualquiera de las capas del sistema y que proporcionan una gran gama de servicios. Entre ellos vamos a destacar, al ser los más empleados, los siguientes:

- **Módulos:** Consideraremos como módulo a cualquier componente que al menos disponga de una vista, modelo y controlador para manejar una funcionalidad o conjunto de funcionalidades específicas para lograr un determinado objetivo.
- **Librerías:** Podrán ser extensiones del núcleo de CI u otras externas integradas, pero se encargarán de manejar diferentes aspectos del sistema y abstraer su uso a bajo nivel (como por ejemplo el manejo de sesiones o de ficheros)
- **Helpers:** Son fragmentos de código encapsulado en funciones, cuyo objetivo es ayudar a realizar una determinada tarea en una determinada capa.

Diagramas

Por otro lado, tal como muestra la imagen 5.4, el propio framework nos proporciona componentes para manejar los controladores con métodos para, por ejemplo, cargar vistas o

cargar modelos. Estos métodos son muy útiles pero se hace necesario añadir una segunda capa de abstracción a nivel del CORE de nuestro sistema que sea independiente y a su vez permita integrar el comportamiento de otros componentes de software. Construyendo esta capa intermedia podremos tenerla como base para cualquier proyecto de software y no solo para este, ya que se encargará de manejar aspectos comunes a todas las aplicaciones de software como:

- Control de acceso.
- Nivel de permisos.
- Internacionalización.
- Display de interfaces en layout.

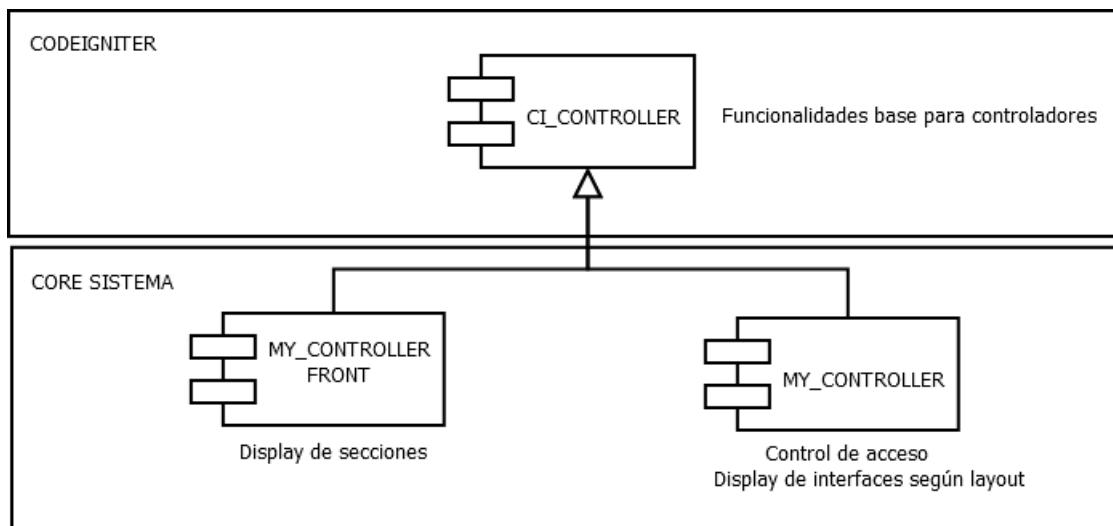


Figura 5.4: Diseño de controladores

El componente de routing del framework será el encargado de determinar qué peticiones son atendidas por cada controlador. Las peticiones de la zona pública, irán siempre precedidas del idioma (es, en, fr, etc.), mientras que las de la zona privada no. Así evitamos 'colisiones' de urls de las secciones públicas con los controladores privados.

Uno de los aspectos más útiles y que me gustaría resaltar de esta capa de abstracción es la utilización del patrón layout. Esta patrón de diseño nos permite abstraer sobre qué plantilla o layout del frontend vamos a desplegar una determinada vista. Así, dependiendo del controlador o del caso de uso en concreto que herede este componente, dispondrá de métodos para desplegar las interfaces sobre la plantilla que desee y además, al encapsularlo

todo en un único método, podemos reemplazar el layout fácilmente, lo que nos permitirá incluir varias plantillas adicionales con facilidad tal y como se explica en las sección 10.5. Además, otras de las ventajas más importantes de este diseño es que los controladores hijos son los que tienen la responsabilidad de controlar la seguridad (aunque la lógica para manejarlas estará implementada en el padre), a diferencia de otros diseños donde se suelen tener controladores estáticos para zonas privadas y zonas públicas. En este caso, se pueden tener funcionalidades tanto públicas como privadas dentro de estos controladores. Esto ha tenido que realizarse así, ya que hay determinadas funcionalidades del módulo del CMS o de autentificación que así lo requieren.

Por otro lado, tal como muestra la imagen 5.5, Codeigniter nos proporciona componentes para manejar los modelos con muchas utilidades diferentes como, por ejemplo, el query builder, que nos permite construir fácilmente y de manera flexible y escalonada, casi cualquier consulta SQL que queramos realizar con un lenguaje OO más próximo al desarrollador. Estos métodos son muy útiles pero se hace necesario añadir una segunda capa de abstracción a nivel del CORE de nuestro sistema que sea independiente y a su vez permita integrar el comportamiento de otros componentes de software.

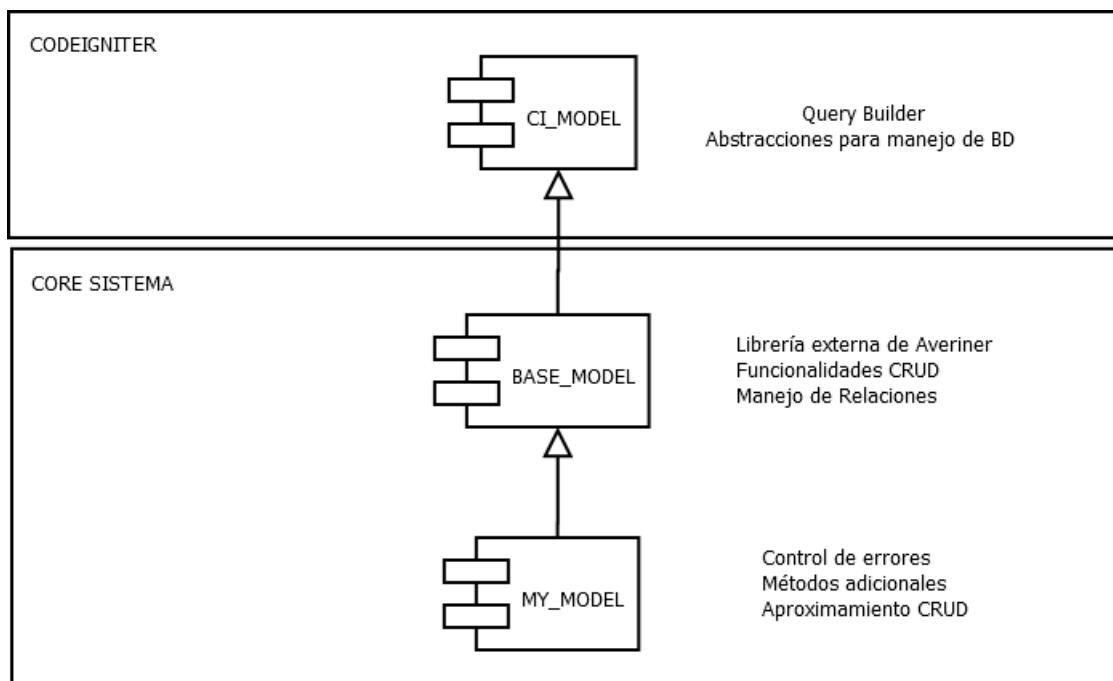


Figura 5.5: Diseño de modelos

Esta segunda capa, a su vez, está dividida en dos niveles de abstracción:

- La primera usa el componente que el autor Averiner [27] define como MY_Model,

la cual no hay que confundir con la siguiente capa de abstracción que explicaremos a continuación. Este componente lo hemos renombrado como Base_Model ya que va a contener funcionalidades CRUD así como de manejo de relaciones entre otras muchas más que nos van a servir como base para la siguiente capa.

- En la segunda capa de abstracción, MY_Model, implementaremos responsabilidades generales pero que están relacionadas con la arquitectura y las estrategias que hemos utilizado para diseñar la base arquitectónica del sistema, como son:
 - El manejo de errores dentro de los modelos.
 - La comprobación de existencia de objetos.
 - El uso de vistas.

En definitiva, teniendo estos niveles de abstracción, podremos disponer de un conjunto de funcionalidades útiles y reutilizables tanto para nuestros controladores como para nuestros modelos y que podrán ser empleadas en futuros desarrollos. Una vez se ha ofrecido esta introducción inicial, se pasa a explicar cómo hemos organizado todos los componentes para implementar tanto la zona privada como la zona pública.

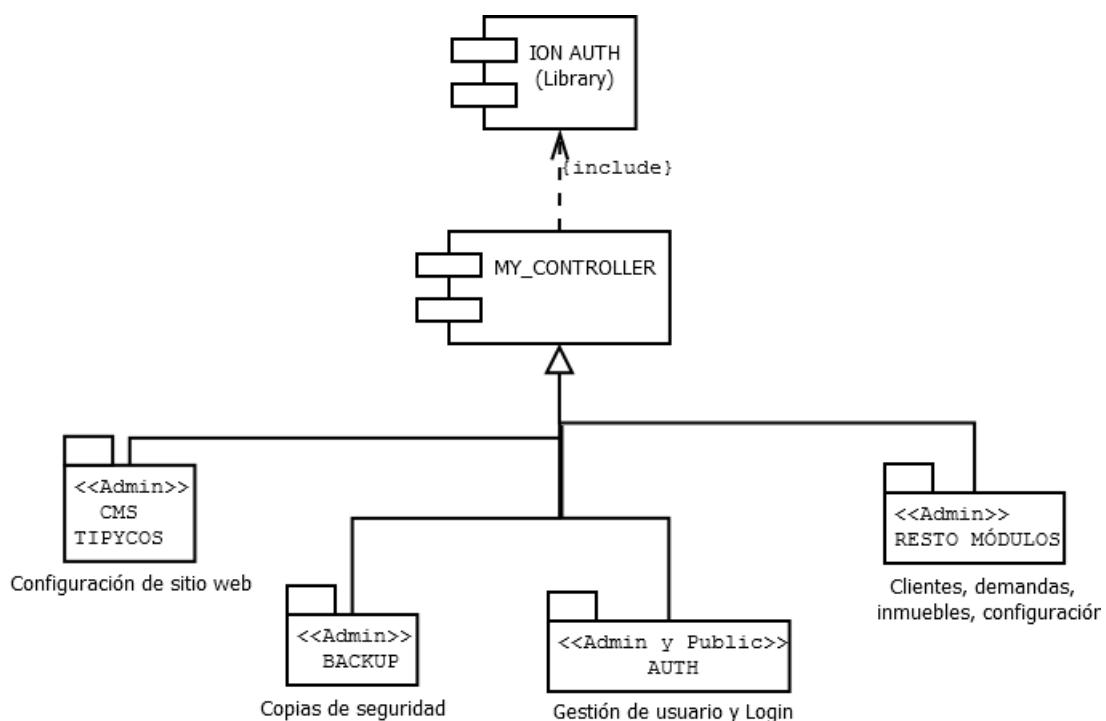


Figura 5.6: Diseño de software de gestión

Cómo se explica en la siguiente sección, el módulo de **ION Auth** será el encargado tanto de verificar los permisos de la zona privada o software de gestión, de la gestión de los usuarios así como la autentificación. Al almacenar a los usuarios a través de este módulo, también será el que determiné quién realiza las opciones sobre otros módulos, como los backups, los inmuebles, los clientes, etc.

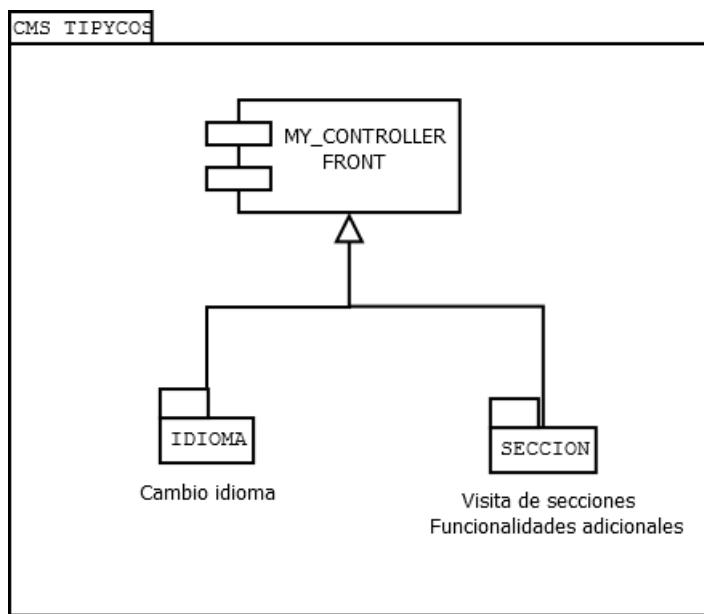


Figura 5.7: Diseño de sitio web

La zona pública o sitio web está compuesta, principalmente, por el CMS de TIPYCOS [24]. Dispondrá de su propia configuración de idioma establecida por el usuario a través del propio CMS en su configuración (por defecto el español), mientras que la zona privada, dispondrá de su propio idioma, por defecto el español y que vendrá definido a través del propio registro del usuario en la base de datos. Además, se observa como el controlador que vamos a emplear será diferente al de la zona privada ya que no tiene que manejar, entre otras cosas, los permisos y la autentificación de los usuarios.

Todo lo que se ha comentado hasta ahora pertenece al ámbito de lo que vamos a definir como backend (controladores y modelos principalmente), pero respecto al frontend (vistas), también hay que comentar algunas cosas. La primera de ellas es el nivel de acceso del componente KCFINDER a los ficheros adjuntos.

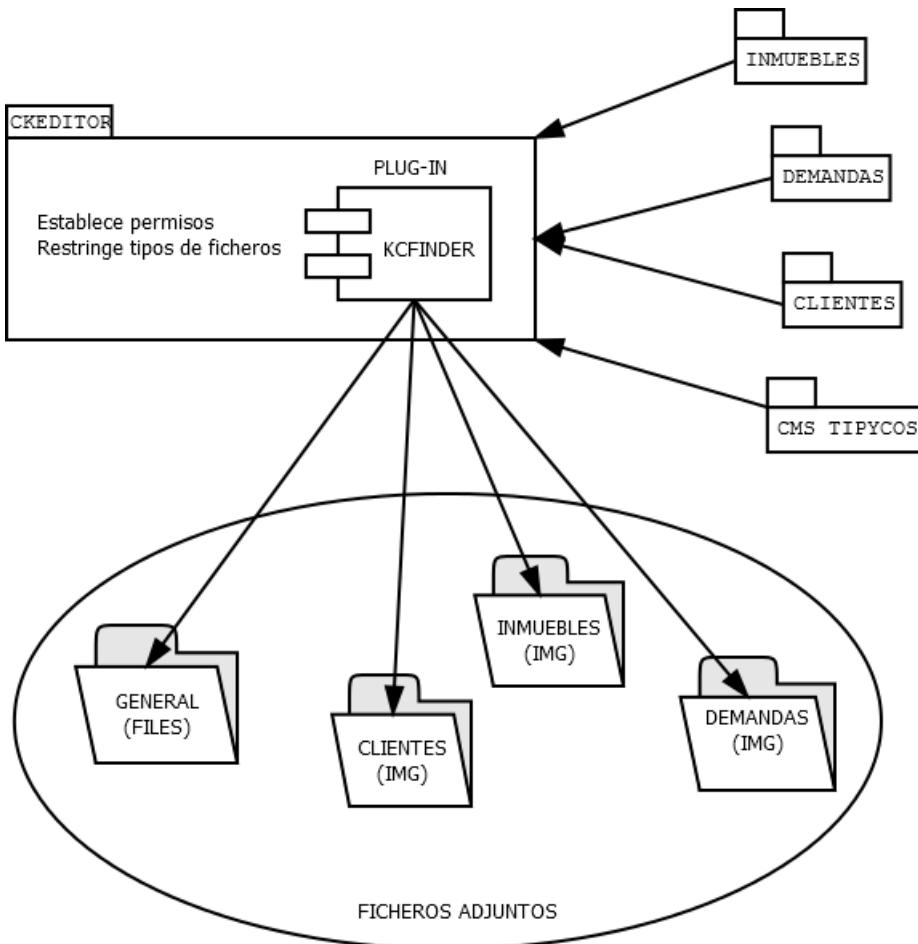


Figura 5.8: Nivel de acceso de KCFINDER a ficheros adjuntos

En la figura anterior, se observa como el componente KCFINDER [29], dependiendo del módulo en el que se use, sólo podrá acceder a los datos adjuntos de ese módulo. Mientras que si este es utilizado dentro de un módulo de ámbito general (como por ejemplo el CMS), accederá a los datos adjuntos generales. Por otro lado también habrá que especificar qué tipos de ficheros queremos adjuntar y visualizar. El módulo general podrá tener cualquier tipo de ficheros, mientras que la principal utilidad de esta herramienta en el resto de módulos será la de incluir, a través de CKEDITOR [28] en un área de texto, imágenes adjuntas en los formularios, como por ejemplo las imágenes de los inmuebles.

Finalmente, me gustaría comentar que se van a utilizar tres layouts diferentes sobre los cuales se van a cargar las interfaces del sistema.

- El primero servirá como plantilla para la zona pública, el cual estará basado en una plantilla simple de Bootstrap y contendrá componentes básicos.
- El segundo servirá como plantilla para la zona privada, el cual estará basado en la

plantilla Ace Template (Bootstrap) y contendrá componentes adicionales.

- El tercero servirá como plantilla para la zona de login, el cual estará basado en Materialize.

En la sección de Frameworks 6.1.1 se detalla más información.

Interfaces con sistemas externos

El sistema interactuará con los siguiente sistemas externos:

- API de Google Maps v3 [18]: Mostrará los mapas de la situación de un inmueble en particular o de un conjunto.
- Google Analytics [48]: Se ha seguido la integración estándar ofrecida por Google en su web para medir las estadísticas de la zona pública.
- Recaptcha de Google [49]: El CMS de TIPYCOS incorpora este componente, pero debe de configurarse en la plataforma para que funcione.
- Youtube: Se realizará una lectura con CURL al registrar los enlaces de un inmueble para validar si están activos y son embedables en la web corporativa.

5.2. Parametrización del software base

Siguiendo los diagramas anteriormente expuestos, las parametrizaciones así como los referimientos necesarios para adaptar los módulos a los requisitos del sistema son:

- Del CMS se implementará:
 - Aunque al principio pensaba que habría que crear toda la sección de inicio y el buscador como respuesta a eventos fijos, TIPYCOS me aconsejó que sería mucho más flexible y rápido reutilizar una plantilla que usaban para mostrar productos de una tienda, añadiéndole dos tipos de bloques personalizados. Uno para manejar listados de inmuebles (como los destacados) y otro para manejar sus búsquedas.
 - Se deberán maquetar todas las interfaces para adaptarla a las diferentes clases usadas por la plantilla de frontend.
 - Manejar la seguridad a través de la librería de ION Auth.

- Se elimina al usuario de la tabla de configuración del CMS para que la configuración sea para la plataforma en general y no para un usuario en concreto, tal como planteaba el enfoque inicial de este módulo.
- Se separa la utilización del idioma, para que sea diferente del asignado en la zona privada.
- Del **Backup** se implementará:
 - La funcionalidad para realizar copias completas del sistema.
 - Se deberán maquetar todas las interfaces para adaptarla a las diferentes clases usadas por la plantilla de frontend.
 - Almacenar el usuario que realiza la copia de seguridad.
 - Manejar la seguridad a través de la librería de ION Auth.
- Del Módulo **ION Auth**:
 - Definir la posibilidad de poner un idioma para los usuarios de forma que sea el idioma utilizado en la plataforma cuando inicie sesión ya que el módulo, en sí mismo, no tiene definición alguna de idioma. Aunque esto sólo estará disponible para el entorno de desarrollo, para el resto de entornos, el idioma será siempre por defecto, el idioma definido en el framework.
 - Se deberán maquetar todas las interfaces para adaptarla a las diferentes clases usadas por la plantilla de frontend.
 - Se implementará la funcionalidad de eliminar un usuario.
 - Por otro lado, al tener una librería incorporada que maneja la información de autenticación, esta será la base para manejar tanto los permisos como la autenticación en el sistema.
 - Se configura la librería para que soporte dos roles y que la clave de registro del usuario sea el correo electrónico. El resto de parámetros se mantendrán por defecto.

De otros componentes:

- De **KCFinder** habrá que personalizar los niveles de acceso y las carpetas así como los tipos de datos habilitados en cada una de ellas para los módulos que se han indicado en el diagrama 5.8.

Del propio framework:

- Habilitaremos la comprobación con CRSF excepto en algunas funcionalidades.
- Almacenaremos la sesión en la base de datos.

5.3. Diseño físico de Datos

En esta etapa, se parte del esquema lógico global obtenido durante la etapa de Análisis y se obtiene una descripción de la implementación de la base de datos. Esta descripción es completamente dependiente del SGBD específico que se vaya a utilizar e incluye las estructuras de almacenamiento y los métodos de acceso que se utilizarán para conseguir un acceso eficiente a los datos. El SGBD que se utilizará para diseñar esta aplicación será MySQL Workbench y de describirá más adelante en el capítulo de Implementación 6.1.

5.3.1. Motor de almacenamiento MySQL

Se comenzará eligiendo el motor de almacenamiento MySQL. El motor de almacenamiento (storage-engine) se encarga de almacenar, manejar y recuperar información de una tabla. La elección de uno u otro dependerá mucho del escenario donde se aplique. Para esta aplicación se ha escogido el motor de almacenamiento InnoDB por los siguientes motivos:

- Capacidad para soportar **transacciones ACID** (Atomicity, Consistency, Isolation, Durability).
- **Integridad de datos:** Previene el alta de datos no adecuados. A menudo, los datos de una tabla están relacionados con los de otras. Las restricciones de clave externa aseguran la integridad de estas relaciones.
- Mayor velocidad en general a la hora de recuperar datos.
- Bloqueo a nivel de registro: si varios usuarios o aplicaciones intentan acceder simultáneamente a la información contenida en una tabla para modificarla, el bloqueo a nivel de registro hace que solo deban quedar a la espera aquellos que intentan modificar los datos de un mismo registro.
- Recuperación automática en caso de crash: Si el servidor sufre una avería, o por alguna otra razón finaliza abruptamente la ejecución de mysql, las tablas InnoDB

se recuperan automáticamente volviendo a ejecutar las transacciones grabadas en el fichero de log de transacciones, haciendo rollback de las que estén incompletas.

5.3.2. Creación de las tablas

Una vez decidido el motor de almacenamiento a utilizar, se pasará a crear las distintas tablas de la base de datos. Después de aplicar el modelo relacional y de pasar el proceso de normalización, se han obtenido 74 tablas para la base de datos. A la creación de dichas tablas, se le sumará la creación de índices y restricciones de integridad que reflejarán las claves foráneas de cada tabla.

Para la mayoría de las tablas con clave primaria simple, se definirá dicha clave como **autoincremental**. Esta decisión está basada en una serie de ventajas que hacen del uso de claves primarias autoincrementales una buena práctica a la hora de trabajar con bases de datos. A continuación se enumeran los motivos en los que está basada esta decisión:

- Facilita la obtención del último registro insertado. Basta con usar la función agregada `MAX (SELECT MAX(id) FROM tabla;)`.
- Permite conservar la integridad referencial entre dos o más tablas relacionadas. Si un campo que actúa como clave foránea en una o más tablas cambia, habría que hacer una operación extra en la base de datos para actualizar este campo en todas las tablas donde estuviese referenciado.
- Optimiza las consultas a la base de datos.
- Da cierta flexibilidad en cuanto a la cardinalidad de las relaciones entre dos entidades. Si esta cardinalidad cambiase después de haber sido definida inicialmente, la solución al problema sería tediosa y en ocasiones imposible al tener que reorganizar las claves primarias de las entidades involucradas.

Se añadirán índices a las tablas para aquellos atributos que hagan de clave foránea. Esto permitirá crear restricciones de integridad para las tablas basadas en estos índices. También se añadirán índices para claves alternativas en algunas de las tablas y para aquellas consultas más frecuentes (como las que se emplean en las estadísticas).

Tablas resultantes

Además de las tablas resultantes de las relaciones especificadas en el modelo lógico 4.1.2 de datos se incluyen estas otras pertenecientes a los sistemas externos integrados así como

otras tablas de apoyo:

- articulos: Noticias registradas.
- articulo_categorias: Categorías asignadas a las noticias.
- articulo_etiquetas: Etiquetas asignadas a las noticias.
- articulos_idiomas: Traducción en diferentes idiomas de las noticias.
- bloque_inmuebles: Configuración de datos de los bloque de tipo inmuebles.
- carrusel: Configuración de datos de los bloque de tipo carrusel.
- categoria_blog_idiomas: Traducción en diferentes idiomas de categorías asignadas a las noticias.
- categoria_blog: Categorías asignadas a las noticias.
- categoria_carrusel_idiomas: Traducción en diferentes idiomas de categorías asignadas a un bloque de tipo carrusel.
- categoria_carrusel: Categorías asignadas a un bloque de tipo carrusel.
- etiquetas: Traducción de cada etiquetas registrada en cada idioma.
- imagen_carrusel_idiomas: Imágenes asignadas a un bloque de tipo carrusel.
- imagen_carrusel: Traducción en diferentes idiomas de imágenes asignadas a un bloque de tipo carrusel.
- votos: Votos realizados en las noticias.
- comentarios_blog: Comentarios realizados de las noticias.
- backup: Copias de seguridad realizadas por el administrador.
- bloque: Bloques contenidos en las diferentes secciones.
- bloque_idiomas: Información dependiente del idioma de los bloques registrados.
- ci_sessions: Sesiones activas y manejadas por Codeigniter.
- config: Preferencias y datos de configuración establecidos por el administrador para el uso de la zona privada.

- config_admin: Preferencias y datos de configuración establecidos por el CMS para la configuración de la zona pública.
- footer_opciones: Opciones posibles del footer.
- footer_opciones_clientes: Opción seleccionada en cada columna del footer.
- footer_texto_idiomas: Información dependiente del idioma de las columnas del footer seleccionadas.
- geocoding: Relación de direcciones y coordenadas de Google Maps que sirven de caché para optimizar el posicionamiento de las direcciones en los mapas de Google.
- idiomas: Ya se incluyó en el análisis debida a su alta interconectividad con tablas relacionadas con las demandas, inmuebles y clientes aunque pertenece al componente del CMS.
- login_attempts: Intentos de acceso al sistema por IP.
- seccion: Secciones del CMS.
- seccion_idiomas: Información dependiente del idioma de las secciones registradas.
- texto: Contenido de los bloques de textos de las secciones.
- texto_idiomas: Información dependiente del idioma de las secciones registradas.
- tipo_bloque: Tipos de bloques que pueden asignarse.
- usuarios se renombra a users.
- grupos se renombra a groups.
- usuarios_grupos se renombra a users_groups.

Estas tres últimas tablas, al igual que ocurre con la de idiomas, fueron incluidas en el análisis debida a su alta interconectividad con otras tablas del sistema, ya que, almacena a los usuarios registrados en toda la aplicación independientemente del componente que se utilice.

Vistas creadas

- `v_backups`: Información de los backups realizados.
- `v_clientes`: Información principal de los clientes. Utilizada para las tablas de datos como las del buscador de clientes.
- `v_clientes_ficheros`: Muestra información de los ficheros adjuntos de los clientes.
- `v_clientes_inmuebles`: Vista usada en estadísticas para los intereses de los clientes.
- `v_demandas`: Información principal de las demandas. Utilizada para las tablas de datos como las del buscador de demandas.
- `v_demandas_ficheros`: Muestra información de los ficheros adjuntos de las demandas y sus tipos.
- `v_clientes_tipos_inmueble`: Vista usada en estadísticas para saber los tipos de inmuebles demandados.
- `v_estados`: Muestra información de los estados registrados.
- `v_inmuebles`: Información principal de los inmuebles. Utilizada para las tablas de datos como las del buscador de inmuebles.
- `v_inmuebles_demandas`: Información principal de los inmuebles asignados a las demandas y los datos relacionados.
- `v_inmuebles_ficheros`: Muestra información de los ficheros adjuntos de los inmuebles.
- `v_tipos_ficheros`: Muestra información de los tipos de ficheros registrados.

5.3.3. Interacción de módulos con el diseño físico

En el siguiente diagrama se ofrece una representación (muy a groso modo) de las comunicaciones entre las estructuras físicas anteriormente nombradas y los módulos. No obstante, como podemos observar en la siguiente figura, aunque una tabla pertenezca a un módulo, luego tiene varios usos por otros módulos.

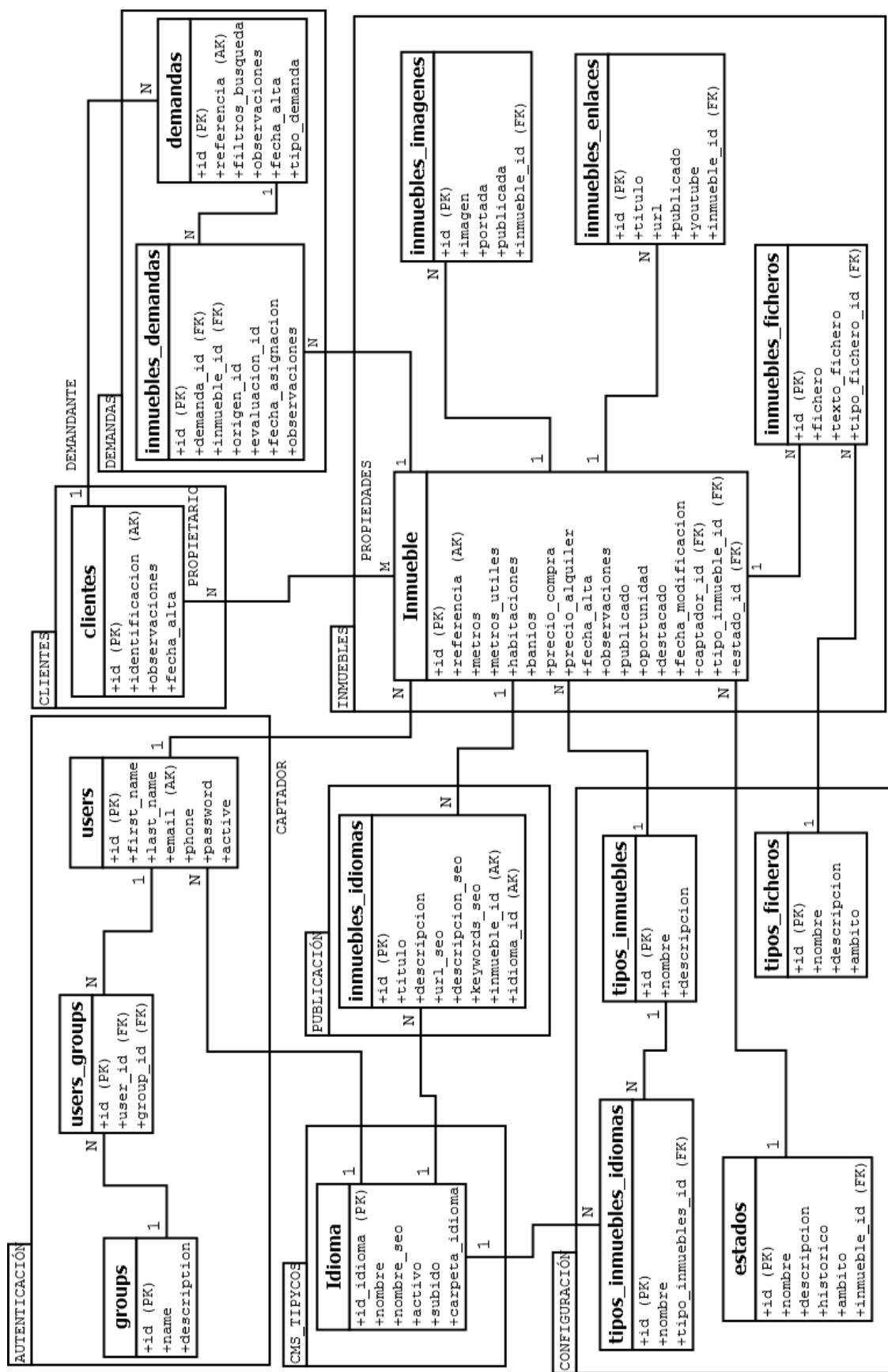


Figura 5.9: Interacción de módulos y composición de tablas

5.4. Diseño detallado de Componentes

No es objetivo de la memoria representar los comportamientos internos de los componentes externos. No obstante, me gustaría añadir este diagrama sobre el proceso que se realiza al acceder a la zona privada, el cual es bastante representativo de la complejidad que entrañan algunas funcionalidades heredadas de estos componentes.

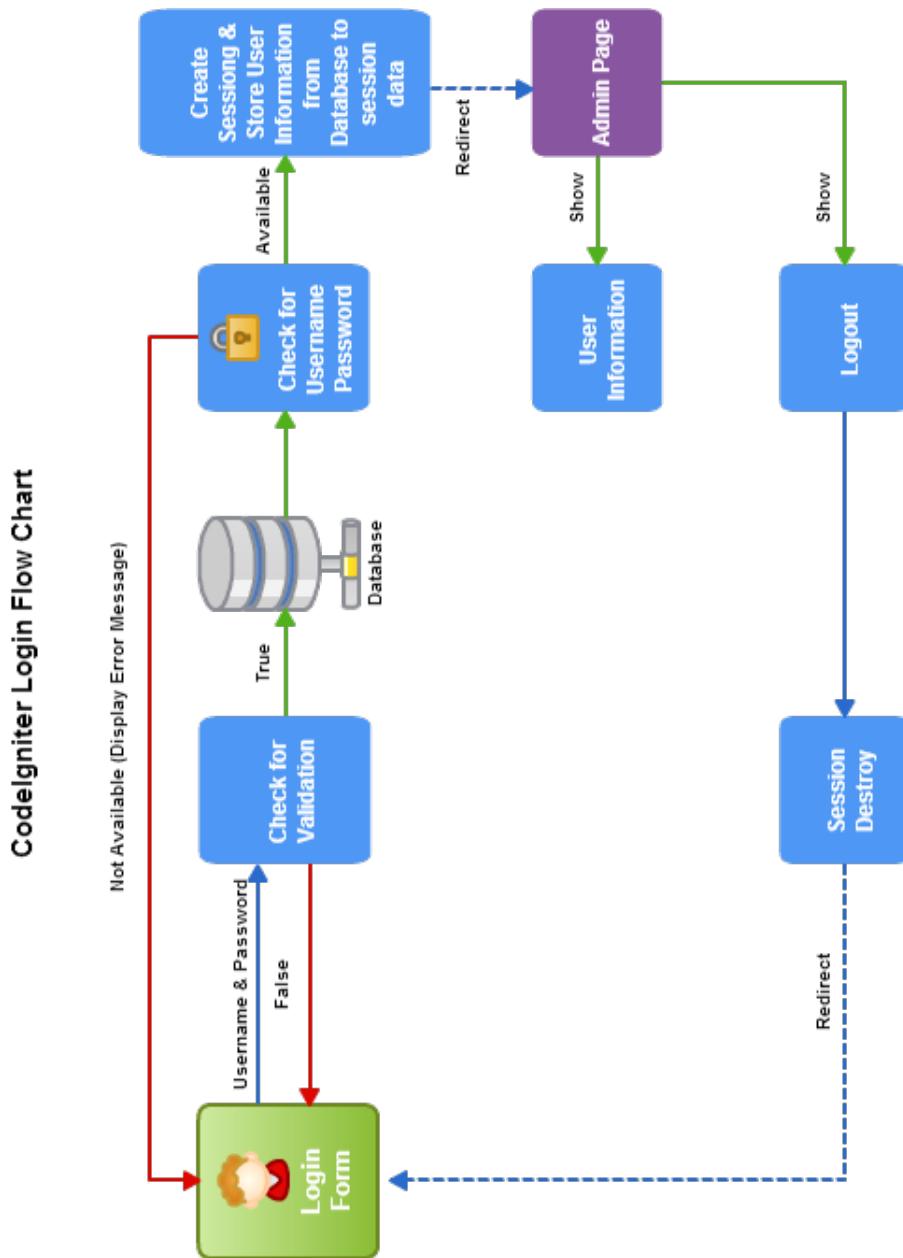


Figura 5.10: Diagrama de acceso al sistema

Por otro lado, a continuación se ofrecen algunos diagramas de secuencia representativos para ilustrar como interactúan los elementos del software. La representación de estos,

dependiendo del caso de uso, tendrá un nivel de abstracción mayor o menor y que podrían contener detalles relacionados con el propio framework de desarrollo.

Para simplificar la implementación de muchas funcionalidades, se han creado plantillas de modelos y controladores con algunas operaciones que serán de uso común en muchas ocasiones. En la siguiente figura se ilustra como funcionaría el escenario normal del caso de uso 'Insertar cliente' adaptando la plantilla mencionada.

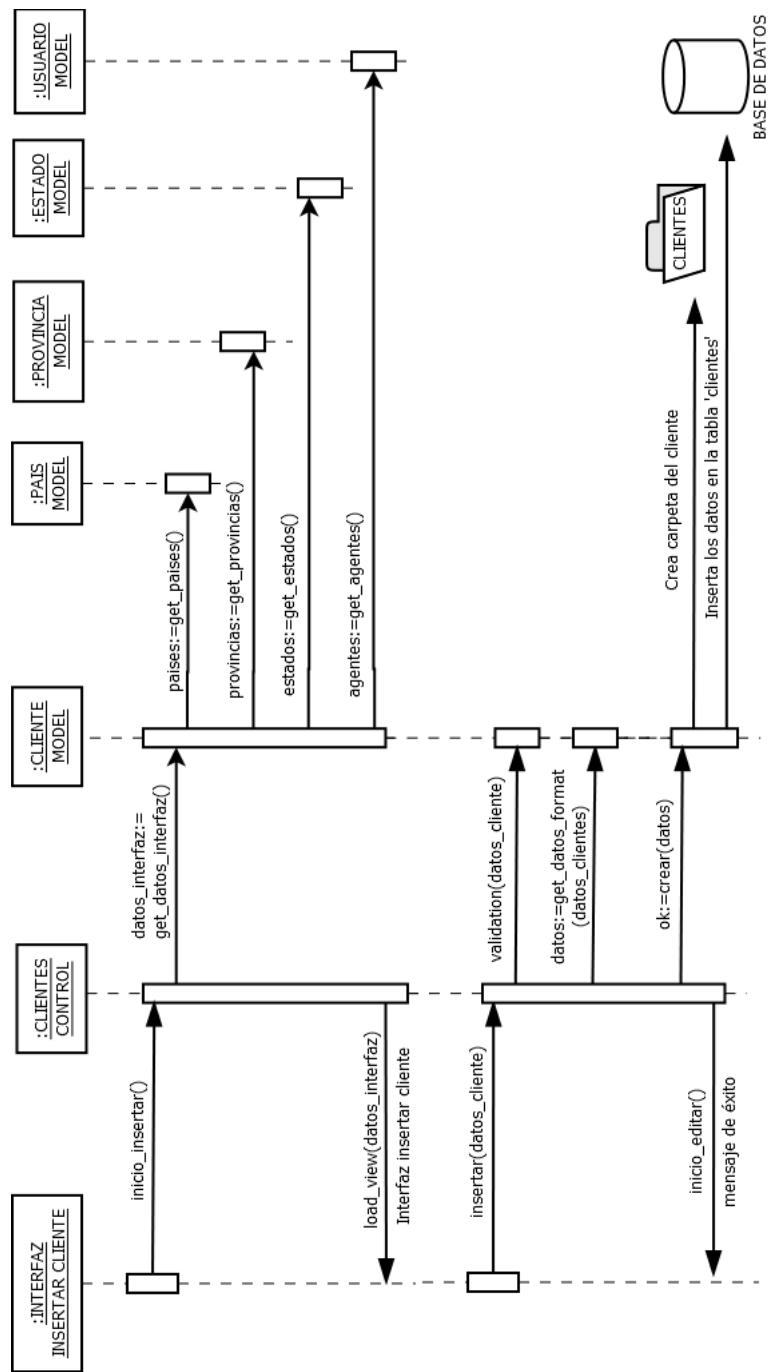


Figura 5.11: Diagrama de secuencia - Insertar Cliente (Flujo normal)

En la siguiente figura se ilustra como funcionaría el escenario de error del caso de uso 'Insertar cliente'.

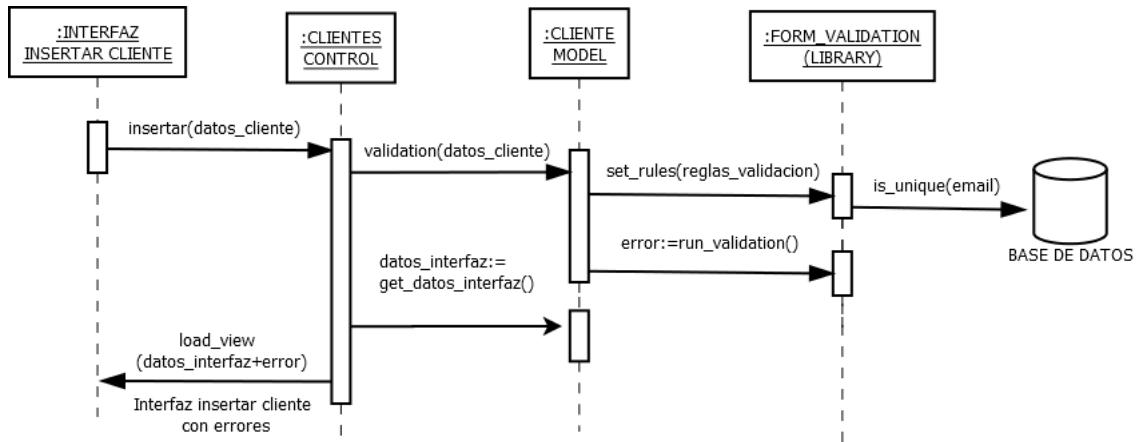


Figura 5.12: Diagrama de secuencia - Insertar Cliente (Error)

Estos dos diagramas son un ejemplo bastante completo de lo que hace el sistema en la mayoría de casos de uso, ya que mucho de su comportamiento se puede definir (con muchos detalles diferentes en cada uno de ellos) como operaciones CRUD.

Como se puede apreciar, siempre va a seguir una misma secuencia (aunque con variaciones, dependiendo del caso de uso):

- Inicio del caso de uso: Será un evento cuyo objetivo es cargar la interfaz inicial del caso de uso, sobre la cual se van a recibir las acciones del usuario. En este caso en concreto es `inicio_insertar()`. Además, para mostrar la interfaz de un cliente es necesario que consulte los diferentes listados que formarán los diferentes select del formulario; Agentes Inmobiliarios, países, provincias y los diferentes estados en los que se encuentra el cliente.
- Acción de datos: Será un evento encargado de realizar una determinada operación en el sistema. En nuestro caso `insertar()`, cuya responsabilidad es la de registrar al usuario en el sistema y que normalmente suele constar de 3 partes:
 1. **Validación:** Se encarga de validar los datos suministrados por el usuario a través de la interfaz. En este ejemplo la representa el método `validation()`. En Codeigniter se suele usar, aunque no es obligatorio, una librería para validar datos de formularios llamada `Form_Validation`. Esta permite definir un conjunto de reglas parametrizables que pueden servir, en muchos casos, para realizar

comprobaciones (incluyendo el acceso a la base de datos) de manera rápida y limpia. Otras veces deberemos definir operaciones dentro del propio modelo que realicen las comprobaciones oportunas y manejas los errores de forma conjunta.

2. **Formatear datos:** Una vez los datos han sido validados hay que adaptar su formato a la representación que tenga la base de datos de los mismos. Esto lo hacemos a través de la operación `get_datos_format()`. Esta fase a veces se realiza junto con la operación del sistema y en este caso, se realiza fuera para aportar mayor flexibilidad, ya que hay otros casos de uso (como importar CSV clientes) que utilizan la siguiente parte (el registro del cliente) pero que tienen una interfaz para obtener los datos y formatearlos diferente, ya que no provienen de un formulario si no de un fichero de datos.
3. **Operación en sistema:** Normalmente suele ser una operación CRUD, en este caso de ejemplo será `crear()`, la cual se encargará de crear la carpeta de datos en el sistema de ficheros (en algunos casos se podría emplear la librería Upload cuando se requieran trabajar con subida de fichero). En el ejemplo anterior se realizará sobre la zona de clientes. Finalmente, se emplearán operaciones de capas superiores para registrar los datos formateados anteriormente en la base de datos, en este caso, en la tabla clientes.

A continuación, se muestra el diagrama de secuencia correspondiente al caso de uso 'Mostrar mapa de inmuebles', el cual se encargará de desplegar en un mapa de google todos los inmuebles que se hayan encontrado previamente con el caso de uso 'Buscar inmuebles'. Este diagrama ya viene parametrizado para determinar si se va a visualizar un mapa público o privado de los inmuebles, en cuyo caso, la única diferencia radicaría en la operación `get_unique_markers()`, que es la encargada de formatear la información de los inmuebles y convertirla en las diferentes marcas (markers) que aparecerán en el mapa.

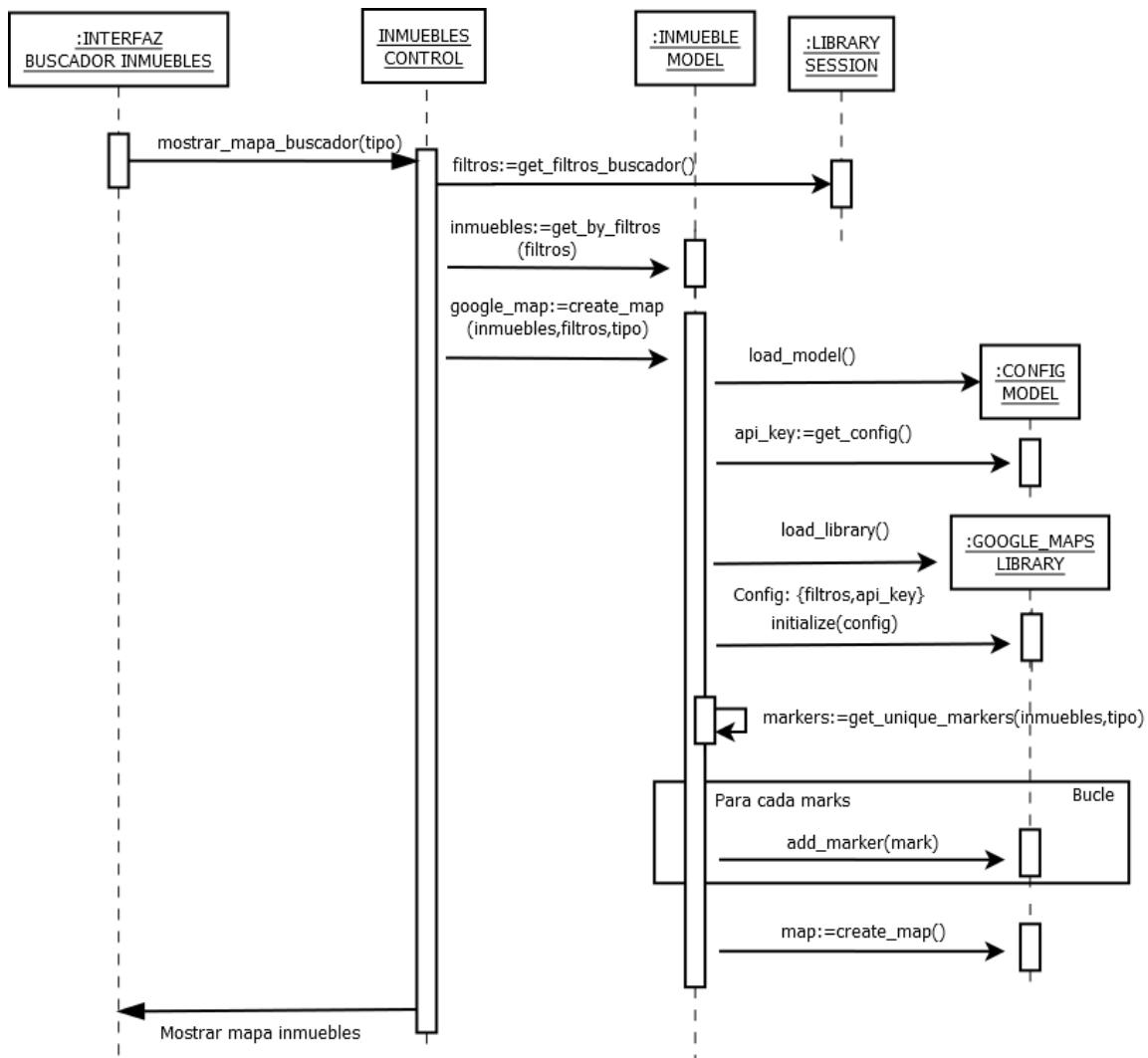


Figura 5.13: Diagrama de secuencia - Mostrar mapa de inmuebles de buscador

Para finalizar, me gustaría añadir uno de los primeros diagramas que hice de esta parte, una mezcla entre diagrama de clases y de estado que ofrece una perspectiva sobre la interacción que debería seguir el proceso para generar cualquier documento del sistema.

Aunque esté simplificado y no represente la arquitectura real 100 %, me sirvió para saber por donde deberían ir los tiros y la forma en la que debería de implementarlo.

Sobre todo se pretende establecer una arquitectura que sirva de base, no sólo para la documentación actual, si no para que en el futuro sea fácil añadir nuevos tipos de documentos.

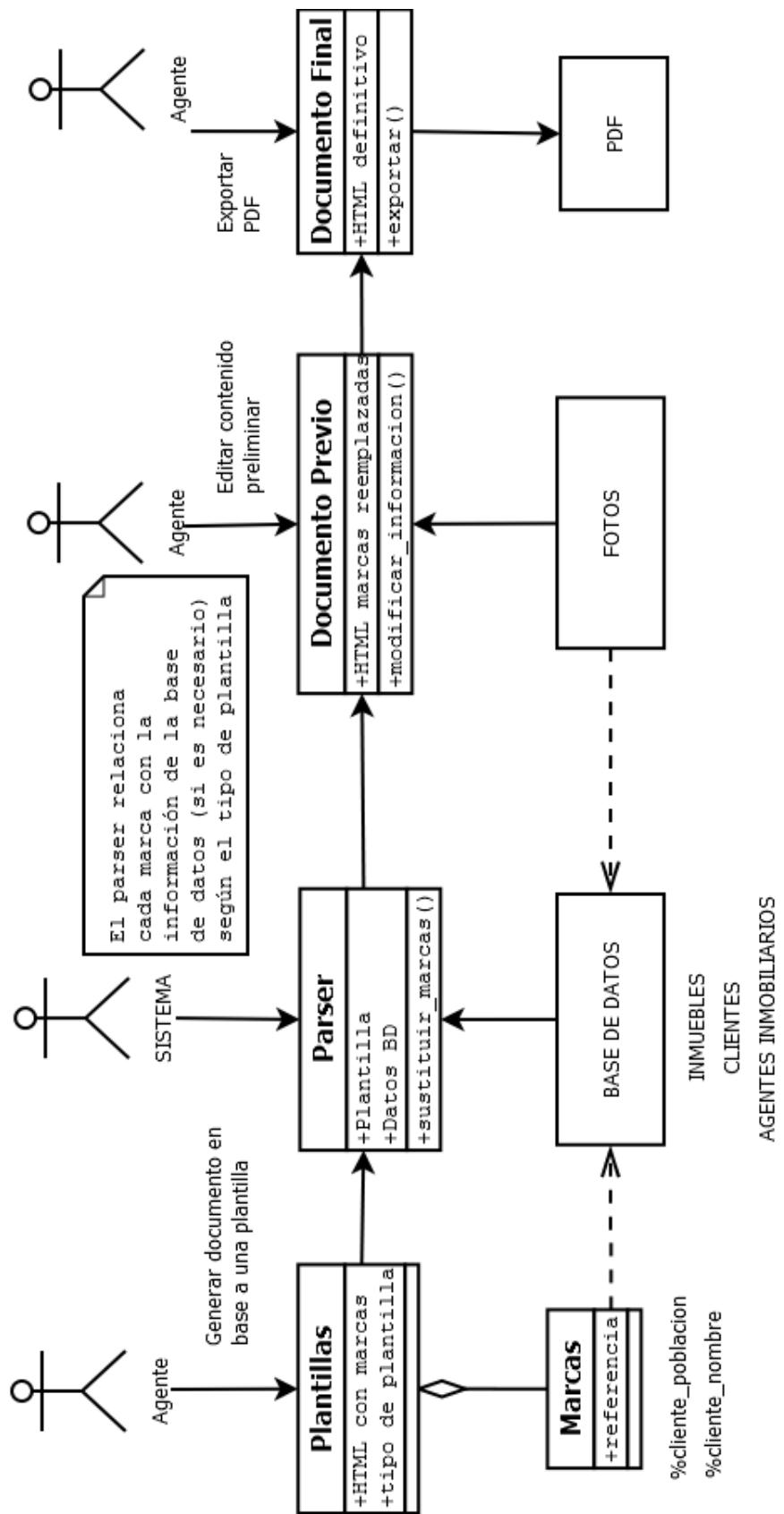


Figura 5.14: Diagrama de clases sobre el proceso de generación de Documentación

5.5. Diseño detallado de la Interfaz de Usuario

En esta sección se detallarán las interfaces entre el sistema y el usuario, incluyendo un prototipo de alta fidelidad con el diseño de la IU. Se definirá el comportamiento de las diferentes pantallas, indicando qué ocurre en los distintos componentes visuales de la interfaz cuando aparecen y qué acciones se disparan cuando el usuario trabaja con ellas.

Debido a la gran cantidad de interfaces, sólo mostraremos un conjunto representativo de estas ya que todas llevan el mismo diseño base y los mismos criterios usabilidad.

| Nombre Completo | CIF/NIE/NIF | Provincia | Municipio | Dirección | Teléfono | E-mail | Fecha alta | Ofe. | Dem. | Opciones |
|--------------------------|-------------|-----------|-----------|-------------------------------------|---------------------|---------------------------|------------|------|------|----------|
| Berasuain Ruiz, Angel | 75777802R | Almería | Almería | Calle pinito del Oro, número 4, 6 C | 956078391-659163979 | angel.berasuain@gmail.com | 26/05/2017 | 4 | 3 | |
| Calvante Diaz, Antonio | 00346592M | Almería | Almería | Calle Obispo Orberá | - | sdasd@asasd.com | 02/06/2017 | 0 | 0 | |
| Dahl, Jonh | 33446605M | | copenague | | 956078391-659163979 | pepito2@perez.com | 25/05/2017 | 1 | 0 | |
| De Benito Diaz, Rosa | 15092216F | Cádiz | Algar | Iglesia | | rosa.benito@test.com | 08/07/2017 | 0 | 0 | |
| Escalante Pérez, Antonio | 21017477P | Cádiz | Cádiz | Avenida marconi, 1 | | test@fechanac.com | 02/06/2017 | 0 | 1 | |

Figura 5.15: Buscador de clientes - Cargar Poblaciones

| Nombre Completo | CIF/NIE/NIF | Provincia | Municipio | Dirección | Teléfono | E-mail | Fecha alta | Ofe. | Dem. | Opciones |
|------------------------|-------------|-----------|-----------|-------------------------------------|---------------------|---------------------------|------------|------|------|----------|
| Berasuain Ruiz, Angel | 75777802R | Almería | Almería | Calle pinito del Oro, número 4, 6 C | 956078391-659163979 | angel.berasuain@gmail.com | 26/05/2017 | 4 | 3 | |
| Calvante Diaz, Antonio | 00346592M | Almería | Almería | Calle Obispo Orberá | - | sdasd@asasd.com | 02/06/2017 | 0 | 0 | |

Figura 5.16: Buscador de clientes - Mostrar Calendario

El buscador constará de diferentes elementos interactivos:

- Al hacer click en el select de provincias se disparará la acción 'cargar municipios' sobre el controlador Common. Esta llamada se realizará a través de AJAX para recargar el select de poblaciones dinámicamente sin tener que refrescar la interfaz.

- Al hacer click en cualquiera de los campos de fecha de alta, se desplegará un calendario marcando la fecha actual y la seleccionada por el usuario.

Por otro lado, dispondrá de varios botones que ejecutan diferentes casos de uso:

- Al hacer click en el botón 'Exportar CSV' se disparará la acción 'exportar CSV' sobre el controlador Clientes. Como consecuencia se devolverá un CSV para que el usuario lo abra o lo descargue.
- Al hacer click en el botón 'Importar CSV' se disparará la acción 'inicio importar CSV' sobre el controlador Clientes cargando la interfaz para seleccionar el fichero CSV a importar.
- Al hacer click en el botón 'Insertar' se disparará la acción 'inicio insertar cliente' sobre el controlador Clientes cargando la interfaz 'Insertar Cliente'.

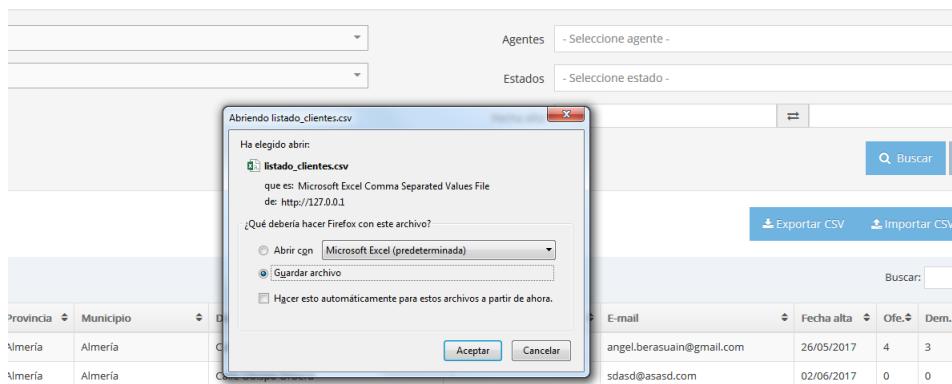


Figura 5.17: Clientes - Exportar CSV

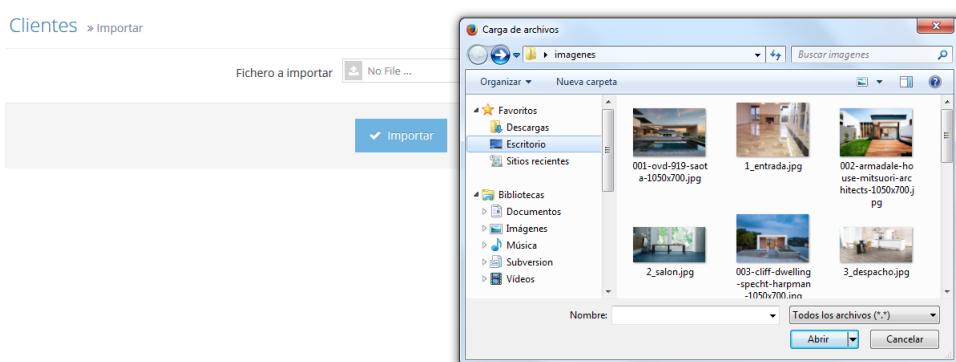


Figura 5.18: Clientes - Importar CSV

A continuación, mostramos como debería quedar una visualización responsive en una tablet de la interfaz de Insertar Cliente:

Clientes » Insertar

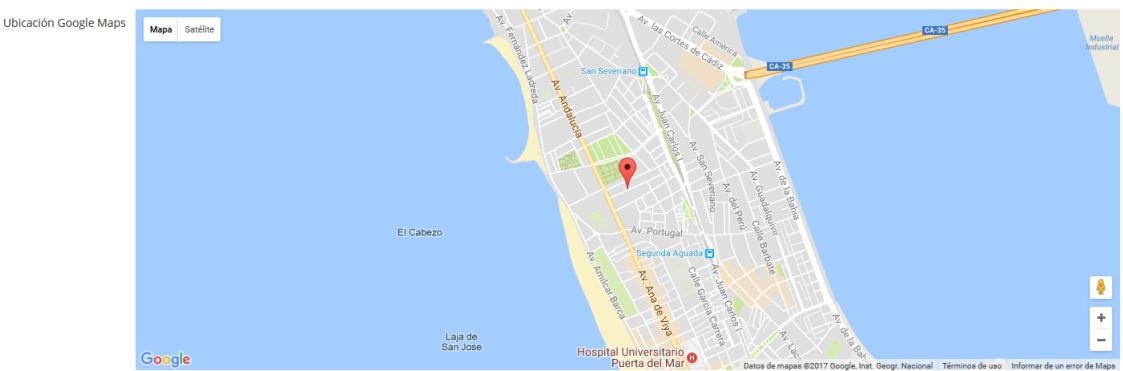


| | |
|---|---|
| DATOS PERSONALES | |
| NIF/NIE/CIF | <input type="text"/> |
| Nombre | <input type="text"/> |
| Apellidos | <input type="text"/> |
| Fecha de nacimiento | <input type="text"/> |
| Introduzca la fecha en formato dd/mm/aaaa (por ejemplo; 19/05/1982) | |
| CONTACTO | |
| País de residencia | <input type="text" value="España"/> |
| Provincia | <input type="text" value="- Seleccione provincia -"/> |
| Población | <input type="text" value="- Seleccione población -"/> |
| Dirección | <input type="text"/> |
| E-mail | <input type="text"/> |
| Teléfonos | <input type="text"/> |

Figura 5.19: Clientes - Interfaz Insertar

| | |
|--------------------|---|
| País de residencia | <input type="text" value="España"/> |
| Provincia | <input type="text" value="Cádiz"/> |
| Población | <input type="text" value="Cádiz"/> |
| Dirección | <input type="text" value="Calle Huerta del obispo"/> |
| E-mail | <input type="text" value="antonio.borrell@correo.com"/> |
| Teléfonos | <input type="text" value="956010203"/> |

Ubicación Google Maps



The map displays a coastal area with several streets labeled in Spanish. A red marker indicates the exact location of the address entered in the form. The map also shows the coastline, a bridge, and various landmarks.

Figura 5.20: Clientes - Visualizar dirección en Google Maps

Cuando se termina de cumplimentar el campo de la dirección del cliente, se disparará el evento 'visualizar dirección en Google Maps' sobre el controlador Common. Esta llamada se realizará a través de AJAX para recargar el mapa de Google Maps sin tener que refrescar

la interfaz.

Si el usuario hace click sobre el botón 'Insertar' se disparará la acción 'insertar cliente' sobre el controlador Clientes, disparando la acción 'inicio editar cliente' mostrando la interfaz 'Editar Cliente' con el mensaje de confirmación de operación con éxito:

The screenshot shows a web-based application interface for managing clients. At the top, there is a navigation bar with three tabs: 'DATOS DEL CLIENTE' (selected), 'FICHEROS ADJUNTOS', and 'FICHA DEL CLIENTE'. Below the navigation bar, the main content area has a title 'Datos del cliente > Editar'. A green success message box displays '- Atención - Elemento insertado'. Underneath, there are three tabs: 'INMUEBLES DEMANDADOS' (selected), 'DEMANDAS', and 'PROPIEDADES'. A note below the tabs states: 'Actualmente no hay inmuebles demandados por el cliente actual'. The 'DATOS PERSONALES' section contains fields for NIF/NIE/CIF (88406917T), Nombre (Antonio), Apellidos (Borrell Sánchez), and Fecha de nacimiento (01/01/1981). A placeholder text at the bottom of the personal data section reads: 'Introduzca la fecha en formato dd/mm/aaaa (por ejemplo; 19/05/1982)'.

Figura 5.21: Clientes - Insertar con éxito

Si existieron errores, mostrará los mensajes de errores en la propia interfaz de 'Insertar Cliente' recuperando los valores de los campos introducidos en la interfaz:

The screenshot shows the 'Insertar' (Insert) page for clients. The main content area features a red error message box with the following text:
- Atención -
El campo NIF/NIE/CIF es obligatorio.
El campo Nombre es obligatorio.
El campo Apellidos es obligatorio.
El campo Correo electrónico es obligatorio.
El campo Población es obligatorio.
El campo Provincia es obligatorio.
El campo Estado es obligatorio.

Figura 5.22: Clientes - Insertar con errores

Capítulo 6

Construcción del Sistema

Este capítulo trata sobre todos los aspectos relacionados con la implementación del sistema en código, haciendo uso de un determinado entorno tecnológico.

6.1. Entorno de Construcción

6.1.1. Frameworks

Bootstrap 3



Figura 6.1: Logo Bootstrap

Bootstrap [33] es un framework de frontend o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web basado en Flat Design. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript adicionales.

Bootstrap tiene un soporte relativamente incompleto para HTML 5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe

un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso.

Para la zona pública se ha empleado el framework de bootstrap 3.2 sin modificaciones, tal como el CMS de TIPYCOS traía originalmente. En cambio, la zona privada usa una plantilla llamada Admin Ace template v1.3.3 [31] que utiliza esta versión de bootstrap, concretamente la 3.3.1. Esta plantilla dispone de multitud de componentes integrados así como una interfaz predefinida para sitios webs basados en administración que, además de dar un aspecto muy profesional al sistema, te permite tener un conjunto de componentes listos para usar y con muchos ejemplos on-line que puedes tomar como referencia.

Materialize 0.97



Figura 6.2: Logo Materialize

Materialize [34] es un marco de trabajo para interfaces web moderno, responsive y basado en Material Design. Creado y diseñado por Google, Material Design es un lenguaje de diseño que combina los principios clásicos del diseño exitoso junto con la innovación y la tecnología. El objetivo de Google consiste en desarrollar un sistema de diseño que permite una experiencia de usuario unificada a través de todos sus productos en cualquier plataforma.

Hay un artículo [35] en el que se explica con detalle la diferencia entre Flat y Material Design y compara estas dos tendencias.

En este proyecto en concreto, sólo se ha empleado para la zona de login de la plataforma ya que fue el primero en evaluarse y las primeras funcionalidades implementadas. Posteriormente, debido a que el CMS de TIPYCOS emplea Bootstrap y la demanda actual del mercado se centra en este tipo de framework, decidí usarlo para practicar su desarrollo. No obstante, como reconocimiento a que he intentado al menos iniciarme en Materialize se han

dejado estas interfaces. Además, en un futuro, debido a la tendencia actual del mercado, es posible que tenga cada vez mayor relevancia y puedan crearse en el futuro plantillas para la zona pública que la usen.

Codeigniter 3.1



Figura 6.3: Logo CodeIgniter

CodeIgniter [32] es un framework para personas que construyen sitios web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si estuviera escribiendo código desde cero, proporcionando un rico conjunto de librerías para tareas comúnmente necesarias, así como una interfaz sencilla y una estructura lógica para acceder a estas librerías. CodeIgniter le permite concentrarse creativamente en su proyecto al minimizar la cantidad de código necesario para una tarea determinada. CodeIgniter ha sido elegido para este proyecto por los siguientes motivos:

1. Posee un rendimiento excepcional, sobre todo al renderizar interfaces.
2. Soporta una amplia compatibilidad con los diferentes hosting del mercado que ejecutan una variedad de versiones y configuraciones de PHP.
3. Necesita prácticamente una configuración casi cero para empezar a trabajar.
4. No hace falta utilizar la consola de comandos.
5. No posee unas reglas de codificación restrictivas. Es muy flexible en cuanto a cómo puedes implementar tus necesidades.
6. No necesita bibliotecas monolíticas a gran escala como PEAR.
7. No necesita un lenguaje de plantilla (aunque un analizador de plantillas está disponible opcionalmente si lo desea).
8. Se centra en evitar la complejidad del diseño, favoreciendo soluciones simples.

9. Ofrece documentación clara y completa.

Sobre todo, el haber encontrado un módulo que proporcione las funcionalidades básicas de un CMS, así como un sistema de autenticación completo y fácil de integrar, fueron puntos clave en su elección.

jQuery



Figura 6.4: Logo Jquery

jQuery [36] es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Las principales características que la definen son:

1. Selección de elementos DOM.
2. Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
3. Eventos.
4. Manipulación de la hoja de estilos CSS.
5. Efectos y animaciones.
6. Animaciones personalizadas.
7. AJAX.

8. Soporta extensiones.
9. Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
10. Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.5.

Las versiones empleadas varían dependiendo del framework utilizado 1.10.1 para la plantilla de la zona pública de Bootstrap, 1.11.1 para la plantilla de Bootstrap de la zona privada, y 2.1.1 para Materialize.

6.1.2. Lenguajes de programación

Los principales lenguajes de programación utilizados para el desarrollo del sistema son: HTML 5, CSS3, PHP 5.5.12, Javascript y la versión de SQL implementada por el sistema de gestión de bases de datos relacional MySQL 5.6.17.

HTML 5



Figura 6.5: Logo HTML 5

HTML 5 es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML 5 especifica dos variantes de sintaxis para HTML: una clásica, HTML, conocida como HTML 5, y una variante XHTML conocida como sintaxis que HTML y XHTML se han desarrollado en paralelo. La versión definitiva de la quinta revisión del estándar se publicó en octubre de 2014.

Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se recomienda al usuario común actualizar su navegador a la versión más nueva, para poder

disfrutar de todo el potencial que provee HTML 5. El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C.

PHP 5.5



Figura 6.6: Logo PHP 5.5

PHP es un lenguaje de programación **del lado del servidor**, diseñado para desarrollo de aplicaciones web de contenido dinámico. Se ha escogido este lenguaje de programación, por los siguientes motivos:

- Es un lenguaje totalmente libre y abierto.
- Tiene un gran parecido con los lenguajes más comunes de programación estructurada tales como C o Perl, ambos estudiados en la carrera.
- Gran cantidad de recursos y documentación existentes.
- Fácil acceso a las bases de datos. Buena integración con MySQL y Oracle.
- Funciona sobre prácticamente todas las plataformas y garantiza una alta velocidad de ejecución.

En resumen, es un lenguaje flexible, versátil, libre y multiplataforma.

CSS 3



Figura 6.7: Logo CSS 3

CSS es un lenguaje que sirve para definir el estilo o apariencia de las páginas web, escritas con HTML (en este caso XHTML). Su principal objetivo es separar el contenido de la forma, y permitir a los diseñadores mantener un control más preciso sobre la apariencia de las páginas.

Este lenguaje, ha evolucionado hasta su última versión CSS3, la cual se ha utilizado para desarrollar esta aplicación. Esta versión, introduce aún más mecanismos de control sobre los elementos de las páginas web que los que ofrecían las versiones 1 y 2.

Javascript

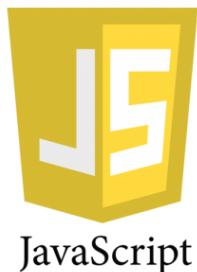


Figura 6.8: Logo JavaScript

Javascript es un lenguaje interpretado (no requiere compilación), utilizado principalmente en páginas web, con una sintaxis semejante a la de los lenguajes Java y C. JavaScript a diferencia de PHP, es un lenguaje **del lado del cliente**, permitiendo crear efectos atractivos y dinámicos en las páginas web.

En el desarrollo de esta aplicación, JavaScript se ha utilizado en conjunción con Jquery, y ha servido para crear funciones que han ayudado a dar dinamismo a la web.

MySQL v.5.6



Figura 6.9: Logo MySQL

SQL es un lenguaje de consulta estructurada (Structured Query Language). Es un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos. Es un lenguaje estándar de comunicación con bases de datos y que permite combinar cualquier tipo de lenguaje con cualquier tipo de base de datos.

Está basado en el álgebra y el cálculo relacional y consiste en la definición, manipulación y control de datos. Su alcance incluye operaciones como inserción, consultas, actualizaciones y borrado de datos.

Para gestionar la base de datos de esta aplicación se ha escogido **MySQL**. MySQL es un sistema de gestión de base de datos relacional (SGBDR) de código abierto, basado en el lenguaje de consulta estructurado SQL. Se ejecuta con rapidez en prácticamente todas las plataformas (Windows, Linux, Unix) y se asocia normalmente con aplicaciones Web. Es un componente muy importante de los sistemas WAMP (Windows, Apache, MySQL, PHP) como lo es este proyecto.

Las principales ventajas de este SGBDR son las siguientes:

- Código abierto.
- Velocidad al realizar operaciones.
- Bajo coste en requerimientos del sistema. Puede ser ejecutado en una máquina con escasos recursos debido a su bajo consumo.
- Facilidad de instalación y configuración.
- Seguridad de los datos.

6.1.3. Herramientas de desarrollo

MySQL Workbench v6.3.6



Figura 6.10: Logo MySQL Workbench

MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL. Es el sucesor de DBDesigner 4 de fabFORCE.net, y reemplaza el anterior conjunto de software, MySQL GUI Tools Bundle.

Las principales ventajas por las que he elegido este SGBD han sido:

- Poder realizar ingeniería inversa del proyecto anterior para tener una base sobre la que empezar a modelar y no tener que hacerla desde cero.
- Con el anterior diagrama como punto de partida se han añadido todas las modificaciones (de forma manual con SQL o usando el software) pudiendo obtener diagramas E/R en diferentes formatos.
- Conectar de forma sencilla y directa con las base de datos de los diferentes servidores que se han empleado.

PHPMyAdmin v4.1.14

Aunque ha tenido un papel mas bien secundario, en tanto en cuanto, se utilizaba esporádicamente, es digno mencionarlo, ya que es una de las herramientas más populares de gestión de bases de datos construida en PHP y además viene instalada por defecto tanto con el paquete de WAMP como en el propio PLESK del servidor de STAGING y muchos otros servidores.

Git v1.9.15 y Github



Figura 6.11: Logo GIT y GITHUB

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux. El mantenimiento del software Git está actualmente (2009) supervisado por Junio Hamano, quien recibe contribuciones al código de alrededor de 280 programadores.

El diseño de Git se basó en BitKeeper y en Monotone. El diseño de Git resulta de la experiencia del diseñador de Linux, Linus Torvalds, manteniendo una enorme cantidad de código distribuida y gestionada por mucha gente, que incide en numerosos detalles de rendimiento, y de la necesidad de rapidez en una primera implementación.

Entre las características más relevantes se encuentran:

- Fuerte apoyo al desarrollo no lineal, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones. Git incluye herramientas específicas para navegar y visualizar un historial de desarrollo no lineal.
- Gestión distribuida. Al igual que Darcs, BitKeeper, Mercurial, SVK, Bazaar y Monotone, Git le da a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales. Los cambios se importan como ramas adicionales y pueden ser fusionados en la misma manera que se hace con la rama local.
- Los almacenes de información pueden publicarse por HTTP, FTP, rsync o mediante un protocolo nativo, ya sea a través de una conexión TCP/IP simple o a través de

cifrado SSH. Git también puede emular servidores CVS, lo que habilita el uso de clientes CVS pre-existentes y módulos IDE para CVS pre-existentes en el acceso de repositorios Git.

- Una presunción fundamental en Git es que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente, conforme se pasa entre varios programadores que lo revisan.
- Los repositorios Subversion y svk se pueden usar directamente con git-svn.
- Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.
- Todas las versiones previas a un cambio determinado, implican la notificación de un cambio posterior en cualquiera de ellas a ese cambio (denominado autenticación criptográfica de historial). Esto existía en Monotone.
- Resulta algo más caro trabajar con ficheros concretos frente a proyectos, eso diferencia el trabajo frente a CVS, que trabaja con base en cambios de fichero, pero mejora el trabajo con afectaciones de código que concurren en operaciones similares en varios archivos.
- Los renombrados se trabajan basándose en similitudes entre ficheros, aparte de nombres de ficheros, pero no se hacen marcas explícitas de cambios de nombre con base en supuestos nombres únicos de nodos de sistema de ficheros, lo que evita posibles, y posiblemente desastrosas, coincidencias de ficheros diferentes en un único nombre.
- Realmacenamiento periódico en paquetes (ficheros). Esto es relativamente eficiente para escritura de cambios y relativamente ineficiente para lectura si el reempaquetado (con base en diferencias) no ocurre cada cierto tiempo.

Por otro lado, se ha utilizada GitHub para ubicar este proyecto así como el proyecto anterior que se elaboró para GESTICADIZ y que ha servido como base de estudio y punto de partida para este. GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome). Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. El código se almacena de

forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago. En la bibliografía se ofrecen los enlaces directos a los repositorios de los proyectos.

SourceTree v1.6.18



Figura 6.12: Logo SourceTree

SourceTree es un cliente de Git gratuito para Windows y Mac de la empresa Atlassian. Este software simplifica la forma de interactuar con los repositorios de Git para que puedas centrarte en el código. Permite visualizar y administrar tus repositorios mediante una sencilla interfaz de usuario.

Además simplifica el control de versiones distribuidas y pone al día a todo el mundo con rapidez. Perfecto para que los usuarios avanzados sean aún más productivos. Revisa los conjuntos de cambios, realiza cambios provisionales, elige entre ramas de forma selectiva y mucho más.

Sobre todo, lo más valioso es que permite visualizar tu código en forma de árbol, pudiendo comparar y rescatar versiones con facilidad. Aunque el propio IDE de Netbeans tiene comandos para interactuar directamente con el repositorio, no ofrece todas las posibilidades, sobre todo gráficas, de este software, además que a veces, suelen fallar al ejecutarse.

WampServer v2.5



Figura 6.13: Logo WAMP Server

WampServer es un entorno de desarrollo web para Windows con el que se pueden crear aplicaciones web con Apache, PHP y bases de datos MySQL. Facilita la instalación de todas estas herramientas mencionadas y la configuración es prácticamente automática. También incluye phpMyAdmin, que es otra herramienta escrita en PHP para manejar la administración de MySQL a través de páginas web, utilizando Internet. Con ella se pueden gestionar las tablas de la base de datos, ejecutar sentencias SQL, administrar claves en campos, privilegios, exportar datos en varios formatos, etc. Las versiones de los componentes de WampServer v2.5 son las siguientes:

- Apache 2.4.9
- MySQL 5.6.17
- PHP 5.5.12
- phpMyAdmin 4.1.14

Netbeans IDE 8.1



Figura 6.14: Logo NetBeans

Netbeans es un entorno de desarrollo libre que permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles. Da soporte a las siguientes tecnologías, entre otras: Java, PHP, Groovy, C/C++, HTML5, etc. Además, puede instalarse en varios sistemas operativos: Windows, Linux, Mac, OS, etc. Es un editor de código muy completo con una interfaz amigable y fácil de entender, sugerencias de código, control de versiones, correcciones semánticas y sintácticas y muchas otras herramientas.

Netbeans facilita la gestión de proyectos grandes o complejos y cuenta con una gran comunidad aún en crecimiento. Por estos motivos se eligió para desarrollar el código de la aplicación. Aunque está escrito en Java, puede servir para cualquier otro lenguaje de pro-

gramación como es el caso para este proyecto, existiendo además numerosos módulos que permiten extender el NetBeans IDE.

Doxxygen v1.8.13



Figura 6.15: Logo Doxygen

Doxxygen es un generador de documentación para C++, C, Java, Objective-C, Python, IDL (versiones Corba y Microsoft), VHDL y en cierta medida para PHP. Dado que es fácilmente adaptable, funciona en la mayoría de sistemas Unix así como en Windows y Mac OS X. La mayor parte del código de Doxygen está escrita por Dimitri van Heesch. Doxygen es un acrónimo de dox(document) gen(generator), generador de documentación para código fuente. Varios proyectos como KDE usan Doxygen para generar la documentación de su API. KDevelop incluye soporte para Doxygen.

Las principales ventajas que he encontrado para su uso han sido:

- Soporte de notación PHPDoc que es la empleada para redactar los comentarios del código del proyecto.
- Múltiples formato de exportación (RTF, Latex, HTML, etc.)
- Incluye un wizard que te facilita mucho la configuración y la adaptabilidad de lo generado a tu proyecto.
- La compilación es rápida y eficiente.
- Permite generar gráficos de dependencia automáticamente tanto de las clases como de los ficheros.

Sólo se ha generado la documentación del núcleo de funcionalidades del software, es decir, la carpeta correspondiente a la aplicación OPENRS. Se han excluido de la documentación las carpetas log, cache y language, ya que no aportan información relevante al proyecto.

Composer v1.3.2



Figura 6.16: Logo Composer

Composer es una herramienta para la gestión de dependencias en PHP. Le permite declarar las librerías de las que depende su proyecto y las administrará (instalará y actualizará). Composer no es un gestor de paquetes en el mismo sentido que Yum o Apt. Sí, se trata de paquetes o librerías, pero los administra por proyecto, instalándolos en un directorio (por ejemplo, vendor) dentro del proyecto. De forma predeterminada, no instala nada globalmente. Por lo tanto, es un administrador de dependencias. Sin embargo, admite un proyecto "global" por conveniencia a través del comando global.

Esta idea no es nueva y Composer está fuertemente inspirado otros gestores de paquetes como el de Ruby. Básicamente Composer supone que:

- Tienes un proyecto que depende de varias librerías.
- Algunas de esas librerías dependen de otras librerías.

¿Qué aporta Composer?:

- Permite declarar las librerías de las que depende el sistema.
- A su vez las dependencias de esas librerías de otras librerías.

Permite averiguar qué versiones de qué paquetes pueden y necesitan ser instalados y las instala en el sistema de información (lo que significa que los descarga en su proyecto).

Filezilla v3.27.1



Figura 6.17: Logo Filezilla

FileZilla es un cliente FTP multiplataforma de código abierto y software libre, licenciado bajo la Licencia Pública General de GNU. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS (FTPS). No hace falta mucho más que añadir porque es ampliamente conocido, no obstante, me gustaría recalcar que se ha hecho un estudio de otras herramientas como FTPPLOY o DEPLOYBOT, pero por motivos técnicos y temporales no ha sido posible implantar en este proyecto, ya que, la subida directa por FTP para realizar el despliegue de una aplicación no es una buena práctica. Aún así, como el proyecto se centra principalmente en adquirir buenas prácticas dentro de un perfil de analista y programador, las competencias propias del despliegue suelen recaer en otros profesionales y áreas como las de sistemas.

6.2. Código Fuente

En este capítulo se va a describir la organización del código fuente, así como la utilidad de los diferentes ficheros y su distribución en paquetes o directorios. Además, se incluye algún extracto significativo de código fuente que sea de interés para ilustrar alguna funcionalidad específica del sistema.

Cabe destacar que el código del proyecto es software libre y se encuentra ubicado en un espacio de trabajo en github [30]. En este espacio de trabajo se ha incluido la memoria, los PCS, las matrices de prueba, una documentación extensiva creada con DOXYGEN así como todo el código tanto de OPENRS como de OPENINMOCMS. En la documentación del código, en su versión HTML, se puede apreciar la estructura de ficheros y la utilidad y organización de cada carpeta así como de los métodos de las diferentes clases y su jerarquía.

Como ejemplo de código se va a describir la funcionalidad de insertar un cliente así como el establecimiento de permisos y funcionalidades CRUD en el constructor del controlador de clientes.

```

3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  require_once APPPATH . 'core/CRUD_controller.php';
6
7  class Clientes extends CRUD_controller
8  {
9
10     function __construct()
11     {
12         $this->_model = "Cliente_model";
13         $this->_controller = "clientes";
14         $this->_view = "clientes";
15
16         parent::__construct();
17
18         // Secure the access
19         $this->_security();
20
21         // Comprobación de acceso
22         $this->utilities->check_security_access_profiles_or(array("session_es_agente"));
23     }

```

Figura 6.18: Implementación de constructor de clientes

El controlador de clientes al implementar funcionalidades CRUD hereda del controlador diseñado para tales tareas. En este definimos tres atributos:

- `$this->_model`: El cual se encarga de indicar qué modelo debe cargarse como principal en el controlador.
- `$this->_controller`: Indica cuál va a ser el controlador principal que se va a usar como base para la navegabilidad así como la recepción de eventos.
- `$this->_view`: Indica cuál es la carpeta de views del framework donde se van a almacenar las vistas de este módulo.

Toda esta parametrización ha sido implementada manualmente, pero aunque ha costado que funcionara inicialmente, luego en muchísimos módulos del sistema ha tenido gran utilidad y ha ahorrado errores al sustituir variables y comprobar los enlaces, eventos lanzados, etc.

Posteriormente se invoca al método `_security()`, que se va a encargar de controlar que para ejecutar cualquier evento del controlador tenga que estar el usuario logueado en el sistema. Finalmente, se añade una segunda restricción de seguridad, pero esta vez por permisos de usuario, para controlar que cada perfil accede a lo que debe. En este caso, como todos los eventos que recibe este controlador se encargan de implementar funcionalidades

relacionadas con la Gestión de clientes se le indica al sistema a través de la operación check_security_access_perfiles() implementada en la librería de utilities (también de fabricación propia), que debe mirar en la variable de sesión si el usuario actual es un agente inmobiliario.

```

103 // insert
104 public function insert($inmueble_id=0)
105 {
106     // Validation
107     if ($this->is_post())
108     {
109         // Inicializamos los datos de validación para reutilizar la validación del cliente
110         $this->form_validation->set_data($this->input->post());
111         // Check
112         if ($this->{$this->_model}->validation())
113         {
114             // Formatted datas
115             $formatted_datas=$this->{$this->_model}->get_formatted_datas();
116             // Insert
117             $last_id=$this->{$this->_model}->create($formatted_datas,$inmueble_id);
118             // Check
119             if ($last_id)
120             {
121                 $this->session->set_flashdata('message', lang('common_success_insert'));
122                 $this->session->set_flashdata('message_color', 'success');
123                 redirect($this->_controller."/edit/".$last_id, 'refresh');
124             } else {
125                 $this->data['message'] = $this->{$this->_model}->get_error();
126             }
127         } else
128         {
129             $this->data['message'] = validation_errors();
130         }
131     }
132     // Set datas
133     $this->_set_datas_html(NULL,$inmueble_id);
134
135     // Render
136     $this->render_private($this->_view . '/insert', $this->data);
137 }
```

Figura 6.19: Implementación de funcionalidad de Insertar Cliente

Como podrá apreciar el lector la implementación es bastante similar al diseño detallado 5.4 realizado en la etapa anterior.

6.3. Scripts de Base de datos

Todos los índices, tablas, vistas, etc. se encuentran ubicados en un único fichero SQL. A continuación se ofrecen un par de capturas para ejemplificar cómo se han creado las tablas así como las vistas:

```
187 -- Estructura de tabla para la tabla 'clientes_fichas'
188 --
189 --
190
191 CREATE TABLE IF NOT EXISTS `clientes_fichas` (
192     `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
193     `plantilla_id` int(11) unsigned NOT NULL,
194     `agente_id` int(11) unsigned NOT NULL,
195     `html` text NOT NULL,
196     `fecha` date NOT NULL,
197     `cliente_id` int(11) unsigned DEFAULT NULL,
198     PRIMARY KEY (`id`),
199     KEY `FK_clientes_fichas_agente_id` (`agente_id`),
200     KEY `FK_clientes_fichas_plantilla_id` (`plantilla_id`),
201     UNIQUE KEY `FK_clientes_fichas_cliente_id` (`cliente_id`)
202 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
203
204 --
205 -- Filtros para la tabla 'clientes_fichas'
206 --
207 ALTER TABLE `clientes_fichas`
208     ADD CONSTRAINT `FK_clientes_fichas_agente_id` FOREIGN KEY (`agente_id`) REFERENCES `users` (`id`) ON UPDATE CASCADE,
209     ADD CONSTRAINT `FK_clientes_fichas_cliente_id` FOREIGN KEY (`cliente_id`) REFERENCES `clientes` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
210     ADD CONSTRAINT `FK_clientes_fichas_plantilla_id` FOREIGN KEY (`plantilla_id`) REFERENCES `plantillas_documentacion` (`id`) ON UPDATE CASCADE;
211
```

Figura 6.20: Creación de tabla de fichas de los clientes

```
32 -- Vista inmuebles
33 CREATE
34     OR REPLACE
35 VIEW `v_inmuebles` AS
36 SELECT
37     inmuebles.*,
38     poblaciones.poblacion AS nombre_poblacion,
39     poblaciones.provincia_id AS provincia_id,
40     provincias.provincia AS nombre_provincia,
41     poblaciones_zonas.nombre AS nombre_zona,
42     tipos_inmueble_idiomas.nombre AS nombre_tipo,
43     tipos_inmueble_idiomas.idioma_id AS idioma_id,
44     tipos_certificacion_energetica.nombre AS nombre_certificacion_energetica,
45     estados.nombre AS nombre_estado,
46     CONCAT_WS(' ', users.last_name, users.first_name) AS nombre_captador,
47     inmuebles_carteles.id AS cartel_id
48     inmuebles_carteles.impreso AS cartel_impresso
49 FROM
50     inmuebles
51     JOIN estados ON inmuebles.estado_id = estados.id
52     JOIN poblaciones ON inmuebles.poblacion_id = poblaciones.id
53     JOIN provincias ON poblaciones.provincia_id = provincias.id
54     LEFT JOIN poblaciones_zonas ON inmuebles.zona_id = poblaciones_zonas.id
55     JOIN tipos_inmueble ON inmuebles.tipo_id = tipos_inmueble.id
56     JOIN tipos_inmueble_idiomas ON tipos_inmueble_idiomas.tipo_inmueble_id = tipos_inmueble.id
57     LEFT JOIN tipos_certificacion_energetica ON inmuebles.certificacion_energetica_id = tipos_certificacion_energetica.id
58     LEFT JOIN users ON inmuebles.captador_id = users.id
59     LEFT JOIN inmuebles_carteles ON inmuebles_carteles.inmueble_id = inmuebles.id;
```

Figura 6.21: Creación de vista de inmuebles

6.4. Buenas prácticas

Uno de las cuestiones que debe plantearse (o al menos intentarlo como yo he hecho con este proyecto) todo desarrollador a la hora de construir software es utilizar buenas prácticas o hábitos que no sólo son particulares de este proyecto si no pueden reutilizarse para otros. Merece mención a parte tenerlas en consideración respecto al resto de secciones de esta parte ya que afecta a múltiples ámbitos.

6.4.1. Codeigniter

Una de las principales fuentes de inspiración fue el tutorial de Joost van Veen [42], el cual fue ampliado con otras prácticas. A continuación se muestra un resumen de algunas partes del tutorial que se han implementado parcial o totalmente:

- Estructura y organización de código:
 - Capas explicadas en diseño.
 - Configuración de entornos por carpetas.
- Depuración y testing:
 - Profiler.
 - Testing library.
 - Firebug.
- Thin controllers, Fat models.
- Seguridad:
 - Almacenamiento de sesión en base de datos.
 - Encriptación de cookie.
 - Cambio de nombre de cookie y carpeta application.
 - Protección ante ataques XSS and CSRF.
 - ION AUTH (Password hashing, login attempts, expire session time, role access control, etc.).
 - Errores guardados en ficheros de logs (ocultación de errores).
 - Sanitization de entrada de datos con HTML.

- Usar `get_env()` en configuraciones para no desvelar contraseñas.
- Sesiones.
- Composer (vendor).
- Caché.

Además, se ha seguido la guía de estilo PHP tal como aconseja el framework en su web [32].

6.4.2. Bases de datos

En general, se han tomado en consideración algunas prácticas [43] (Usar EXPLAIN, habilitar la caché, Añadir índices autoincrementales, poner `limit 1` con resultados únicos, etc.).

Además se han utilizado otras como:

- Librería PDO para conexiones simples.
- Librería CRUD [27] (Un modelo-Una tabla-Una vista).
- Migraciones usando los SQLs de los PCS.

6.4.3. Otras prácticas

Otras prácticas que se han utilizado:

- Comprobación de precondiciones de casos de uso.
- Comprobaciones de concurrencia.
- Uso de otros recursos del Framework (autoloading, manejo de emails, ficheros, etc.)
- Patrones de diseño (layout, singleton, mvc...)
- Subida de ficheros comprobando el tamaño límite establecido en el servidor.
- Encriptación de nombre de ficheros adjuntos.
- Carpetas y ficheros creados con permisos adecuados.
- Corrección de algoritmo para obtención de IP.

- Criterio de mínima cantidad de información almacenada (adjuntos e imágenes, la documentación generada se almacena en HTML en la base de datos y se exporta cuando uno desea).
- Como se verá en el Manual de implantación y explotación, el despliegue y configuración de la aplicación es flexible y sencillo.
- Uso de htaccess para tener urls amigables y mejorar seguridad.
- Buenas prácticas implementadas en CMS TIPYCOS (Posicionamiento SEO usando keywords y descripciones, urls amigables, etc.).
- Uso de rutas relativas (para almacenamiento de ficheros, para la información contenida en los HTML, etc.)

Todas ellas basadas en la experiencia personal y en la guía general sobre mejores prácticas en PHP [41].

Capítulo 7

Pruebas del Sistema

En este capítulo se presenta el plan de pruebas del sistema de información, incluyendo los diferentes tipos de pruebas que se han llevado a cabo, ya sean manuales (mediante listas de comprobación) o automatizadas mediante algún software específico de pruebas.

7.1. Estrategia

Las pruebas pretenden comprobar, en la medida de lo posible, que el sistema funciona correctamente en cada capa de diseño, así como en cada nivel de abstracción y que se comunica correctamente con los componentes externos. Sobre todo, debido al gran volumen de componente utilizados, tanto externos como internos al framework, las pruebas llevadas a cabo para asegurar la efectiva integración de un componente en el sistema son las que han tenido más peso en todo el proceso de pruebas.

Las pruebas unitarias y de integración han sido codificadas, mientras que para el resto de pruebas se han utilizado otros métodos que se describen en los apartados correspondientes. Principalmente utilizando el navegador y pasando una lista de comprobación. Dado la cantidad de componentes y de funcionalidades implementadas, no todas las pruebas van a ser registradas y se realizarán sobre la marcha.

Los resultados y sus interpretaciones se evalúan en las matrices de prueba o en las descripciones de código de aquellas que han sido implementadas siguiendo los estándares que marca el framework. Adicionalmente, queda por depurar más exhaustivamente la integración del CMS de TIPYCOS debido a su complejidad y lo difícil que resulta la adaptabilidad de la interfaz a los diferentes dispositivos.

7.2. Entorno de Pruebas

Las pruebas se realizarán a través del propio navegador y dispositivo utilizado para manejar el sistema por parte del usuario. Dependiendo del nivel de la prueba, algunas utilizarán la librería de testing del propio framework y otras se realizarán a través de listas de verificación. Por motivos de tiempo, no se han utilizado componentes externos como PHPUNIT (para pruebas unitarias principalmente o de integración) o Selenium (para pruebas funcionales), también se valoró el uso JMETER para realizar pruebas de concurrencia y de estrés, pero, al igual que con las anteriores, se descartó finalmente su uso. No obstante todas estas se valorarán para futuras ampliaciones del proyecto. Algunas de las herramientas que se emplearon son:

- Firebug: plug-in para el navegador Mozilla que proporciona múltiples herramientas para desarrolladores.
- Profiler: CogIgniter lleva integrado una librería para desplegar estadísticas durante la ejecución de una llamada url como pueden ser las consultas a la base de datos realizadas, los valores de las variables POST y GET, etc.
- Librería testing: El framework proporciona una pequeña librería para poder comprobar tipos y valores, algo muy simple pero que suele ser efectivo para pruebas unitarias.

7.3. Roles

Para realizar las pruebas han participado:

- El desarrollador del sistema.
- Usuarios con perfiles de demandantes (para evaluar la zona pública).
- Los desarrolladores de la empresa TIPYCOS para comprobar que el CMS y su integración, sobre todo, en la zona pública, funcionan correctamente. Además han probado la portabilidad del sistema en sus equipos locales.
- Un agente inmobiliario de GESTICADIZ.
- Un colaborador externo de GESTICADIZ.

7.4. Niveles de Pruebas

7.4.1. Pruebas Unitarias

Las pruebas unitarias tienen por objetivo localizar errores en cada nuevo artefacto software desarrollado, antes que se produzca la integración con el resto de artefactos del sistema. Para realizar estas pruebas se han seguido dos métodos principalmente:

- Programación directa a través de controladores de test usando la librería de testing.
- Inclusión de marcas en el código (normalmente utilizando las instrucciones ECHO o VARDUMP) para depurar resultados o ejecuciones.

Este último método se ha utilizado mucho también en las pruebas de integración ya que permite verificar los resultados en determinados puntos de los algoritmos implementados. También cabría mencionar el uso del propio PROFILER para aquellas operaciones que interactuaban con la base de datos.

7.4.2. Pruebas de Integración

Este tipo de pruebas tienen por objetivo localizar errores en módulos o subsistemas completos, analizando la interacción entre varios artefactos software. Como comentaba en la sección de estrategias, se han realizado multitud de pruebas de integración, prácticamente para cada librería empleada en el sistema verificando la comunicación no sólo con el framework (si es que no estaba adaptada al mismo), si no también, con los modelos donde se iban a utilizar principalmente.

Otro tipo de pruebas de integración realizadas también han consistido en verificar la comunicación con otros elementos externos al sistema como pueden ser la API de Google Maps, Google Analytics, Recaptcha o Youtube. Este tipo de pruebas se describen en algunos PCS (cuando ha hecho falta o en las matrices de pruebas).

Finalmente, la comunicación de los elementos usando en AJAX ha sido otro de las pruebas más realizadas en el sistema. Para ello se ha utilizado el navegador con el plug-in de firebug, el cual contiene muchas utilidades para realizar pruebas, entre ellas, verificar la realización de llamadas asíncronas (resultados enviados, devueltos, etc.)

7.4.3. Pruebas de Sistema

Estas pruebas tienen como objetivo asegurar que el sistema cumple con todos los requisitos establecidos: funcionales, de almacenamiento, reglas de negocio y no funcionales. Generalmente, se suelen desarrollar en un entorno específico para pruebas como podría ser Selenium. No obstante, debido a la complejidad de algunas pruebas, de la curva de aprendizaje de la herramienta y el limitado tiempo disponible, se han simulado a través del propio framework con controladores de prueba (como se han especificado en las pruebas unitarias) o a través de las listas de comprobación a las cuales hemos llamado Matrices de prueba.

Pruebas Funcionales

Con estas pruebas se analiza el buen funcionamiento de la implementación de los flujos normales y alternativos de los distintos casos de uso del sistema. La estrategia seguida fue tratar al mismo como una caja negra, interactuando con él como lo haría un usuario. De esta manera, se fueron recorriendo cada uno de los menús comprobando que cada botón nos llevaba a la interfaz correspondiente y que, en cada una de estas pantallas, todos los elementos estaban y funcionaban correctamente.

Para ello, se han realizado, a través del dispositivo y navegador correspondiente, la secuencia de pasos para llevar a cabo los diferentes escenarios de cada caso de uso comprobando lo que estipulaba la matriz de prueba. También ha ayudado, aunque en menor medida, el propio PROFILER del framework ya que muestra información relevante para la depuración de las diferentes llamadas urls.

Pruebas No Funcionales

Estas pruebas pretenden comprobar el funcionamiento del sistema respecto a los requisitos no funcionales establecidos:

- Adaptación de la interfaz a múltiples dispositivos y navegadores: Estas pruebas se han realizado probando en diversos dispositivos (descritos en el apartado de planificación) con diferentes tamaños de pantalla, resoluciones y versiones del sistema. Se han utilizado los navegadores más utilizados actualmente como Mozilla, Internet Explorer, Chrome, etc.
- Facilidad a la hora de introducir los datos iniciales en el sistema: Se ha comprobado

que los CSV eran los suficientemente amplios en cuanto a campos y que los errores mostrados eran comprensibles. También se han facilitado datos por defecto en algunas interfaces para llenar el menor número de campos posibles.

- Ofrecer una imagen moderna y actual de la empresa a través de la web: Sobre todo se ha comprobado que el funcionamiento responsive de la zona pública y la plantilla inicial tenga un formato flexible.
- Disponer de un sistema intuitivo, rápido, fácil de usar y con funcionalidades básicas y prácticas: Para comprobar el tiempo de respuesta se ha hecho un uso constante tanto del PROFILER como de la medición de consulta usando el comando EXPLAIN de MYSQL. Por otro lado, el empleo de la propia plantilla responsive ha ayudado mucho a definir y comprobar que el software era fácil de usar.

En las matrices de prueba, en las cuales se han descrito algunas pruebas, se ha observado las siguientes incompatibilidades (que se anotan como mejoras futuras si son posibles):

- En Internet Explorer la impresión de informes a través de CKeditor no funciona correctamente. Lo resolveremos integrando una librería que haga de parser de HTML a PDF como DOOMPDF.
- Tanto CKeditor como KCfinder no funcionan correctamente en algunos navegadores de smartphones. Actualmente estoy investigando este problema ya que posiblemente con una actualización de ambos componentes a sus versiones más recientes se resolvería el problema.

A continuación se resumen otras pruebas no funcionales comentadas anteriormente:

- **Pruebas de carga:** Principalmente se han probado que en las tablas que usaban el componente datatables.js puede soportar sin problemas la ordenación y paginación de un número elevado de registros. Por otro lado, como se indica luego en el caso práctico, se han realizado pruebas de rendimiento para comprobar que el número de inmuebles no era limitante a la hora de visualizarlos en un mapa de Google.
- **Pruebas de Portabilidad:** Estas pruebas van dirigidas a probar que el sistema puede ser utilizado en diferentes S.O. y en múltiples dispositivos. Para ello se ha hecho un despliegue del sistema en diferentes entornos (desarrollo y staging) que tienen entornos tecnológicos diferentes (Windows y Linux, respectivamente). También uno

de los desarrolladores de TIPYCOS se ofreció a instalarlo en un portátil con sistema operativo MAC OS X. Por otro lado, se han empleado múltiples navegadores (los más comunes utilizados en sus últimas versiones).

- **Pruebas de concurrencia:** Este tipo de pruebas tienen por objetivo comprobar en determinadas funcionalidades qué ocurre cuando dos usuarios las usan simultáneamente. Muchas de estas se han especificado en las propias matrices de pruebas. Normalmente el proceso consistía en abrir dos navegadores diferentes ejecutando la misma funcionalidad, probando que las precondiciones del casos de uso se cumplían y mostraba los errores adecuados.

Pruebas de Regresión

Las pruebas de regresión consiste en la repetición de ciertas pruebas para comprobar que los nuevos cambios que se vayan introduciendo no originen errores en el software ya probado. Normalmente se han ido estipulando durante las propias matrices de prueba atendiendo a si la funcionalidad evaluada dependía de otra en toda o parte de la misma. Para ello, en muchos de las matrices, al comienzo de la prueba, se han indicado las funcionalidades afectados y cómo había que evaluarlas o qué aspectos relevantes había que tener en cuenta.

7.4.4. Pruebas de Aceptación

El objetivo de estas pruebas es demostrar que el producto está listo para el paso a producción. Estas pruebas fueron realizadas al final del proceso construcción por un colaborador inmobiliario de la empresa de GESTICADIZ, además de otras personas con conocimiento del mercado inmobiliario. Otras personas se ofrecieron voluntarios para probar la zona pública y dar su opinión. La empresa TIPYCOS, también se ofreció voluntaria para probar la zona de administración así como la zona pública. No obstante, aunque se tomó como base las explicaciones del manual de usuario, ya que hace un repaso de las funcionalidades más importantes, fueron probando el sistema como estimaron oportuno.

7.5. Ejemplo práctico de integración de componentes: Libre- ría de Google Maps

El proceso a seguir se ha explicado en el apartado de Metodología de desarrollo de la Planificación. Expondremos un ejemplo práctico con la librería de Google Maps:

1. **Evaluación:** Comprobé que disponía al menos de una funcionalidad para visualizar una dirección dada y que funcionaba, al menos, síncronamente.
2. **Adaptación:** La llamada ASÍNCRONA del componente no funcionaba correctamente. En mi empeño de realizar una llamada asíncrona con AJAX localicé una adaptación realizada por otro usuario y pasé a implementarla y probarla y verificar que funcionaba correctamente en mi sistema.
3. **Integración con funcionalidades:** Posteriormente, se pasaron a implementar funcionalidades específicas dentro del modelo del Inmueble para visualizar correctamente cada marca del mapa según si la vista era pública o privada, es decir, si se deseaba ver la información tal como se vería el inmueble en la zona pública o con sus datos reales.
4. **Optimización:** Una vez realizado los apartados anteriores y que las pruebas funcionales fueran exitosas, se comprobó que se tardaba demasiado en desplegar un mapa cargado con muchos inmuebles ya que, por cada inmueble, se estaba haciendo una consulta directa a la API de Google para obtener sus coordenadas correspondientes. Por tanto, se pasó a realizar una optimización para que la utilización fuera viable en el sistema cuya implementación se detalla en los PCS. Haciendo un breve resumen, esta consistía en la utilización la opción de CACHING que incluía la librería y por cada llamada a la API de Google Maps además se almacenaban sus coordenadas en una tabla de GEOCODING, la cual relacionaba unas determinadas coordenadas con una dirección exacta. Por tanto, la primera vez que realizaba esta acción tardaba un tiempo, pero en las posteriores llamadas, al utilizar el GEOCODING, la transformación de la dirección en coordenadas era instantánea. Además con esto matamos dos pájaros de un tiro, ya que además de aumentar el rendimiento de esta funcionalidad extraordinariamente, evitábamos alcanzar siempre el límite diario que dispone la API de Google Maps de 25000 llamadas diarias.

Parte III

Epílogo

Capítulo 8

Manual de implantación y explotación

8.1. Introducción

Este manual va dirigido a toda aquella persona que desee instalar la plataforma en un servidor web. Se requieren conocimientos, al menos básicos, sobre cómo funcionan los entornos cliente-servidor y el acceso y configuración de servidores para realizar una instalación básica.

Si desea hacer un uso más avanzado del sistema o modificar determinados comportamientos, necesitará conocimientos sobre diseño Responsive así como del resto de tecnologías y componentes empleados en la realización del proyecto.

8.2. Requisitos previos

Los requisitos hardware y software para la correcta instalación del sistema han sido especificados en la sección de Arquitectura del sistema 5.1.1. No obstante, si es necesario aclarar que realice algunos **procedimientos para que el sistema interactúe correctamente con los sistemas externos**, principalmente, con diferentes **componentes de Google**:

- Necesitará generar una **API-KEY o clave de autenticación** para **Google Maps** tal como se indica en [47]. Este paso es **obligatorio**.
- Necesitará configurar el sistema de seguimiento de su cuenta de **Google Analytics**. Para ello siga los pasos indicados en [48]. Este paso es **opcional**. Si no realiza este

paso no recibirá información estadística del uso de su web corporativa.

- Necesitará generar la **clave secreta** y la **clave de su sitio web** para el empleo de **Recaptcha**. Este paso es **obligatorio**.
- Necesitará tener una **cuenta de correo personal** creada donde recibir los correos enviados por la plataforma. Recomiendo que use cualquiera de Gmail.

De los procedimientos anteriores, **NO es necesario que los integre en la plataforma**, ya que están integrados, sólo es necesario generar las diferentes claves que se detallan en cada uno de ellos y posteriormente introducirlos en la plataforma tal y como se detalle en el procedimiento de instalación.

Finalmente, el servidor debe tener habilitados al menos los siguientes módulos o extensiones (que normalmente, ya vienen activos en cualquier servidor web o con la instalación de cualquier paquete como XAMPP o WAMP SERVER).

- Apache: Rewrite y PHP5 (ya que el sistema utiliza un .htaccess para eliminar el controlador frontal de CodeIgniter, el conocido index.php, de las url).
- PHP: CURL, OPENSSL, FILEINFO, MBSTRING, PDO_MYSQL, MYSQLI.
- MYSQL: Se recomienda que active el cacheo de consultas y que esté optimizado para el uso de tablas en INNODB así como otras buenas prácticas que se detallan (parcialmente) en la bibliografía [43].

La instalación que ofrece por defecto WAMP SERVER 2.5, contiene todos estos módulos o extensiones ya activos.

8.3. Inventario de componentes

8.3.1. Relación de componentes de backend

La lista 8.1 contiene sólo los componente externos, no todos los empleados, ya que Codeigniter también trae varios:

| ASPECTO | CATEGORÍA | COMPONENTE |
|--------------------------|-----------------|---------------------|
| WYISWYG | Librería | Ckeditor + KCfinder |
| CSV | Librería | CSVReader |
| Copias seguridad | Módulo | MYBackup |
| Autenticación y permisos | Módulo-Librería | ION Auth |
| QR Code | Librería | Ciqrkode + Libern |
| Google Maps | Librería | Google Maps |
| Ficheros | Librería | MYUpload |
| Modelos | Librería | MYModel |
| Utilidades varias | Librería | Utilities |
| Validaciones Formularios | Librería | MYFormValidation |
| Gestión de contenidos | Módulo | CMS TIPYCOS |

Tabla 8.1: Comparativa de componentes de backend

8.3.2. Relación de componentes de frontend

La lista 8.2 todos los componentes empleados según si se han integrado, si ya estaban incluidos en la plantilla utilizada o si se va a realizar una implementación manual:

| ASPECTO | IMPLEMENTACIÓN | COMPONENTE |
|------------------|----------------|---|
| WYISWYG | CMS TIPYCOS | Ckeditor |
| Ficheros | CMS TIPYCOS | KCfinder (integrado en CKeditor) + Dropzone |
| Estadísticas web | Manual | Google Analytics |
| Paginación | Ace Template | Datatables |
| Imágenes | Ace Template | Colorbox |
| Formularios | Ace Template | Chosen select, Date range picker, etc. |
| Gráficos | Ace Template | Jquery flot |
| Iconos | Ace Template | Font Awesome |

Tabla 8.2: Comparativa de componentes de frontend

Adicionalmente a estos, la versión de CodeIgniter empleada es la 3.1 para el backend, y

para el frontend se han empleado bootstrap 3.3.1 (plantilla Ace Admin), bootstrap 3.2 para la plantilla de la web corporativa, y Materialize 0.97 para la zona de login.

8.4. Procedimientos de instalación

8.4.1. Inicial

Lo primero que deberá realizar será la **instalación del servidor web**. Con cumplir los requisitos previos descritos anteriormente puede hacerlo como desee. No obstante, para el desarrollo del proyecto se ha utilizado el paquete de software **WAMP SERVER 2.5** siguiendo los siguientes descritos aquí [50].

Posteriormente, deberá **importar la base de datos**. Para ello acceda la carpeta de instalación y ejecute el script openrs.sql en su SGBD.

El sistema vendrá con una web completamente creada con datos ficticios pero que puede tomar como base para configurar su sitio web y observar el comportamiento del sistema. Los datos de configuración obtenidos en el paso anterior deberá introducirlos en la sección de preferencias de la configuración.

Después deberá ejecutar un script para **adaptar las rutas internas introducidas en los diferentes WYSIWYG a su web**. Para ello acceda al script de la carpeta de instalación y ejecute el script adaptacion.sql en el SGBD de su servidor.

Finalmente, deberá **configurar la conexión de la base de datos y el tipo de entorno** en el servidor en el que despliegue la aplicación (normalmente producción) siguiendo los estándares marcados por Codeigniter.

Se recomienda añadir un **certificado SSL** en el servidor para mayor seguridad. El sistema está adaptado para usarse bajo protocolo HTTPS (o al menos lo probado en desarrollo). Si quiere forzar el uso de HTTPS en su web, descomente la opción correspondiente en el fichero htaccess de la carpeta raíz de la aplicación.

Recuerde que las carpetas deberán estar con los permisos 755 y los ficheros en 644.

8.4.2. Migración desde otro servidor con diferente dominio

Si va realizar una instalación en otro servidor con diferente dominio y ubicación de donde realizó la copia de seguridad del sistema, primero deberá realizar una instalación completa desde cero (tal como se detalló anteriormente), y posteriormente importar la base de datos obtenido de la plataforma y su carpeta uploads. La justificación de esto es que, la copia

de seguridad de la base de datos realizada por la plataforma, por motivos técnicos, no contiene las vistas.

Posteriormente, deberá regenerar aquellos carteles que desee volver a imprimir o que haya impreso y tengo en su escaparate. Esto se debe a que el QR que imprimió anteriormente contiene una dirección de su dominio anterior. Lo ideal para evitar esto es que lo redireccione a su nuevo servidor, ya que, salvo el dominio, el resto de rutas son invariables.

8.5. Pruebas de implantación

Se aconseja ejecutar las pruebas de regresión del sistema, incidiendo en los siguientes aspectos:

- Visualizar y desplegar todos los mapas del sistema.
- Realizar un envío de correo a través del formulario de olvido contraseña.
- Realizar envío de correo desde la zona pública a la inmobiliaria y verificar que el captcha se visualiza y funciona correctamente (recuerde que este captcha, dependiendo de cómo lo haya configurado, es invisible o no).
- Probar a registrar un video youtube y publicarlo y visualizarlo en el detalle.
- Visualizar estadísticas de tráfico generado por la web corporativa en su cuenta de Google Analytics.
- Visualizar estadísticas de inicio en la zona privada.
- Subir ficheros (tanto en los WYSIWYG como la sección de adjuntos).

8.6. Procedimientos de operación y nivel de servicio

Se recomienda crear una **cuenta de correo** en el servidor dónde instale la plataforma y configurar y utilizar el protocolo SMTP. Si hace esto, deberá configurar el componente ION AUTH [25] y CMS TIPYCOS [24] tal como se indican en sus web.

También podrá configurar el propio framework de backend, si desea tal como se indica en la web [32], para, por ejemplo, el límite de tiempo de sesión, añadir más seguridad a las cookies o la forma de ejecutarse la protección CSRF.

No obstante, recomiendo no modificar los valores por defecto, incluidos con la instalación.

Si ocurriese algún problema al realizar backups (ya que estos dependen de los parámetros configurados en el servidor como la variable `memory_limit` entre otros), puede hacerlo manualmente con los dos siguientes pasos:

- Realizando una **copia de seguridad de la base de datos a través de PHPMyADMIN** (que suele venir por defecto con cualquier paquete o instalada en cualquier servidor). Es muy importante que indique al hacer la copia o al importarla, que omita la comprobación de claves foráneas, ya que, en caso contrario, se produciría un error por las **FOREIGN KEYS** especificadas en las diferentes tablas. Active también la opción para eliminar tablas o vistas si ya existen si va a realizar una instalación sobre una base de datos ya existente. Es posible que también deba eliminar el código de los **DEFINERS** de las diferentes vistas si los usuarios del servidor origen y destino de la base de datos son diferentes.
- Comprimiendo la carpeta **uploads** que contiene todos los datos que haya subido a la plataforma y descargándola del servidor o descargando la carpeta directamente.

Todas estas consideraciones ya han sido tenido en cuenta en las copias de seguridad que puede realizar manualmente a través de la plataforma.

Finalmente, comentar que los logs podrán ser consultados en la carpeta habilitada por defecto a tal efecto. No obstante, usted puede modificar este comportamiento si lo desea tal como especifica el framework.

8.7. Limitaciones conocidas

Realizando las pruebas en los diferentes entornos y dispositivos se han encontrado las siguientes limitaciones:

- En algunos móviles, incluso usando navegadores como Chrome, el componente CKEDITOR (es decir los WYSIWYG), no se visualizan correctamente y aparece el código HTML directamente en el textarea.
- Se ha observado que en algunas tablet, al pulsar en imprimir el marco del WYSIWYG, se bloquea el navegador. En siguientes versiones incluiremos una exportación separada directa a PDF a través de una librería externa.

- No todas las resoluciones posibles han sido contempladas en la plantilla web de la zona pública, por lo que es posible, que deba adaptar su funcionamiento a través de CSS a determinadas resoluciones.
- Las subidas de los diferentes documentos del sistema estará limitada por el tamaño de la variable **post_max_size** configurada en el servidor.

Capítulo 9

Manual de usuario

9.1. Introducción

OPENRS es un software pensado para satisfacer las necesidades de una inmobiliaria pequeña o mediana. Dentro de la organización estándar de una inmobiliaria, este software se dirige principalmente a los agentes inmobiliarios. Usted dispondrá de dos zonas bien diferenciadas:

- La zona privada o software de gestión: Desde esta sección podrá gestionar los datos de las propiedades o inmuebles que oferte, su cartera de clientes (tanto demandantes como ofertantes) y las demandas tanto automáticas como manuales. Además, contará con una zona de administración en la que, a parte de la configuración de los datos que se van a utilizar en el sistema y sus preferencias, dispondrá de un CMS para la configuración del aspecto y los contenidos de la zona pública.
- La zona pública o web corporativa: Básicamente, esta es la web de su empresa desde la cual podrá anunciar sus inmuebles así como cualquier otro servicio que desee. Además dispondrá de algunas secciones predefinidas útiles para la promoción de la empresa como noticias, contacto o el buscador de inmuebles, pero tendrá la libertad de agregar cualquier otra que considere apropiada.

Existirán tres roles de usuarios:

- **Visitante de la web**, principalmente, serán posibles futuros clientes (ofertantes y demandantes): interactuarán con la zona pública pero tendrán un rol pasivo. Podrán visualizar los datos definidos desde la zona privada como públicos y ponerse en con-

tacto con la inmobiliaria, pero no dispondrán de una zona privada como ocurre con los agentes inmobiliarios.

- **Agente inmobiliario:** encargado de registrar los datos en el sistema y visualizar las principales estadística de uso del mismo. Tenga en cuenta que el agente se ha definido como un administrador de los datos registrados por la inmobiliaria, es decir, por defecto manipulará los datos asignados a él mismo aunque tendrá la potestad para acceder y modificar cualquier otro dato que afecte a la dinámica de la inmobiliaria, esto quiere decir, estadísticas, inmuebles (así como su publicación en la web corporativa), clientes y demandas.
- **Administrador:** Su principal cometido será el mantenimiento de la zona pública, a través de un pequeño CMS, así como la configuración de los datos que podrán ser empleados por los agentes inmobiliarios en las secciones habilitadas. Entre ellos podemos destacar las plantillas de documentación, los tipos de inmuebles que oferta la inmobiliaria, los posibles estados en los que se pueden encontrar los inmuebles, clientes o demandas, etc. Finalmente, tendrá la posibilidad de gestionar los usuarios que acceden a la plataforma así como sus permisos y la gestión de las copias de seguridad del sistema.

Debido a que la aplicación se divide en zonas y roles bien diferenciados, explicaremos el manual acorde a las mismas. Para que las figuras se visualicen correctamente en el manual se ha utilizado un tamaño de ventana medio. No obstante, la aplicación al ser responsive, se adaptará en la mayoría de casos, a todos los dispositivos y tamaños que pueda utilizar.

9.2. Zona privada

9.2.1. Acceso al sistema

Lo primero que deberá realizar será acceder al sistema. Para ello deberá introducir su nombre de usuario y contraseña en la siguiente interfaz:

The screenshot shows the login page for the OPENRS system. At the top, there is a blue header bar with the text "OPENRS" on the left and "Documentación" and "Repositorio" on the right. Below the header, the page title "Ingresar" is displayed. A message "Por favor, introduce tu email/usuario y contraseña." is present. There are two input fields: one for "Email/Usuario:" and one for "Contraseña:". Below these fields is a checkbox labeled "Recuérdame:". A green "INGRESAR" button is located below the password field. To the right of the "INGRESAR" button, there is a link "¿Has olvidado tu contraseña?". At the bottom of the page, there is a yellow banner with the text "Bienvenido a OPENRS" and a descriptive message: "Empieza a disfrutar de las ventajas de usar este software pensado para PYMES inmobiliarias españolas y define tu web corporativa así como todos los datos necesarios para la gestión de tu negocio."

Figura 9.1: Interfaz de acceso al sistema

Por defecto, se habilitará una cuenta de administrador admin@admin.com con contraseña admin@admin.com, el cual, se recomienda encarecidamente, que cambie su contraseña nada más entrar en la plataforma así como que habilite una cuenta de administrador alternativa a la misma. Habrá otros usuarios creados a modo de prueba.

También dispondrá de una opción para recordar su contraseña en caso de haberla olvidado. Para ello, pulse en el enlace **¿Has olvidado tu contraseña?** y aparecerá la siguiente interfaz:

He olvidado mi Contraseña

Por favor, introduce tu Email para que podamos enviarte un email para restablecer tu contraseña.

Email:

angel.berasuain@gmail.com

ENVIAR

Bienvenido a OPENRS

Empieza a disfrutar de las ventajas de usar este software pensado para PYMES inmobiliarias españolas y define tu web corporativa así como todos los datos necesarios para la gestión de tu negocio.

Figura 9.2: Interfaz de recuperación de contraseña olvidada

Aquí deberá introducir el correo electrónico de la cuenta que desea recuperar la contraseña. Una vez introducido, pulse en **Enviar** y se le remitirá a la dirección de correo electrónico introducida un email como el descrito en la siguiente imagen, desde el cual se le proporcionará un enlace para poder restaurar la contraseña de la cuenta de usuario:



Reestablecer contraseña para [angel.berasuain@gmail.com](#)

Por favor ingresa en este link para [Restablecer Tu Contraseña](#).

Figura 9.3: Correo de recuperación de contraseña olvidada

En la plataforma aparecerá el mensaje **Nueva contraseña enviada por email**.

NOTA: Dependiendo de la configuración que haya establecido en el servidor donde se encuentre alojada la plataforma así como los filtros anti-spam que disponga la cuenta receptora del correo electrónico, el email podría ir a la carpeta de SPAM y tardar unos minutos en llegar. Por tanto, revise esta carpeta si no lo recibe en su bandeja de entrada.

Una vez visualice el email correctamente, pulse sobre el enlace **Restablecer tu contraseña**. Este le llevará nuevamente a la plataforma pero a la siguiente interfaz:

The screenshot shows the 'Cambiar Contraseña' (Change Password) page of the OPENRS platform. At the top, there is a blue header bar with the 'OPENRS' logo on the left and 'Documentación' and 'Repositorio' links on the right. Below the header, the page title 'Cambiar Contraseña' is displayed. A note below the title says 'Nueva Contraseña (de al menos 8 caracteres de longitud):'. Two input fields follow, both containing '*****'. Below these fields is a note 'Confirmar Nueva Contraseña:' followed by another input field containing '*****'. At the bottom of the form is a green 'GUARDAR' (Save) button. Below the form, there is a yellow banner with the text 'Bienvenido a OPENRS' and a descriptive message: 'Empléala a disfrutar de las ventajas de usar este software pensado para PYMES inmobiliarias españolas y define tu web corporativa así como todos los datos necesarios para la gestión de tu negocio.'

Figura 9.4: Interfaz de recuperación de cambiar contraseña

Desde la misma, podrá nuevamente introducir una nueva contraseña para la cuenta de usuario vinculada al correo electrónico introducido previamente. Una vez haya cambiado la contraseña, aparecerá el mensaje **Contraseña renovada con éxito** y podrá acceder al sistema con la nueva contraseña introducida.

Finalmente, en el menú superior de la interfaz de acceso al sistema, se han habilitado dos enlaces para visitar el repositorio del proyecto (el cual incluye este manual) así como la documentación on-line del código por si desea modificarlo o contribuir con su mejora.

9.2.2. Estructura general

Una vez iniciada sesión en el sistema, podrá apreciar el panel de estadísticas 9.6. Esta interfaz, así como todas las de la zona privada están compuestas por un menú lateral de opciones. Cada vez que navegue por una opción esta quedará marcada. Así podrá cambiar a la sección que desee rápidamente o volver al inicio de una sección. Otras interfaces con más niveles de profundidad, como veremos en la sección del administrador, dispondrán de breadcrumbs (migas de pan) para facilitar la navegación.

En el margen superior derecho dispondrá siempre de dos opciones adicionales:

- **Perfil:** Desde el cual podrá cambiar sus datos de inicio de sesión así como su nombre, apellidos o teléfono de contacto.
- **Salir:** Le permitirá desconectarse del sistema con seguridad. Recuerde usar esta opción siempre antes que cerrar la pestaña del navegador ya que su sesión seguirá estando abierta por un período que variará de la configuración realizada sobre la plataforma (por defecto de 2 horas).

Además verá el símbolo de una rueda dentada. Esta sólo aparecerá en las versiones de dispositivos con interfaces medias o largas, y servirá para configurar diferentes aspectos de visualización de las interfaces. Por ejemplo a continuación se muestra la página como quedaría si marcásemos todas las opciones menos la de orientar la web a la derecha:

| Ref. | Tipo | Municipio | Zona | Dirección | Precio Compra | Precio Alquiler | Met. | Hab. | Bañ. | Estado | Fecha alta | Prop. | Dem. | Pend. |
|---------------|---------|-----------|-------------------------|---------------------------|---------------|-----------------|------|------|------|-----------|------------|-------|------|-------|
| 599d8c6668ff8 | Oficina | Almería | Catedral | Calle Obispo Orberá, 5 | 135.000 | 500 | 400 | 4 | 1 | Captación | 23/08/2017 | 0 | 0 | 0 |
| REF0012 | Piso | Cádiz | | Calle García de Sola, 1 | 150.000 | 0 | 80 | 2 | 1 | Captación | 17/08/2017 | 0 | 0 | 0 |
| REF0011 | Piso | Cádiz | | Pablo Ruiz Picasso, 4, 4J | 360.000 | 0 | 120 | 4 | 2 | Captación | 17/08/2017 | 0 | 0 | 0 |
| 5943f77c20f59 | Duplex | Cádiz | Ayuntamiento - Catedral | avenida amílcar barca, 2 | 100.000 | 0 | 60 | 1 | 1 | Captación | 15/08/2017 | 0 | 1 | 1 |
| REF0001 | Piso | Almería | | calle granada | 200.000 | 0 | 50 | 2 | 1 | Alquilado | 01/07/2017 | 1 | 1 | 0 |

| Ref. | Tipo | Municipio | Zona | Dirección | Precio Compra | Precio Alquiler | Met. | Hab. | Bañ. | Estado | Fecha mod. | Prop. | Dem. | Pend. |
|---------------|-------------|--------------|----------|------------------------|---------------|-----------------|------|------|------|-----------|------------|-------|------|-------|
| 5944ea26a1ab8 | Unifamiliar | San Fernando | | Calle real, 1 | 0 | 600 | 60 | 1 | 1 | Captación | 17/08/2017 | 0 | 0 | 0 |
| REF0005 | Oficina | Almería | Catedral | Calle Obispo Orberá, 5 | 135.000 | 500 | 400 | 4 | 1 | Captación | 01/06/2017 | 1 | 2 | 1 |

| Ref. dem. | Ref. inm. | Tipo | Municipio | Zona | Dirección | Precio Compra | Precio Alquiler | Metros | Hab. | Bañ. | Estado | Opciones |
|-----------|---------------|--------|-----------|-------------------------|--------------------------|---------------|-----------------|--------|------|------|-----------|----------|
| DEM0001 | 5943f77c20f59 | Duplex | Cádiz | Ayuntamiento - Catedral | avenida amílcar barca, 2 | 100.000 | 0 | 60 | 1 | 1 | Captación | |

Figura 9.5: Estadísticas: Ejemplo de configuración de vista interfaces

Finalmente, abajo dispondrá del pie de página de la zona privada con el Copyright del software. Todas estas secciones permanecerán siempre como parte de la plantilla de la zona privada, sólo se cargará una nueva interfaz en la parte central a medida que vaya navegando por las diferentes opciones del sistema.

9.2.3. Perfil de Agente Inmobiliario

En los siguientes ejemplos, se han introducido datos adicionales para exemplificar la funcionalidad del sistema. La plataforma, inicialmente, vendrá con un sistema completamente configurado (tal como se encuentra actualmente en el servidor de pruebas del software) para que usted tenga unos datos con los que poder trabajar y tener como base para configurar su web y su sistema como desee.

Estadísticas

Como mencionamos antes, nada más inicie sesión en la plataforma accederá al denominado como Inicio en el menú lateral izquierdo o Dashboard (panel de estadísticas). Este panel, por defecto, está pensado para mostrar la información más relevante para el agente inmobiliario tanto de forma gráfica como de forma textual. Está dividido varias pestañas una por cada módulo que conforman los datos que puede manipular en el sistema y ofrece una visión estadística general de la información registrada en la plataforma acorde a los criterios más importantes de cada sección. Por defecto se le mostrará la información de los inmuebles vigentes registrados desde su punto de vista, es decir, lo que se denomina en el título de la página como **Estadísticas personales**, viene a representar cómo está la información de los elementos asignados a usted y que aún no forman parte del histórico del sistema (para saber a qué se refiere con este término véase el capítulo de configuración de los estados de la aplicación).

Por cada gráfico, dispondrá una serie de opciones para poder consultar, no sólo los inmuebles vigentes, sino sólo los históricos o hacer una evaluación global. Al hacer clicks en ellos, se cargarán dinámicamente (sin refrescar la página) las estadísticas del bloque correspondiente.

Además de estos gráficos se ofrece una serie de tablas para visualizar los últimos inmuebles registrados, los modificados, aquellos pendientes de evaluar en demandas o que han sido propuestos para visita. Desde estas podrá navegar directamente al inmueble o demanda que desee consultar.

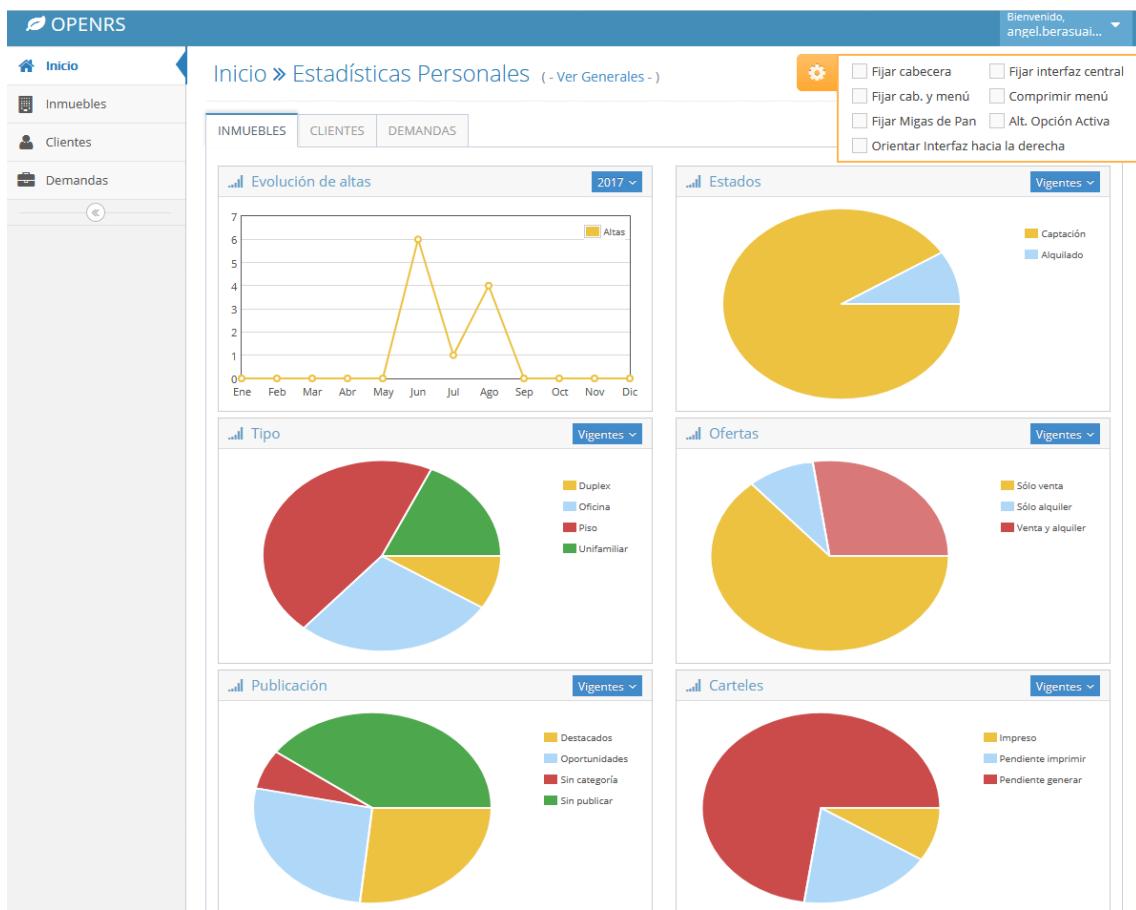


Figura 9.6: Estadísticas Personales: Inmuebles

Por ejemplo, de la imagen anterior podríamos deducir, de un simple vistazo, que la mayoría de inmuebles se han introducido en Junio, que está en período de captación, que la mayoría son duplex aunque hay bastantes oficinas, que la mayoría están a la venta, que hay muchos sin publicar en la web corporativa o que la mayoría de sus carteles publicitarios están pendientes de generar. A continuación se exponen las interfaces del resto de pestañas:

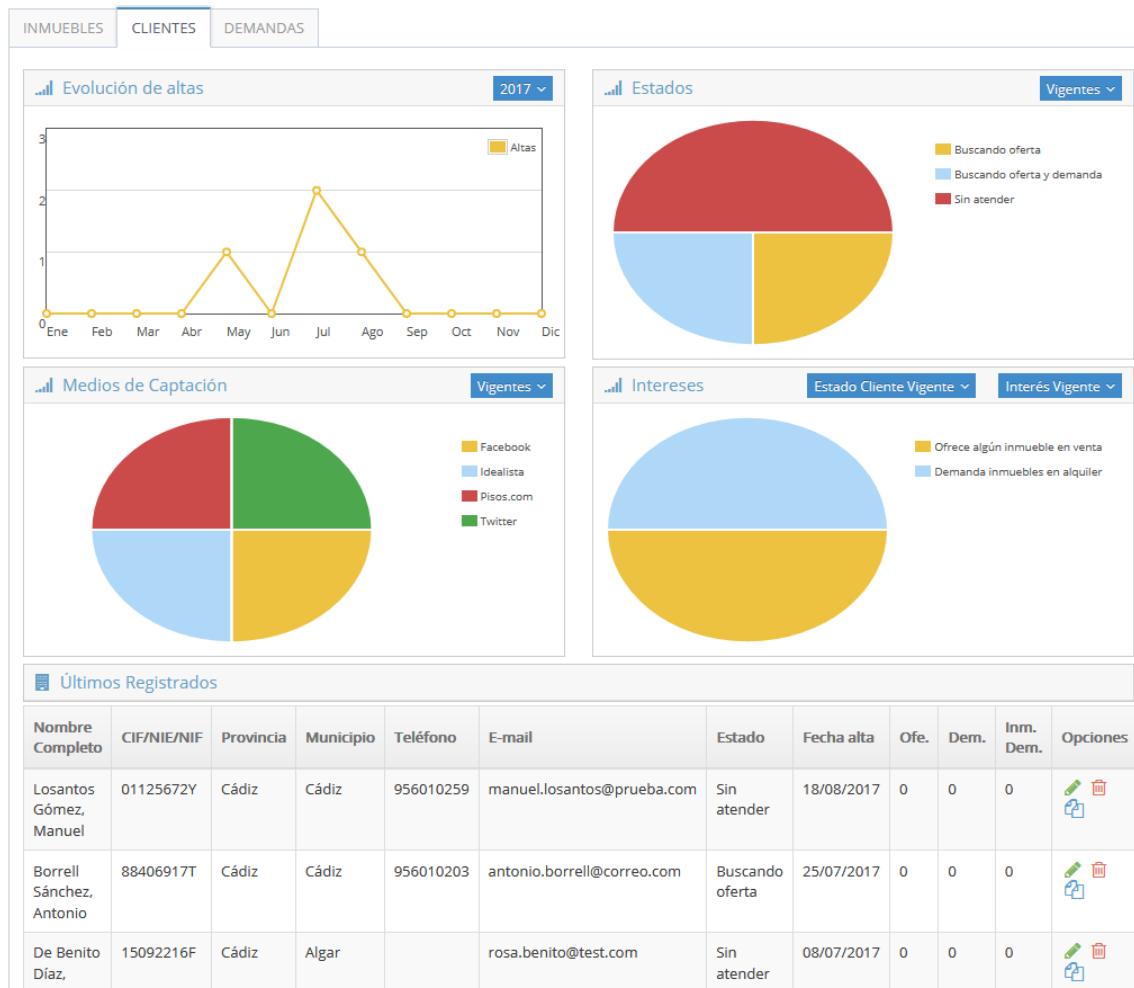


Figura 9.7: Estadísticas Personales: Clientes

Inicio » Estadísticas Personales (- Ver Generales -)

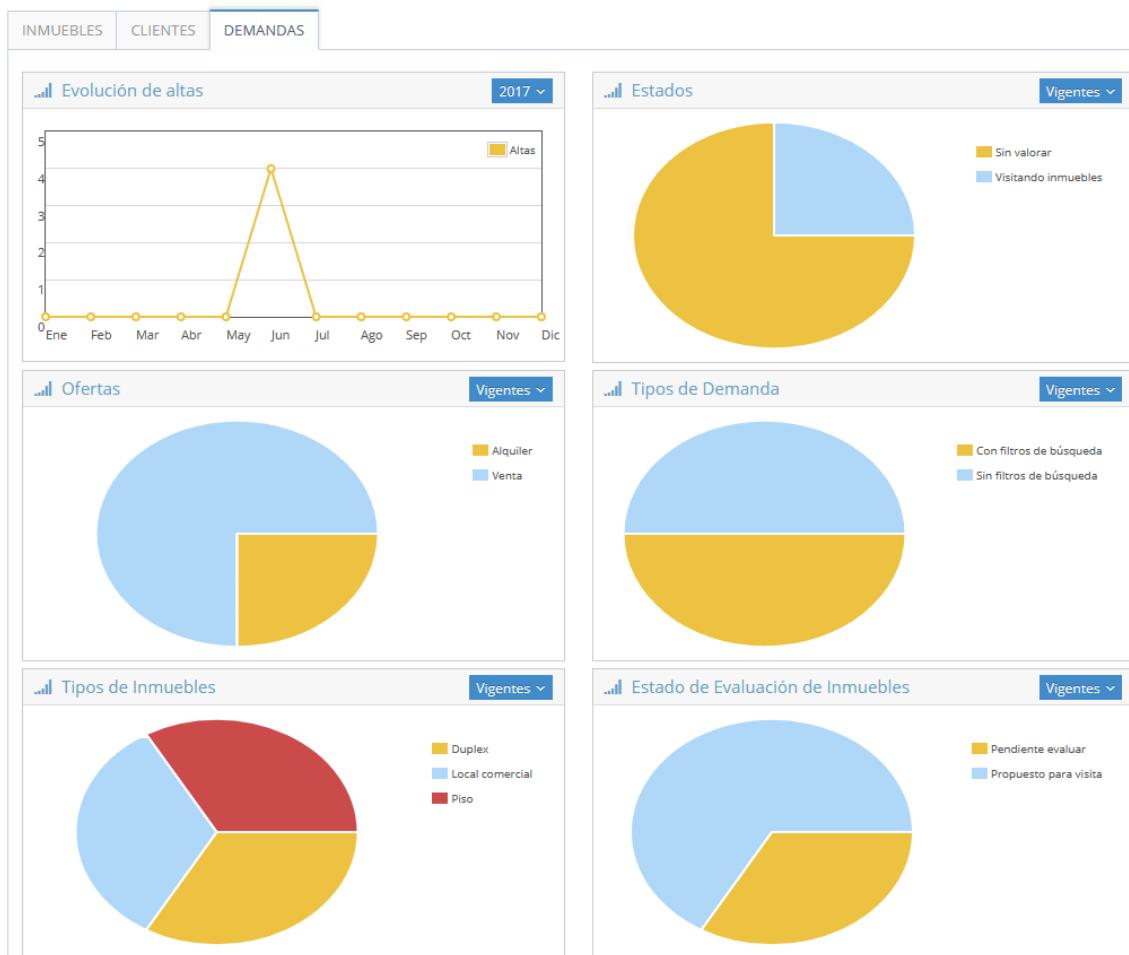


Figura 9.8: Estadísticas Personales: Demandas

Finalmente si pulsa, sobre **Ver estadísticas generales**, podrá visualizar las estadísticas de todos los datos registrados en el sistema, no sólo los asignados a usted. Esta interfaz es la única que puede visualizar el administrador y se muestra en la siguiente figura 9.47.

Inmuebles

Haremos un análisis más detallado de este módulo, al ser el más importante, explicando todos sus funcionalidades. Los demás tendrán la misma organización y las mismas funcionalidades básicas pero adaptadas a su ámbito de actuación.

Por defecto, al igual que con las estadísticas, se muestran los inmuebles asignados a usted tal y como muestra la siguiente figura 9.9. Este listado pretende mostrar la vista comercial del inmueble, es decir, muestra aquellos datos más interesantes desde el punto de vista del arrendamiento o la compra-venta. Las columnas más importantes en este aspecto son:

- **Prop.**: muestra el número total de propietarios del inmueble.
- **Dem.**: muestra el número total de demandas que tiene el inmueble.
- **Pend.**: muestra el número total de demandas en las que el inmueble está pendiente de evaluar.
- **Vis.**: muestra el número total de demandas en las que el inmueble está propuesto para visita.

Desde el mismo podrá editar, eliminar o duplicar un inmueble. Al duplicarlo, la plataforma sólo duplicará los datos de la ficha principal del inmueble (pero no sus datos de publicación o imágenes adjuntas por ejemplo). Si todo fue bien, se le redirigirá a la interfaz de edición del inmueble, desde la cual, podrá completar el resto de datos que desee.

También puede ordenar el listado dinámicamente haciendo click en las diferentes columnas por las que desee ordenar, así como filtrar cualquier búsqueda realizada a través del campo **Buscar**, introduciendo las palabras que desee.

Inmuebles

Mostrar: 100 ▶ registros

| Ref. | ▲ Tipo | ▼ Municipio | ◆ Zona | ◀ Dirección | ◀ Met.♦ | Precio Compra ♦ | Precio Alquiler ♦ | Hab.♦ | Bañ.♦ | Estado | ◀ Fecha alta ♦ | Prop.♦ | Dem.♦ | Pend.♦ | Vis.♦ | Opciones ♦ | |
|---------------|-------------|--------------|---------------------------------|------------------------------|---------|-----------------|-------------------|-------|-------|------------|----------------|--------|-------|--------|-------|------------|--|
| 5943772059 | Duplex | Cádiz | Ayuntamiento - Catedral | avenida amilcar barca, 2 | 100.000 | 0 | 60 | 1 | 1 | Capitación | 15/08/2017 | 0 | 1 | 1 | 0 | | |
| 5944ea26a1ab8 | Unifamiliar | San Fernando | | Calle real, 1 | 0 | 600 | 60 | 1 | 1 | Capitación | 17/06/2016 | 0 | 0 | 0 | 0 | | |
| 59515c52a7091 | Oficina | Almería | Catedral | Calle Obispo Orberá, 1, bajo | 135.000 | 500 | 400 | 4 | 1 | Capitación | 26/06/2017 | 0 | 0 | 0 | 0 | | |
| 59515c6d021ea | Oficina | Almería | Catedral | Calle Obispo Orberá, 1, bajo | 135.000 | 500 | 400 | 4 | 1 | Vendido | 26/06/2017 | 0 | 0 | 0 | 0 | | |
| 599e8c8ce3586 | Piso | Cádiz | Centro Histórico - Plaza España | Columba | 150.000 | 0 | 80 | 2 | 1 | Capitación | 24/08/2017 | 0 | 0 | 0 | 0 | | |
| REF0001 | Piso | Almería | Rambla Lorca | calle granada | 200.000 | 0 | 50 | 2 | 1 | Alquilado | 01/07/2017 | 1 | 1 | 0 | 1 | | |
| REF0002 | Piso | Almería | | carrereta granada | 300.000 | 0 | 70 | 3 | 2 | Capitación | 02/06/2017 | 0 | 0 | 0 | 0 | | |
| REF0003 | Piso | Almería | | avenida del mediterraneo | 400.000 | 0 | 100 | 4 | 3 | Capitación | 03/06/2017 | 1 | 0 | 0 | 0 | | |
| REF0005 | Oficina | Almería | Catedral | Calle Obispo Orberá, 5 | 135.000 | 500 | 400 | 4 | 1 | Capitación | 09/06/2017 | 1 | 2 | 1 | 1 | | |
| REF0006 | Unifamiliar | Cádiz | | Avenida marconi, 10 | 125.000 | 0 | 60 | 1 | 1 | Capitación | 12/06/2017 | 0 | 0 | 0 | 0 | | |
| REF0011 | Piso | Cádiz | | Pablo Ruiz Picasso, 4, 4J | 360.000 | 0 | 120 | 4 | 2 | Capitación | 17/08/2017 | 0 | 0 | 0 | 0 | | |
| REF0012 | Piso | Cádiz | | Calle García de Sola, 1 | 150.000 | 0 | 80 | 2 | 1 | Capitación | 17/08/2017 | 0 | 0 | 0 | 0 | | |

Mostrando registros del 1 al 12 de un total de 12 registros

Buscar:

Q Busqueda Avanzada Exportar CSV Ver datos publicación Importar CSV Ver mapa real Insertar

Figura 9.9: Inmuebles: Buscador

Adicionalmente, en la parte superior de la interfaz dispondrá de una botonera con diferentes acciones. La primera de ellas, la **Búsqueda Avanzada**, mostrará todas las opciones por las cuales puede filtrar los datos.

The screenshot shows the 'Inmuebles' advanced search interface. It is divided into three main sections: 'Datos generales' (General Data), 'Publicación' (Publication), and 'Datos específicos' (Specific Data). Each section contains several dropdown menus for filtering search results. At the bottom right of the form are two buttons: 'Buscar' (Search) and 'Limpiar' (Clear).

| Datos generales | |
|--------------------------|------------------------------|
| Provincias | - Seleccione provincia - |
| Poblaciones | |
| Zonas | |
| Tipo | - Seleccione tipo - |
| Agentes Asignados | Berasuain Ruiz, Ángel Luis |
| Certificación energética | - Seleccione certificación - |
| Estados | - Seleccione estado - |
| Fecha alta | |

| Publicación | |
|-------------|-----------------------|
| Publicado | - Indiferente - |
| Ofertas | - Seleccione oferta - |
| Destacado | - Indiferente - |
| Oportunidad | - Indiferente - |

| Datos específicos | |
|---------------------|------------------------------------|
| Habitaciones | |
| Precios | |
| Modificación precio | - Seleccione modificación precio - |
| Baños | |
| Metros | |

Buscar Limpiar

Figura 9.10: Inmuebles: Búsqueda Avanzada

Si deja las opciones con los valores **- seleccione X -**, entonces no se aplicará el filtro de datos. Si pulsa en el botón **Limpiar**, restaurará los filtros por defecto. Además cada búsqueda que haga quedará guardada en la sesión en curso, por lo que los resultados que encontrará en la búsqueda anterior, aún cambiando de sección, se mantendrán. Si se encuentran resultados, se habilitarán las siguientes opciones:

- **Ver datos de publicación:** Al hacer click en esta opción se desplegará dinámicamente una lista de los inmuebles desde el punto de vista de los datos que afectan a la publicación del mismo en la zona pública con diferentes opciones para poder acceder directamente a los datos que necesita manipular. Por ejemplo, si no tiene portada,

podrías pulsar en la columna de Portada en el valor **Sin especificar**, y accederías directamente al listado de imágenes del inmueble. Aquellas direcciones públicas (sobre las que se va a centrar el mapa de Google en la zona pública), que coincidan con las direcciones reales en las que está ubicado el inmueble, se mostrarán en negrita por si usted desea cambiarla por otra. La leyenda de las columnas más relevante es:

- **Pub.**: indica si el inmueble está o no publicado. Si pincha en su valor le llevará a la zona de publicación del inmueble.
- **Dest.**: indica si se ha marcado como Destacado. Si pincha en su valor le llevará a la zona de publicación del inmueble.
- **Oport.**: indica si se ha marcado como Oportunidad. Si pincha en su valor le llevará a la zona de publicación del inmueble.
- **URL**: indica si se ha especificado la url-seo que determina en enlace del inmueble en la zona pública. Si pincha en **Ver** le llevará al detalle público del inmueble incluso si no está publicado. Si pincha en **NO** le llevará a la zona de publicación del inmueble.
- **Portada**: muestra la imagen de portada del inmueble. Si pincha en la imagen abrirá una nueva pestaña mostrando un zoom de la misma. Si pincha en **Sin especificar**, se le derivará al listado de imágenes.
- **Img.**: muestra el número de imágenes publicadas (no el total). Si no hay imágenes mostrará 0 y si pincha sobre él, se le derivará a la zona de subida de imágenes [9.30](#). En caso contrario, se mostrará un número superior a 0 y si pincha en él, se le derivará al listado de imágenes del inmueble [9.28](#).
- **Cartel**: muestra el estado de generación en el que se encuentra el cartel. Si ha sido generado, se le derivará a la zona de edición del cartel [9.35](#) y podrá imprimirlo o modificarlo si lo desea. Si no ha sido generado, se le derivará a la zona de creación del cartel [9.34](#).
- **Vídeo**: muestra si existe video publicado del inmueble. Si no lo hay mostrará **NO** y si pincha sobre él, se le derivará a la interfaz de insertar enlace [9.33](#). En caso contrario, se mostrará un **Ver** y si pincha en él, se abrirá una nueva pestaña en el navegador con la reproducción del vídeo en Youtube.
- **Enlaces**: muestra el número de enlaces publicados (no el total). Si no hay imágenes mostrará 0 y si pincha sobre él, se le derivará a la interfaz de insertar

enlace 9.33. En caso contrario, se mostrará un número superior a 0 y si pincha en él, se le derivará al listado de enlaces del inmueble 9.32.

- **Ver mapa público:** Al hacer click en esta opción se desplegará dinámicamente un mapa de Google 9.12 sobre el cual se posicionarán todos los inmuebles que tengan dirección pública especificada. Este mapa puede tener varios usos, pero el principal de ellos es proporcionar una vista de los inmuebles general antes incluso de estar publicados. En general, al igual que con el mapa real, puede filtrar primero con la búsqueda avanzada, los inmuebles que deseé, y posteriormente usar cualquiera de estas opciones.

Adicionalmente, si pincha sobre cada inmueble, se desplegará una ventana de información tal y como la vería un visitante de la zona pública excepto por el enlace de editar un inmueble que se agrega como atajo rápido por si quiere editarlo directamente.

- **Ver mapa real:** Al hacer click en esta opción se desplegará dinámicamente un mapa de Google 9.13 sobre el cual se posicionarán todos los inmuebles respecto sus direcciones reales. Este mapa puede tener varios usos, pero el principal de ellos es proporcionar una vista de la ubicación real del mercado inmobiliario para valorar dónde captar más clientes o dónde menos.

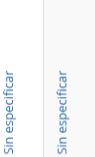
| Ref. | Tipo | Municipio | Zona | Direc. Pública | Precio Compra | Precio Alquiler | Met. Hab. | Bañ. | Pub. | Oport. | URL Ver | Portada | Img. | Cartel | Vídeos | Enlaces |
|----------------|-------------|--------------|-------------------------------------|-------------------------------|---------------|-----------------|-----------|------|------|--------|---------|---|------|--------------------|--------|---------|
| 59437772059 | Duplex | Cádiz | Ayuntamiento - Catedral | avenida amilcar barca, 2 | 100.000 | 0 | 60 | 1 | 1 | Sí | NO |  | 31 | Pendiente imprimir | NO | 0 |
| 5944ea26a1ab8 | Unifamiliar | San Fernando | | Calle real | 0 | 600 | 160 | 2 | 2 | Sí | NO |  | 2 | Pendiente generar | NO | 0 |
| 59515d52a7091 | Oficina | Almería | Catedral | Calle Obispo Orberá | 135.000 | 500 | 400 | 4 | 1 | Sí | Sí |  | 1 | Impreso | NO | 0 |
| 59515d6dd021ea | Oficina | Almería | Catedral | Calle Obispo Orberá | 135.000 | 500 | 400 | 4 | 1 | NO | Sí |  | 0 | Pendiente generar | NO | 0 |
| 599d8c666ff8 | Oficina | Almería | Catedral | Calle Obispo Orberá | 135.000 | 500 | 400 | 4 | 1 | Sí | Sí |  | 0 | Pendiente generar | NO | 0 |
| REF0001 | Piso | Almería | | | 200.000 | 0 | 50 | 2 | 1 | NO | NO |  | 10 | Pendiente generar | NO | 0 |
| REF0002 | Piso | Almería | | centro comercial mediterraneo | 300.000 | 0 | 70 | 3 | 2 | NO | NO |  | 10 | Pendiente generar | NO | 0 |
| REF0003 | Piso | Almería | | | 400.000 | 0 | 100 | 4 | 3 | NO | NO |  | 0 | Pendiente generar | NO | 0 |
| REF0005 | Oficina | Almería | Catedral | Calle Obispo Orberá | 135.000 | 500 | 400 | 4 | 1 | Sí | Sí |  | 14 | Pendiente imprimir | Ver | 0 |
| REF0006 | Unifamiliar | Cádiz | Playa S. M del Mar - Playa Victoria | mercadona | 125.000 | 0 | 60 | 1 | 1 | Sí | Sí |  | 4 | Pendiente generar | NO | 0 |

Figura 9.11: Inmuebles: Vista de publicación

Mapa con direcciones pública (vista web corporativa)

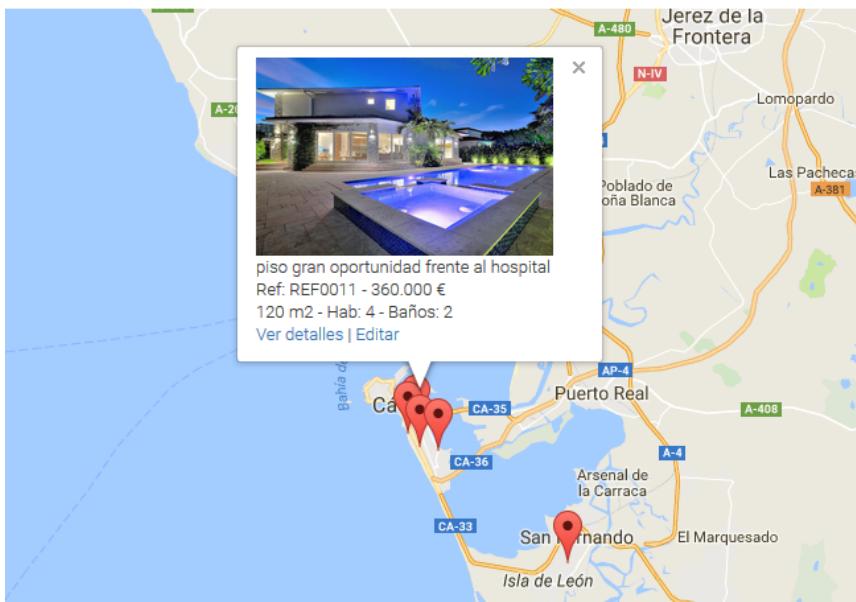


Figura 9.12: Inmuebles: Vista pública en Google Maps

Mapa con direcciones reales

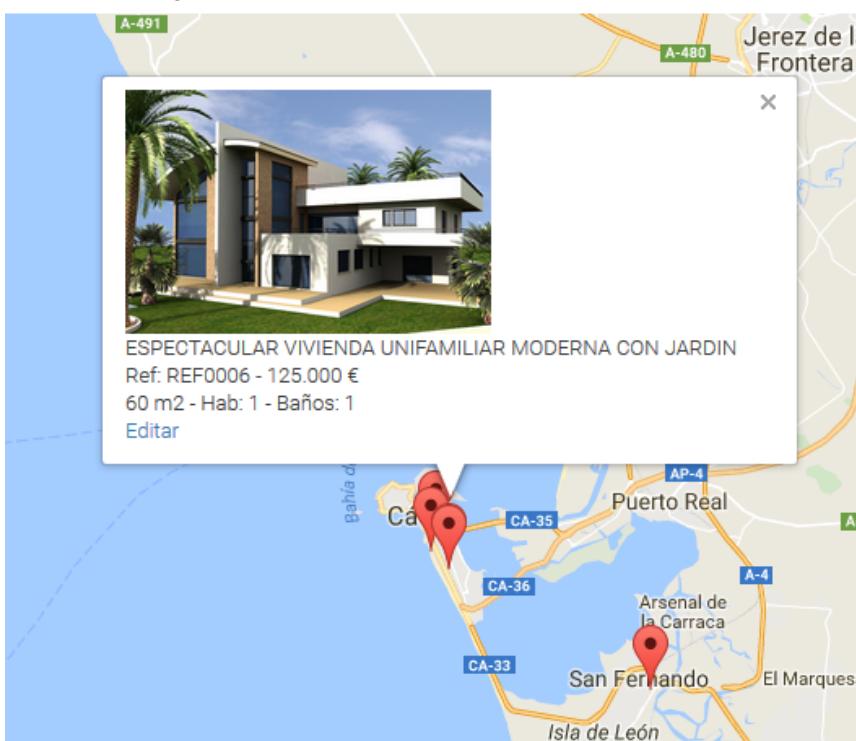


Figura 9.13: Inmuebles: Vista real en Google Maps

Uno de los detalles importantes de la visualización de los mapas de inmuebles, y que los diferencia de otras plataformas, es que además de disponer de estas dos vistas mostradas anteriormente, también es capaz de detectar y mostrar elegantemente aquellas direcciones que coinciden en un mismo punto del mapa de Google. Constar que no se comprueba si la dirección en sí es coincidente, si no si la traducción en coordenadas de la dirección del inmueble que realiza Google es exactamente la misma.

Mapa con direcciones pública (vista web corporativa)

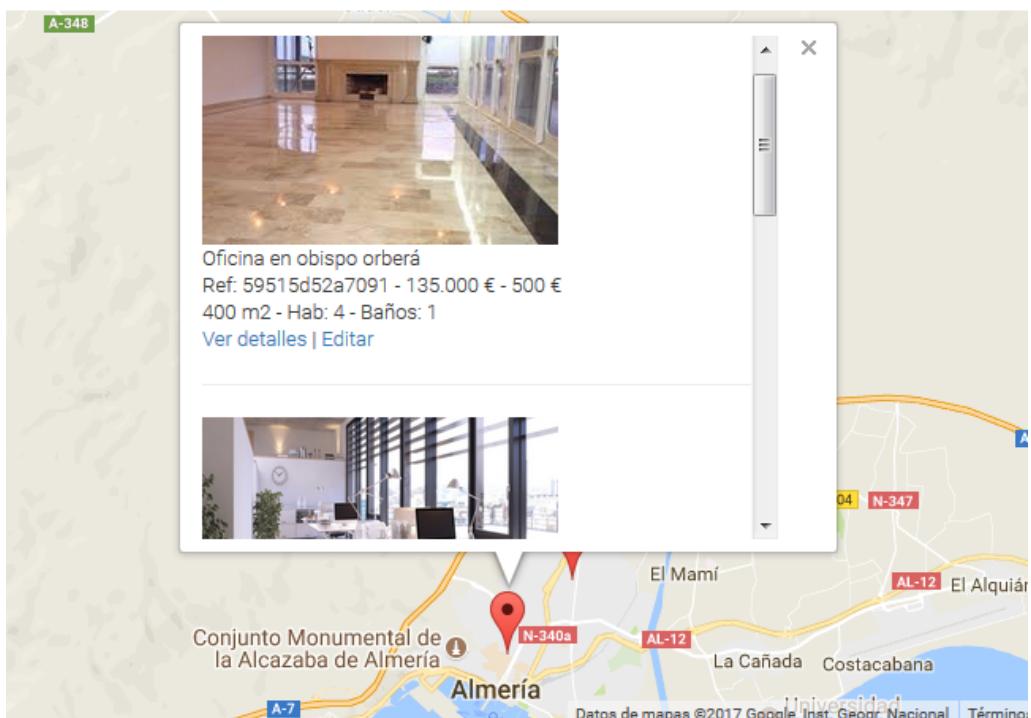


Figura 9.14: Inmuebles: Vista de coincidencias de marcas en Google Maps

Por otro lado existen otras opciones para realizar con los inmuebles mostrados desde el buscador. La exportación a CSV le permitirá obtener un listado idéntico al mostrado por la tabla por defecto de inmuebles. Si hace click en el botón **Exportar CSV** aparecerá la siguiente ventana emergente desde la cual podrá descargar el CSV con la información de los inmuebles:

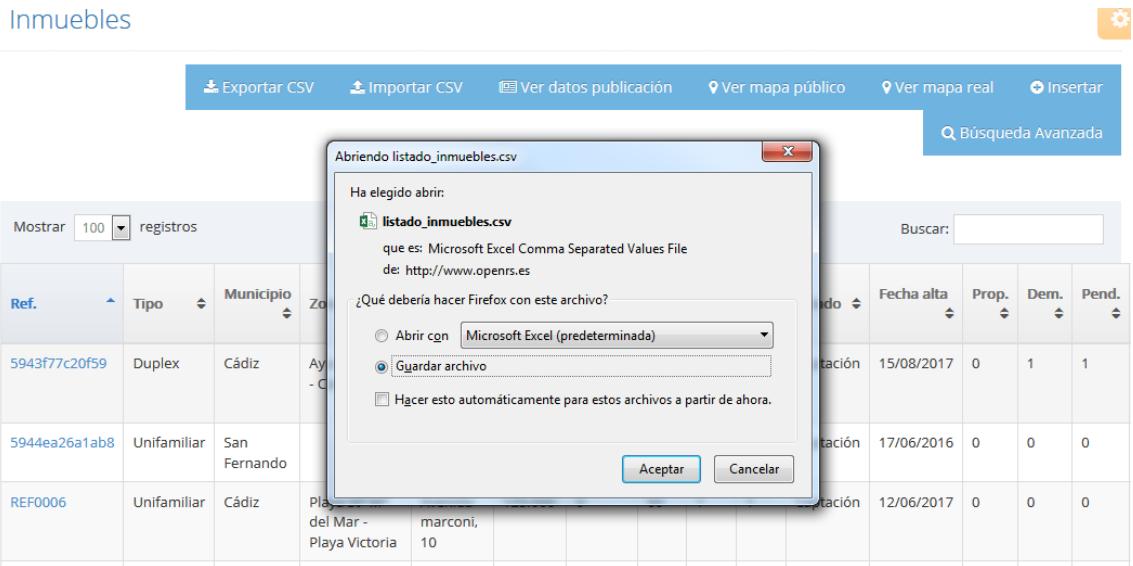


Figura 9.15: Inmuebles: Exportar CSV

El formato del CSV puede visualizarlo en los anexos a esta documentación.

Si desea registrar un inmueble, podrá hacerlo a través del botón **Insertar** que aparece en el listado de inmuebles. Se le mostrará el formulario de insertar un inmueble tal como se muestra en las figuras 9.16 y 9.17.

Nótese que cada vez que cambie el texto del campo dirección se mostrará su situación exacta en el mapa de Google para que usted pueda comprobar al instante, si la dirección introducida es correcta o no. Si no se mostrase ningún punto o apareciese un punto situado en medio del océano, significa que la dirección no ha sido mapeada correctamente desde Google Maps. Esto no impide dar de alta el inmueble pero se recomienda que se modifique la dirección hasta que se muestre un punto válido del mapa, de lo contrario, cuando desee visualizar el inmueble en cualquier mapa de Google, este no se mostrará.

Recuerde que se han habilitado sencillos buscadores en campos con múltiples registros como la provincia y la población para que le sea más simple seleccionar el valor adecuado. Una vez haya cumplimentado correctamente, al menos, los datos obligatorios, se le redirigirá a la interfaz de edición del inmueble 9.21 para ampliar su información.



| DATOS GENERALES | |
|-----------------------|--|
| (*) Referencia | REF0015 Puede introducir cualquier secuencia de caracteres alfanumérica que no esté ya registrada en el sistema |
| (*) Tipo | Ático |
| (*) Fecha de alta | 29/08/2017 Introduzca la fecha en formato dd/mm/aaaa (por ejemplo; 19/05/1982) |
| UBICACIÓN | |
| (*) Provincia | Cádiz |
| (*) Población | Cádiz |
| Zona | Centro Histórico - Plaza España |
| (*) Dirección | Calle costa rica, 1 Deberá introducir la dirección especificando el tipo de vía al principio, si lo desea, el nombre de la vía, el número de portal, piso, etc. No utilice símbolos ordinales como ° o ª. Todo separado por un espacio y comas simples ",". Si lo desea agregue el C.P. al final, también separado por ",". Por ejemplo, Calle Pinito del Oro, 4, 6 C, 04009. La población y la provincia no deben escribirse, ya que se tomarán de los datos introducidos anteriormente. |
| Ubicación Google Maps |  |

Figura 9.16: Inmuebles: Formulario de alta de inmueble (Parte 1)

| DATOS ESPECÍFICOS | |
|---|--|
| (*) Metros | <input type="text"/> |
| (*) Metros útiles | <input type="text"/> |
| (*) Habitaciones | <input type="text"/> |
| (*) Baños | <input type="text"/> |
| Precio compra | <input type="text"/> |
| Sin decimales ni signos de puntuación. Si especifica un precio positivo el inmueble estará en venta. | |
| Precio compra anterior | <input type="text"/> |
| Sin decimales ni signos de puntuación. Si especifica un precio positivo mayor que el de compra estará reflejando una bajada de precio. Si especifica un precio menor será una subida. Si no desea especificar ni subida ni bajada deje el campo a 0 o vacío | |
| Precio Alquiler | <input type="text"/> |
| Sin decimales ni signos de puntuación. Si especifica un precio positivo el inmueble estará en alquiler. El precio sólo puede ser mensual a no ser que especifique usted lo contrario en la descripción. Si además el precio de compra es positivo, el inmueble está a la venta o en alquiler. | |
| Precio alquiler anterior | <input type="text"/> |
| Sin decimales ni signos de puntuación. Si especifica un precio positivo mayor que el de alquiler estará reflejando una bajada de precio. Si especifica un precio menor será una subida. Si no desea especificar ni subida ni bajada deje el campo a 0 o vacío | |
| (*) Certificación energética | <input type="text" value="En trámite"/> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 2px;"> De acuerdo al RD 235/2013, de 5 de abril, te recordamos que, en función del tipo de inmueble de que se trate y del consumo previsto, debes indicar su nivel de certificación de eficiencia energética en el desplegable de la ficha del anuncio. Para más información puedes acceder aquí. </div> |
| Año construcción | <input type="text"/> |
| Introduzca el año en formato aaaa (por ejemplo: 1982) | |
| DATOS AUXILIARES | |
| (*) Estado | <input type="text" value="- Seleccione estado -"/> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 2px;"></div> |
| Agente Asignado | <input type="text" value="Berasuain Ruiz, Ángel Luis"/> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 2px;"></div> |
| Observaciones | <input type="text"/> |

Figura 9.17: Inmuebles: Formulario de alta de inmueble (Parte 2)

No obstante, se ha habilitado un sencillo método de importación de múltiples inmuebles. Si hace click en **Importar CSV**, se mostrará la interfaz para subir el CSV. Una vez seleccione el fichero a subir la interfaz quedará así:

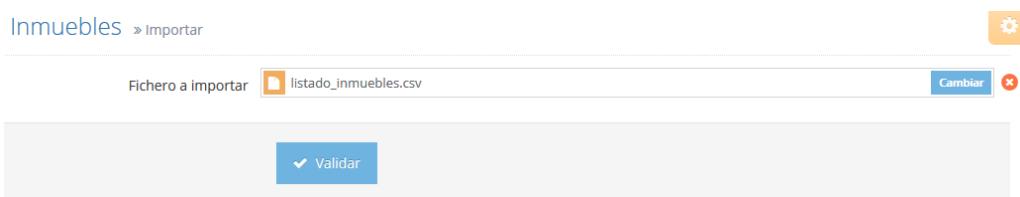


Figura 9.18: Inmuebles: Subir CSV para importación

El formato del CSV es el mismo que el que se ha utilizado en la exportación a CSV, pero eliminando los campos sobrantes que no pueden ser importados (puede visualizar los

formato en los anexos a esta documentación).

Cuando pulse en el botón **Validar** la plataforma realizar dos conjuntos de validaciones diferentes sobre los datos introducidos:

- **Validaciones sobre la conversión de los datos:** Estas validaciones tienen como objetivo validar aquellos datos pertenecen a selectores (como por ejemplo, provincias o poblaciones) así como otras comprobaciones como la repetición de la unicidad del código de referencia del inmueble dentro del propio listado CSV. Estos errores o advertencias se mostrarán en las propias celdas de la tabla de datos, tal y como muestra la imagen de la figura 9.19. Las advertencias se muestran en verde, mientras que los errores se muestran en naranja.
- **Validaciones sobre la semántica y formato de los datos:** Estas validaciones son las mismas que se realizan durante el proceso de registro manual de un inmueble a través del botón **Insertar** comentado anteriormente. Estos errores se mostrarán en la columna de **Otros errores encontrados** siempre y cuando no se hayan encontrado errores previamente en la fase anterior.

Si el registro contiene algún tipo de error, se mostrará sombreado en color rosa claro. En caso contrario, no aparecerá sombreado. Si pulsa sobre el botón **Importar Válidos**, realizará la importación sólo de aquellos registros que hayan superado las validaciones. Si pulsa sobre el botón **Importar Otro CSV**, descartará la validación actual y el sistema le redirigirá de nuevo a la funcionalidad de Importar CSV.

Finalmente, si confirma la importación de los registros válidos, la plataforma registrará los mismos y mostrará una interfaz con un resumen de la importación realizada. Esta constará del mismo listado que se mostró durante el proceso de validación pero con la diferencia que sombreará en verde aquellos inmuebles en los que ha tenido éxito registrando sus datos tal y como muestra la figura 9.20.

Inmuebles ➤ Validación Importación CSV

100 ▶ registros

Referencia ▲ Tipo ▲ Fecha Alta ▲ Provincia ▲ Municipio ▲ Zona ▲ Dirección ▲ Metros ▲ Habit. ▲ Baños ▲ Metros Útiles ▲ Cert. Energi. ▲ Precio Alquiler Ant. ▲ Precio Compra Ant. ▲ Precio Compra Últ. ▲ Año Const. ▲ Estado ▲ Observaciones ▲ Otros errores encontrados ▲

Importar Válidos Importar Otro CSV

Buscar: _____

| ímo | Referencia | Tipo | Fecha Alta | Provincia | Municipio | Zona | Dirección | Metros | Hab. | Baños | Metros Útiles | Cert. Energi. | Precio Alquiler Ant. | Precio Compra Ant. | Precio Compra Últ. | Año Const. | Estado | Observaciones | Otros errores encontrados |
|--------------------|--------------|-------------|------------|-----------|--------------|---------------------------------------|--------------------------|--------|------|-------|---------------|---------------|----------------------|--------------------|--------------------|------------|--------|---------------|---------------------------|
| borrell@correo.com | 59446a26a1a8 | Unifamiliar | 17/06/2016 | Cádiz | San Fernando | Sin especificar | Calle real, 1 | 160 | 110 | 2 | 2 | 0 | 0 | 600 | 0 | En trámite | 0 | 1982 | Captación |
| REF0012 | REF0012 | Piso | 17/08/2017 | Cádiz | Cádiz | La Paz - Segunda Agnada - Loreto | Calle García de Sola, 1 | 80 | 70 | 2 | 1 | 150000 | 0 | 0 | 0 | Exento | 0 | 1950 | Captación |
| REF0011 | REF0011 | Piso | 17/08/2017 | Cádiz | Cádiz | Sin especificar | Pablo Ruiz Picasso, 4,4J | 120 | 105 | 4 | 2 | 360000 | 0 | 0 | 0 | B | 200 | 1981 | Captación |
| REF0006 | REF0006 | Unifamiliar | 12/06/2017 | Cádiz | Lepa | Playa Sct M* del Mar - Playa Victoria | Avenida marconi, 10 | 60 | 50 | 1 | 1 | 125000 | 150000 | 0 | 0 | En trámite | 0 | 1995 | Captación |
| REF0006 | REF0006 | Duplex | 15/08/2017 | Cádiz | Cádiz | Ayuntamiento - Catedral | avenida animita barca, 2 | 60 | 50 | 1 | 1 | 100000 | 90000 | 0 | 0 | D | 19 | 1990 | Captación |

Todos los registros del 1 al 5 de un total de 5 registros

Anterior 1 Siguiente

Figura 9.19: Inmuebles: Resultado validación de CSV

Figura 9.20: Inmuebles: Resultado importación de CSV

Nótese que, además de introducir los datos del formulario de alta del inmueble, es posible asignar automáticamente el propietario del inmueble si lo especificó en la columna correspondiente del CSV.

Una vez hemos introducido los datos por cualquiera de las dos vías anteriores, podremos ampliar la información del inmueble a través de la edición del mismo. Como se muestra en la siguiente interfaz 9.21, la edición contiene un menú superior con las siguientes opciones:

- **DATOS DEL INMUEBLE:** Contiene la ficha principal del inmueble, la cual se compone de: los datos generales (que se especificaron en la interfaz de registro del inmueble), el estado de publicación en la zona pública, así como un listado de las demandas, propietarios, características y sitios cercanos asignados.
- **IMÁGENES:** Sección para poder subir y publicar imágenes.
- **ENLACES:** Sección para poder registrar y publicar enlaces de interés así como videos en Youtube.
- **CARTEL:** Sección que le permitirá crear el cartel a través de una plantilla ya prediseñada y en un determinado idioma, así como editar un cartel ya generado e imprimirla en PDF o directamente en la impresora.
- **FICHEROS ADJUNTOS:** Sección que le permitirá adjuntar cualquier tipo de fichero que le sea de utilidad.
- **FICHA DEL INMUEBLE:** Sección que le permitirá crear la ficha del inmueble a través de una plantilla ya prediseñada y en un determinado idioma, así como editar una ficha ya generada e imprimirla en PDF o directamente en la impresora. A efectos prácticos, la ficha es exactamente igual que el cartel pero no tiene la posibilidad de generar código QR.

Cada vez que se sitúe en una sección esta quedará marcada pero podrá navegar fácilmente hacia cualquier otra.

DATOS DEL INMUEBLE

IMÁGENES ENLACES CARTEL FICHEROS ADJUNTOS FICHA DEL INMUEBLE

Datos del inmueble REF0019 » Editar (Última vez actualizado: 29/08/2017)

DEMANDAS PROPIETARIOS CARACTERÍSTICAS SITIOS CERCANOS

+ Asociar a demanda existente + Crear Demanda

| Ref. | Cliente | Tipos Inmuebles | Lugar | Precios | Metros | Hab. | Baños | Observaciones | Estado Evaluación | Fecha asig. | Opciones |
|---------|-----------------------|---------------------------|---------------------------------|----------------|--------|------|-------|-----------------------------------|-------------------|-------------|----------|
| DEM0001 | Berasuain Ruiz, Angel | Piso, Unifamiliar, Duplex | Cádiz (Ayuntamiento - Catedral) | 50.000-150.000 | 50-120 | 1-2 | 1 | Desea un inmueble bonito y amplio | Pendiente evaluar | 31/07/2017 | |

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

Figura 9.21: Inmuebles: Editar (Parte 1)

Desde la pestaña de demandas y propietarios se muestran los datos asignados y se puede navegar directamente pulsando en la referencia de la demanda o en el nombre de cliente. Recuerde que el sistema ha sido diseñado de forma circular, es decir, desde cada detalle de cada sección podrá navegar a los detalles de la otra.

Datos del inmueble » Asociar demandas

Mostrar 100 registros Buscar:

| Ref. | Cliente | Tipos Inmuebles | Lugar | Precios | Metros | Hab. | Baños | Observaciones | Fecha alta |
|-------------|--------------------------|-----------------|-------|---------|--------|------|-------|----------------------|------------|
| DEM0002 | Guerrero, Diego | - | - | - | - | - | - | Oficina en el centro | 19/06/2017 |
| DEMINMUEBLE | Escalante Pérez, Antonio | - | - | - | - | - | - | - | 26/06/2017 |

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

+ Asociar demandas

Figura 9.22: Inmuebles: Asignar demandas

Además desde estas pestañas tendrá la posibilidad no sólo de navegar o editar una demanda, si no que podrá asignar nuevas demandas así como crear demandas desde cero con el inmueble ya vinculado.

Demandas ➔ Insertar 

Se ha seleccionado el inmueble con referencia [5943f77c20f59](#) para ser asignado a la demanda actual

| DATOS GENERALES | |
|-------------------|--|
| (*) Referencia | <input type="text"/> |
| (*) Cliente | - Seleccione cliente - |
| (*) Oferta | - Seleccione oferta - |
| (*) Tipo demanda | - Seleccione tipo demanda - |
| (*) Fecha de alta | <input type="text" value="29/08/2017"/> Introduzca la fecha en formato dd/mm/aaaa (por ejemplo; 19/05/1982) |

Figura 9.23: Inmuebles: Crear demanda

La interfaz de la creación de la demanda a través del inmueble sólo se diferencia de la creación normal en que muestra un mensaje indicando que se va a vincular el inmueble seleccionado a la demanda que se va registrar. Para los propietarios se siguen los mismos procedimientos.

Desde la pestaña de características y sitios cercanos podrá asignar o desasignar aquellos que considere oportunos. Sólo con hacer click o quitar la opción seleccionada quedará guardado el cambio en el inmueble, no tendrá necesidad de pulsar editar para guardar los datos del formulario completo.

| DEMANDAS | PROPIETARIOS | CARACTERÍSTICAS | SITIOS CERCANOS | | |
|----------|--------------|--|---|---|--|
| | | <input checked="" type="checkbox"/> Aire acondicionado <input checked="" type="checkbox"/> Buen estado <input type="checkbox"/> Garaje <input checked="" type="checkbox"/> Mobiliario <input type="checkbox"/> Orientación Sur <input type="checkbox"/> Por reformar <input checked="" type="checkbox"/> Última planta | <input checked="" type="checkbox"/> Altas calidades <input type="checkbox"/> Cocina <input type="checkbox"/> Interior <input type="checkbox"/> Orientación Este <input type="checkbox"/> Piscina <input checked="" type="checkbox"/> Salón comedor | <input type="checkbox"/> Armarios empotrados <input type="checkbox"/> Electrodomésticos <input type="checkbox"/> Jacuzzi <input checked="" type="checkbox"/> Orientación Norte <input type="checkbox"/> Planta Baja <input type="checkbox"/> Terraza | <input type="checkbox"/> Ascensor <input type="checkbox"/> Exterior <input type="checkbox"/> Jardín <input type="checkbox"/> Orientación Oeste <input type="checkbox"/> Planta intermedia <input type="checkbox"/> Trastero |

Figura 9.24: Inmuebles: Editar (Parte 2)

PUBLICACIÓN

A diferencia del formulario de inserción, el de edición incorpora un apartado para publicar el inmueble. Este contiene toda la información necesaria para poder definir los datos que aparecerán en la zona pública en diferentes idiomas. Primero deberá definir unos datos generales como son:

- **Publicado:** determina si está publicado el inmueble.
- **Destacado:** determina si se va a publicar en la sección de Destacados.
- **Oportunidad:** determina si se va a publicar en la sección de Oportunidad.
- **Dirección pública:** este campo determina en qué lugar va a mostrar Google Maps, en la zona pública, que está ubicado el inmueble. Usted puede indicar una zona aproximada a la real o la misma dirección real del inmueble si lo desea. Al igual que el campo dirección, cada vez que modifique su valor se consultará la posición en Google Maps y se mostrará en un mapa.

DATOS ZONA PÚBLICA (- Ver vista pública -)

| | |
|--|--|
| Publicado <input checked="" type="checkbox"/> | |
| Destacado <input type="checkbox"/> | |
| Oportunidad <input type="checkbox"/> | |
| (*2) Dirección pública <input type="text" value="avenida amilcar barca, 2"/> | |
| Ubicación Google Maps | <div style="border: 1px solid #ccc; padding: 5px; width: 500px; height: 350px; position: relative;"><div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%;">A Google Map showing the location of the property. The map is centered on Avenida Amílcar Barca, 2, which is highlighted with a red dot. The surrounding area includes the Jardines de Varela park, the El Cabezo hill, and various streets like Av. Andaluza, Av. Portugal, and Av. Juan Carlos I. The map also shows the coastline and the CA-35 highway. The Google logo is visible in the bottom left corner of the map area.</div></div> |

Figura 9.25: Inmuebles: Publicación (Parte 1)

También podrá visualizar el detalle del inmueble en la zona pública haciendo click en **Ver vista pública** incluso si no está publicado. Posteriormente, deberá definir la información en cada idioma:

Figura 9.26: Inmuebles: Publicación (Parte 2)

Los campos indicados como SEO, se utilizan para mejorar el posicionamiento de la página del detalle del inmueble en la zona pública en los motores de búsqueda. El campo descripción SEO, al ser una descripción corta, aparecerá en las ventanas de información cuando haga click sobre un inmueble mostrado en un mapa de Google. En cambio, el título y la descripción aparecerán como datos principales en los listados de la zona pública así como en su detalle.

IMÁGENES

Al acceder a la sección de imágenes se le mostrará un listado con las imágenes subidas al inmueble actual tal como muestra la siguiente figura:

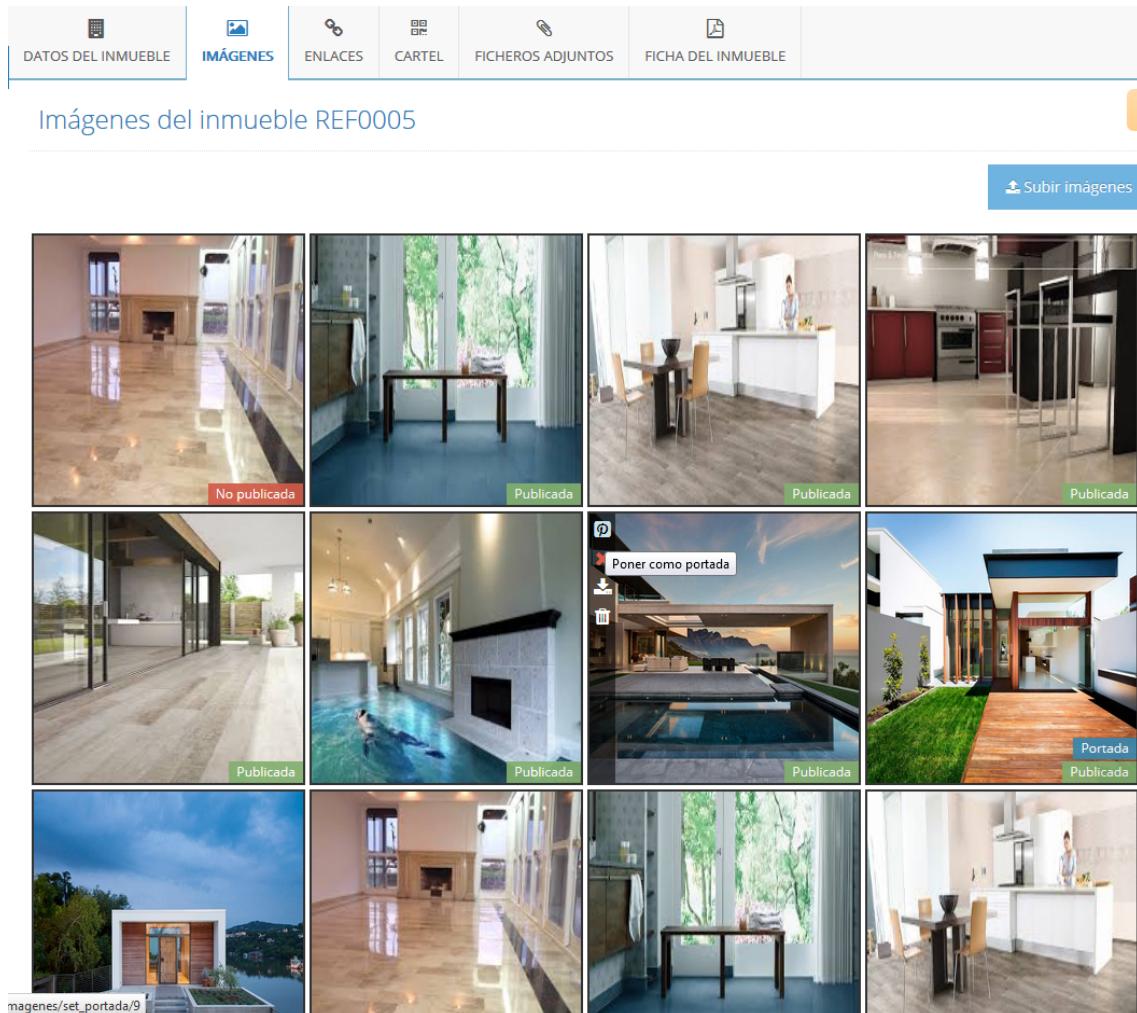


Figura 9.27: Inmuebles: Listado de imágenes

Cada imagen mostrará una marca con el estado en que se encuentra, y podrá realizar una serie de acciones sobre ella, dependiendo del estado en que esté:

- **Ponerla como portada:** Al hacer esta acción pasará a ser la imagen principal que se muestra tanto en la zona pública como en la vista de publicación.
- **Quitar de publicación:** La imagen la podrá visualizar cualquier agente inmobiliario que accede al inmueble pero no aparecerá en la zona pública.
- **Publicar:** La acción contraria a la anteriormente explicada.

- **Descargar:** Aparecerá una ventana emergente para descargar la imagen directamente a su dispositivo.
- **Borrar:** Elimina la imagen de la plataforma. No podrá ser recuperada a menos que haya una copia de seguridad realizada con anterioridad. La portada no podrá ser borrada en ningún caso.

Además, la plataforma tiene la posibilidad de mostrar este listado en diferentes formatos según el dispositivo tal y como se muestra en las siguientes figuras. Si hace click en cualquier imagen hará zoom en un elegante slider que mostrará el tamaño real de la imagen. Como podrá apreciar, en la esquina superior izquierda quedarán minimizados los menús tanto de la toda la aplicación como las secciones del propio inmueble.

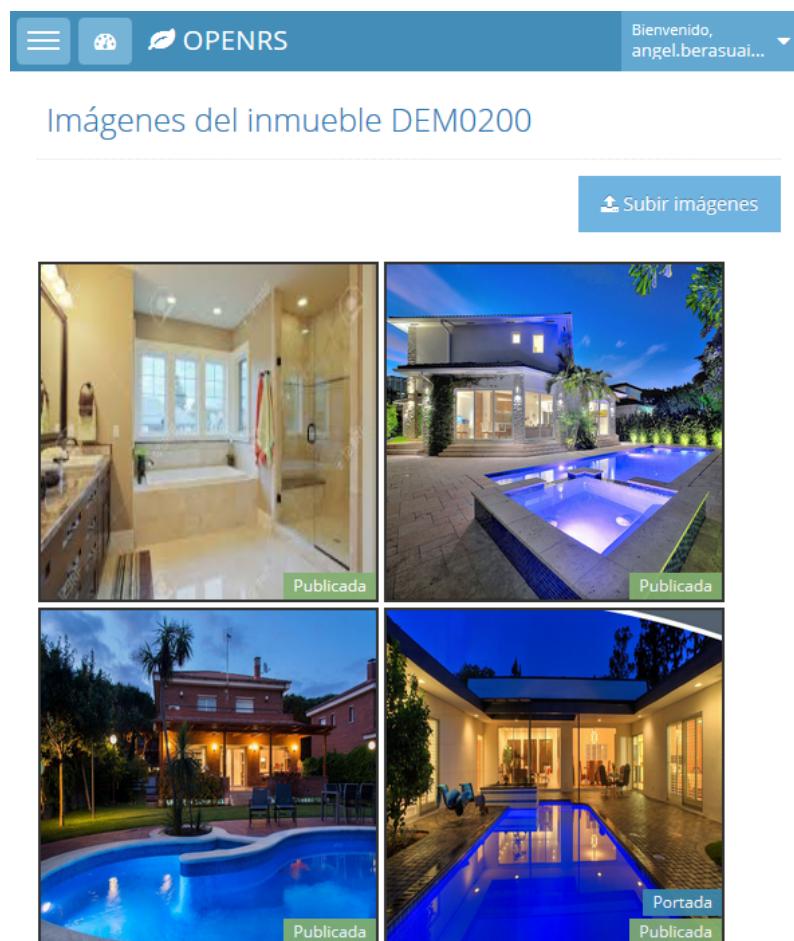


Figura 9.28: Inmuebles: Listado responsive de imágenes

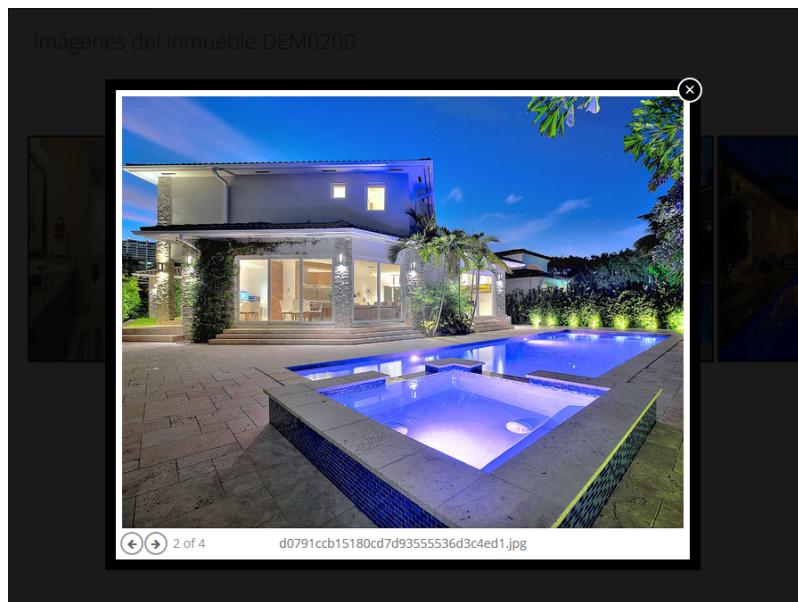


Figura 9.29: Inmuebles: Slider de imágenes

Si pulsa sobre el enlace **Subir imágenes** se le mostrará la siguiente interfaz:



Figura 9.30: Inmuebles: Subir imágenes

Desde esta podrá subir imágenes de forma dinámica, arrastrando una carpeta completa o seleccionándolas individualmente y arrastrándolas a su interior. Si arrastra una carpeta completa no se mostrará una imagen en miniatura de su progreso tal y como se muestra en la siguiente figura, si no que en su lugar aparecerá sólo el nombre del fichero:

Imágenes del inmueble DEM0200 ➤ Subir imágenes

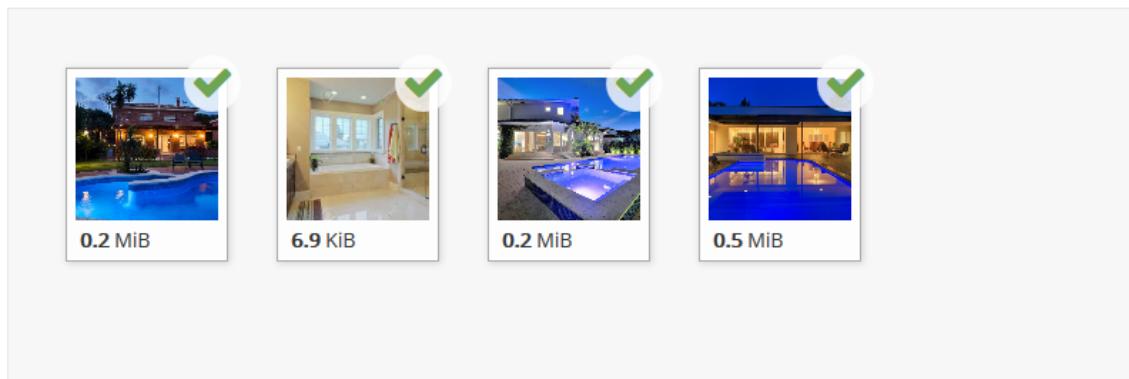


Figura 9.31: Inmuebles: Progreso al subir imágenes

Si sube imágenes que ya estaban subidas al inmueble, el software las duplicará y creará una copia con un nombre diferente. Una vez las imágenes han sido subidas, deberá hacer click en el título de la sección para volver al listado de imágenes y poder realizar las acciones que desee.

ENLACES

| Título | Publicado | Youtube | Opciones |
|----------------------------|-------------------------------------|-------------------------------------|----------|
| Video youtube NO publicado | x | <input checked="" type="checkbox"/> | |
| Video youtube publicado | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |

Figura 9.32: Inmuebles: Listado de enlaces

En la sección de enlaces podrá registrar dos tipos diferentes de enlaces externos:

- **Youtube:** Si marca esta opción estará registrando un video. Al hacerlo se hará una comprobación de que el recurso indicado sigue estando activo y que puede ser embedible dentro de un reproductor de vídeo. En caso contrario, el sistema le avisará de tal circunstancia.
- **Publicado:** Si marca esta opción estará publicado el enlace en la zona pública. Sólo puede haber un video publicado por cada inmueble pero tantos enlaces como desee

que no sean vídeo, como por ejemplo, los enlaces a las páginas de anuncio en portales inmobiliarios.

El campo título sólo es el texto descriptivo que aparecerá del enlace.

The screenshot shows a form titled 'Enlaces de REF0005 » Insertar'. It includes fields for 'Título' (Title), 'Dirección web (URL)' (Web address (URL)), and checkboxes for 'Publicado' (Published) and 'Video Youtube' (YouTube video). At the bottom are 'Insertar' (Insert) and 'Reset' buttons.

Figura 9.33: Inmuebles: Insertar enlace

CARTEL

El enlace de esta sección le redirigirá a la interfaz de creación del cartel automáticamente si este no ha sido creado con anterioridad, en caso contrario, se le mostrará la interfaz de edición. Sólo podrá generar un cartel por inmueble. Antes de generar el cartel, se recomienda que haya llenado el apartado de publicación del inmueble para que pueda generarse correctamente el código QR. Si la plantilla que seleccione como base para generar el cartel no tiene código QR, entonces no es necesario.

The screenshot shows a form titled 'Cartel de 59515d6d021ea'. It includes dropdowns for 'Plantilla' (Template) set to 'Cartel Ejemplo' and 'Idioma' (Language) set to 'Español'. At the bottom are 'Insertar' (Insert) and 'Reset' buttons.

Figura 9.34: Inmuebles: Generar cartel

Una vez haya generado el cartel en un determinado idioma conforme a una plantilla determinada, el sistema generará automáticamente el cartel y le mostrará la interfaz de edición por si desea realizar algún retoque del mismo.

Cartel de REF0005

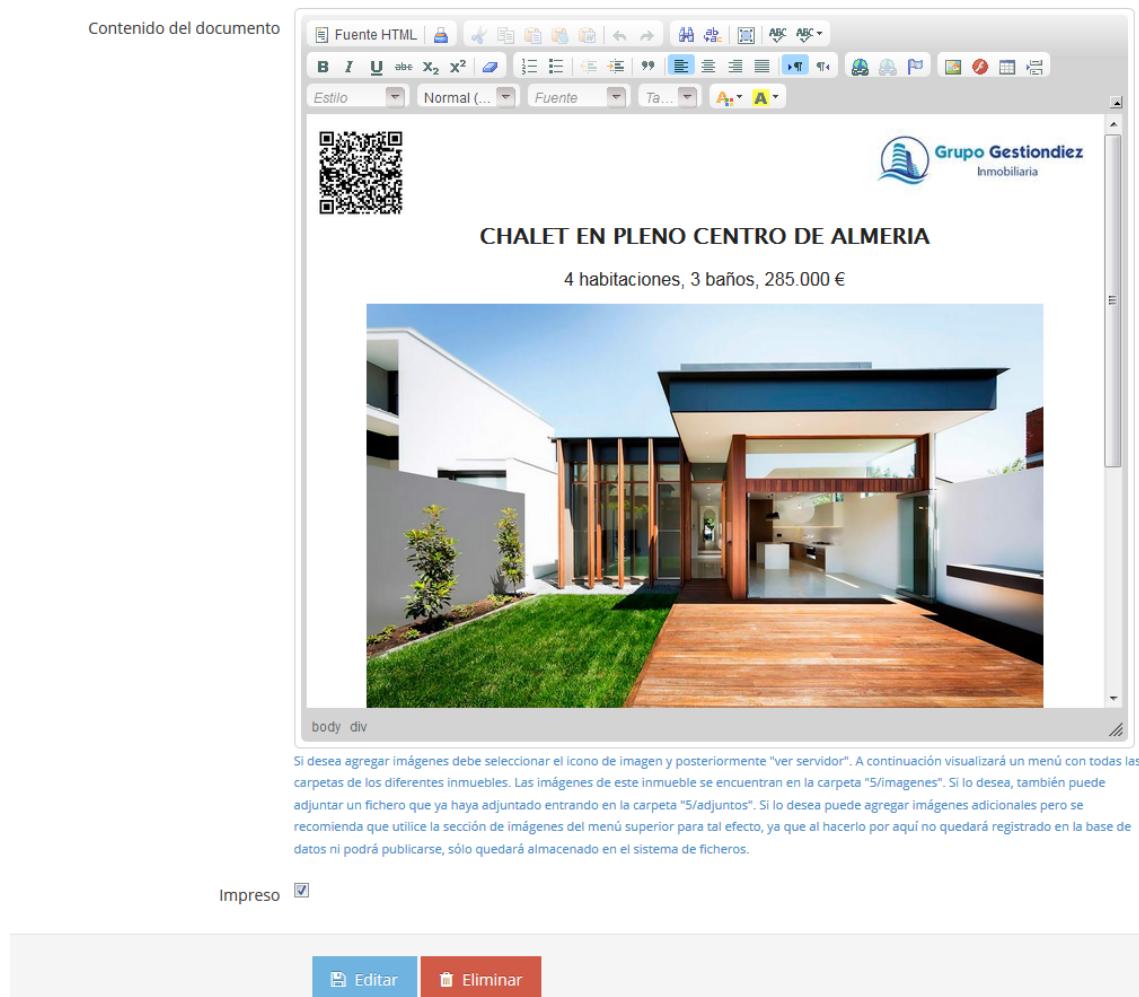


Figura 9.35: Inmuebles: Editar cartel

Puede utilizar el editor **WYSIWYG** para modificar cualquier dato e incluso agregar imágenes. También tendrá la posibilidad de indicar si ya se ha impreso el cartel, lo que indica que ya está publicado en su escaparate. Al pulsar en el botón de impresión se le abrirá la ventana de impresión. Si dispone de una impresora virtual PDF instalada en su dispositivo (lo cual hoy día es lo más común) podrá generarlo en PDF, si no sólo podrá generarlo en las impresoras a las que tenga acceso su dispositivo. Finalmente, recuerde que después de realizar cualquier cambio debe pulsar en **Editar** para que se guarden los cambios en la plataforma. Si pulsa en **Eliminar**, la plataforma borrará toda la información generada del cartel y le redirigirá a la interfaz de generación.

FICHEROS ADJUNTOS

| Ficheros del inmueble 5943f77c20f59 ➤ Listado | | |
|--|-------|----------|
| ✖ Insertar | | |
| Nombre | Tipo | Opciones |
| nota simple | Otros | |
| Mostrando registros del 1 al 1 de un total de 1 registros | | |
| Anterior 1 Siguiente | | |

Figura 9.36: Inmuebles: Listado de ficheros adjuntos

El campo **Texto del fichero** sólo es el texto descriptivo que aparecerá del fichero en los listados. Esto es necesario ya que el nombre real del fichero, por su seguridad, se renombra con un hash aleatorio, así nadie, salvo aquellos usuarios autentificados en el sistema, podrán acceder al fichero adjunto.

El campo tipo, definido por el administrador, servirá para clasificar el fichero y así tener información estadística de qué tipo de ficheros se han adjuntado y cuales no.

| | |
|---|---|
| Texto del fichero | <input type="text"/> |
| Fichero a adjuntar | <input type="button" value="Sin fichero ..."/> Seleccionar |
| Tipo | <input type="button" value="- Seleccione tipo -"/> |
| ✓ Insertar Reset | |

Figura 9.37: Inmuebles: Adjuntar fichero

FICHA DEL INMUEBLE

La ficha del inmuble funciona exactamente igual que el cartel, excepto por la salvedad de que no puede anotarse si está impreso o no y que nunca generará código QR, por lo que, los datos de publicación del inmuble no tienen efecto.

Clientes

De este apartado se ha hecho una explicación con bastante detalle en el Diseño detallado de la interfaz de usuario 5.5 por lo que no se ampliará demasiada información. Además tenga en cuenta que, todos los módulos que controla el agente inmobiliario se han diseñado siguiendo las mismas métricas y estándares, por lo que adjuntar un fichero o generar la ficha de un cliente funcionará del mismo modo y las interfaces serán las mismas. No obstante, sería necesario aclarar algunos detalles que no se explican en el diseño detallado.

La captura de pantalla muestra la sección 'Clientes' con un formulario de búsqueda avanzada. Los campos incluyen:

- Paises: - Seleccione pais -
- Agentes: Berasuain Ruiz, Ángel Luis
- Provincias: - Seleccione provincia -
- Estados: - Seleccione estado -
- Poblaciones: (campo vacío)
- Medios captación: - Seleccione medio -
- Intereses: Ofrece algún inmueble en venta
- Tipo de interés: - Seleccione - (selección actualizada a 'La demanda u oferta está vigente')
- Fecha alta: (campo vacío)
- Botones: Buscar, Limpiar
- Botones de acción: Búsqueda Avanzada, Exportar CSV, Importar CSV, Insertar

Figura 9.38: Clientes: Búsqueda avanzada

Al realizar la búsqueda avanzada, el campo **Intereses** va a consultar respecto lo que una persona tiene asociado en su ficha de cliente. Si una persona es propietaria de un inmueble que está en venta, entonces **Ofrece algún inmueble en venta**, por ejemplo. Pero claro, esos inmuebles y demandas tienen unos estados que pueden ser históricos (o antiguos) o vigentes. Para dilucidar si es algo que ocurrió en el pasado o es algo vigente, se utiliza el campo **Tipo de interés**.

Las columnas que aparecen en el listado de clientes tienen el siguiente significado:

- Ofe.: Representa el número de propiedades que tiene asignadas el usuario.
- Dem.: Representa el número de demandas que ha realizado el usuario.
- Inm. Dem.: Contará el número de inmuebles que hayan sido asignados a las demandas que haya realizado el usuario.

| | |
|--|--|
| DATOS PERSONALES | |
| (*) NIF/NIE/CIF | <input type="text"/> |
| Introduzca el número sin guiones y todo en mayúscula (por ejemplo 75777802R o G32843955) | |
| (*) Nombre | <input type="text"/> |
| (*) Apellidos | <input type="text"/> |
| Fecha de nacimiento | <input type="text"/> |
| Introduzca la fecha en formato dd/mm/aaaa (por ejemplo; 19/05/1982) | |
| CONTACTO | |
| (*) País de residencia | <input type="text" value="España"/> |
| (*1) Provincia | <input type="text" value="- Seleccione provincia -"/> |
| (*1) Población | <input type="text" value="- Seleccione población -"/> |
| (*) Dirección | <input type="text"/> <small>Deberá introducir la dirección especificando el tipo de vía al principio, si lo desea, el nombre de la vía, el número de portal, piso, etc. No utilice símbolos ordinales como º o ª. Todo separado por un espacio y comas simples ",". Si lo desea agregue el C.P. al final, también separado por ",". Por ejemplo, Calle Pinito del Oro, 4, 6 C, 04009. La población y la provincia no deben escribirse si introduce una dirección Española, ya que se tomarán de los datos introducidos anteriormente. En caso contrario, agregue la población y/o región extranjera al final de la dirección. Por ejemplo Kastaniej, 1, Copenhague.</small> |
| (*) E-mail | <input type="text"/> |
| Teléfonos | <input type="text"/> |
| DATOS AUXILIARES | |
| (*) Estado | <input type="text" value="- Seleccione estado -"/> |
| (*1) Medio captación | <input type="text" value="- Seleccione medio -"/> |
| Agente Asignado | <input type="text" value="Berasuain Ruiz, Ángel Luis"/> |

Figura 9.39: Clientes: Insertar

Por otro lado, recuerde que en el formulario del cliente, a diferencia de como ocurre con el inmueble, aunque se haga un posicionamiento en un mapa de Google de la dirección del cliente, si esta no es del todo correcta no tiene mayor relevancia, ya que, este dirección no va a ser posicionada en un mapa público ni en ningún otro salvo ese. También debe tener en cuenta que el NIF/CIF/NIE que introduzca va a ser validado sintácticamente, esto quiere decir, que debe tener una sintaxis correcta y el número debe ser válido. Esta comprobación se omitirá si el país de residencia es extranjero, ya que la numeración de cada país varía.

Demandas

El principal objetivo de esta sección es registrar el interés concreto de un determinado cliente, ya sea, de forma automática (haciendo un cruce por parte del sistema) o manual (proponiendo el agente inmobiliario un conjunto de inmuebles a su elección).

Siguiendo las mismas directrices explicadas en la sección de Clientes, sólo se van a resaltar los detalles más importantes. Este módulo también cuenta con un completo buscador tal y como se aprecia en la siguiente figura:

The form is divided into three main sections: 'Datos generales' (General Data), 'Ubicación' (Location), and 'Datos específicos' (Specific Data).
Datos generales: Contains dropdown menus for 'Clientes' (Clients), 'Tipo inmueble' (Type of property), 'Ofertas' (Offers), 'Tipo demanda' (Demand type), 'Agente asignado' (Assigned agent), 'Certificación energética' (Energy certification), and 'Estados' (States). It also includes a date range input for 'Fecha alta' (High date).
Ubicación: Contains dropdown menus for 'Provincias' (Provinces), 'Poblaciones' (Populations), and 'Zonas' (Areas).
Datos específicos: Contains input fields for 'Habitaciones' (Rooms), 'Baños' (Bathrooms), 'Precios' (Prices), and 'Metros' (Meters). Below these fields are two buttons: 'Buscar' (Search) and 'Limpiar' (Clear).

Figura 9.40: Demandas: Búsqueda avanzada

Las columnas que aparecen en el listado de demandas tienen el siguiente significado:

- **Pro.:** muestra el número total de inmuebles propuestos (tanto por parte del agente como del sistema).
- **Pend.:** muestra el número total de inmuebles que están pendientes de evaluar.
- **Vis.:** muestra el número total de inmuebles que están propuestos para visita.

Mostrar registros
Buscar:

Búsqueda Avanzada
 Exportar CSV

| Ref. | Cliente | Tipos Inmuebles | Lugar | Precios | Met. Hab. | Baños | Observaciones | Estado | Fecha alta | Pro. | Pend. | Vis. | Opc. |
|---------------|-----------------------------|--------------------------|---|----------------|--------------|-------|---------------|-------------------------|------------|------|-------|------|------|
| CADIZ0002 | De Benito Diaz, Rosa | - | Cádiz (Provincia) | - | - | - | - | Buscando Inmuebles | 29/08/2017 | 10 | 10 | 0 | |
| DEM0002 | Guerrero, Diego | - | | - | - | - | - | Oficina en el centro | 19/06/2017 | 3 | 1 | 2 | |
| DEM0003 | Morato, David | Piso, Local comercial | | 60.000-120.000 | - | - | - | Sin valorar | 21/06/2017 | 5 | 5 | 0 | |
| DEMALMERIA001 | Gómez Nieto, David | Duplex | Cádiz (Ayuntamiento - Catedral, Centro Histórico - Plaza España, Cortadura - Zona Franca) | - | - | - | - | Sin valorar | 21/06/2017 | 5 | 5 | 0 | |
| DEMINNUEBLE | Escalante Pérez, Antonio | - | | - | - | - | - | Visitando Inmuebles | 26/06/2017 | 1 | 0 | 1 | |

Mostrando registros del 1 al 5 de un total de 5 registros
Anterior
1 Siguiente

Figura 9.41: Demandas: Buscador

A diferencia de los módulos anteriores, no tendrá la posibilidad de importar los datos por

CSV. Podrá elegir dos tipos de demandas:

| DATOS GENERALES | |
|-------------------|---|
| (*) Referencia | DEM0002 |
| (*) Cliente | Guerrero, Diego - Ver datos del cliente - |
| (*) Oferta | Alquiler |
| (*) Tipo demanda | Sin filtros de búsqueda |
| (*) Fecha de alta | 19/06/2017 Introduzca la fecha en formato dd/mm/aaaa (por ejemplo; 19/05/1982) |
| DATOS AUXILIARES | |
| Estado | Sin valorar |
| Agente asignado | Berasuain Ruiz, Ángel Luis |
| Observaciones | Oficina en el centro |

Figura 9.42: Demandas: Sin filtros de búsqueda

Datos de la demanda » Proponer Inmuebles 

| Mostrar 100 registros Buscar: <input type="text"/> | | | | | | | | | | | |
|--|---------------|-----------|--------------------|-------------------------|-----------------------------|---------------|-----------------|--------|------|-------|-----------|
| | Referencia | Tipología | Municipio | Zona | Dirección | Precio Compra | Precio Alquiler | Metros | Hab. | Baños | Estado |
| <input type="checkbox"/> | REF0005 | Oficina | Almería | Catedral | Calle Obispo Orberá, 5 | 135.000 | 500 | 400 | 4 | 1 | Captación |
| <input type="checkbox"/> | 59515d52a7091 | Oficina | Almería | Catedral | Calle Obispo Orberá,1, bajo | 135.000 | 500 | 400 | 4 | 1 | Captación |
| <input type="checkbox"/> | 5943f77c20f59 | Duplex | Cádiz | Ayuntamiento - Catedral | avenida amilcar barca, 2 | 100.000 | 0 | 60 | 1 | 1 | Captación |
| <input type="checkbox"/> | HUERCAL-0001 | Chalet | Huércal de Almería | | C/Tortola, 7 | 600.000 | 0 | 500 | 4 | 2 | Captación |

Figura 9.43: Demandas: Proponer inmuebles

La primera de ellas, la demanda **sin filtros de búsqueda**, tiene como objetivo asignar inmuebles que un cliente determinado le ha solicitado explícitamente, o bien, usted cree que le pueden interesar.

| INMUEBLES PROPUESTOS | | | | | | | | | | | | | | |
|---|-----------------------|-------------|------|---------------|-----------|-----------|-------------------------|-----------------------------|---------------|-----------------|------|------|------|----------|
| Ver en mapa Proponer Inmuebles | | | | | | | | | | | | | | |
| Mostrar <input type="text" value="100"/> registros <input type="text" value="Buscar:"/> | | | | | | | | | | | | | | |
| Origen* | Estado | Fecha asig. | Obs. | Ref. | Tipología | Municipio | Zona | Dirección | Precio Compra | Precio Alquiler | Met. | Hab. | Bañ. | Opciones |
| Agente | Propuesto para visita | 30/08/2017 | | REF0001 | Piso | Almería | Rambla Lorca | calle granada | 200.000 | 0 | 50 | 2 | 1 | |
| OPENRS | Pendiente evaluar | 29/08/2017 | | REF0005 | Oficina | Almería | Catedral | Calle Obispo Orberá, 5 | 135.000 | 500 | 400 | 4 | 1 | |
| OPENRS | Pendiente evaluar | 29/08/2017 | | 59515d52a7091 | Oficina | Almería | Catedral | Calle Obispo Orberá,1, bajo | 135.000 | 500 | 400 | 4 | 1 | |
| OPENRS | Pendiente evaluar | 29/08/2017 | | 5943f77c20f59 | Duplex | Cádiz | Ayuntamiento - Catedral | avenida amilcar barca, 2 | 100.000 | 0 | 60 | 1 | 1 | |
| OPENRS | Pendiente evaluar | 29/08/2017 | | 595f8dec1f807 | Duplex | Cádiz | Ayuntamiento - Catedral | Calle Chile, 2 | 100.000 | 0 | 60 | 1 | 1 | |
| OPENRS | Pendiente evaluar | 29/08/2017 | | 59a47541758ab | Duplex | Cádiz | Ayuntamiento - Catedral | avenida amilcar barca, 2 | 100.000 | 0 | 60 | 1 | 1 | |

Mostrando registros del 1 al 6 de un total de 6 registros

Anterior 1 Siguiente

Figura 9.44: Demandas: Inmuebles propuestos

Modificar datos de inmueble propuesto

| | |
|---------------|--|
| Estado | <input type="text" value="Pendiente evaluar"/> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 2px;"> Pendiente evaluar Proponer para visita Pendiente decisión cliente Descartado por agente Interesa cliente No Interesa cliente </div> |
| Observaciones | |
| | <input type="button" value="Cancelar"/> <input type="button" value="Aplicar"/> |

Figura 9.45: Demandas: Modificar inmueble propuesto

Para ello, primero, deberá **Proponer inmuebles** haciendo click en el botón del mismo nombre. Se le mostrará un listado de inmuebles que el cliente aún no tiene asignado en la demanda y podrá asignarlos en bloque.

Una vez los inmuebles han sido propuestos se mostrarán en la pestaña correspondiente y además de visualizar sus datos podrá ver en qué estado se encuentran. Además, tendrá la posibilidad de **modificar su estado** pulsando en **editar** y sus observaciones tal como muestra la figura 9.45.

Los posibles estados que puede asignar a cada inmueble vienen definidos en la imagen 9.45.

Nótese que el sistema va a informar sólo sobre los inmuebles en los estados **Pendiente de evaluar** y **Propuesto para visita**. El resto de estados son meramente informativos para usted.

También es importante saber que la columna origen le va a informar quién ha propuesto el inmueble. Si tiene el valor **OPENRS**, habrá sido propuesto por el sistema (ahora se verá cómo), en caso contrario, pondrá **Agente**, y habrá sido producto de una asignación a través de Proponer inmuebles.

Finalmente, si desea que el sistema haga una comparativa con los inmuebles registrados, podrá optar por la demanda **con filtros de búsqueda**.

Tenga en cuenta que el cruce de los filtros establecidos con los inmuebles registrados se va a realizar automáticamente cuando se de cualquiera de estas acciones:

- Registrar o editar un inmueble.
- Duplicar un inmueble.
- Registrar o editar la demanda.
- Duplicar la demanda.

Además al cruzar, todas aquellos inmuebles pendientes de evaluar que hayan sido propuestos y tengan como origen OPENRS, serán eliminados si al evaluar el filtro de búsqueda ya no cumplen los requisitos. Si desea eliminar un inmueble propuesto por el sistema, no utilice el botón **Eliminar**, ya que el sistema, cada vez que edite la demanda volverá a emparejarlo. Para ello, mejor cámbielo a cualquier otro estado que no sea pendiente de evaluar.

| UBICACIÓN | |
|--|--|
| Provincia | Cádiz |
| Población | - Seleccione población - |
| Zona | <small>Actualmente no existen zonas registradas para la población seleccionada</small> |
| DATOS ESPECÍFICOS | |
| Tipo inmueble | <input type="checkbox"/> Ático <input type="checkbox"/> Casa rústica <input type="checkbox"/> Chalet <input type="checkbox"/> Cortijo |
| Precio (desde) | 0 |
| Precio (hasta) | 0 |
| Habitaciones (desde) | 0 |
| Habitaciones (hasta) | 0 |
| Baños (desde) | 0 |
| Baños (hasta) | 0 |
| Metros (desde) | 0 |
| Metros (hasta) | 0 |
| Año construcción (desde) | 0 |
| <small>Introduzca el año en formato aaaa (por ejemplo: 1982)</small> | |
| Año construcción (hasta) | 0 |

Figura 9.46: Demandas: Con filtros de búsqueda

9.2.4. Perfil de Administrador

Estadísticas

El administrador sólo tendrá acceso a visualizar las estadísticas generales del sistema sin acceso a las tablas de datos, ya que, no tiene actividad en la zona del agente inmobiliario.

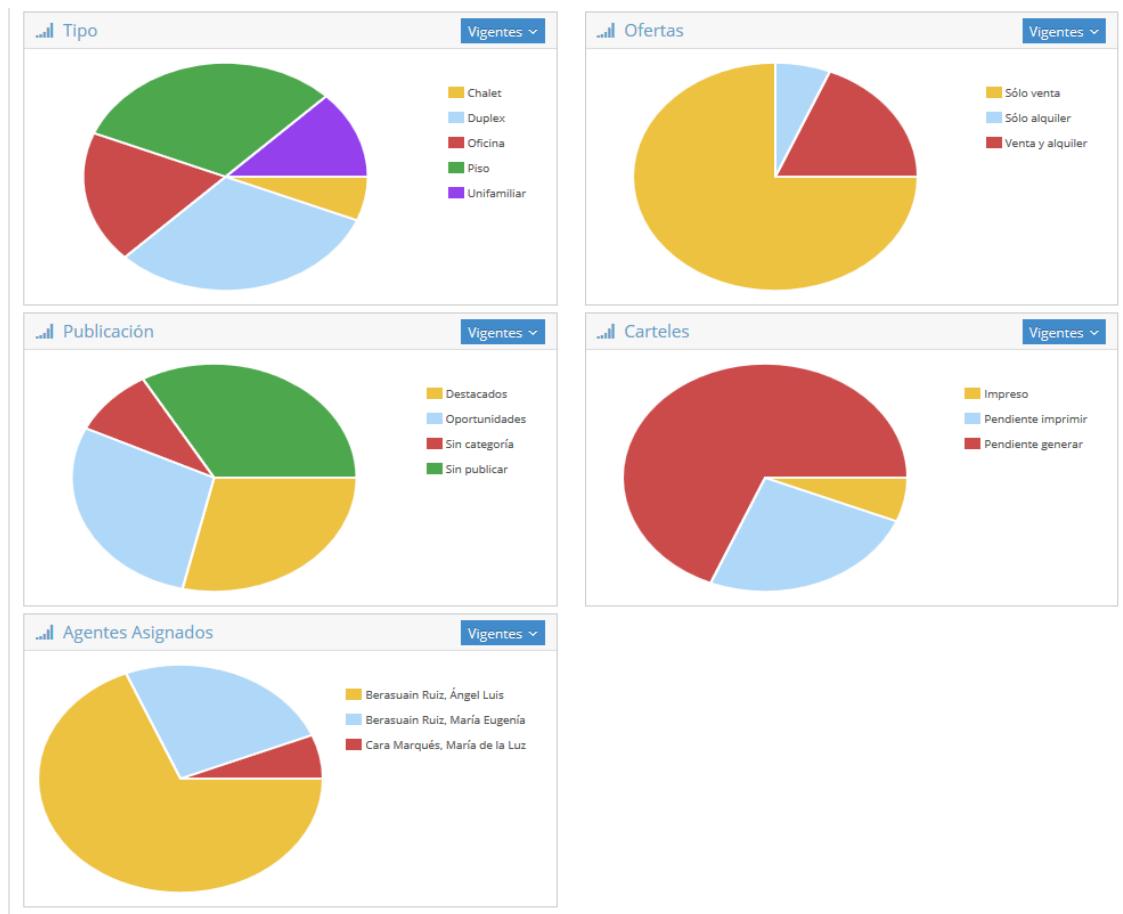


Figura 9.47: Estadísticas Generales: Inmuebles

Además, para cada pestaña, se habilitará un nuevo diagrama, **Agentes Asignados**, que permite visualizar la información agrupada según los diferentes agentes inmobiliarios registrados en el sistema.

Usuarios

En este módulo usted podrá registrar los usuarios que tendrán acceso a la plataforma. Por defecto los usuarios se darán de alta con el rol de agente inmobiliario, pero desde la edición podrá asignarle los permisos que considere oportunos. También podrá deshabilitarlos haciendo click en su estado.

Mostrar 100 registros Buscar:

| Nombre | Apellidos | Email | Grupos | Estado | Acción |
|-----------------|----------------|---------------------------|-----------------|--------|--------|
| Admin | istrator | admin@admin.com | admin | Activo | |
| Antonio | López | klaimir@hotmail.com | agente | Activo | |
| Irene | Forja Barriga | irelfita@hotmail.com | admin agente | Activo | |
| Maria de la Luz | Cara Marqués | mluz1712@gmail.com | admin agente | Activo | |
| Maria Eugenia | Berasuain Ruiz | mberasuain@gmail.com | admin agente | Activo | |
| Ángel Luis | Berasuain Ruiz | angel.berasuain@gmail.com | admin agente | Activo | |

Mostrando registros del 1 al 6 de un total de 6 registros

Anterior 1 Siguiente

Figura 9.48: Usuarios: Listado

Editar Usuario

Por favor introduzca la nueva información del usuario.

| | |
|--|--|
| Nombre: | <input type="text" value="Irene"/> |
| Apellidos: | <input type="text" value="Forja Barriga"/> |
| Teléfono: | <input type="text" value="611223344"/> |
| Contraseña: (si quieres cambiarla) | <input type="text"/> |
| Confirmar contraseña: (si quieres cambiarla) | <input type="text"/> |

Miembro de grupos

- admin
- agente

Guardar Usuario

Figura 9.49: Usuarios: Editar

La interfaz anterior es la misma que aparecerá cuando pulse sobre **Perfil** en el menú superior derecho.

Nótese que para poder eliminar un usuario no puede tener datos asociados (es decir, que esté asignado a demandas, inmueble, clientes, etc.). Esta restricción es algo que se dará en todos los apartados de la sección de la Configuración 9.2.4, ya que si no, quedarían datos inconsistentes en el sistema.

Copias de seguridad

Desde este módulo podrá realizar copias de seguridad completas (tanto de los ficheros que haya subido a la plataforma como de la base de datos) pero no restaurarlas. Para ello deberá seguir los procedimientos explicados en el Manual de Instalación 8.4.

Copias de seguridad 

| Fecha | Tipo | Realizada por | Opciones |
|---------------------|---------------|----------------------------|----------|
| 09/08/2017 09:51:47 | Base de datos | Berasuain Ruiz, Ángel Luis | |
| 09/08/2017 09:51:47 | Ficheros | Berasuain Ruiz, Ángel Luis | |
| 23/08/2017 13:54:57 | Base de datos | Berasuain Ruiz, Ángel Luis | |
| 23/08/2017 13:55:01 | Ficheros | Berasuain Ruiz, Ángel Luis | |

Mostrando registros del 1 al 4 de un total de 4 registros Anterior 1 Siguiente

Tipo de compresión:

Figura 9.50: Copias de Seguridad: Listado

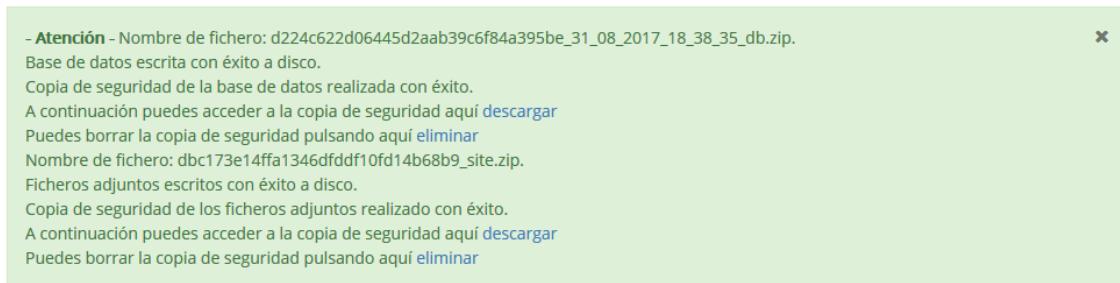


Figura 9.51: Copias de Seguridad: Realizar

Desde el propio listado tendrá la posibilidad de realizar las copias de seguridad en diferentes formatos de compresión (GZIP y ZIP) haciendo click en **Realizar copia de seguridad**. Posteriormente, recibirá el mensaje de la figura 9.51, desde el cual podrá descargarlas o eliminarlas. Estas acciones también podrá hacerlas desde el listado general.

Configuración

Desde esta sección podrá configurar los siguientes datos:

- Estados.
- Plantillas de documentación.
- Tipos de ficheros.
- Poblaciones y Zonas.
- Preferencias.
- Clientes:
 - Medios de captación.
- Inmuebles:
 - Tipos.
 - Características.
 - Sitios cercanos.

La sección de configuración de Zona pública 9.2.4, a pesar de ser parte de la configuración del sistema, como afecta sólamente a la web corporativa, se ha recogido en otro apartado. Recuerde que en la sección de **Preferencias** deberá introducir las claves obtenidas previamente de los diferentes componentes de Google que interactúan con la plataforma para que esta pueda funcionar correctamente.

Los estados representan en qué situación de gestión se encuentra un determinado inmueble, cliente o demanda.

| Mostrar 100 registros Buscar: | | | | | |
|-------------------------------|------------------------------|-------------------------------------|---|---|--|
| Ámbito | Nombre | Histórico | Descripción | Opciones | |
| Clientes | Buscando oferta | <input checked="" type="checkbox"/> | Se están realizando acciones para encontrar oferta |   | |
| Clientes | Inactivo | <input checked="" type="checkbox"/> | El cliente ya no está activo, forma parte del histórico |   | |
| Clientes | Buscando demanda | <input checked="" type="checkbox"/> | Se están realizando acciones para encontrar demanda |   | |
| Clientes | Buscando oferta y demanda | <input checked="" type="checkbox"/> | Buscando oferta y demanda |   | |
| Clientes | Sin atender | <input checked="" type="checkbox"/> | Pendiente de revisar la situación del cliente |   | |
| Demandas | Sin valorar | <input checked="" type="checkbox"/> | Aún no se ha comenzado con el proceso de la demanda |   | |
| Demandas | Buscando inmuebles | <input checked="" type="checkbox"/> | Buscando inmuebles que cumplan las expectativas |   | |
| Demandas | Visitando inmuebles | <input checked="" type="checkbox"/> | Proceso de visitas de inmuebles de interés |   | |
| Demandas | Cerrada con éxito | <input checked="" type="checkbox"/> | La demanda ha concluido con éxito |   | |
| Demandas | Pendiente de ficha de visita | <input checked="" type="checkbox"/> | Pendiente de ficha de visita |   | |
| Demandas | Cerrada con fracaso | <input checked="" type="checkbox"/> | La demanda ha concluido con fracaso |   | |
| Inmuebles | Captación | <input checked="" type="checkbox"/> | Buscando demandantes |   | |
| Inmuebles | Vendido | <input checked="" type="checkbox"/> | El inmueble está vendido |   | |
| Inmuebles | En construcción | <input checked="" type="checkbox"/> | Aún se está construyendo |   | |
| Inmuebles | Alquilado | <input checked="" type="checkbox"/> | El inmueble está alquilado |   | |

Mostrando registros del 1 al 15 de un total de 15 registros

Anterior 1 Siguiente

Figura 9.52: Configuración: Estados

Usted puede orientarlos como desee, en la imagen anterior se proponen unos cuantos por defecto, pero puede modificarlos como desee, ya que su texto no afecta a ninguna funcionalidad del sistema. No obstante, recuerde que los textos sí deben ser únicos, como ocurre con la mayoría del resto de datos configurables del sistema, ya que, en las importaciones por CSV se realiza una comparación literal en español.

No obstante, si debe tener en cuenta el check de histórico que determina que un determinado elemento ya no forma parte de la actividad actual de la inmobiliaria (en otras palabras, podríamos decir que para usted ya está inactivo).

Este check si tiene consecuencias en el sistema, como se ha explicado en la sección de Demandas 9.44, ya que, aquellos inmuebles o demandas que estén en un estado histórico no se incluirán dentro de la operación del cruce de inmuebles con demandas. Por otro lado, también tiene un valioso valor para las estadísticas, ya que cada una de ellas está ponderada por este valor.

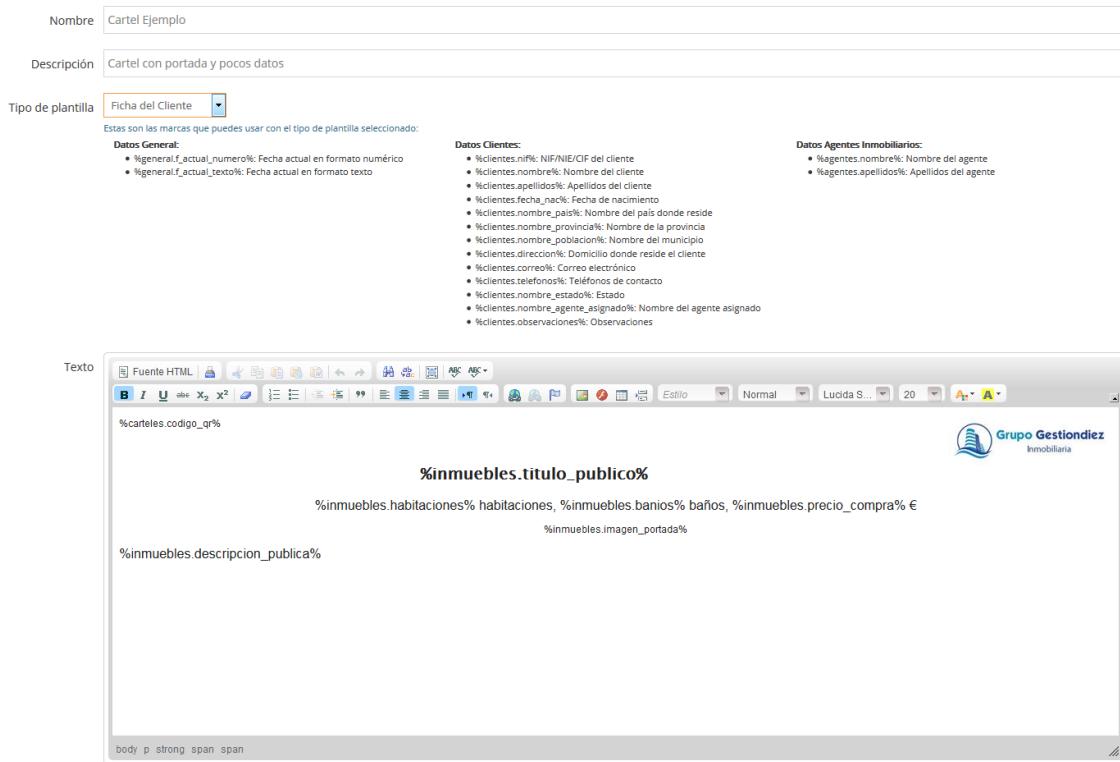


Figura 9.53: Configuración: Plantilla de documentación

Las plantillas, como se ha explicado con detalle en el Análisis, van a estar compuesta por un conjunto de marcas únicas que determinarán que valores se van a sustituir de según qué entidad de datos.

Según vaya eligiendo una determinada plantilla, se cargarán las marcas que puede utilizar. Si quiere hacer una plantilla parecida a otra ya existente, dispone de un botón en el listado para duplicarla, por lo que le ahorrará mucho trabajo.

Recuerde que a través del WYSIWYG incorporado, podrá utilizar cualquier imagen que haya sido subida en la sección de la zona pública o cargar cualquier imagen que desee, además de configurar el texto con el formato que considere más adecuado.

| Nombre | Opciones |
|---------------------------------------|----------|
| Ayuntamiento - Catedral | |
| Bahía Blanca | |
| Centro Histórico - Plaza España | |
| Cortadura - Zona Franca | |
| La Caleta - La Viña | |
| La Laguna | |
| La Paz - Segunda Aguada - Loreto | |
| Mentidero - Teatro Falla - Alameda | |
| Playa Stº Mº del Mar - Playa Victoria | |
| San José - Varela | |

Figura 9.54: Configuración: Zonas de municipios

En general, usted podrá enfocar el sistema como le sea más conveniente, poniendo los nombres que considere más adecuados y representativos para el correcto funcionamiento y gestión de su actividad diaria.

Aquellos datos que se muestran en la zona pública (como todos los de la sección de inmuebles, por ejemplo), vendrán agrupados por los diferentes idiomas activos en la plataforma y se mostrarán pestañas por cada uno de ellos de forma similar a como ocurre con la publicación del inmueble.

Configuración de zona pública

La zona pública ha sido desarrollada integrando el CMS de la empresa TIPYCOS. Por tanto, tanto la funcionalidad como sus interfaces pueden ser revisadas con detalle accediendo a la documentación de su web [24]:

- **Idiomas:** Podrá habilitar o deshabilitar así como subir los idiomas que consideré oportunos. Estos idiomas se suben a través de un fichero comprimido que incluye las diferentes traducciones. Estas traducciones deberá realizarlas usted mismo si no están subidas en el repositorio del software. Por defecto, se incluyen traducciones en inglés y en español.
- **Aspecto:** Podrá configurar datos básicos tales como el nombre de la web, sus colores corporativos así como su logotipo.
- **Pie:** El pie de página se puede dividir en tres columnas y en cada una de ellas podrá elegir una serie de tipos de datos a introducir (como su enlace a Facebook, Twitter, textos personalizados, etc.).
- **Secciones:** En este apartado podrá configurar las secciones que aparecen en el menú de cabecera así como su contenido. Siguiendo directrices sencillas para que se parezca a otros CMS muy conocidos como Wordpress, usted primero deberá dar de alta la sección y posteriormente su contenido. El contenido de cada sección se divide en bloques de unos determinados tipos. Cada tipo de bloque contendrá unas funcionalidades básicas que puede realizar. OPENRS ha ampliado la cantidad de bloques que ofrecía el software original con los siguientes:
 - **Inmuebles:** Permite mostrar inmuebles acorde a las dos categorías definidas en la sección de Publicación del inmueble, Destacados y Oportunidades. Estos podrán mostrarse en grupos de 3, 4, 6 y 8.
 - **Buscador:** Sirve para mostrar el filtro de búsqueda de los inmuebles 9.64. Una vez pulse en **Buscar**, mostrará el listado de todos los inmuebles registrados, independientemente de su categoría tal y como se muestra en la figura 9.66
- **Noticias:** Contiene toda la funcionalidad para publicar su propio Blog de noticias dividido en una serie de artículos en diferentes idiomas.
- **Ver web:** Enlace directo para visualizar el contenido actual de la web corporativa.

Si desea cambiar los colores de texto, botones, etc. Tendrá que configurar manualmente el CSS de la zona pública tal como se explica en el Manual de Implementación. 8.1.

Gestionar idiomas - OPENRS (Open Real Estate Software)



| Idioma | Seo1 | Seo2 | Activo | Opciones |
|---------|------|------|-------------------------------------|--|
| Español | es | es | <input checked="" type="checkbox"/> | <input type="button" value="Guardar"/>  |
| english | en | en | <input checked="" type="checkbox"/> | <input type="button" value="Guardar"/>  |

Figura 9.55: Gestión de Zona Pública: Idiomas

Gestionar Aspecto

Nombre web

Cabecera fija No Si

Color cabecera

Color fuente cabecera

Color borde cabecera

Color fondo

Color fuente fondo

Color pie

Color fuente pie

Logo actual



GestiCadiz
Servicios Inmobiliarios

Modificar logo No Si

Imagen

Figura 9.56: Gestión de Zona Pública: Aspecto

Listar secciones



Ordenar Nueva

| Prioridad | Título | Menu | Estado | Opciones |
|-----------|-----------|------|-----------|----------|
| 1 | INICIO | 1 | Publicado | |
| 2 | NOSOTROS | 1 | Publicado | |
| 5 | CONTACTO | 1 | Publicado | |
| 4 | SERVICIOS | 1 | Publicado | |
| 3 | NOTICIAS | 1 | Publicado | |

Figura 9.57: Gestión de Zona Pública: Secciones

Español
english

Título

INICIO

URL SEO

inicio

Descripción SEO

OpenRS

Palabras clave

inicio, portada, openrs

Título SEO

Inicio - OpenRS

Estado

Publicado

Tipo sección

auto manual blog contacto desplegable

Visible menú

Visible footer

Desplegable

Figura 9.58: Gestión de Zona Pública: Detalle de sección

Paso 1 - Datos del bloque

Paso 2 - Contenido del bloque

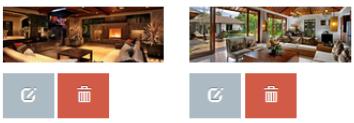
Datos de galería

| | |
|-----------------------|----------|
| Tipo de galería | Carrusel |
| Imágenes por página | 9 |
| Nº máximo de imágenes | 0 |
| Número de columnas | 3 |

Guardar

Imágenes

Ordenar imágenes



Categorías

No hay categorías para el carrusel seleccionado.

Añadir imagen

Figura 9.59: Gestión de Zona Pública: Tipo de bloque Slider

Listado de artículos

Mostrar 100 registros

Buscar:

Creado **Visto / Votos** **Imagen** **Título** **Url Seo** **Categorías** **Etiquetas** **Estado** **Opciones**

| Creado | Visto / Votos | Imagen | Título | Url Seo | Categorías | Etiquetas | Estado | Opciones |
|------------------------|---------------|---|---|-----------------|------------|-----------|-----------|---|
| 2017-08-30 17:11:48 | 17 / 1 |  | Hola mundo | hola_mundo | | | Publicado |   |
| 2017-08-31 20:13:45 | 9 / 0 |  | Mapa por provincias para saber dónde se vendieron más y menos viviendas nuevas y usadas | mapa_provincias | | | Publicado |   |

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

Figura 9.60: Gestión de Zona Pública: Noticias

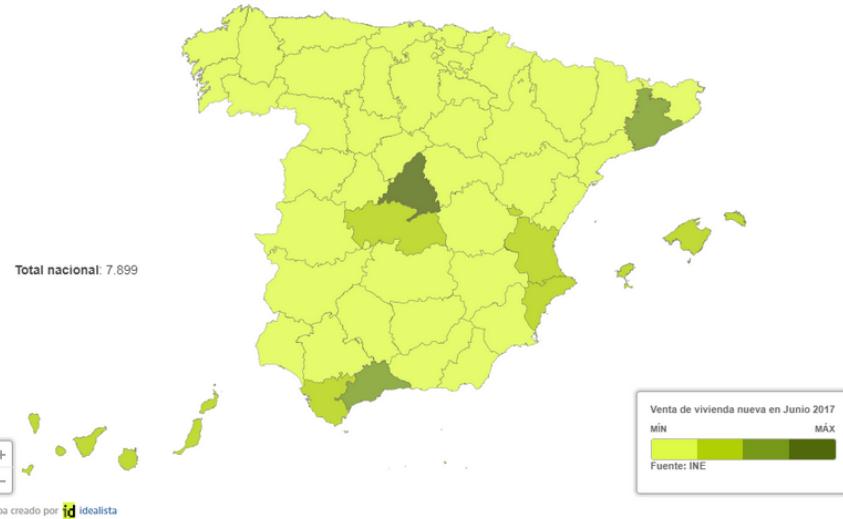
Español english

Editar Artículo

Título Mapa por provincias para saber dónde se vendieron más y menos viviendas nuevas y usadas

Fecha de creación 2017-08-31 20:13:4

Imagen actual **La venta de vivienda nueva por provincias**
(Junio 2017)



Modificar logo No Si

Imagen Seleccionar archivo (.png, .jpg, .jpeg)
 No se ha seleccionado ningún archivo.

Url Seo

Figura 9.61: Gestión de Zona Pública: Detalle Noticia

9.3. Web Corporativa

A continuación se va a proponer una plantilla de la zona pública según los elementos configurados en la sección anterior. No obstante esta configuración es sólo una sugerencia de cómo podría quedar organizada la web corporativa. El administrador, según los componentes que se han explicado anteriormente, tendrá bastante libertad para ir modificando, como considere oportuno, el aspecto, la organización y el contenido de la misma.

En este diseño inicial se ha configurado una **Cabecera** con la secciones que se proponen a continuación y el logo corporativo.

Posteriormente, se ha creado una sección de **Inicio** de tipo Manual con cuatro bloques:

- **Slider 9.63**: compuesto por un bloque tipo Carrusel.
- **Buscador 9.64**: compuesto por un bloque tipo Buscador. Este bloque, al hacer cualquier búsqueda muestra sus resultados [9.66](#).
- **Oportunidades 9.65**: compuesto por un bloque tipo Inmuebles y configurado para mostrar 3 ofertas.
- **Destacados**: compuesto por un bloque tipo Inmuebles y configurado para mostrar 3 destacados.

El **Detalle** de cada inmueble [9.70](#) utiliza una plantilla fija. No obstante, según los datos que haya configurado en la zona privada se irá mostrando la cantidad de información necesaria. Se han incluido dos secciones de tipo Manual con un bloque de tipo Texto, **Servicios 9.76** y **Nosotros 9.75** para promocionar la empresa.

Se ha creado una sección de tipo Blog compuesto por un bloque del mismo tipo, que contiene toda la funcionalidad para interactuar con las **Noticias 9.77**.

Se ha incluido una sección de tipo **Contacto** con un bloque de tipo Texto que contiene un Iframe con los datos de localización de la inmobiliaria por Google Maps [\[46\]](#).

9.3.1. Cabecera

Figura 9.62: Web Corporativa: Cabecera

9.3.2. Inicio



Figura 9.63: Web Corporativa: Slider

| | | | | | |
|------------------|---------------------|---------------------|--------------|------------|---------------|
| Referencia | - Seleccione oferta | - Seleccione tipo - | Cádiz | Cádiz | Bahía Blanca |
| - Habitaciones - | - Baños - | Precio desde | Precio hasta | Sup. desde | BUSCAR |

Figura 9.64: Web Corporativa: Buscador de inmuebles

OPORTUNIDADES

| | | |
|---|--|--|
| VENTA Casa grande en pleno centro de Cádiz En esta vivienda podrás disfrutar de la tranquilidad de saber que tus hijos juegan sin peligro, mien... Ref. REF0004 90 m ² 1 2 50.000,00 € VER | ALQUILER ESPECTACULAR VIVIENDA UNIFAMILIAR ESPECTACULAR VIVIENDA UNIFAMILIAR MODERNA CON JARDÍN Ref. 5944ea26a1ab8 160 m ² 2 2 600,00 € / mes VER | VENTA OFICINA MUY CENTRICA Y LUMINOSA OFICINA EN PLENO CENTRO DE ALMERIA NUEVA A ESTRENAR Ref. 59515d6d021ea 400 m ² 2 4 135.000,00 € VER |
|---|--|--|

Figura 9.65: Web Corporativa: Oportunidades

GestiCadiz
Servicios Inmobiliarios

INICIO NOSOTROS NOTICIAS SERVICIOS CONTACTO ES ▾

Referencia - Seleccione oferta - Seleccione tipo - Almeria
 - Habitaciones - - Baños - Precio desde Precio hasta Sup. desde **BUSCAR**

Ver en mapa



VENTA Y ALQUILER
OPORTUNIDAD

CHALET EN PLENO CENTRO DE ALMERIA

Chalet en pleno centro de almería con varias dependencias

Ref. REF0005



VENTA Y ALQUILER
OPORTUNIDAD

Oficina en obispo orberá

Oficina en obispo orberá

Ref. 59515d52a7091



VENTA

PRECIOSO CHALET CON PISCINA EN HUERCAL DE ALMERIA

CHALET CON 4 DORMITORIOS, GARAJE Y PISCINA

Ref. HUERCAL-0001

Figura 9.66: Web Corporativa: Resultados de búsqueda

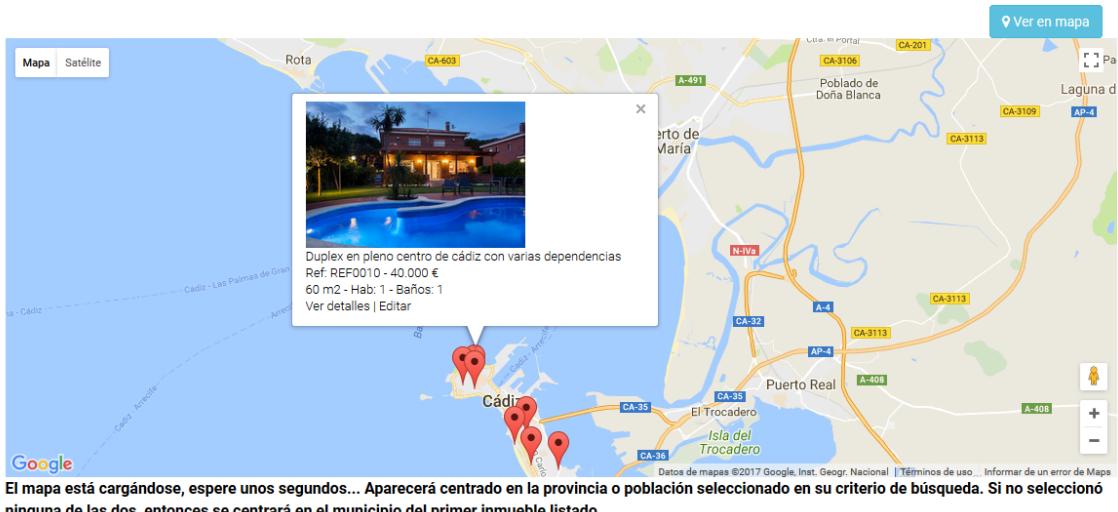


Figura 9.67: Web Corporativa: Mapa interactivo

9.3.3. Detalle del inmueble

El inmueble se encuentra dividido en dos zonas:

- La primera contiene un menú con la información de la ubicación, galería de imágenes, video, enlaces y contacto.
- La segunda se correspondería con la ficha del inmueble.

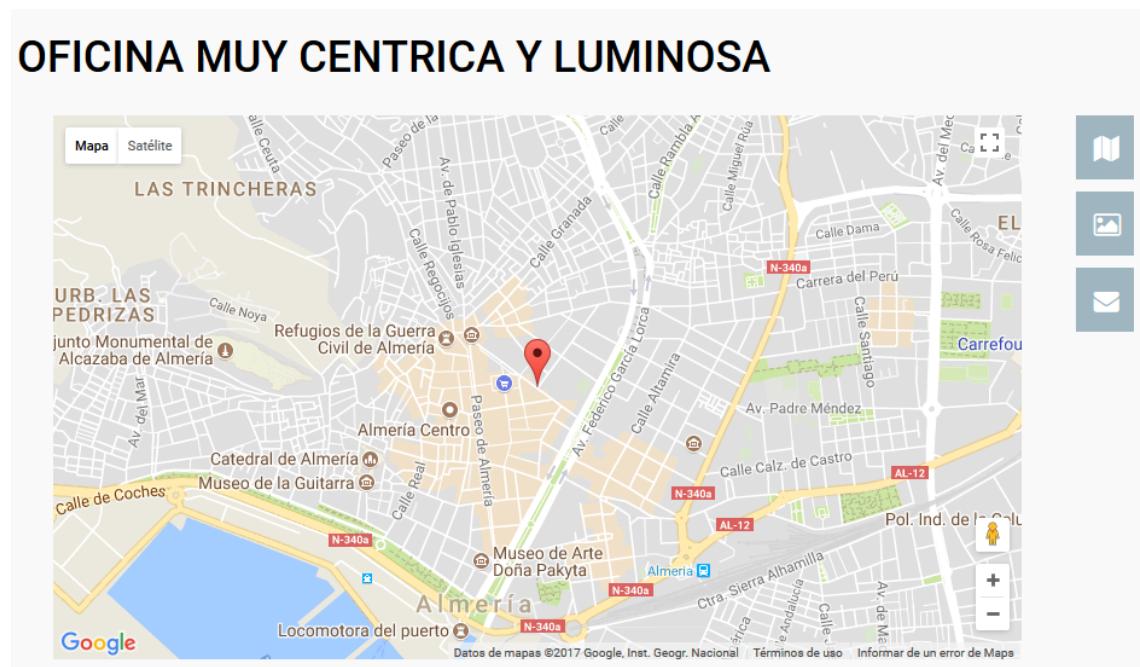


Figura 9.68: Web Corporativa: Mapa

CHALET EN PLENO CENTRO DE ALMERIA



Figura 9.69: Web Corporativa: Vídeo

CHALET EN PLENO CENTRO DE ALMERIA

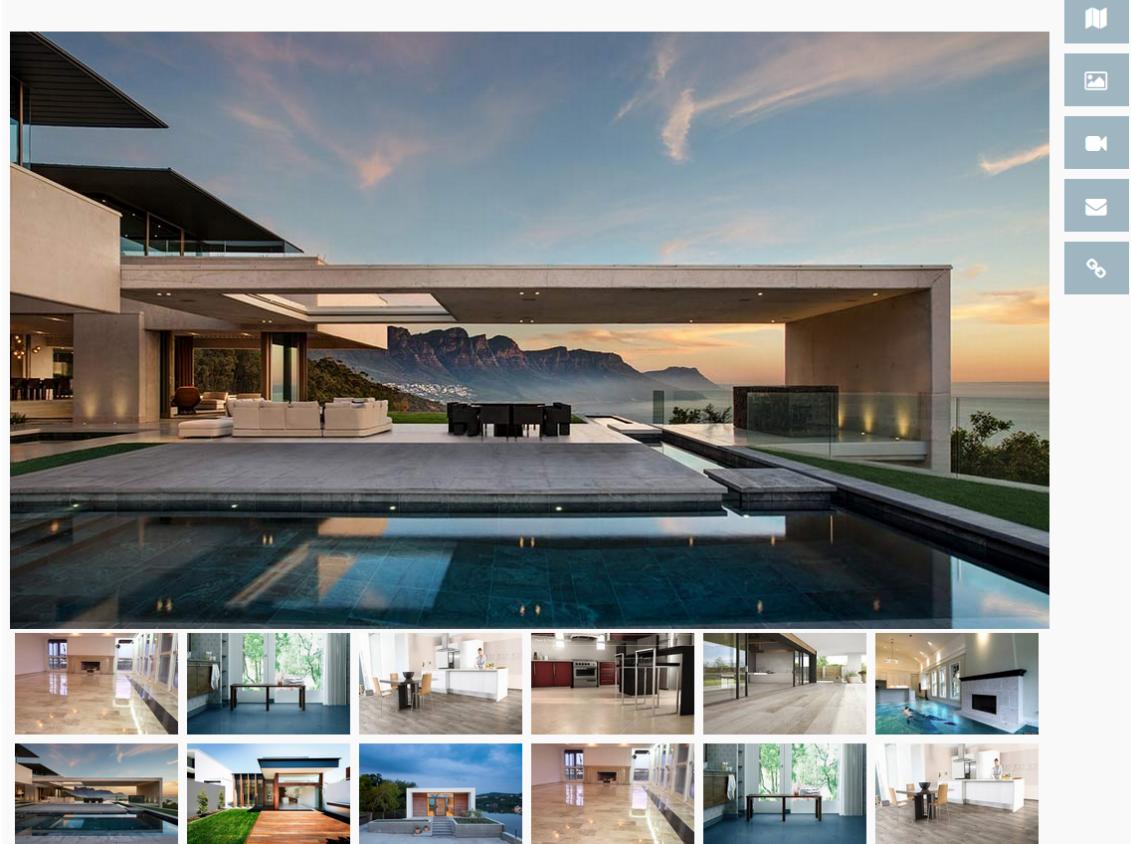


Figura 9.70: Web Corporativa: Galería de imágenes

CHALET EN PLENO CENTRO DE ALMERIA

- Ver inmueble en Idealista -
- Ver inmueble en Fotocasa -

Figura 9.71: Web Corporativa: Enlaces

En la siguiente imagen se muestra como el formulario de contacto está protegido por un captcha de Google. Normalmente no tendrá que introducirlo ya que estará oculto en el margen inferior derecho, pero a veces se solicita que introduzca unos determinados datos.

The screenshot shows a contact form for a chalet listing. On the left, there are input fields for name ('Antonio López'), email ('antonio@prueba.com'), and a message ('dadasdasdasdad'). A green 'Enviar' button is at the bottom. To the right is a CAPTCHA challenge titled 'Selecciona todas las imágenes de escaparates de tiendas'. It displays a 3x3 grid of nine images, with the middle column being the correct ones. Below the grid are three icons: a circular arrow, a person, and a circle with a dot. To the right of the icons is a blue 'VERIFICAR' button. At the top right of the page are 'CONTACTO' and 'ES' dropdown menus. On the far right, there is a vertical column of five icons: a map, a photo, a video camera, and an envelope.

Figura 9.72: Web Corporativa: Contacto del inmueble

A diferencia de la zona de contacto, el correo remitido desde el inmueble vendrá con la referencia indicada para poder visualizar el interés del mismo, aunque ambos correos se remitirán a la misma cuenta de usuario especificada en la configuración del sistema.

A continuación, se muestra el resto de datos del inmueble que abarcan su precio, descripción, sitios cercanos, características y su certificación energética. Todos el detalle del inmueble puede ser visualizado por el agente inmobiliario desde la sección de Publicación del inmueble incluso antes de estar publicado, lo cual, le permite tener una vista preliminar de cómo quedaría el inmueble.

Descripción

Este **amplio ático** nos recibe con un amplio salón comedor, una espaciosa cocina, y una amplia terraza con orientación sur y oeste, desde esta terraza se tienen vistas al mar Mediterráneo. La cocina es amplia, con un diseño abierto hacia el salón, y el área de comedor tiene grandes ventanas y acceso a la terraza. Las habitaciones están en un área independiente, ambas con orientación al sur, y con posibilidad de incluir accesos directos a la terraza desde las habitaciones. Este área independiente la completa un baño y un espacio de lavado.

El ático tiene una terraza-solarium privada de 43 metros cuadrados, con vistas al sur y al oeste. En esta terraza se disfrutan de espectaculares vistas al mar Mediterráneo y a las montañas al norte de Torrox.

La ubicación permite disfrutar de las playas de Torrox Costa, que se encuentran a tan sólo 50m., con la tranquilidad de estar en la zona de El Peñoncillo, y a solo 5 minutos de bancos, supermercados y todos los servicios necesarios.

Precio compra

~~300.000,00 €~~

~~285.000,00 €~~

Precio alquiler

~~1.525,00 € / mes~~

~~1.000,00 € / mes~~

Datos Generales

| | |
|-----------------------|--------|
| Tipo Propiedad | Chalet |
| Superficie Construida | 300 |
| Superficie Útil | 220 |
| Baños: | 3 |
| Dormitorios: | 4 |

Figura 9.73: Web Corporativa: Datos del Inmueble (Parte 1)

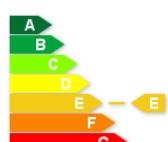
Características

| | |
|-------------------|--------------------|
| Cocina | Mobiliario |
| Ascensor | Aire acondicionado |
| Interior | Exterior |
| Orientación Oeste | Última planta |
| Buen estado | |

Sitios Cercanos

| | |
|--------------|---------------------------|
| Colegio | Hospital |
| Farmacia | Bus |
| Restaurantes | Zonas de interés cultural |

Calificación energética



84 kWh

Figura 9.74: Web Corporativa: Datos del Inmueble (Parte 2)

9.3.4. Nosotros

Bienvenidos a Gesticadiz.es

GESTICADIZ, nace en el 2001, nuestro personal se ha dedicado con firme propósito en dar un servicio inmobiliario integral a nuestros clientes y a mejorar día a día para dar el asesoramiento más comprometido.

Si desea **ALQUILAR o COMPRAR** una vivienda, plaza de garaje, trastero, solar, terreno ... Venga a vernos, seguro que en nuestra cartera de inmuebles tenemos lo que le interesa. Queremos dar satisfacción a la demanda actual y lo que es más importante, queremos seguir estando al servicio de nuestros clientes. Para resolver cualquier duda envíenos un MAIL a gesticadiz@gmail.com, con las necesidades que desean para el alquiler o compra de su inmueble: espacio, división, planta, años de construcción, situación, calidades, precio... y nos pondremos a trabajar inmediatamente para contestarle en la mayor brevedad posible y así hacer posible su deseo.



Figura 9.75: Web Corporativa: Nosotros

9.3.5. Servicios



Cádiz y Provincia

Nuestros trabajos abarcan diversos campos:

- Proyectos Obra nueva.
- Proyectos de Rehabilitación.
- Gestión y Dirección de obras.
- Adecuación de locales.
- Delineación y Diseño de interiores.



Alquileres o ventas:

Nuestra empresa Gesticadiz es la inmobiliaria de Cádiz con la mejor oferta de la provincia para comprar, alquilar o vender pisos, casas, apartamentos, oficinas, locales, etc. Tenemos amplia experiencia en el asesoramiento para la venta o alquiler de su inmueble. Nuestras capacidades de negociación son clave para que nos prefiera, a la hora de asegurar y maximizar sus ventajas competitivas en el negocio.

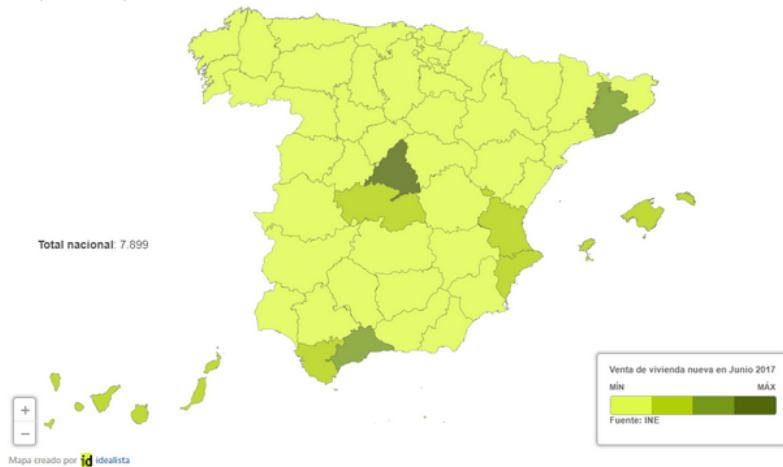
Figura 9.76: Web Corporativa: Servicios

9.3.6. Noticias

Mapa por provincias para saber dónde se vendieron más y menos viviendas nuevas y usadas

Descripción de Mapa por provincias para saber dónde se vendieron más y menos viviendas nuevas y usadas
[...Seguir leyendo](#)

La venta de vivienda nueva por provincias
(Junio 2017)



MÁS VISTOS

[Hola mundo](#)

Mapa por provincias para saber dónde se vendieron más y menos viviendas nuevas y usadas

MÁS VOTADOS

[Hola mundo](#)

CATEGORÍAS

[Etiquetas](#)

Hola mundo

[hola mundo](#) ...[Seguir leyendo](#)



Figura 9.77: Web Corporativa: Listado de noticias

Como podrá apreciar en las imágenes, el blog de noticias incluye muchísimas funcionalidades, como especificar etiquetas, navegar entre las diferentes noticias, añadir comentarios, o tuitearlos, visualizar un detalle completo de la noticias, etc.

El pasado mes de junio se vendieron 44.135 viviendas, un 19,3% más que hace un año, pero un 1,4% menos que el mes anterior, según datos del INE. Aunque la venta de casas de segunda mano es el líder indiscutible, se vendieron 7.899 casas nuevas, niveles que no se veían desde enero de 2015. Y Madrid se lleva la palma al acaparar un 23% del total de viviendas nuevas transmitidas. Por su parte, se transmitieron 36.236 casas usadas, un 19,2% más interanual y Madrid volvió a ser líder.

Para ver de cerca cómo se han repartido las compraventas de viviendas y qué provincias han sido las más activas hemos preparado dos mapas interactivos con las transmisiones de casas usadas y nuevas.

Cabe destacar el papel de la provincia de Madrid, que ha sido la protagonista en junio: en compraventa de vivienda nueva ha acaparado el 23% del total de operaciones, al registrar 1.815 ventas de obra nueva. Le siguen, pero de lejos, Barcelona, con 589 compraventas, y Málaga con 521 operaciones.

En cambio, en el lado contrario, se sitúan Melilla con una vivienda nueva vendida, Zamora, con 6 casas a estrenar transmitidas, y Cuenca, con 8 operaciones.

En cuanto a la venta de viviendas usadas, Madrid volvió a liderar las operaciones con 5.483 unidades, un 15% de las 36.236 casas traspasadas. Le siguen, de nuevo, Barcelona, con 4.327 ventas y Málaga con 2.305 operaciones. En el otro lado de la balanza, se encuentran Ceuta con 17 ventas, Melilla, con 44, y Soria, con 46 casas vendidas.



Etiquetas:

[<< Hola mundo](#) |

ARTÍCULOS RELACIONADOS

Escribe un comentario

Email

Nick

comentario

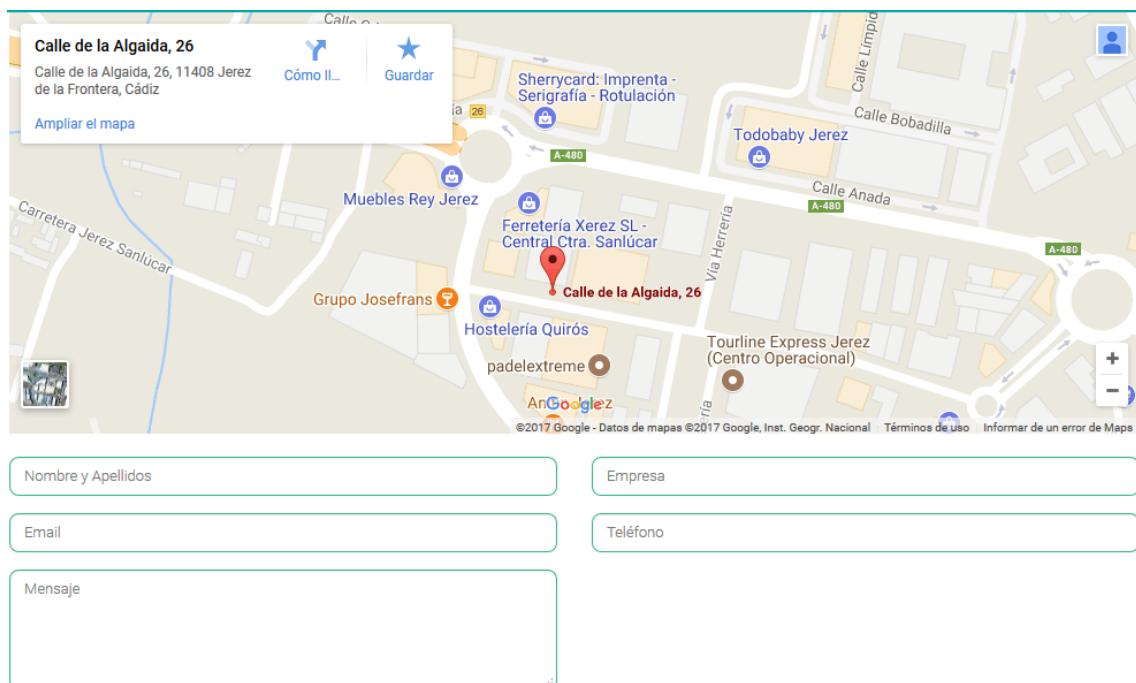
[Enviar](#)

Comentarios

No existen comentarios para mostrar

Figura 9.78: Web Corporativa: Detalle de noticia

9.3.7. Contacto



A screenshot of a Google Maps interface showing the location of 'Calle de la Algaida, 26'. The map includes labels for 'Calle de la Algaida, 26', 'Sherrycard: Imprenta - Serigrafia - Rotulación', 'Ferretería Xerez SL - Central Ctra. Sanlúcar', 'Todobaby Jerez', 'Muebles Rey Jerez', 'Grupo Josefrans', 'Hostelería Quirós', 'padelextreme', 'Tourline Express Jerez (Centro Operacional)', and 'Calle Limpia'. Below the map are several input fields: 'Nombre y Apellidos' (Name and Surname), 'Email', 'Mensaje' (Message), 'Empresa' (Company), 'Teléfono' (Phone), and a large green 'Enviar' (Send) button.

Figura 9.79: Web Corporativa: Contacto

9.3.8. Pie de página

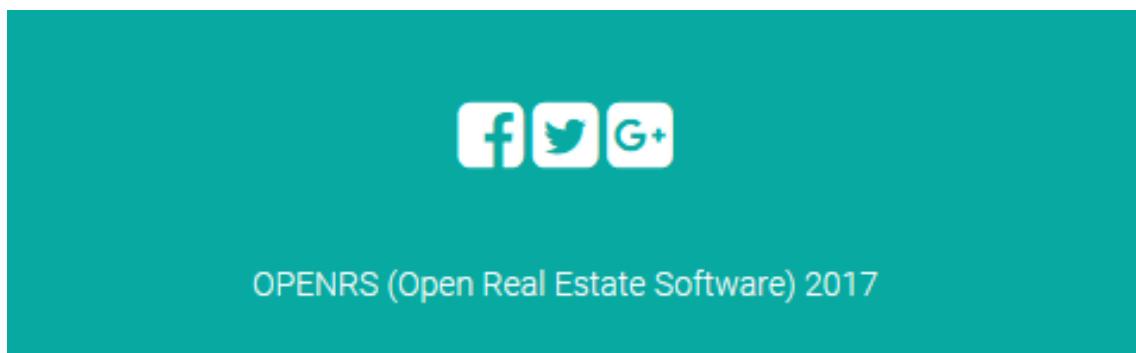


Figura 9.80: Web Corporativa: Pie de página

Capítulo 10

Conclusiones

En este último capítulo se detallan las lecciones aprendidas tras el desarrollo del presente proyecto y se identifican las posibles oportunidades de mejora sobre el software desarrollado.

10.1. Dimensiones del proyecto

Para que el lector se haga una idea de las dimensiones del proyecto aquí van algunos datos significativos:

- Dos frameworks de frontend y un framework de backend utilizados.
- 3 módulos añadidos, personalizados e integrados.
- 7 librerías externas añadidas.
- Más de 330 commits realizados.
- 40 ramas de desarrollo.
- Más de 80000 líneas de código programadas (sólo contando la carpeta de aplicación del framework con ficheros php y sin librerías, en total el sistema contiene más de 190000).
- 74 tablas.
- 12 vistas.
- Más de 120 índices.
- Más de 10000 registros en la base de datos (5 MB).

10.2. Objetivos alcanzados

En resumen, podríamos decir que los principales objetivos tanto personales (vea el último apartado para más información) como del sistema se han alcanzado, aunque la herramienta, a pesar de su complejidad, todavía es bastante básica en algunos aspectos y necesita de mejoras para ser más competitiva respecto a los productos comerciales instalados en el mercado.

Entre los objetivos marcados para el sistema han quedado sin realizar, se destacan los siguientes:

- Uno de los grandes objetivos que no pudo ser logrado por motivos técnicos fue la integración con portales inmobiliarios, ya que las APIs, al menos en los portales que he preguntado (Fotocasa, Idealista y Pisos.com) son privativos. Sólo Idealista dispone de una API pero para realizar consultas, la cual, no es útil para este objetivo. En los anexos a este proyecto se adjunta ya que tiene interés para futuras ampliaciones.
- Ficha de visita: Por una parte, por los riesgos acaecidos y por otra, por que esta funcionalidad está encuadrada dentro de las acciones que un agente puede realizar y para que pudiera estar completa debería contemplar otras acciones además de estar integrada con otros sistemas externos como Google Calendar.
- APIs: Realmente este objetivo no debería de haber aparecido en esta versión, ya que no hay ningún componente que las pueda emplear. No obstante, como se preveía dejar una versión que pudiera estar interconectada con otros sistemas, se puso inicialmente.
- Mejoras: Aquí entra en juego lo que se comenta en la sección de dificultades y tiene que ver con el alcance así como el futuro modelo de negocio a implantar y radica en; ¿Hasta dónde debe llegar de flexible, de personalizable, de automático o de funcionalidades el sistema para esta versión inicial?. En el apartado de mejoras se ofrecen más detalles.

No obstante también se han realizado mejoras o incluido funcionalidades que no estaban en los objetivos iniciales:

- Mejora en la subida de imágenes: Inicialmente sólo se iban a subir (todas juntas a la vez en un único envío sincrono) e iban a quedar publicadas automáticamente. Luego, se refinó y se añadieron las opciones actuales a las imágenes así como la posibilidad

de subirlas elegantemente y de forma asíncrona a través del componente Dropzone.js estando ahora limitadas su tamaño individualmente y no por el envío completo.

- Subida y parametrización de documentos: Los documentos iban a subirse sólo para los clientes e inmuebles y no iban a llevar un estado personalizable asociado. Tampoco se contempló la subida de demandas.
- Plantillas de documentos: Sólo se iban a incluir un par de plantilla para la ficha de visita y el cartel con unos campos básicos.
- Integración con Google Analytics: Ni la posibilidad de configurar la cuenta a través del sistema.
- Visualización de varios inmuebles en Google Maps: Sólo se planteó inicialmente una visión dentro del detalle del inmueble en la parte pública y privada.
- El sistema de demanda iba a ser manual exclusivamente y se añadió un cruce automático y se introdujo un filtro de búsqueda mucho más amplio.
- Las marcas usadas para generar documentación iban a ser fijas y sólo las más representativas.
- La zona pública no iba a incluir un módulo de noticias inicialmente. Además se ha mejorado la flexibilidad y el aspecto modificando el módulo base de TIPYCOS.
- Otras mejoras y ampliaciones: En los PCS pueden encontrar más información al respecto.

GESTICADIZ, por su parte, se encuentra satisfecha por el producto, incluso considera que tiene más funcionalidades que las que necesita realmente. No obstante, valorará en las próximas semanas su implantación definitiva, ya que, a su juicio, considera que, a pesar de todas las mejoras implementadas orientadas a facilitar la introducción de datos en el sistema, hay que introducir muchos datos y realizar muchas configuraciones para hacer un uso eficiente del sistema.

10.3. Dificultades

- **Ámbito, necesidades y herramientas actuales:** La cantidad de herramientas y saturación de estas dentro del sector inmobiliario hace que la valoración y la selección

de las principales necesidades sea compleja y consuma una gran cantidad de tiempo.

- **Flexibilidad vs Automatización:** Uno de los principales problemas que he tenido ha sido, sobre todo a la hora de construir, aunque también a la hora de analizar el sistema, es ¿Cuánto de flexible tiene que ser una funcionalidad respecto al grado de automatización que te proporciona? o ¿Cuánto cuesta realizar una determinada tarea de una manera más manual y flexible respecto a una que valga para muchos casos pero tenga un mayor grado de automatización?. La respuesta es muy pero que muy complicada de dar ya que hay que tener muchos aspectos en cuenta como el nivel de uso de la funcionalidad, el grado de flexibilidad que necesita, tus propios hábitos al construir o los del cliente al utilizar el producto. Como decía es una pregunta muy compleja, y más en este producto que tiene un enfoque dentro del mercado no de un cliente en concreto. Mi principal conclusión ha sido que en todo momento hay que usar el sentido común y hacer las cosas cómo uno se sienta más cómodo luego, tanto usándolas como construyéndolas, aunque esto implique que luego no sea tan flexible o tan automática o que no siga las mejores prácticas.
- **Valoración y aprendizaje de componentes:** Como se verá en el siguiente capítulo, he tenido que aprender y sobre todo, **VALORAR**, una cantidad de tecnologías, componentes y metodologías bestial (aunque suena mal la palabra). Mezclar todo esto en sus diversos ámbitos y niveles de abstracción es una tarea ardua y compleja.
- **Integración de componentes seleccionados:** Cuando se integran componentes externos siempre surgen problemas como por ejemplo que algunos ya vengan adaptados para CI 3, y otro no como el CMS de TIPYCOS por nombrar uno de ellos. A parte de esto, cuando integras un componente hay que testearlo unitariamente (que muchas veces implica programarlo como test de integración) y luego adaptarlo a tu funcionalidad específica, y luego, si procede, depende del caso, hay que documentar todo esto. La laboriosidad de todo esto es tremenda.
- **Implantación en entornos reales:** Esto es un viejo fantasma que siempre surge en todos los proyectos de software. Tu código funciona perfectamente en local pero arriba en staging o en producción no funciona (o al revés). Ha habido muchas pequeñas cosas que solucionar que se detallan en los PCS.
- **Alcance del sistema:** Es muy complicado determinar hasta dónde debe llegar el

sistema en cada funcionalidad para aportar una perspectiva lo suficientemente buena para que cualquier pequeña o mediana inmobiliaria ponga su negocio en tus manos. Esto es malo en el sentido que puede quedarse muy corto para algunas o demasiado grande para otras.

- **Riesgos imprevistos:** Que otro cliente te presione para hacer otro proyecto cuando no lo tenías previsto o que enfermes (entre otros problemas de salud como la ansiedad, la falta de sueño, etc.), entre otros motivos, hace que tengas que improvisar y reorganizar todo tu calendario con la pérdida de funcionalidades y calidad en general del proyecto que esto implica.
- **Deathline:** La fecha de entrega del proyecto era fija y sin posibilidad de retrasarse un sólo día, esto ha sido un factor de presión increíble y con el cual he tenido que lidiar durante todo el proyecto.
- **Test en múltiples entornos y dispositivos:** Sobre todo en las fases finales del proyecto, la cantidad diferentes de navegadores, dispositivos, resoluciones, etc. en los que se debería de probar la aplicación es incalculable. Además, el proyecto ha sido probado en al menos una distribución de los sistemas operativos actuales y en todos los posibles entornos.
- **Nivel de exigencia:** El nivel de exigencia tanto a nivel funcional como tecnológico ha sido muy elevado para un proyecto de estas características, debido a que se ha querido conseguir un producto que pueda rivalizar en un mercado comercial tan agresivo como el inmobiliario.
- **Desarrollo completamente a distancia:** Entrevista, presentación, reuniones, desarrollo de versiones, etc. Todo ha sido realizado a través de herramientas on-line o a distancia sin apenas ninguna interacción presencial.

10.4. Lecciones aprendidas

A continuación se realiza una comparativa de los dos proyectos que se han realizado en el ámbito de las inmobiliarias para otorgar una visión global de todo lo aprendido así como de todo lo utilizado para la elaboración del proyecto:

| CATEGORÍA | OPENINMOCMS | OPENRS |
|----------------------------|------------------------------|---------------------------------------|
| WYSIWYG | - | TIPYCOS (Ckeditor + KCfinder) |
| Noticias | SimplePie (RSS) | TIPYCOS (Bootstrap) |
| PDF | FPDF + otras | DOMPDF (futuro) |
| CSV | - | CSVReader |
| Copias seguridad | MySQLDump + MYSQLDUMPER | MYBackup |
| Autentificación y permisos | Manual | ION Auth |
| Estadísticas web | Phpmvisites | Google Analytics |
| Videos | HTML - almacenado en sistema | HMTL 5 - Youtube |
| QR Code | PHPQRCode | Ciqrkode |
| Google Maps | EasyGoogleMap | BIOSTALL |
| Base de datos | ADODB | CodeIgniter + MYModel |
| Sesiones | Static Sesion | CodeIgniter |
| Captcha | SecurImage | Recaptcha |
| Correos | PHPMailer | Codeigniter |
| Helpers | Manual | Codeigniter |
| Ficheros | Manual | Codeigniter + MYUpload + Dropzone.js |
| Validaciones | Manual | Codeigniter (FormValidation + Models) |
| Configuración | Manual | Codeigniter |
| Internacionalización | Translator | Codeigniter + Manual |
| Paginación | Paginator | Codeigniter + Datatables.js |
| Imágenes | XHTML | Colorbox.js |
| Formularios | XHTML | Ace Template (Varios) |
| Gráficos | JPGraph | Ace Template |
| Iconos | Imágenes propias | Ace Template (Font Awesome) |

Tabla 10.1: Comparativa de componentes

| CATEGORÍA | OPENINMOCMS | OPENRS |
|---------------------|-------------|---------------------------|
| PHP | 5.2 / 5.3 | 5.5/5.6 |
| MYSQL | 5.2 | 5.6 |
| Javascript | Nativo | Jquery |
| HTML | XHTML 1.0 | HTML 5 |
| Framework Front-end | - | Bootstrap 3 y Materialize |
| Framework Back-end | Manual | Codeigniter 3.1 |
| Repositorio | SVN | GIT + GitHub |
| AJAX | Javascript | Jquery |

Tabla 10.2: Comparativa de tecnologías

| CATEGORÍA | OPENINMOCMS | OPENRS |
|------------------------|-----------------|-----------------------------------|
| Repositorio | SVN | GIT |
| Modelo de proceso | Incremental | Espiral |
| Metodología desarrollo | Métrica v3 | Métrica v3, MAGERIT y MADEJA |
| Documentación | Word | Word + Latex |
| SGBD | PhpMyadmin | MYSQL Workbench |
| IDE | Notepad++ | Netbeans 8 |
| Servidor desarrollo | APPSEERV 2.5.10 | WAMP 2.5 |
| Servidor staging | - | Openrs.es + Plesk |
| Documentación código | PHPDOC | Doxxygen + PHPDOC |
| Pruebas | Manuales | Testing Codeigniter |
| Calidad y seguimiento | PCS | Diario retos + PCS + GIT + Trello |
| Comunicación | - | Skype + Trello |
| Modelo de negocio | - | TIMON + ampliaciones + plantillas |

Tabla 10.3: Comparativa de procesos y herramientas

10.5. Trabajo futuro

En relación a las tecnologías y componentes empleadas que deberán implantarse o valorarse:

- PHP 7.
- Bootstrap 4.
- Codeigniter 4.
- Crear un entorno tecnológico totalmente LAMP.
- Aprender SCRUM y aplicarlo eficientemente.
- Integración de AWS (Amazon Web Services).

En cuanto a la metodología de trabajo, sólo mencionar que, además de seguir mejorando el modelo propuesto para GIT, el uso de TRELLO u otras herramientas on-line para la coordinación del proyecto se hace indispensable.

Se podría crear un modelo de negocio basado en versión premium o expansiones separadas, pero en cualquiera de los casos, debería de contener como mínimo, los siguientes elementos:

- Subida de inmuebles a portales inmobiliarios (Fotocasa, Idealista y Pisos.com)
- Plantillas zona pública en Material Design (Materialize) y Boostrap.
- Crear API y módulos en diferentes CMS (principalmente en Wordpress).
- Módulo de Acciones (Agenda integrada con Google Calendar).
- Mejoras indicadas en anexos (Más flexibilidad, más información, más estadísticas, etc.)
- Módulo para Clientes (facturas, comunicación directa con inmobiliaria, envío de datos de inmueble, etc.).
- Ampliación de Documentación generada.

10.6. Valoración personal

Para finalizar me gustaría realizar algunas valoraciones de índole personal:

- He intentado obtener un producto único en su ámbito y creo que lo he conseguido y me siento tremadamente orgulloso.
- La herramienta tiene un aspecto, globalmente, al menos inicialmente y desde la humildad, fantástico, totalmente responsive, muy flexible e intuitiva y con un gran potencial. En resumen, hasta la fecha, el mejor proyecto que he hecho en mi vida valorándolo en su conjunto.
- Los conocimientos y las buenas prácticas adquiridas me van a permitir adaptarme mejor a la realidad del mercado como analista y programador.
- He hecho el proyecto con mucho cariño y con muchísima dedicación, y aunque no vaya a convertirse en mi modo de vida, espero sacar algún rédito económico en el futuro.
- Este proyecto ha supuesto un esfuerzo desmedido y totalmente desproporcional para lo que se supone que es un proyecto fin de carrera, ya que el primer software, lo que pretendía ser OPENINMOCMS y que lleva usando GESTICADIZ durante más de 4 años, ya es, por sí mismo, un PFC que perfectamente podría haber entregado y que tiene muchísimo trabajo, en algunas funcionalidades, más que este incluso. Lo que me gustaría transmitir es que este proyecto es la consecución de dos proyectos, no sólo de uno.
- Se han conseguido resultados similares a las herramientas comerciales e incluso, en algunas funcionalidades, superiores.
- Este proyecto cierra una etapa de mi vida que lleva estancada más de 10 años. Por mucho que el lector lea este documento y todos sus anexos, sea cual sea la valoración que haga del mismo, jamás podrá llegar a entender la presión, el sufrimiento, las frustraciones, la dedicación, el esfuerzo y las implicaciones personales, sobre todo psicológicas, que este proyecto tiene para mí.

Si has llegado hasta aquí, muchas gracias por tu tiempo.

Bibliografía y referencias en internet

- [1] Ministerio de Administraciones Públicas: *Métrica V.3 : Metodología de Planificación y desarrollo de sistemas de información* http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html. 2005.
- [2] Roger S. Pressman: *Ingeniería del software, un enfoque práctico*. McGrawHill, Edición 6, 2005.
- [3] Ian Somerville: *Ingeniería de software*. Pearson, Edición 9, 2011.
- [4] Craig Larman: *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Pearson Educación, Edición 2, 2003.
- [5] Página web oficial de Wikipedia (versión en inglés): www.wikipedia.en [Último acceso: 07/09/17].
- [6] Página web oficial de MADEJA: www.madeja.es [Último acceso: 13/07/17].
- [7] Página web oficial del Instituto Nacional de Estadística: www.ine.es [Último acceso: 10/05/17].
- [8] Página web oficial de Colegio Profesional de Ingenieros en Informática de Andalucía: <http://cpiiand.es> [Último acceso: 13/06/17].
- [9] Comparador de franquicias inmobiliarias <http://www.franquiciator.es/franquicias/inmobiliarias/> [Último acceso: 13/06/17].
- [10] Página web oficial de php: www.php.net [Último acceso: 27/08/17].
- [11] Página web oficial de mysql: www.mysql.com [Último acceso: 25/08/17].
- [12] Página web oficial de apache: www.apache.com [Último acceso: 06/06/17].
- [13] Página web oficial W3C: www.w3c.es [Último acceso: 13/06/17].

- [14] Página web con múltiples foros y manuales utilizados en la implementación: www.lawebdelprogramador.com [Último acceso: 10/07/17].
- [15] Página web para consulta de errores y dudas de implementación: www.desarrolloweb.com [Último acceso: 10/06/17].
- [16] Página web para consulta de errores y dudas de implementación: www.stackoverflow.com [Último acceso: 06/09/17].
- [17] Página web con múltiples foros y manuales utilizados para realizar la documentación en Latex: tex.sharelatex.com [Último acceso: 01/09/17].
- [18] Repositorio librería para Codeigniter Google Maps V3: <https://github.com/BIOSTALL/CodeIgniter-Google-Maps-V3-API-Library> [Último acceso: 20/08/17].
- [19] Página web oficial de Inmofactory: <http://www.inmofactory.com/> [Último acceso: 20/07/17].
- [20] Página web oficial de Inmoenter: <http://www.inmoenter.com/> [Último acceso: 02/09/17].
- [21] Ejemplo on-line de plantilla New York de CRM Inmoenter: <http://www.realestateagent.es> [Último acceso: 02/09/17].
- [22] Licencia GPL v3.0: <https://www.gnu.org/licenses/gpl-3.0.en.html> [Último acceso: 13/06/17].
- [23] Web actual de GESTICADIZ: <http://www.gesticadiz.es> [Último acceso: 07/09/17].
- [24] Repositorio de CMS de TIPYCOS: <https://bitbucket.org/Alleon86/cms-tipycos> [Último acceso: 07/09/17].
- [25] Repositorio de ION AUTH: <https://github.com/benedmunds/CodeIgniter-Ion-Auth> [Último acceso: 13/05/17].
- [26] Tutorial sobre backups en Codeigniter 3: <http://www.cumacoder.com/2016/01/backup-restore-database-codeigniter-3.html> [Último acceso: 01/09/17].
- [27] Repositorio de MYModel de Averiner: https://github.com/avenirer/CodeIgniter-MY_Model [Último acceso: 01/08/17].

- [28] Web oficial de CKEDITOR: <http://ckeditor.com/> [Último acceso: 05/07/17].
- [29] Web oficial de KCFINDER: <https://kcfinder.sunhater.com/> [Último acceso: 05/07/17].
- [30] Repositorio oficial de OPENRS: <https://github.com/klaimir/openrs> [Último acceso: 07/09/17].
- [31] Plantilla de frontend Ace Admin v.1.3.3: <https://themelot.net/ace-v1-3-3-responsive-admin-template.html> [Último acceso: 07/09/17].
- [32] Página web oficial de Codeigniter: www.codeigniter.com [Último acceso: 07/09/17].
- [33] Página web oficial de Bootstrap: www.bootstrap.com [Último acceso: 02/09/17].
- [34] Página web oficial de Materialize: <http://materializecss.com> [Último acceso: 03/09/17].
- [35] Artículo Flat Design vs Material Design: Similitudes y Diferencias: <https://www.silocreativo.com/flat-design-vs-material-design/> [Último acceso: 04/09/17].
- [36] Página web oficial de jQuery: <https://jquery.com/> [Último acceso: 05/07/17].
- [37] Tutoriales online sobre múltiples tecnologías web [https://www.w3schools.com/](https://www.w3schools.com) [Último acceso: 02/09/17].
- [38] Sistema para la evaluación de competencias en LMS mediante servicios web <http://rodin.uca.es/xmlui/handle/10498/16373> [Último acceso: 31/07/17].
- [39] Miguel López Morales *Divandroid*. UCA, 2014.
- [40] Página web oficial de contador de líneas de código <http://cloc.sourceforge.net> [Último acceso: 07/09/17].
- [41] Guía de buenas prácticas en PHP <https://phpbestpractices.org/> [Último acceso: 03/08/17].
- [42] CodeIgniter Best Practices with Joost van Veen <https://code.tutsplus.com/courses/codeigniter-best-practices> [Último acceso: 03/08/17].
- [43] Guía de buenas prácticas en MYSQL <https://code.tutsplus.com/tutorials/top-20-mysql-best-practices--net-7855> [Último acceso: 03/08/17].

- [44] Tutoriales online de múltiples tecnologías <https://code.tutsplus.com> [Último acceso: 01/09/17].
- [45] Sitio web con diapositivas de diferente temática <https://es.slideshare.net/> [Último acceso: 15/08/17].
- [46] Insertar un mapa o compartir una ubicación de Google <https://support.google.com/maps/answer/144361> [Último acceso: 31/08/17].
- [47] Soporte Google: Obtener una clave o autenticación de Google Maps <https://developers.google.com/maps/documentation/geocoding/get-api-key?hl=es-419> [Último acceso: 31/08/17].
- [48] Soporte Google: Configurar el seguimiento de Analytics <https://support.google.com/analytics/answer/1008080?hl=es> [Último acceso: 31/08/17].
- [49] Artículo: Cómo Integrar No CAPTCHA reCAPTCHA en tu Sitio Web <https://webdesign.tutsplus.com/es/tutorials/how-to-integrate-no-captcha-recaptcha-in-your-website--cms-23024> [Último acceso: 31/08/17].
- [50] Artículo: Cómo instalar WAMP SERVER 2.5 <http://www.softwaredegestionlibre.com/2015/01/como-instalar-wampserver-2-5-en-windows-7.html> [Último acceso: 31/08/17].

Información sobre Licencia

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document,
but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License.

Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or

distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the

same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one. The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly

and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation. Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice. Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.