



Universitat Autònoma
de Barcelona

GEOPOSICIONAMIENTO DE ELEMENTOS EN UN MAPA VIRTUAL

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Gestió

realitzat per

Ignacio Urionabarrenechea Español

i dirigit per

Xavier Verge Mestre

Escola Universitària d'Informàtica

Sabadell, Juny de 2009

Resumen

Las siguientes páginas recogen aquellos conceptos claves en las diferentes fases de desarrollo de una aplicación web de geoposicionamiento virtual de elementos mediante servicios de mapas gratuitos para la empresa de transportes MRW; síntesis además, que completa la realización del proyecto de final de carrera de Ingeniería Técnica en Informática de Gestión de la Escuela Universitaria de Informática de Sabadell.

MRW Maps, así bautizada, surge de la necesidad de generar un espacio interno común donde se pueda consultar información relacional de franquicias sin la obligación de tener grandes conocimientos de informática ni de las ciencias geográficas.

Se trata por tanto, de estudiar las posibilidades que ofrece desarrollar una herramienta de estas características sobre un mapa de Google Maps.

Tabla de contenidos

Capítulo 1. Introducción	1
1.1 Presentación	1
1.2 Situación Actual	3
1.3 Motivación y Objetivos	5
1.4 Organización de la memoria	7
Capítulo 2. Estudio de Viabilidad	8
2.1 Introducción	8
2.2 Objeto	9
2.2.1 Situación Actual	9
2.2.2 Perfil Cliente/Usuario	10
2.2.3 Objetivos.....	10
2.3 Descripción del Sistema	11
2.3.1 Descripción	11
2.3.2 Modelo de desarrollo	12
2.3.3 Requerimientos funcionales.....	14
2.3.4 Requerimientos no funcionales.....	14
2.4 Valor de Negocio.....	15
2.4.1 Análisis DAFO.....	15
2.5 Planificación	17
2.5.1 Planificación Inicial	17
2.5.2 Diagrama de Gantt	18
2.6 Análisis de Costes.....	19
2.6.1 Recursos Necesarios.....	19
2.6.2 Coste Total.....	23
2.7 Evaluación de Riesgos	24
2.8 Conclusiones	25

Capítulo 3. Análisis y Especificación	26
3.1 Introducción.....	26
3.2 Requisitos funcionales	27
3.3 Requisitos no funcionales	28
3.4 Modelo de casos de uso	30
3.4.1 Diagrama de casos de uso	30
3.4.2 Especificación de los casos de uso	31
Capítulo 4. Conocimientos Previos.....	40
4.1 Fundamentos teóricos	40
4.1.1 Google Maps.....	40
4.1.2 Ext JS	42
4.2 Fundamentos prácticos.....	44
4.2.1 API de Google Maps	44
4.2.2 Empezar con Ext JS	47
4.3 Tecnologías de desarrollo	50
Capítulo 5. Diseño	56
5.1 Introducción.....	56
5.2 Diseño de la interfaz gráfica	57
5.2.1 Introducción.....	57
5.2.2 Interfaz de Usuario	59
Capítulo 6. Implementación y Pruebas.....	65
6.1 Implementación.....	65
6.2 Pruebas	66
6.2.1 Introducción.....	66
6.2.2 Pruebas de Usabilidad	66
6.2.3 Pruebas de Rendimiento	67
6.2.4 Pruebas de Compatibilidad	68
Capítulo 7. Conclusiones.....	70
Capítulo 8. Futuras líneas de trabajo.....	72
Capítulo 9. Bibliografía	75

Capítulo 1. Introducción

1.1 Presentación

Como empresa líder en el transporte urgente de paquetería, con más de 800 franquicias y con más de 40 centros logísticos distribuidos entre Andorra, España, Gibraltar, Portugal y Venezuela, MRW y su Departamento de Tecnologías de la Información estudia y desarrolla nuevos servicios, prestaciones y aplicaciones que mejoren y satisfagan aquellas necesidades de sus clientes, internos y externos.

Su modelo de negocio se basa en una red comercial de franquicias, que a la postre es la clave de su éxito. A merced de lo que pueda parecer, el valor de negocio de estas empresas se basa en la información, información acerca de sus operaciones internas y de su ambiente externo.

Paralelamente, el desarrollo de aplicaciones web, considerada una de las áreas más activas y dinámicas existentes actualmente en el desarrollo de software, ha favorecido a la aparición de nuevos paradigmas de desarrollo que permiten crear aplicaciones web híbridas incrustando contenidos de una o más fuentes de información.

A diferencia de hace unos años, donde basaban su contenido en documentos estáticos, estas aplicaciones generan sus contenidos de forma dinámica combinando datos de diferentes fuentes con servicios de geoposicionamiento como Google Maps, videos de Youtube, fuentes RSS¹, fotos, compartición de archivos...

¹RSS (Sindicación de contenidos en la Web): Es un formato de documento basado en XML que se utiliza para actualizar las novedades y noticias de un sitio web.

Con estas ideas iniciales, el caso de estudio que se expone a continuación tiene como objetivo aunar en una misma aplicación mediante el servicio de geoposicionamiento que ofrece Google de manera gratuita, toda una serie de datos referentes a las franquicias y que el cliente pueda abastecerse virtualmente de información que necesite para realizar gestiones de forma rápida, intuitiva y fácil.

1.2 Situación Actual

Actualmente, la aparición de nuevas tecnologías colaborativas está revolucionando la forma de ver Internet. Estas tecnologías siguen la tendencia y el concepto marcados por la Web 2.0². A su vez, estas tecnologías no desplazan a las anteriores sino que además interactúan entre ellas y dan lugar a nuevas configuraciones que basan su contenido en lo que el usuario, como protagonista principal, desee mostrar y, que se conoce como *mashup*³.

Proveniente de la música, mashup es un término inglés utilizado cuando un DJ mezcla diferentes tipos de música con la finalidad de conseguir una pieza musical completamente nueva y que difiere de las originales.

No obstante, en la informática, su significado se asemeja bastante. También conocido como remezcla, es la combinación de datos de terceros existentes en diferentes proveedores de datos para crear un nuevo contenido completo y mejorado, consumiendo servicios directamente a través del protocolo HTTP⁴, tal como define el portal la Wikipedia. Para entender un poco más este nuevo concepto, la siguiente figura muestra la arquitectura de un mashup, siempre compuesta de tres partes:

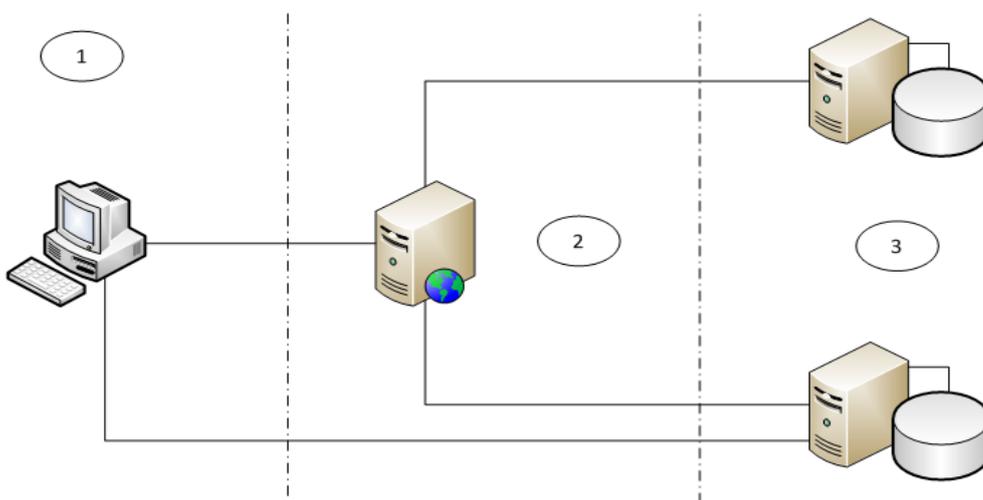


Fig. Arquitectura de un mashup

²Web 2.0: Término acuñado por O'Reilly en 2004 para referirse a la segunda generación de contenido web, basada en comunidades de usuarios, redes sociales, wikis o blogs.

³[http://es.wikipedia.org/wiki/Mashup_\(aplicación_web_híbrida\)](http://es.wikipedia.org/wiki/Mashup_(aplicación_web_híbrida))

⁴HTTP (Protocolo de transferencia de Hipertexto): Es el protocolo utilizado por un cliente y un servidor para comunicarse.

El usuario mediante su navegador (1) accede al sitio web del mashup, que es la aplicación web que contiene toda la información y de la que no es dueña (2). El contenido de éste se obtiene accediendo a los proveedores de datos y servicios vía una API⁵ y diferentes protocolos web (3).

La aplicación que se realizará para MRW será un sistema de localización con servicios de mapas que embeberá información de franquicias en una misma aplicación.

Actualmente en la red la mayor parte de los mashup están basados en la utilización de mapas, por lo que su estructura frecuentemente es la misma, es decir, integración de servicios ya creados de cartografía, con diferentes niveles de profundidad y geoposicionamiento (latitud y longitud), mediante los cuales el propio usuario interactúa.

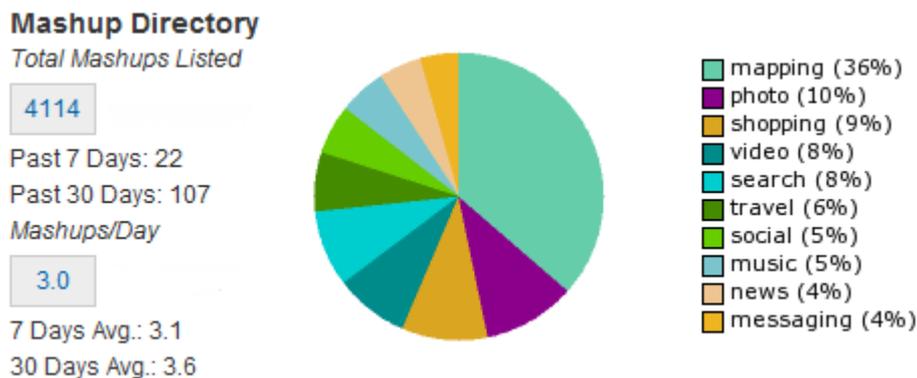


Fig. Directorio de mashups registrados en programmable.com

Según el sitio web www.programmableweb.com/mashups, portal dedicado al desarrollo de páginas Web 2.0, lleva registrados a día de hoy más de 4100 mashups, el 36% de los cuales utiliza servicios de mapa, como se observa en la figura anterior.

Por lo tanto, ya no basta simplemente con leer la información almacenada en una base de datos, sino que el éxito consiste en añadir nuevas funcionalidades a nuestras aplicaciones que muestren esta información utilizando un entorno creativo e ingenioso, mediante el cual el usuario se sienta partícipe.

⁵API (Interfaz de programación de aplicaciones): Es el conjunto de funciones y procedimientos que facilitan el intercambio de datos entre aplicaciones.

1.3 Motivación y Objetivos

El punto de partida del proceso de aprendizaje de una persona dicen que se multiplica cuando algo que le interesa realmente estimula en él actitudes de curiosidad. La experiencia demuestra que una persona responde proactivamente a estos estímulos o motivaciones, posibilitando no sólo el aprendizaje de procedimientos y destrezas, actitudes y/o valores, sino también asimilando todos sus conceptos.

Como parte final de una actividad universitaria la idea de desarrollar una herramienta web en el marco real de una empresa ofrece al proyectista la oportunidad de incorporarse en un proyecto de creación de software desde su inicio. El proceso de creación permite también conocer en primera persona cómo se trabaja en un entorno de desarrollo profesional que cuenta con un gran equipo de personas.

La idea inicial de utilizar Google Maps como plataforma de geoposicionamiento consistirá en estudiar las diferentes posibilidades, capacidades y limitaciones que puede ofrecer una herramienta como esta para una empresa de transportes como MRW.

Esta idea lleva a plantear una serie de objetivos:

Es necesario diferenciar dos tipos. Por un lado, el perfil profesional, que analiza la finalidad de lo que se pretende conseguir. Y, por otro lado, el personal, objetivos por los cuales el proyectista realiza este caso de estudio:

A nivel profesional:

- Satisfacer sus propias necesidades y completar sus inquietudes.

El valor añadido de esta aplicación dependerá de cómo se muestren los datos y de cuál sea su uso. Se debe entender no únicamente como un producto informático sino como un ejemplo de espacios o técnicas colaborativas entre todos sus integrantes. Además la información proporcionada por el portal tendrá mucho más valor cuando esté integrada o complementada junto con otras herramientas. En ese punto sus necesidades e inquietudes serán medibles.

- Seleccionar y filtrar información útil en el mapa, mediante búsquedas geográficas avanzadas.

- Visualización geográfica conjunta de diferentes elementos con niveles de profundidad diversos.

A nivel personal:

- Acercar al proyectista al mundo laboral.
- Profundizar en el aprendizaje de las estructuras de datos. Analizar las estructuras aplicadas a la recuperación asociativa de información.
- La manera en que se colocan los elementos dentro de la estructura afectará la forma en que se realicen los accesos a cada elemento
- Acercar al proyectista a un sector en clara expansión de forma que sirva de introducción para una posible definición del perfil laboral.
- Elaborar y presentar un trabajo como ejercicio integrador de los conocimientos técnicos adquiridos durante los años de estudio.
- Aprender, adquirir y conocer nuevas técnicas de trabajo y nuevos conocimientos de las diferentes herramientas de desarrollo de software, aplicables en el marco real de una empresa.
- Involucrarse en un modelo de negocio, formando parte del mismo, adquiriendo nuevas técnicas de trabajo y nuevos conocimientos.

En resumen, desarrollar e implementar un innovador y dinámico portal que permita registrar y clasificar por zonas geográficas todas las franquicias y así obtener datos de éstas mediante diferentes niveles de zoom, generan una gran dosis de inquietudes siguiendo un poco con la introducción en este.

A su vez, ofrece una buena oportunidad para poner en práctica aquellos conocimientos recibidos durante los años anteriores de estudio y de potenciar el aprendizaje de diferentes entornos de desarrollo y tecnologías dentro del mundo del desarrollo basado en Web.

No obstante la motivación y el objetivo principal es obtener la titulación de Ingeniería Técnica en Informática de Gestión.

1.4 Organización de la memoria

La memoria constará de 9 capítulos:

Se dividirá en diferentes secciones para ir profundizando y conociendo paulatinamente todas las fases que se llevan a cabo en el desarrollo de una aplicación. Además estos capítulos irán acompañados de esquemas, imágenes y tablas para un mejor entendimiento.

Para comenzar, será importante conocer la idea inicial propuesta por la necesidad de una empresa de transportes y concluirá con una breve descripción de las partes que conformarán este documento, no sin antes definir brevemente los objetivos y motivaciones a cumplir a corto plazo.

A continuación, después de esta breve introducción, en el siguiente capítulo se buscará analizar la viabilidad económica, organizativa y técnica del proyecto, basándose en cálculos aproximativos. Además se incluirá un análisis de mercado previo al desarrollo de la herramienta.

Los siguientes capítulos 3, 4, 5 y 6 recogen las fases del ciclo de vida clásico de la ingeniería del software.

En el capítulo 3 de análisis y especificación, se definirán los requerimientos de la aplicación y se definirán casos de uso.

Una vez realizado el análisis y especificación de los requisitos del sistema se centrará, ya en el capítulo 4 y 5, en conocer más a fondo que tecnologías se utilizarán para realizar el desarrollo y diseño de la interfaz y se mostrará un prototipo de aplicación en base a estas tecnologías descritas.

En los capítulos restantes 6, 7 y 8 se realizarán las pruebas de validación necesarias previa configuración e instalación, se especificarán casos de prueba que demuestren la integridad y compatibilidad, se llevarán a cabo conclusiones y se expondrán futuras líneas de trabajo.

Por último, en el capítulo 9 se hará referencia a las publicaciones impresas y a los recursos electrónicos utilizados durante este caso de estudio.

Capítulo 2. Estudio de Viabilidad

2.1 Introducción

Como en todo negocio, los análisis económicos necesarios que justifiquen la viabilidad de la idea inicial deberán evaluarse a través de resultados medibles.

Con este estudio se busca analizar la viabilidad económica, organizativa, operativa y técnica del proyecto. Por ello, basándose en cifras estimativas, se analizará a continuación, el tiempo, el trabajo y los recursos necesarios (en términos de recursos materiales y humanos) que supondrá este proyecto informático; se evaluará además la viabilidad económica y se analizarán los costes para efectuar el desarrollo de la aplicación.

La solución, aparte de determinar si el proyecto que se lleva a cabo tiene garantías de finalizar con éxito, servirá de guía para definir las especificaciones del sistema.

Estas estimaciones se podrán comparar con los datos reales obtenidos tras la finalización del proyecto.

2.2 Objeto

2.2.1 Situación Actual

En los últimos años la idea de mashup en Internet se asocia a la palabra éxito y su crecimiento exponencial es imparable. Como anteriormente se ha podido observar, en la red ya existen más de 4.000, de los cuales casi 1400 utilizan mapas como base de su contenido.

Si a toda esta asociación de nuevas tecnologías, le sumamos la gran cantidad de información que MRW almacena en su base de datos, englobar en un mismo portal información de más de 800 franquicias, de todas las plataformas y puntos de interés relacionados con la compañía, nace la idea principal de este proyecto, ya bautizado como 'MRW Maps'.

La idea es que el portal represente geoinformación (con un margen de error mínimo) del total de franquicias por comunidad autónoma, provincia o municipio, con diferentes niveles de zoom o simplemente permita insertar marcas que indiquen que existen competidores o clientes potenciales cercanos, entre otras muchas funciones.

Utilizar por tanto la API de Google Maps es una idea innovadora dentro de la compañía, ya que actualmente uno de los métodos para obtener información sobre las franquicias es un programa COBOL⁶ sobre DOS, no obstante, no deja de ser un sistema efectivo aunque transmite la sensación de ser algo antiguo y poco dinámico.

Por tanto poder mostrar, limpiar, filtrar, personalizar y optimizar toda esta información virtualmente en un mapa será garantía de resultados. A partir de aquí las posibilidades son infinitas.

⁶COBOL (Lenguaje Común Orientado a Negocios): Fue desarrollado para reemplazar a los lenguajes ensamblador, es el lenguaje más utilizado a nivel de gestión.

2.2.2 Perfil Cliente/Usuario

Esta herramienta inicialmente no tendrá un destinatario concreto. Será uno de los nuevos servicios que estarán disponibles en la red interna de la empresa, por lo que ha sido especialmente pensada, diseñada e implementada para que cualquier usuario con acceso a la red y con escasos conocimientos informáticos pueda acceder y utilizarla.

Al tratarse de un acceso enfocado al propio empleado, la aplicación se creará utilizando una interfaz gráfica amigable e intuitiva que concuerde con los servicios que se están desarrollando paralelamente en la compañía. Por lo tanto el perfil de cliente y el perfil de usuario pueden considerarse el mismo.

Será importante tener en cuenta el papel de las franquicias para futuras ampliaciones. Protagonistas principales de la aplicación y como socios, clientes y proveedores que son, será imprescindible que dispongan de perfiles de usuario con privilegios para administrar el contenido del portal.

2.2.3 Objetivos

Los objetivos específicos planteados en el proyecto con un horizonte de 4 meses se vinculan al desarrollo de una aplicación enfocada en el geoposicionamiento.

Hoy en día, en muy pocos pasos se pueden realizar mashups de mapas fácilmente y sin tener conocimientos de programación. Por ello un mashup debe presentar una interfaz gráfica amigable y práctica, de fácil navegación, maleable y que muestre al usuario información correctamente estructurada.

Partiendo de las premisas expuestas en el punto 1.3 de este mismo documento, los objetivos serán construir un portal que aúne todas estas tecnologías descritas a través del cual diferentes usuarios puedan acceder simultáneamente a la información.

Una vez el usuario esté dentro de la aplicación podrá ver un mapa de España que, mediante una serie de iconos mostrarán el total de franquicias existentes en la red de MRW a cuatro niveles de profundidad: Comunidad autónoma, provincia, municipio y franquicia.

2.3 Descripción del Sistema

2.3.1 Descripción

El objetivo principal de esta aplicación está claramente definido. Sabemos que su función es situar en un mapa virtual diferentes elementos. Aprovechando que Google dispone de un servidor de aplicaciones de mapas en Web, bien conocido como Google Maps y que la API es de uso libre, permitiendo incluso la modificación de cualquier aspecto de la interfaz original, el framework⁷ elegido para la creación de nuestro sitio Web es Ext JS.

Como bien es sabido, gran parte de las aplicaciones de Google utilizan un gran número de archivos de JavaScript, razón por la cual la elección de la librería Ext JS fue unánime. Mediante esta librería se diseñará un sitio web con una estructura homogénea, de fácil acceso a sus módulos de información con un menú lateral izquierdo de tipo acordeón, un menú lateral derecho con pestañas y una región central donde se carga el mapa.

En líneas generales, el usuario podrá visualizar en el mapa todas las franquicias que forman parte de la marca y que actualmente competen, cargar plataformas, insertar empresas competidoras y guardar esas marcas en formato XML⁸.

La recuperación y transformación de estos datos se realiza mediante JSON⁹, proporcionando una manera sencilla de convertir objetos a JavaScript aliviando la carga de la comunicación que supone realizarlo con XML.

De manera más detallada en el capítulo 5 de este documento se exponen más extensamente las características principales de librerías y lenguajes que han aparecido anteriormente como son el API de Google Maps, Ext JS, JavaScript y JSON y, de las funcionalidades propias que podría adoptar esta aplicación.

⁷Framework: Es un esquema, esqueleto o patrón para el desarrollo e implementación de una aplicación.

⁸XML (Lenguaje de Marcas Ampliable): Es una especificación/lenguaje de programación diseñado especialmente para los documentos web.

⁹JSON (Notación de objetos JavaScript): Es un formato ligero de intercambio de datos.

2.3.2 Modelo de desarrollo

En todo proyecto de desarrollo de software definir una metodología es muy importante ya que permite establecer que estrategia se debe seguir para alcanzar los objetivos marcados inicialmente.

Siguiendo el modelo tradicional de desarrollo de software, también conocido como ciclo de vida clásico, su desarrollo se realiza secuencialmente en diferentes fases, de forma que la etapa posterior no se inicia hasta que la inmediatamente anterior no finaliza.

Durante este proceso y para comprobar la capacidad de adaptación de la herramienta y su eficacia, debido a que su desarrollo e implementación está pensado para utilizar diferentes módulos, librerías y tecnologías que deben ser correctamente combinadas, se crearán diseños rápidos o prototipos para medir su rendimiento. Las fases que se seguirán son las siguientes:

1. Análisis y requerimientos

Este es el primer paso en el sistema clásico. Consiste en comprender que va a hacer el software dentro de las necesidades que tiene la empresa y comprobar si su desarrollo es viable o no.

2. Especificación

Consiste en plasmar mediante UML¹⁰ las necesidades y requerimientos del usuario, para poder realizar el diseño posterior del sistema.

3. Diseño

Se definirá con total detalle cómo se desarrollará el sistema para que soporte todos los requisitos, incluyendo requisitos no funcionales y otras restricciones.

¹⁰UML (Lenguaje Unificado de Modelado): Es un lenguaje de modelado de sistemas de software para definir, especificar, construir y documentar un sistema.

4. Codificación

A partir de la fase de diseño, se lleva a cabo toda la construcción del sistema. Un programador es el responsable de desarrollar por completo las funcionalidades definidas durante las etapas anteriores.

5. Prueba

No es más que comprobar que el sistema obtenido es el resultado final de la idea propuesta inicialmente. Se emplea de manera experimental para comprobar que el sistema no tiene ningún error y además satisface y cumple con las necesidades y expectativas de los usuarios. Si se detecta algún error se debe volver a la fase responsable del error y corregirlo tanto en esa fase como en las posteriores si se ven afectadas por dicho cambio.

* Prototipo

Como se ha especificado anteriormente, la construcción de prototipos representa una estrategia de desarrollo. Es por ello que incluye el desarrollo interactivo o en continua evolución, donde el usuario participa de forma directa en el proceso.

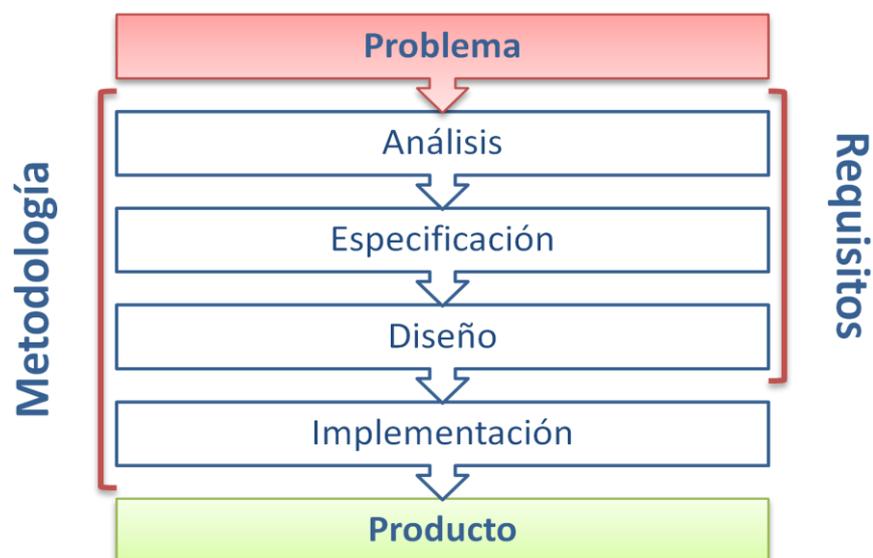


Fig. Modelo de desarrollo secuencial

2.3.3 Requerimientos funcionales

Puede ser una forma de describir lo que el sistema debe hacer. Por objeto, este proyecto, permite acceder en un sólo clic a toda una serie de información relacional de franquicias y empresas asociadas mediante diferentes menús de navegación y métodos de búsqueda avanzada.

2.3.4 Requerimientos no funcionales

Se especifican las restricciones impuestas por el cliente o por el propio problema y que pueden afectar al diseño de la aplicación:

Principalmente destacan:

Usabilidad

Es un factor muy importante a tener en cuenta ya que la aplicación, como se ha comentado anteriormente, será utilizada internamente en la red de la empresa por sus propios empleados, por lo que el sistema debe ser fácil de usar y de comprender para cualquier tipo de persona.

Rendimiento

El sistema podrá ofrecer simultáneamente la misma información a más de un usuario desde diferentes terminales de acceso. De esta manera cualquier dispositivo actual o futuro que pueda manejar un visor de Internet y se pueda conectar a la red, podría estar conectado al sistema de la compañía.

Adaptabilidad

Debido al método de implementación seleccionado, el proyecto deberá ser fácilmente modificable y con posibilidades de ser ampliado.

2.4 Valor de Negocio

2.4.1 Análisis DAFO

Es una herramienta que permite comprender y evaluar el valor que puede aportar, estratégicamente, al negocio tanto desde un punto de vista interno como externo, cuales son aquellos puntos fuertes y débiles que facilitan la toma de decisiones estratégicas en una organización.

Considerando la finalidad de esta herramienta es posible aplicarla en los diferentes procesos de implantación de desarrollo de software.

Aún así, sabiendo que la aplicación que se desarrollará se implantará internamente es importante saber cuál sería el comportamiento del mercado en el caso de exponerla al exterior, siempre enfocado a la captación de clientes potenciales.

Por tanto, el siguiente paso consiste en establecer los objetivos de la empresa y marcar brevemente cuáles son las fortalezas, debilidades, oportunidades y amenazas de esta herramienta siguiendo las indicaciones de la siguiente tabla:

Factor	Notación	Definición
Debilidades	D	Aspectos que limitan la efectividad y que constituyen una amenaza para el proyecto o desarrollo estratégico de una organización.
Amenazas	A	Riesgos potenciales externos al proyecto. De otro modo, es todo aquello que puede impedir su implantación o reducir su efectividad.
Fortalezas	F	Recursos potenciales que pueden servir para explotar oportunidades y que proyectos de otros de igual clase.
Oportunidades	O	Ventaja que producirá en un futuro tras su implantación

Análisis Interno – Factores controlables

Análisis Externo – Factores no controlables

Debilidades:

- Escasa reacción de los usuarios.

Amenazas:

- La evolución del mundo digital es continua y evoluciona muy rápidamente. Estos avances provocan cambios en la forma de utilizar este tipo de herramientas, por lo que puede provocar un uso indebido por parte de los usuarios y a su vez crear rechazo.

Fortalezas:

- Interacción con los clientes: Los clientes pueden proporcionar información sobre ellos mismos y son un canal de comunicación instantáneo. A mayor número de clientes mayor cantidad de información.
- Sistema de identificación virtual de
- Rasgos y características de la propia herramienta que facilitan su flexibilidad y capacidad de aceptación e innovación, además de la posibilidad de ampliar muchas características que ésta posee.

Oportunidades:

- Auge tecnológico.
- Penetración y uso de Internet: Los niveles actuales de penetración y uso de Internet en la empresa permiten la expansión de este tipo de servicios. Además estas herramientas 2.0 abren un abanico enorme de posibilidades.
- Su escalabilidad permite integrar fácilmente nuevos productos y/o servicios.
- Esta herramienta se nutrirá de la información proporcionada por los diferentes departamentos y franquicias, por lo que puede haber departamentos que inicialmente no estén involucrados pero que a la larga pueden verse relacionados indirectamente con alguna de las funcionalidades que ésta proporciona, por lo que podría atender a un mayor número de usuarios.

Desde una perspectiva organizacional, el impacto que producen los sistemas y tecnologías de la información sobre sus clientes se puede analizar desde un punto de vista interno y externo. Asimismo, será importante intentar minimizar las amenazas, tomar conciencia de éstas y explotar aquellas fortalezas y oportunidades para que la propia información se convierta para MRW en un recurso estratégico y en un valor de negocio añadido.

2.5 Planificación

2.5.1 Planificación Inicial

Desde el momento en que se plantea una necesidad, se fijan metas y se confecciona un camino para lograrlas. Es importante tener una planificación previa de las tareas que se deben llevar a cabo para cumplir todos los objetivos.

Por tanto, cada fase dependerá de la anterior, es decir, hasta que no finalice una fase no se podrá pasar a la siguiente, exceptuando la fase de pruebas, que se realizará simultáneamente junto a la fase de implementación ya que se irán realizando pruebas a medida que se vaya implementando las diferentes funcionalidades definidas en la fase de especificación.

A continuación se muestra la planificación de todas las tareas, indicando la duración en días y las fechas de inicio y final de la tarea. Seguidamente en el punto 2.5.2, se mostrará el diagrama de Gantt correspondiente a la descripción textual.

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> Geoposicionamiento de elementos en un mapa virtual	109,5 días?	lun 03/11/08	mié 08/04/09
<input type="checkbox"/> Estudio Previo	6,75 días?	lun 03/11/08	mar 11/11/08
Definición	1,25 días?	lun 03/11/08	mar 04/11/08
Documentación	4,5 días?	mar 04/11/08	lun 10/11/08
Análisis de plataforma de representación	1 día?	lun 10/11/08	mar 11/11/08
<input type="checkbox"/> Estudio de Viabilidad	8 días?	mar 11/11/08	vie 21/11/08
Objetivos	1 día?	mar 11/11/08	mié 12/11/08
Descripción de las necesidades del sistema	1 día?	mié 12/11/08	jue 13/11/08
Planificación	1 día?	jue 13/11/08	vie 14/11/08
Presupuesto	1 día?	vie 14/11/08	lun 17/11/08
Redacción del documento de EV	4 días?	lun 17/11/08	vie 21/11/08
<input type="checkbox"/> Realización del Proyecto	71,75 días?	vie 21/11/08	vie 06/03/09
<input type="checkbox"/> Análisis y Especificación	6,25 días?	vie 21/11/08	lun 01/12/08
Requisitos del sistema	2,25 días?	vie 21/11/08	mar 25/11/08
Modelo de Casos de Uso	4 días?	mié 26/11/08	lun 01/12/08
<input type="checkbox"/> Diseño	10 días?	mar 02/12/08	lun 15/12/08
Arquitectura básica del sistema	3 días?	mar 02/12/08	jue 04/12/08
Diseño de la Interfaz Web	7 días?	vie 05/12/08	lun 15/12/08
Implementación	50 días?	mar 16/12/08	jue 26/02/09
<input type="checkbox"/> Plan de Pruebas	5,5 días?	vie 27/02/09	vie 06/03/09
Pruebas de funcionamiento	3 días?	vie 27/02/09	mar 03/03/09
Corrección de errores	2 días?	mié 04/03/09	jue 05/03/09
Configuración	0,5 días?	vie 06/03/09	vie 06/03/09
<input type="checkbox"/> Documentación	23 días?	vie 06/03/09	mié 08/04/09
Redacción de la memoria	20 días?	vie 06/03/09	vie 03/04/09
Revisión	3 días?	vie 03/04/09	mié 08/04/09

Se estima que se realizará el desarrollo del proyecto por un analista-programador-documentador que dedica 5 horas diarias de lunes a viernes. Debido a los diferentes compromisos académicos y laborales por parte del proyectista es posible que se requieran horas de trabajo, por eso que cualquier modificación de la planificación inicial quedará reflejada en futuras correcciones.

2.5.2 Diagrama de Gantt

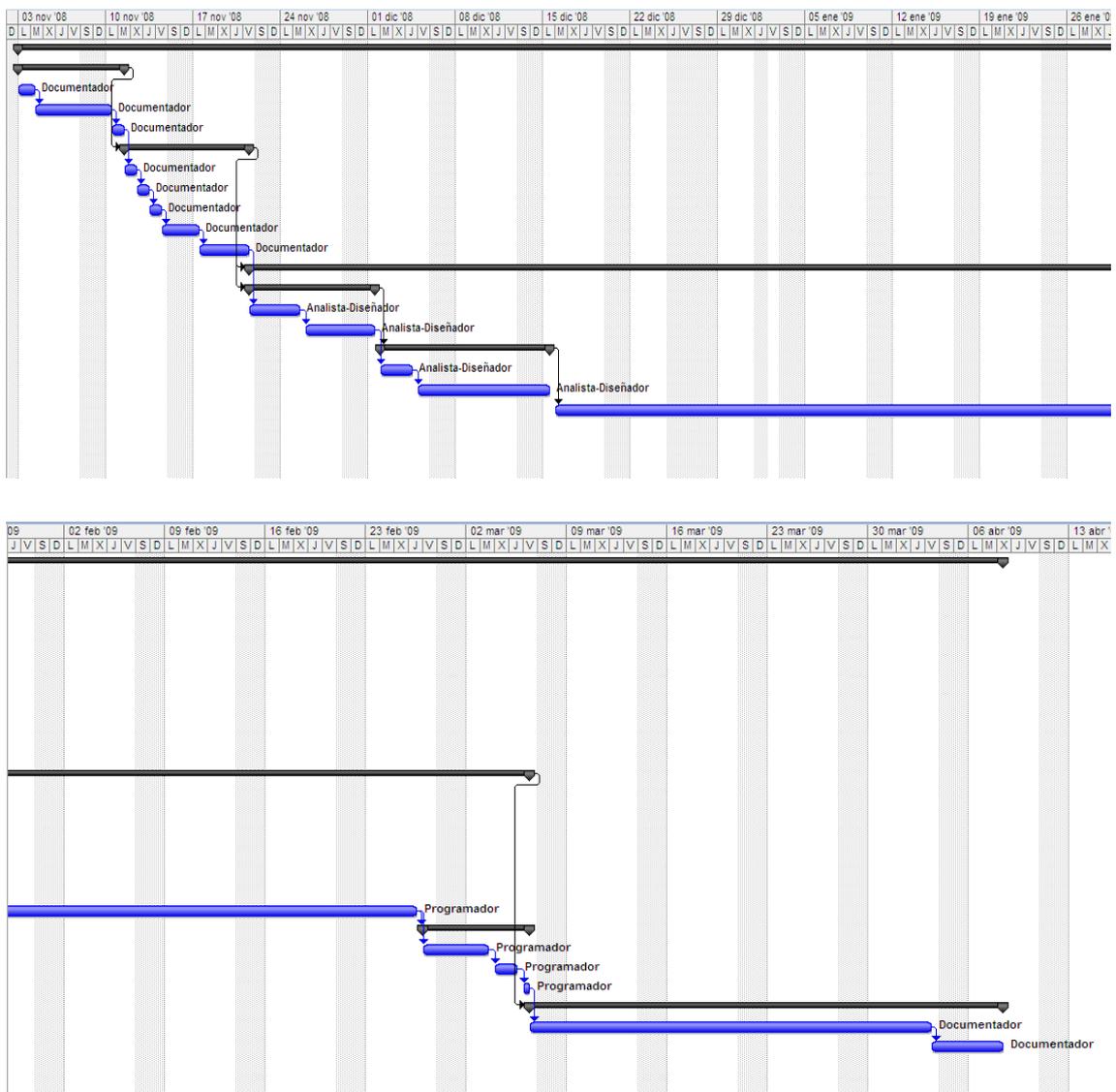


Fig. Diagrama de Gantt de la planificación inicial prevista.

2.6 Análisis de Costes

2.6.1 Recursos Necesarios

Para la elaboración y posterior valoración económica es importante indicar que recursos o herramientas serán necesarios en el desarrollo del proyecto, desglosados en recursos humanos (costes del personal encargado del desarrollo) y, recursos materiales (costes de las herramientas utilizadas para su creación).

2.6.1.1 Recursos Humanos

Pese a ser un proyecto que desarrollará íntegramente una persona, en todo desarrollo de software es importante definir el perfil profesional de los participantes en la creación y la documentación de la misma.

- **Director del proyecto:** Responsable encargado de asesorar, planificar, evaluar y corregir el desarrollo de la idea inicial.
- **Analista/Diseñador:** Su perfil es puramente técnico. Analiza la situación de la empresa con el propósito de especificar y diseñar los requerimientos de los usuarios. Diferenciamos:
 - **Analista de requerimientos:** Participa en la toma de requerimientos de negocio orientada a la implantación de sistemas informáticos.
 - **Analista de sistemas:** Controla y examina el sistema de información de una empresa con el propósito de proponer soluciones óptimas a nuevos desarrollos mediante sistemas computacionales.
- **Programador:** Responsable de implementar las especificaciones del analista en código ejecutable.
- **Equipo de pruebas:** Equipo encargado de realizar pruebas de validación y calidad del software, más allá de comprobar que funciona. Explorar situaciones donde se pueden encontrar fallos y comunicar al equipo de desarrollo los defectos encontrados en la aplicación. Puede estar formado por un solo integrante.

- **Documentador:** Redacta los informes previos, la documentación final y los manuales de usuario (si son necesarios).
- **Formador:** Consiste en transmitir conocimientos teóricos y prácticos que puedan ser útiles en el aprendizaje y manejo de la aplicación.

Sin embargo, para la planificación del desarrollo del proyecto actual será realizado por tres perfiles de profesionales:

Analista/Diseñador – Programador – Documentador

Teniendo en cuenta la planificación inicial y los roles asignados, la siguiente tabla detalla el coste relativo a los recursos humanos y su coste por hora:

Recursos Humanos	Coste por hora
AD - Analista / Diseñador	70,00 €
P - Programador	40,00 €
D - Documentador	30,00 €

Descripción actividad	Recurso
Estudio Previo	D
Estudio de Viabilidad	D
Análisis y Especificación	AD
Diseño	AD
Implementación	P
Plan de Pruebas	P
Documentación	D

Nombre de tarea	Duración	Costo
<input type="checkbox"/> Geoposicionamiento de elementos en un mapa virtual	109,5 días?	22.450,00 €
<input type="checkbox"/> Estudio Previo	6,75 días?	1.012,50 €
<input type="checkbox"/> Estudio de Viabilidad	8 días?	1.200,00 €
<input type="checkbox"/> Realización del Proyecto	71,75 días?	16.787,50 €
<input type="checkbox"/> Análisis y Especificación	6,25 días?	2.187,50 €
<input type="checkbox"/> Diseño	10 días?	3.500,00 €
Implementación	50 días?	10.000,00 €
<input type="checkbox"/> Plan de Pruebas	5,5 días?	1.100,00 €
<input type="checkbox"/> Documentación	23 días?	3.450,00 €

2.6.1.2 Recursos Hardware

PC usuario-proyectista:

- Procesador Intel Core 2 Duo CPU E6750 @ 2.66 GHz
- Memoria RAM: 2 GB
- Disco duro: 400 GB
- Tarjeta de red con conexión a Internet
- Monitor TFT de 22"
- Ratón y teclado

Servidor Virtualizado:

- Procesador Intel Core 2 Duo CPU E6750 @ 2.66 GHz
- Memoria RAM: 2 GB
- Disco duro: 32 GB

Recursos Hardware	Amortización
Equipo de sobremesa HP Pavilion	599,00 €
Monitor de pantalla plana panorámica HP w2228v de 22"	179,00 €
TOTAL	778,00 €

2.6.1.3 Recursos Software

Servidor Virtualizado:

- Virtualización: VMware Server 2.0 Free Edition
- Sistema Operativo: Ubuntu Server 8.0.4

PC usuario-proyectista:

- Sistema Operativo: Windows Vista Business.
- Navegador: Internet Explorer 7, Mozilla Firefox, entre otros.
- Adobe Flash Player

Generación de la documentación:

- Microsoft Office Standard 2007
- Microsoft Visio Standard 2007
- Microsoft Project 2007

Entornos de Programación:

- Aptana Studio Community Edition: IDE de desarrollo para aplicaciones web 2.0, con especial soporte para AJAX/JavaScript. Versión gratuita basada en programas similares como Eclipse.
- Ext JS: Librería de Java Script para simular aplicaciones de escritorio

PC usuario-cliente:

También es importante indicar las características o requisitos mínimos en cuanto a software que desde el PC de un usuario-cliente serán imprescindibles para el correcto funcionamiento de la aplicación:

- Sistema Operativo: Windows XP o superior.
- Navegador: Internet Explorer 7 o superior, Mozilla Firefox, entre otros.
- Adobe Flash Player

Recursos Software	Precio
Microsoft Windows Vista Business*	0,00 €
Microsoft Office Standard 2007**	569,00 €
Microsoft Visio Standard 2007**	329,95 €
Microsoft Project 2007**	779,00 €
Librería Ext JS	0,00 €
IDE Aptana Studio Community Edition	0,00 €

*Licencia del sistema operativo Microsoft Windows Vista Business no contemplada en el precio, ya que viene incluida en la compra del PC del usuario-proyectista.

**El precio incluido en la tabla anterior es el coste total por la compra del software. El precio real deberá ser la amortización de dichos programas durante el desarrollo del sistema tal como se indica en la siguiente tabla:

Suponiendo una vida útil de 60 meses = 5 años y 4 meses de desarrollo de la aplicación.

Recursos Software	Amortización
Microsoft Office Standard 2007	$(569,00/60)*4 = 37,93 \text{ €}$
Microsoft Visio Standard 2007	$(329,95/60)*4 = 21,99 \text{ €}$
Microsoft Project 2007	$(779,00/60)*4 = 51,93 \text{ €}$
TOTAL	111,85 €

2.6.2 Coste Total

Recursos	Coste
Recursos Humanos	22.450,00 €
Recursos Hardware	778,00 €
Recursos Software	111,85 €
TOTAL	23.339,85 €

2.7 Evaluación de Riesgos

Como en todo proyecto, la evaluación de riesgos potenciales es una actividad preventiva para identificar aquellas acciones que hay que modificar para evitar posibles fallos. Consiste en identificar aquellos factores/riesgos internos y externos que indiquen directamente en el correcto funcionamiento de la aplicación.

Al tratarse de un software para Internet su funcionamiento está expuesto a la conexión con ésta y con el servidor de datos, por lo que se trata de un riesgo únicamente predecible si se produce un fallo en la comunicación.

Otro de los riesgos que hay que tener presente y cómo se observa día a día, es la incompatibilidad entre los navegadores. Éstos están en constante desarrollo y a ciencia cierta es imposible saber cuál de ellos va a usar cada visita, por lo que es imprescindible que la aplicación sea totalmente funcional independientemente del navegador que utilice el cliente final. Para ello, durante el desarrollo se irán realizando constantes pruebas entre los navegadores más populares.

Revisar periódicamente el estado del proyecto en aspectos como organización y administración, cumplimiento del plan de trabajo y de los niveles de servicio establecidos son entre otros muchos puntos clave para que no se ocasionen pérdidas de tiempo, dinero y esfuerzo.

Otro de los factores de riesgo que hay que evaluar en un estudio es la formación del que lo realiza. El riesgo recae en el programador ya que es la persona encargada de plasmar las especificaciones en código ejecutable y el tiempo en el aprendizaje de las herramientas se antoja como vital.

Cabe reseñar que en futuras ampliaciones de la aplicación existirán usuarios con roles específicos y con diferentes permisos para realizar modificaciones sobre el mapa, por lo que el acceso deberá estar protegido por contraseña cifrada. Se tendrá que evitar que un usuario pueda acceder a contenidos donde no tiene permisos.

2.8 Conclusiones

Con la creación de esta aplicación el número de servicios que ofrece el portal de trabajo colaborativo y gestión documental de la compañía, aumentará de forma notable en cuanto a cantidad pero sobretodo en calidad.

Esta nueva funcionalidad facilitará a sus empleados y respectivos departamentos la opción de visualizar cualitativamente información de su interés de manera virtual, además de ofrecer a los clientes (franquicias) la opción de poder analizar con mayor precisión a sus clientes potenciales; por lo que la implementación y aceptación del proyecto se plantean para MRW como una mejora competitiva.

Así, después de plantear el estudio de viabilidad de forma equilibrada mediante un análisis del tiempo, trabajo y recursos necesarios, podemos responder que el proyecto es viable.

Capítulo 3. Análisis y Especificación

3.1 Introducción

El tratamiento de requisitos es el proceso mediante el cual se especifican y validan los servicios que debe proporcionar el sistema así como las restricciones sobre las que se deberá operar. *(Ferreira & Loucopoulos, 2001)*

Es en esta fase donde se analizarán los objetivos y las necesidades del cliente.

Se conoce también como etapa de determinación de requisitos.

El proceso comienza con la captura de requisitos determinando un conjunto de características propias del sistema software que se va a desarrollar y que satisfaga estos requerimientos.

La importancia de esta fase es esencial puesto que los errores más comunes y más costosos de reparar, así como los que más tiempo consumen se deben a una inadecuada ingeniería de requisitos.

Se distinguen dos tipos de requisitos, los requisitos funcionales y los requisitos no funcionales. A continuación se detallarán de forma más amplia.

3.2 Requisitos funcionales

Los requisitos funcionales describen las características o funcionalidades propias del software, es decir, las entradas y salidas de datos y procesos que debe haber. Son entendidos como las capacidades que debe disponer una aplicación con el fin de resolver un problema o cubrir unas necesidades. Se distinguen:

- La aplicación se basa en la información de cada una de las franquicias existentes en la red. Por tanto el objetivo principal de la aplicación consiste en mostrar en un mapa esta información, ofreciendo al usuario estos datos mediante diferentes tipos vistas.
- Al iniciar la aplicación el usuario podrá seleccionar datos directamente incrustados en un mapa o podrá ir filtrándolos dependiendo de su elección mediante una estructura en forma de árbol.
- La gestión de la competencia es otro de los módulos importantes que describen lo que el sistema deberá realizar. Deberá permitir al usuario introducir marcas que indiquen que en esas coordenadas existe un punto de competencia.

3.3 Requisitos no funcionales

Aparte de los requisitos funcionales que se exponen en el apartado anterior, existen otros requerimientos que también se tienen que tener en cuenta en el diseño del sistema.

Conocidos también como requisitos de calidad, los requisitos no funcionales, se definen para especificar las cualidades generales que debe tener el sistema al realizar su función. No están relacionados con funciones del sistema y se refieren a varios factores, como interfaz de usuario, documentación, consideraciones de hardware, características de calidad, económicas o políticas si el caso lo requiere:

Para esta herramienta será necesario definir los siguientes requisitos no funcionales:

- La herramienta deberá ser funcional independientemente del navegador que utilice el cliente. Se realizará el desarrollo en base a los navegadores más populares.
- La aplicación requerirá en todo momento acceso a Internet, por lo que uno de los requisitos principales será disponer de una tarjeta de red con acceso a Internet. Además para que funcione el servicio de mapas será necesario contar con el reproductor de Flash instalado.
- La interfaz de usuario estará desarrollada mediante una estructura homogénea utilizando el framework Ext JS, por tanto, deberá ser fácilmente modificable y con posibilidades de ser ampliada siguiendo la estructura y estilo web 2.0 marcados al inicio del desarrollo sin que pierda calidad en los servicios ofrecidos.
- El sistema debe ser fácil de usar y comprender ya que será un nexo de información entre las franquicias y los empleados de MRW justificando un entorno totalmente amigable.
- La respuesta de la aplicación en aspectos como la navegabilidad, visualización de información o inserción de marcas no debe alargarse en demasía en el tiempo, obteniendo una respuesta rápida y ágil en todo momento.

-
- La herramienta deberá garantizar la fiabilidad de la información que muestra el mapa en todo momento
 - En ampliaciones futuras, la aplicación presentará procesos de autenticación de usuarios por lo que identificar los objetivos de seguridad tales como confidencialidad, integridad y disponibilidad deberán ser contemplados y provistos en la aplicación.

3.4 Modelo de casos de uso

3.4.1 Diagrama de casos de uso

La especificación es el paso siguiente al análisis y sirve para describir cómo modelar la funcionalidad del sistema mediante casos de uso.

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal para invocar una cierta funcionalidad sobre el propio sistema.

Cada caso de uso puede proporcionar uno o más escenarios que indican cómo debería interactuar el sistema con el actor para conseguir un objetivo específico.

La especificación formal de un caso de uso incluye 3 elementos básicos:

- Actores: Representa los papeles que un usuario del sistema puede desempeñar.
- Restricciones, requisitos y escenarios
- El propio caso de uso o servicios que el sistema ejecutará y que describen una secuencia de eventos realizados por el sistema.

Además, un caso de uso puede 'incluir' la funcionalidad de otro caso de uso o puede 'extender' otro caso de uso con su propio comportamiento, como se puede observar en la siguiente figura:

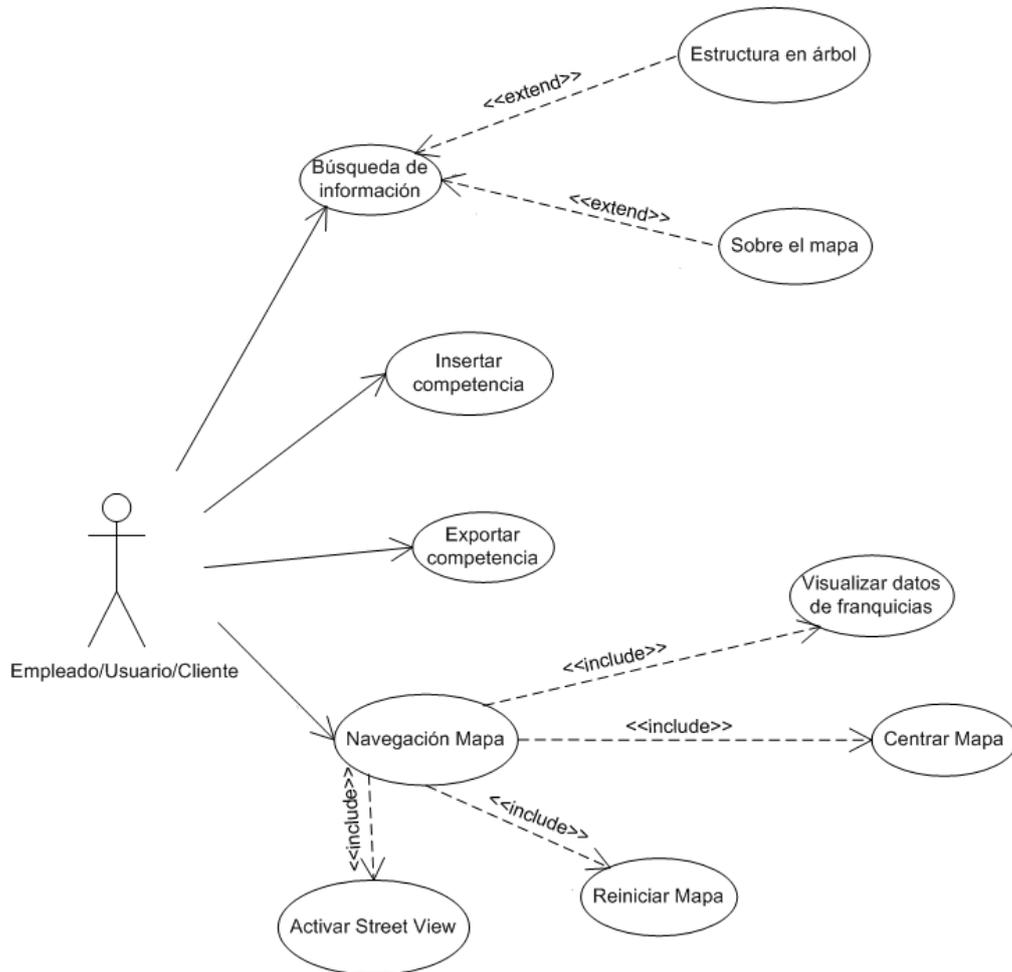


Fig. Diagrama de Casos de Uso

3.4.2 Especificación de los casos de uso

La especificación de casos de uso define el comportamiento del sistema en cada funcionalidad e interacción con el usuario de forma completa, precisa y variable. Debe describir correctamente todos los requisitos, sin importar el cómo se desarrolla y definir la interacción básica y las posibles alternativas.

Para cada caso de uso que aparece en los diagramas, se especificará la siguiente información:

- Descripción: Se detallará de forma textual que es lo que realiza el caso de uso.
- Actores: Actores que interactuarán con el sistema.

- Precondiciones: Hechos o acciones que deberán cumplirse previamente para que se pueda llevar a cabo un flujo.
- Poscondiciones: Hechos o acciones que se cumplirán si el flujo de eventos es el principal.
- Flujo principal: Hechos o acciones que se llevarán a cabo de forma exitosa, según su ejecución normal.
- Flujo alternativo: Hechos o acciones que indicarán cuando el sistema realiza aquellas ejecuciones menos frecuentes.

3.4.2.1 Caso de uso: Búsqueda de información por una estructura en árbol

Descripción

El usuario al iniciar la aplicación dispone de diferentes opciones para realizar la búsqueda de información. Una de ellas permite al usuario moverse con gran facilidad por una estructura de carpetas similar a la de los sistemas operativos y que representan el modo de visualización de las franquicias a 4 niveles: Por comunidad autónoma, por provincia, por municipio y por franquicia.

Actores

Empleado/Usuario/Cliente

Precondiciones

Se ha cargado el servicio de mapas en la aplicación y se ha recuperado la información que contiene los datos del árbol.

Flujo principal

1. El usuario inicia la aplicación desde el portal interno.
2. El sistema carga el mapa de Google Maps en la región central de la aplicación.
3. El sistema recupera los datos del fichero JSON y los muestra con una estructura en árbol en el panel lateral.
4. El usuario puede desplazarse por el árbol mediante los símbolos \pm para realizar

la búsqueda de información.

Poscondiciones

El usuario puede desplazarse por el árbol para buscar información.

3.4.2.2 Caso de uso: Búsqueda de información por el mapa

Descripción

Otra de las opciones que dispondrá el usuario cuando entre en la aplicación es realizar la búsqueda de información desplazándose directamente por el mapa. El usuario podrá moverse utilizando los botones de zoom.

Actores

Empleado/Usuario/Cliente

Precondiciones

Se carga el servicio de mapas en la aplicación y se recupera la información que contiene los datos del mapa. Esta información se representa mediante iconos, por lo que será necesario que se carguen las imágenes que lo componen.

Flujo principal

1. El usuario inicia la aplicación desde el portal interno.
2. El sistema carga el mapa de Google Maps en la región central de aplicación.
3. El sistema recupera los datos del fichero JSON y los muestra mediante unos iconos. Cuando el usuario realiza búsquedas de información en un nivel de zoom diferente al inicial el mapa le muestra información diferente.

Poscondiciones

El usuario puede moverse por el mapa para buscar información.

3.4.2.3 Caso de uso: Insertar competencia

Descripción

El usuario tendrá la posibilidad de insertar marcas en aquellos puntos geográficos que conozca que hay locales de la competencia. Dispondrá de tres posibilidades:

- a) Insertar marcas introduciendo la dirección.
- b) Insertar marcas mediante las coordenadas de latitud y longitud.
- c) Insertar marcas mediante un icono directamente sobre el mapa.

Actores

Empleado/Usuario/Cliente

Precondiciones

La aplicación se carga satisfactoriamente.

Flujo principal

1. El usuario escoge la opción insertar competencia mediante una pestaña situada en el inferior del panel.
2. El usuario escoge una de las tres opciones que le ofrece la aplicación.
3. Dependiendo de la elección:
 - a) El usuario introduce la dirección correcta y el sistema pinta una marca en el mapa.
 - b) El usuario introduce la latitud y la longitud y el sistema pinta una marca en el mapa.
 - c) El sistema le informa que ya puede realizar un clic en el mapa para introducir una marca.
El usuario realiza un 'clic' en el mapa.
El sistema pinta una marca en el mapa y le devuelve la dirección.

Flujo alternativo

3. El usuario introduce por error una dirección inexistente.

4. El sistema rebota un error indicando que la dirección no existe y que introduzca otra de nuevo.

Poscondiciones

El usuario escoge una de las tres opciones para insertar las marcas e inserta un punto en el mapa.

3.4.2.4 Caso de uso: Exportar competencia**Descripción**

El usuario tendrá la posibilidad de guardar las marcas que ha insertado previamente en el mapa exportándolas en un fichero XML para utilizarlo en otros sistemas.

Actores

Empleado/Usuario/Cliente

Precondiciones

El usuario ha insertado una marca en el mapa.

Flujo principal

1. El usuario introduce una marca en el mapa.
2. El usuario escoge la opción copiar datos al área de texto.
3. El sistema muestra el panel previamente escondido con los datos que acaba de insertar.
4. El usuario realiza la acción de exportar mediante un botón.
5. El sistema muestra el fichero con los datos guardados en formato XML.

Pos condiciones

Los datos insertados se muestran en formato XML.

3.4.2.5 Caso de uso: Visualizar datos de franquicias

Descripción El usuario puede visualizar en cualquier momento los datos de las franquicias. Datos como el nombre de la compañía, la dirección o, el código postal entre otros.
Actores Empleado/Usuario/Cliente
Precondiciones Se ha cargado el servicio de mapas en la aplicación y se ha recuperado la información que contiene los datos de las franquicias. Estos datos se muestran en el mapa mediante iconos o en una estructura en árbol del panel lateral.
Flujo principal <ol style="list-style-type: none">1. El usuario encuentra aquella franquicia de la cual desea obtener información2. El usuario realiza un 'clic' sobre el icono que muestra el número de la franquicia en cuestión.3. El sistema le muestra un globo informativo con todos los datos de la misma y una opción que le permite buscar el itinerario para poder llegar.
Poscondiciones El usuario visualiza la información de una franquicia.

3.4.2.6 Caso de uso: Activar modo callejero

Descripción El usuario en cualquier momento puede visualizar cualquier información disponible en el mapa mediante un callejero fotográfico. Deberá seleccionar la opción que estará en la barra de herramientas superior.

Actores

Empleado/Usuario/Cliente

Precondiciones

Se ha cargado el servicio de mapas en la aplicación.

El usuario tiene instalado el reproductor Flash para visualizar imágenes en 360 grados.

14 es el zoom mínimo requerido para visualizar esta vista.

Flujo principal

1. El usuario solicita al sistema que le muestre la opción vista callejera.
2. El sistema le muestra la opción vista callejera en una ventana acomodada en la región centro del mapa y le marca un icono para que el usuario pueda desplazarse.
3. El usuario desplaza el icono.
4. El sistema refresca la vista anterior y le muestra la nueva vista en la misma ventana.

Flujo alternativo - 1

1. El usuario solicita al sistema que le muestre la opción vista callejera.
2. Si el nivel de zoom es menor a 14 el sistema rebota un error indicando el mínimo zoom requerido para mostrar esta vista.

Flujo alternativo - 2

1. El usuario solicita al sistema que le muestre la opción vista callejera.
2. Si en esa población no existe la opción vista callejera el sistema rebota un error informándole el mínimo zoom requerido para mostrar esta vista.

Poscondiciones

El usuario puede visualizar información de forma panorámica mediante la vista callejera.

3.4.2.7 Caso de uso: Centrar mapa

Descripción El usuario puede en cualquier momento centrar la vista en el mapa haciendo 'clic' con el botón derecho del ratón.
Actores Empleado/Usuario/Cliente
Precondiciones Se ha cargado el servicio de mapas en la aplicación.
Flujo principal <ol style="list-style-type: none">1. El usuario realiza un 'clic' sobre una zona del mapa.2. El sistema le muestra un menú contextual con la opción Centrar Aquí.3. El usuario vuelve a 'clicar' aceptando la acción.4. El sistema automáticamente llama a la función que permite situar el mapa centrado en unas determinadas coordenadas.
Poscondiciones El sistema centra el mapa en la posición indicada por el usuario.

3.4.2.8 Caso de uso: Reiniciar mapa

Descripción El usuario puede en cualquier momento reiniciar el mapa para que vuelva al estado inicial de nuevo.
Actores Empleado/Usuario/Cliente
Precondiciones

Se ha cargado el servicio de mapas en la aplicación.

Flujo principal

5. El usuario realiza un 'clic' sobre el botón reiniciar mapa.
6. El sistema recarga el mapa volviendo otra vez al estado inicial.

Poscondiciones

El mapa se reinicia y se pierden aquellos cambios realizados sobre él.

Capítulo 4. Conocimientos Previos

4.1 Fundamentos teóricos

4.1.1 Google Maps

Google Maps es el nombre que recibe el servicio de mapas gratuito de Google. Anunciado por primera vez en febrero del 2005, Google Maps es un servidor de aplicaciones de mapas en web que ofrece imágenes de mapas vectoriales desplazables, fotos satelitales y más recientemente, desde de Mayo del 2007, imágenes casi esféricas a nivel de calle, permitiendo a los usuarios ver partes de las ciudades seleccionadas y sus áreas circundantes y que se conoce como Google Street View.

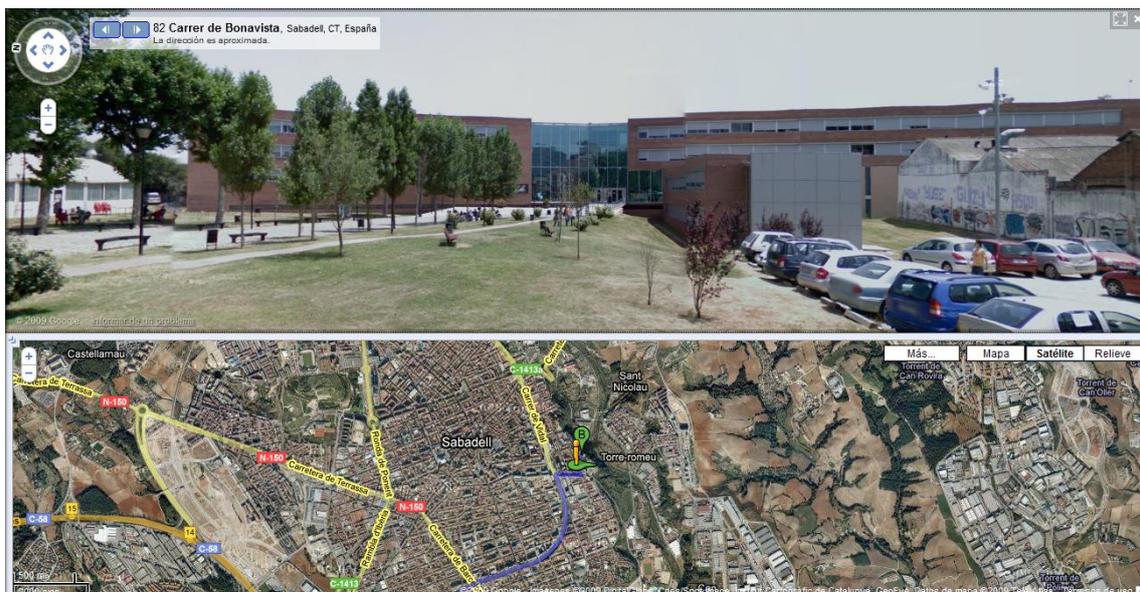


Fig. Vista panorámica con Google Street View de la Esola Universitaria d'Informàtica de Sabadell.

La misión geográfica por tanto, es organizar toda la información que tiene que ver con una localización, en términos de latitud y longitud, y hacerla fácil y accesible mediante mapas.

Bien es sabido que muchas de las aplicaciones de Google Maps, utilizan un gran número de archivos JavaScript y XML. De ahí, uno de los muchos motivos para desarrollar esta aplicación usando los servicios de mapas actualizados que Google Maps ofrece libremente, ya que todas las aplicaciones que se desarrollarán en este portal se ejecutarán en el navegador del cliente, por tanto, el uso de lenguajes como JavaScript favorecerán a la unión de todas ellas en una misma herramienta.

Para lograr la conectividad asíncrona, Google aplica la tecnología AJAX¹¹ característica que evitará que el usuario tenga que refrescar todo el contenido de la página a cada movimiento del mapa.

4.1.1.1 Características básicas

Aparte de las características comentadas en el punto anterior, Google Maps ofrece las siguientes posibilidades. Brevemente se resumen:

- **Búsqueda:** Se puede realizar una búsqueda introduciendo el nombre de la calle o una población. Ese punto es marcado por un indicador en forma de pin, el cual es una imagen transparente sobre el mapa. Además, permite encontrar una gran variedad de restaurantes, teatros, hoteles y negocios desde una misma web, similar al concepto que tienen sistemas de posicionamiento global por satélite.
- **Control sobre el mapa:** Con Google Maps se pueden hacer acercamientos o alejamientos para mostrar el mapa a diferentes niveles de zoom.
- **Codificación geográfica inversa:** Desde muy poco tiempo atrás, Google ofrece esta característica que consiste en devolver un objeto JSON

¹¹AJAX (JavaScript asíncrono y XML): Es una técnica de desarrollo web para crear aplicaciones interactivas ejecutadas en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

estructurado de la consulta de una dirección mediante dirección más cercana dentro de un determinado margen.

- Rutas o Direcciones: Google permite establecer la ruta itinerante entre dos puntos o dos direcciones. Establece dos formas predeterminadas de viaje: vehículo o a pié. Por defecto son instrucciones para ir en coche.
- Servicios: Google Maps ofrece un amplio número de servicios como Panorámico (ofrece al usuario la posibilidad de conocer diferentes zonas geográficas mediante el uso de fotografías), entradas de la Wikipedia, webcams o la situación actual del tráfico (por ahora sólo en algunas ciudades, ninguna en España).

En resumen, Google ofrece un servicio de mapas de gran versatilidad que permite integrar en una página web muchas de estas características mediante su API.

4.1.2 Ext JS

Es la librería más avanzada para el desarrollo rápido de aplicaciones, que ofrece herramientas para crear interfaces gráficas totalmente novedosas y una arquitectura flexible.

De acuerdo a la definición de la página web, Ext JS es una librería que permite construir aplicaciones complejas en Internet. Se caracteriza por:

- Componentes UI (Interfaz de usuario) de alto nivel y personalizables: Menús, tablas, layouts, paneles, formularios, pestañas...
- Modelo de componentes extensibles.
- API fácil de usar.
- Extensa comunidad de usuarios
- Su licencia es Open Source aunque también disponen de una licencia comercial.
- Código reutilizable

Es una excelente forma de programar en JavaScript ya que provee los objetos necesarios para recorrer y manipular los elementos DOM de una manera mucho más sencilla, añadiendo además conjuntos de elementos gráficos muy cuidados y todo lo necesario para construir atractivos desarrollos al estilo Web 2.0.

El uso de esta librería es tremendamente útil, son casi todas ventajas. Su potencia radica en la rica colección de componentes para el diseño de interfaces de usuario del lado del cliente haciendo uso extensivo de AJAX y, de la potencia de procesamiento permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.

4.2 Fundamentos prácticos

4.2.1 API de Google Maps

El API de Google Maps proporciona diferentes utilidades para crear innovadoras aplicaciones geográficas. Se definen como pequeños programas que se instalan en el servidor que hace el mashup. El API contacta con el sitio que provee de datos (Google Maps) y le pide que le envíe los datos que en aquel momento requiere el usuario mostrándolos mediante el navegador.

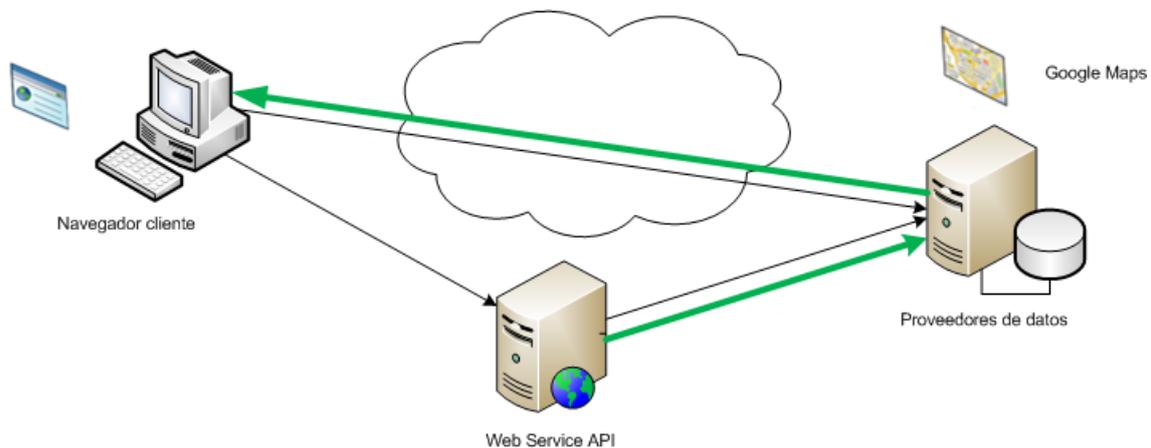


Fig. Web API

A partir de aquí, se definen aquellas funciones que deberán utilizarse para desarrollar la plataforma de mapas en el portal de MRW. No pretende ser un manual de referencia del API sino una ayuda para que el proceso de creación del mapa en el portal sea lo más sencillo posible.

Antes de comenzar, será necesario disponer de una clave. Para conseguirla se deberá registrar la dirección web que contendrá el mapa en el siguiente enlace aceptando unos términos y condiciones que se pueden encontrar en la siguiente dirección web:

<http://www.google.com/apis/maps/signup.html>

Esta clave será únicamente válida para un directorio o dominio y estará vinculada a una cuenta de Google.

Una vez realizado este proceso previo será necesario crear el documento HTML que contendrá la clave y que hará referencia al API y, reservar un espacio en el documento para incrustar el mapa. Se deberá asignar un identificador y especificar ancho y largo. La idea es que al formar parte de una región el ancho y el alto deben especificarse en %.

Una vez reservado espacio para el mapa en la web, será necesario crear una instancia de la clase tipo GMap2.

index.html

```
<script type="text/javascript"
    src="http://maps.google.com/maps?file=api&v=2&key=CLAVE_MRW">
</script>
...
<body onload="cargar_mapa();" onunload="GUnload();">
<div id="mapa" style="width: 100%; height: 100%"></div>
</body>
...
```

interface.js

```
var map=null;
function cargar_mapa(){
...
map = new GMap2(document.getElementById("mapa"));
...
}
```

A partir de aquí, ya se podrá acceder a las funcionalidades que nos brinda el API de Google MAPS. Se pueden añadir controles, seleccionar diferentes tipos de mapa, añadir ventanas de información, crear eventos, rutas...

Para tener una visión más amplia será imprescindible consultar la documentación oficial disponible en el siguiente enlace:

<http://code.google.com/intl/es/apis/maps/documentation/reference.html>

Existen muchas funcionalidades descritas ampliamente en la web que permiten tener un control más exhaustivo sobre el mapa. Entre todas ellas será imprescindible acoplar en la aplicación las que a continuación se describen. Serán vistas como las funcionalidades más importantes a desarrollar y sobre las que se basa la originalidad de la aplicación a realizar para MRW:

- **Marker Manager:** Aglutina las funcionalidades de gestión de marcas de modo que se puedan definir para cada uno de los puntos (o conjuntos de ellos) un rango de zooms mediante los cuales tendrán visibilidad. Permitirá definir según el nivel de zoom diferentes vistas de las franquicias.

```
var mmanager = new MarkerManager(mapa); //Inicialización de la variable

//Se añaden las marcas en una array para visualizarlas a diferentes niveles de zoom.
mmanager.addMarkers(lista_Markers, zoomMin, zoomMax);

mmanager.refresh(); //Refresca el gestor al cambiar el nivel de zoom
```

- **El objeto GStreetViewPanorama:** Como el nombre indica, se trata de proporcionar al mapa vistas panorámicas de 360 grados. Para incorporar Street View será necesario crear un contenedor (“pano”) para el visor Flash y un objeto GStreetViewPanorama, además de indicar una orientación de modo que especifiquen la ubicación deseada.

```
var myPano = new GStreetviewPanorama(document.getElementById("pano")); //Contenedor
point = new GLatLng(42.345573,-71.098326); //Localización geográfica
myPOV = {yaw:370.64659986187695,pitch:-20}; //Orientación
myPano.setLocationAndPOV(point, myPOV); //Superposición de la vista panorámica
```

- **Codificación Geográfica:** Esta funcionalidad permitirá realizar búsquedas geográficas avanzadas. Su funcionamiento se basa en transformar una dirección (cadena de texto) en un objeto GLatLng (latitud y longitud).

```
var geocoder = new GClientGeocoder();

function showAddress(direccion) {
  geocoder.getLatLng(
    direccion,
    function(point) {
      if (!point) {
        alert(direccion + " no encontrada");
      } else {
        map.setCenter(point, 13); //Centra el punto en el mapa
        var marker = new GMarker(point); //Crea una marca
        map.addOverlay(marker); //Pinta en el mapa la marca creada
        marker.openInfoWindowHtml(direccion); //Añade una ventana de información
      }
    }
  );
}
```

- **Análisis de datos y código XML:** Esta funcionalidad permitirá descargar archivos de datos para mostrar su contenido en el mapa.

```
//Descarga el fichero 'data.xml' que contiene una lista de coordenadas de latitud y
longitud en XML
GDownloadUrl("data.xml", function(data, responseCode) {
  var xml = GXml.parse(data);

  var markers = xml.documentElement.getElementsByTagName("marker");

  //Recorre el fichero en busca de tags que contengan: marker, guardando su localización
  for (var i = 0; i < markers.length; i++) {
    var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
                             parseFloat(markers[i].getAttribute("lng")));

    map.addOverlay(new GMarker(point)); //Pinta las marcas en el mapa
  }
});
```

4.2.2 Empezar con Ext JS

Para comenzar será necesario tener instalado algún programa para el desarrollo de aplicaciones web o bien usar un servidor web local mediante programas que facilitan la instalación de Apache¹² y otros lenguajes.

¹²Apache: Servidor HTTP de código abierto multiplataforma que implementa la idea de sitio virtual mediante el protocolo HTTP.

Seguidamente será necesario importar el framework para poder comenzar a desarrollar con éste. Será necesario importar los siguientes archivos en la cabecera del mismo documento HTML (de ahora en adelante *index.html*) creado anteriormente para introducir la clave y los controles de la API de Google Maps como se muestra a continuación:

- `ext-2.2/ext-all.js`: Archivo donde se encuentran todos los componentes del framework.
- `ext-2.2/resources/css/ext-all.css`: Archivo necesario para dotar de un estilo gráfico a todos estos componentes y controles.
- `ext-2.2/adaptor/ext/ext-base.js`: Archivo que contiene la librería para manejo de DOM, AJAX, animaciones, etc.

`index.html`

```
...  
  
<link rel="stylesheet" type="text/css" href="../ext-2.2/resources/css/ext-all.css"/>  
<script type="text/javascript" src="../ext-2.2/adaptor/ext/ext-base.js"></script>  
<script type="text/javascript" src="../ext-2.2/ext-all.js"></script>  
  
<script type="text/javascript"  
  
    src="http://maps.google.com/maps?file=api&v=2&key=CLAVE_MRW">  
  
</script>  
  
...  
  
<body onload="cargar_mapa();" onunload="GUnload();">  
  
<div id="mapa" style="width: 100%; height: 100%"></div>  
  
</body>  
  
...
```

El paso siguiente es cargar el método `OnReady`. Es probablemente el primer método que se usa en todas las páginas. Este método es llamado automáticamente una vez que el DOM ha sido cargado por completo, garantizando que cualquier elemento de la página al que se quiere referenciar ya está disponible cuando se ejecuta el script.

```
interface.js

var map=null;

function cargar_mapa(){

Ext.onReady(function(){//Objeto Ext

If(GBrowserIsCompatible()){

map = new GMap2(document.getElementById("mapa"));

}else{
  alert ("Disculpa. Tu navegador no es compatible con Google Maps)
}

}
...
});
```

Tener claro que son y como interactúa cada uno de los componentes permitirá obtener aplicaciones similares a las que se describen en el siguiente capítulo.

Aspectos como los que a continuación se resumen serán frecuentes en el desarrollo de esta aplicación. Por tanto será importante conocer la documentación de ésta y otras librerías tanto en el proceso de aprendizaje como en el desarrollo de casos prácticos ya que es mucho más eficaz y rápido encontrar soluciones en estos documentos oficiales.

- Objeto Ext: Es el objeto principal de la librería, siempre debe hacerse referencia a ella mediante este objeto.
- Viewport: El objeto Viewport representa el contenedor principal y es allí donde se ponen todos los componentes.
- Layout: Indica cómo se alinea cada uno de los componentes.

Para poder obtener aplicaciones web de alta originalidad será necesario visitar frecuentemente la siguiente dirección, ya que la documentación suele renovarse y ampliarse con frecuencia:

<http://extjs.com/deploy/dev/docs/>

4.3 Tecnologías de desarrollo

4.3.1 HTML

HTML es el lenguaje con el que se definen las páginas web. Es el acrónimo de Hypertext Markup Language o lo que es lo mismo, lenguaje basado en marcas diseñado para la creación y publicación de páginas web. Es una sintaxis para definir los elementos que se quieren mostrar en una página, su estructura y ubicación, así como las relaciones entre ellos y con otros elementos de la web mediante hipervínculos. Además el lenguaje HTML es un estándar reconocido hoy en día en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C.

La primera descripción de HTML publicada por primera vez fue en el año 1991 y su desarrollador es el físico británico Tim Berners-Lee. Después evolucionaría hasta llegar a cuatro versiones más. HTML 4.0 fue la última en 1998.

Junto a Robert Cailliau, Berners-Lee es uno de los precursores de lo que se conoce hoy como World Wide Web, además de ser fundador del W3C citado anteriormente.

Principalmente se basa en atributos y etiquetas proporcionando la estructura al documento HTML e indicando al navegador cómo se presenta un sitio web. Por lo general, los elementos están formados por una etiqueta de inicio, el contenido, y una etiqueta de cierre: `<etiqueta>Texto</etiqueta>`

La principal ventaja de este tipo de lenguajes es que son muy sencillos de leer y permiten ser interpretados por cualquier sistema electrónico de hoy en día.

Originariamente estos documentos solo contenían texto e imágenes pero a medida que el desarrollo del estándar HTML avanzaba las páginas que se creaban incluían otro tipo de información sobre aspectos como: tipos de letra, colores, párrafos, márgenes..., por lo que se definió el lenguaje CSS, que a continuación se resume.

4.3.2 CSS

Las hojas de estilo, conocidas como CSS (Cascade Style Sheet) ofrecen un control total sobre el estilo y el formato del documento HTML, permitiéndolo entender mejor y reduciendo considerablemente su tamaño.

Originariamente, el W3C propuso la creación de estas hojas de estilo como estándar web pero, no fue hasta el año 1995 cuando Lie y Bos, sus creadores, se unieron para definir lo que sería la primera versión oficial de CSS nivel 1 y que más tarde publicaría el propio organismo que lanzó la propuesta.

CSS por tanto es el mecanismo que permite separar los contenidos definidos mediante HTML y el aspecto de éstos en la pantalla del usuario, es decir, separa la estructura de un documento de su presentación.

Una de las ventajas más importantes es que al separar contenido de su presentación los documentos creados son más flexibles, ya que se adaptan mejor a las diferentes plataformas, con lo que aumenta considerablemente su accesibilidad y adaptabilidad. Además la información puede ir incluida también en el documento HTML mediante el atributo `<style>`.

4.3.3 JavaScript

JavaScript es un lenguaje de scripts, interpretado, multiplataforma y orientado a objetos que se utiliza principalmente para crear páginas web dinámicas que no necesitan ser compiladas para su ejecución. En otras palabras, los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador y sin necesidad de procesos intermedios ni accesos al servidor.

Es el lenguaje más utilizado por los programadores web para realizar partes del desarrollo de una web ya que permite enlazarse o añadirse a los documentos web proporcionando un control total y dinámico sobre ellos. Además, también permite controlar las aplicaciones que lo ejecutan, habitualmente navegadores.

JavaScript nace de un anuncio conjunto entre Sun Microsystems y Netscape, allá por Diciembre de 1995. Netscape, empresa creadora del conocido navegador web Netscape Navigator comenzó a desarrollar un lenguaje de programación al que llamó LiveScript que permitía crear pequeños programas en las páginas y que era mucho más

sencillo de utilizar que Java. Para evitar una guerra de tecnologías en 1997 se envió la especificación al organismo ECMA¹³ para su estandarización. Además el W3C diseñó el DOM¹⁴ para evitar incompatibilidades.

Para incluir JavaScript en cualquier documento web únicamente es necesario añadir en el documento HTML la siguiente cabecera que hará referencia al script:

```
<script type="text/javascript" src="archivo.js"></script>
```

4.3.4 JSON

JSON, es un formato ligero de intercambio de datos basado en JavaScript. Sus siglas especifican que el intercambio de estos datos se ha creado a partir de un subconjunto de la notación de objetos en JavaScript.

Se emplea habitualmente en entornos donde el intercambio y el tamaño de datos entre cliente y servidor de forma asíncrona es de vital importancia, de ahí su uso en Google por ejemplo.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un *objeto*, registro, estructura, diccionario, tabla hash o lista de claves
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como vectores, listas o secuencias.

```
var dirección = {
  "dirección"      : "Gran Via de l'Hospitalet",
  "ciudad"         : "Hospitalet de Llobregat",
  "codigoPostal"  : 08908
};
```

¹³ECMA: Es una organización internacional que contribuye activamente a la estandarización a nivel mundial de la tecnología de la información y las telecomunicaciones.

¹⁴DOM (Modelo de Objetos del Documento): Interfaz de programación de aplicaciones que proporciona un conjunto de estándares para representar documentos HTML y XML.

Como se observa la sintaxis puede ser muy conveniente para escribir código más compacto y portable, de forma que lo hacen muy útil

4.3.5 XML

XML es un lenguaje con una importante función de intercambio entre plataformas, estructuración y envío de datos en la web, como si de una base de datos se tratara.

Es un lenguaje en continuo desarrollo, desde que en 1998 el organismo W3C lo estandarizara. Se basa en la utilización de etiquetas, de la misma manera que lo hace HTML, por lo que favorece la operatividad entre los mismos.

Sus siglas significan lenguaje de marcas extensible, por lo que esta basado en un conjunto de reglas que definen etiquetas semánticas para organizar un documento, o lo que es lo mismo, se trata de un lenguaje diseñado precisamente para hacer frente a los problemas de compatibilidad y adaptabilidad de las nuevas tecnologías a Internet.

Se considera uno de los lenguajes más potentes en servicios web e intercambio de información ya que busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible.

4.3.6 JSON vs XML

Desde la aparición de JSON, su rápido crecimiento y aceptación en la red en los últimos años ha abierto un gran debate sobre que formato es mejor: XML o JSON.

Una de las ventajas principales de JSON, es que es mucho más entendible y fácil de escribir por la forma en que presenta la información. En la figura siguiente se puede observar esto:

<pre>[{"value": "1", "text": "Barcelona"}, {"value": "2", "text": "Madrid"}, {"value": "3", "text": "Sevilla"}, {"value": "4", "text": "Bilbao"},]</pre>	<pre><?xml version="1.0" ?> <selectChoice> <entry> <optionText>1</optionText> <optionValue>Barcelona</optionValue> </entry> <entry> <optionText>2</optionText> <optionValue>Madrid</optionValue> </entry> <entry> <optionText>3</optionText> <optionValue>Sevilla</optionValue> </entry> <entry> <optionText>4</optionText> <optionValue>Bilbao</optionValue> </entry> </selectChoice></pre>
--	--

Fig. Comparativa de formatos entre JSON y XML respectivamente

La diferencia de peso entre los dos formatos es considerable sobre todo para el uso con AJAX.

Características y diferencias:

En la siguiente tabla se especifican de manera detallada similitudes y diferencias que existen entre estos dos formatos de intercambio de datos:

Fuente: <http://msdn.microsoft.com/es-es/library/bb299886.aspx#EEAA>

CARACTERÍSTICAS	XML	JSON
Tipos de datos	No ofrece ninguna noción de tipos de datos. Hay que usar el esquema XML para agregar información de tipos.	Ofrece tipos de datos escalares y la capacidad de expresar datos estructurados a través de matrices y objetos.
Compatibilidad con objetos y matrices	Los objetos y las matrices se deben expresar mediante convenciones, con frecuencia a través del uso combinado de atributos y elementos.	Compatibilidad nativa con objetos y matrices.
Espacios de nombres	Admite espacios de nombres	Sin concepto de espacios de nombres. Los conflictos para asignar nombres se evitan generalmente mediante el anidado de objetos.

Formato	Complejo. Se requiere un mayor esfuerzo para decidir cómo asignar tipos de aplicación a elementos y atributos XML. Puede generar muchas discusiones sobre si es preferible un enfoque centrado en elementos o en atributos.	Sencillas. Proporciona una asignación mucho más directa para los datos de aplicación. La única excepción puede ser la ausencia del literal de fecha/hora.
Tamaño	Los documentos tienden a tener gran tamaño, especialmente cuando se usa para el formato el enfoque centrado en elementos	La sintaxis es muy concisa y da como resultado texto con formato en el que la mayor parte del espacio.
Análisis en JavaScript	Se necesita una implementación de XML DOM y código de aplicación adicional para volver a asignar texto en objetos JavaScript.	No se necesita código de aplicación adicional para analizar texto; se puede usar la función eval de JavaScript.
Curva de Aprendizaje	Generalmente tiende a necesitar del uso de varias tecnologías en colaboración	Pila de tecnología muy sencilla conocida por los desarrolladores con experiencia en JavaScript u otros lenguajes de programación dinámicos.

Fig. Diferencias de características claves entre XML y JSON

Ambos formatos son perfectamente sustitutivos y complementarios a la vez, por lo tanto, decidir sobre que formato es mejor no consiste en comprobar cuál es el que ofrece más características. Se debe determinar como la mejor elección aquel que lo convierta en la opción destacada para un determinado caso en concreto.

En resumen, parece que JSON representa mejor, más ordenada y entendible la información por lo que es el formato adecuado en entornos AJAX donde, evitándose el proceso de parseo de los datos XML, requiere menos codificación y procesamiento.

Capítulo 5. Diseño

5.1 Introducción

La fase de diseño de software es aquella etapa que define con total detalle cómo se desarrollará el sistema para que realice las funcionalidades descritas en las fases anteriores, implementando todos los requisitos contenidos en el modelo de análisis y especificación.

Basándonos en las tecnologías más actuales para el desarrollo de aplicaciones web, la idea consistirá en realizar una aplicación online estructurada mediante HTML, de la misma manera que lo hace la librería de JavaScript Ext JS para el apartado gráfico y que Google Maps servirá como nexo para especificar todos aquellos elementos que MRW quiera registrar en un mapa, permitirán obtener un portal con un 'look & feel' definido, claro y de alto nivel.

Por tanto, antes de comenzar con el desarrollo será importante conocer aquellos aspectos necesarios para llevar a cabo una aplicación de este tipo y mostrar algunas de las pautas para seguir con la tendencia marcada por la Web 2.0.

Durante las siguientes páginas, se fijarán aspectos más detallados de cuáles serán los módulos, mostrando todo esto en una interfaz gráfica de usuario que permita visualizar al lector cuál es la potencia que puede llegar a tener una aplicación de esta magnitud, además de proporcionar una idea de desarrollo completamente detallada sobre cómo comenzar a codificar esta aplicación.

5.2 Diseño de la interfaz gráfica

5.2.1 Introducción

De forma abreviada, también se le denomina GUI (Graphic User Interface), es uno de los apartados más cuidados por los diseñadores y desarrolladores de sistemas de información.

Con la llegada de la Web 2.0, la preocupación por el aspecto final de una aplicación web ha pasado a un primer plano. Para el usuario, la interfaz gráfica es el sistema por lo que, aspectos como usabilidad, accesibilidad y estética son vitales para que el sistema tenga una buena aceptación y no produzca rechazo.

El diseño de la interfaz gráfica siempre dependerá del tipo de aplicación que se desarrolla pero todas siguen unas reglas básicas que a continuación se exponen:

Para entender mejor estas reglas, lo más importante es ponerse en el lugar del usuario final, ya que el éxito de estas aplicaciones se basan en que el usuario pueda interactuar y establecer un contacto lo más rápido posible con la herramienta.

- El objeto principal o idea que se intenta plasmar en una web, ha de ser de fácil identificación. La manera de mostrar la información o el contenido es una de las partes más cuidadas a la hora de realizar el diseño.
- Facilidad de comprensión, aprendizaje y uso de los elementos que la componen, evitando así saturación y confusión al usuario.
- Representación fija y permanente de un determinado contexto de acción.
- Mantener el mismo estilo a lo largo y ancho de la web.
- Diseño ergonómico mediante menús, barras de acciones e iconos de fácil acceso. Además, tanto el texto como los gráficos y los colores utilizados deben corresponderse al contenido de la acción que realizan.
- Operaciones rápidas, incrementales y reversibles, con efectos inmediatos.
- Tratamiento del error bien cuidado y adecuado al nivel del usuario.

En líneas generales, se ha pensado en un entorno donde el mapa con la representación de los elementos mediante iconos indicando el número de franquicias por zona, ocupe la mayor parte de la ventana del navegador. Además, se ha buscado mostrar toda esta información de una manera limpia y ordenada evitando la sobrecarga de controles para realizar acciones que puedan provocar confusiones.

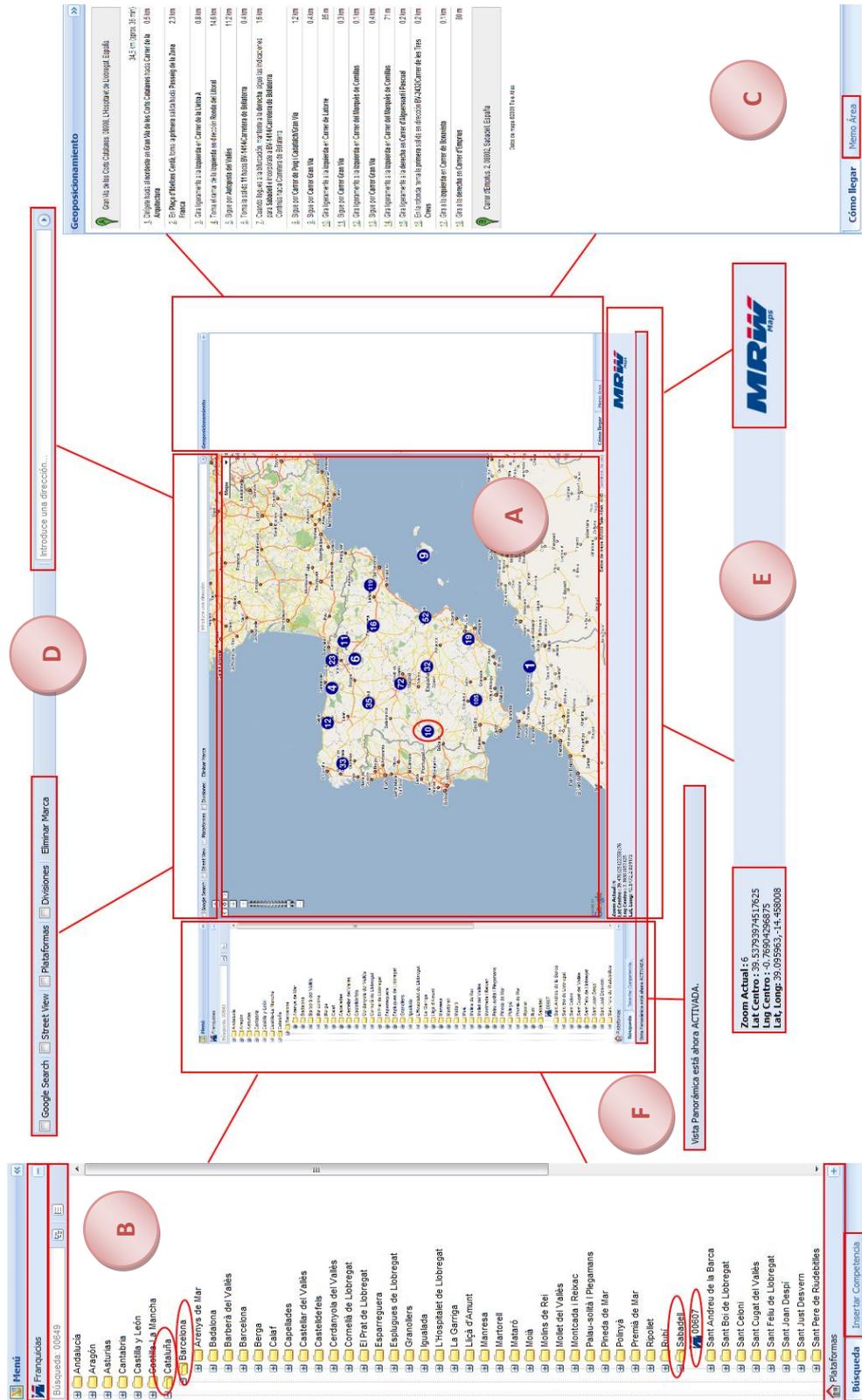
Se añadirá una barra de herramientas en la cabecera de la página con diferentes opciones mediante las cuales el usuario podrá interactuar con el mapa, seleccionando y deseleccionando aquellas que desee mostrar. Se creará además una región sud con información del mapa para que el usuario en todo momento pueda controlar sus movimientos. Además, este tipo de aplicaciones deberán contar con un feedback informativo que alerte al usuario de todos los movimientos que está realizando y cuáles han sido sus últimas acciones realizadas. Por tanto se introducirá también una barra de estado para este fin.

En cuanto a los menús laterales se refiere, ambos en diferentes regiones este y oeste, se definirán aquellos elementos que dependen directamente de las acciones que realice el usuario. Consistirá en un árbol de navegación a varios niveles de profundidad, además de incluir diferentes opciones para insertar puntos de competencia y paneles de información para saber cómo llegar, así como la posibilidad de exportar datos en un fichero XML.

Por tanto, por todo lo anterior, la elección para el desarrollo de la interfaz gráfica de la herramienta fue Ext JS. Como se expone anteriormente en el punto 4.2.2, esta librería de JavaScript ofrece soluciones para desarrollar herramientas totalmente adaptables a las necesidades del cliente, soluciones que cuentan con elementos gráficos que enriquecen la experiencia del mismo.

A continuación se podrá visualizar un prototipo basado en los requerimientos que se han planteado durante el desarrollo de este estudio y que se analizan con detalle.

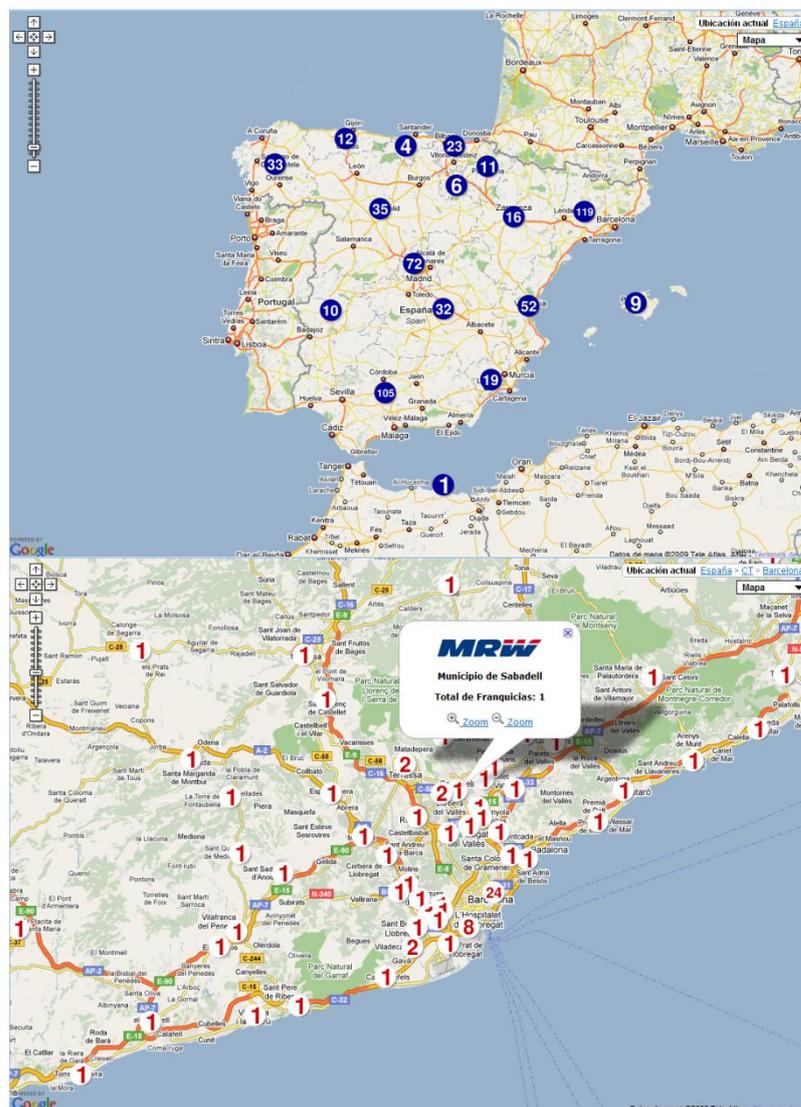
5.2.2 Interfaz de Usuario



Se pueden distinguir varias regiones:

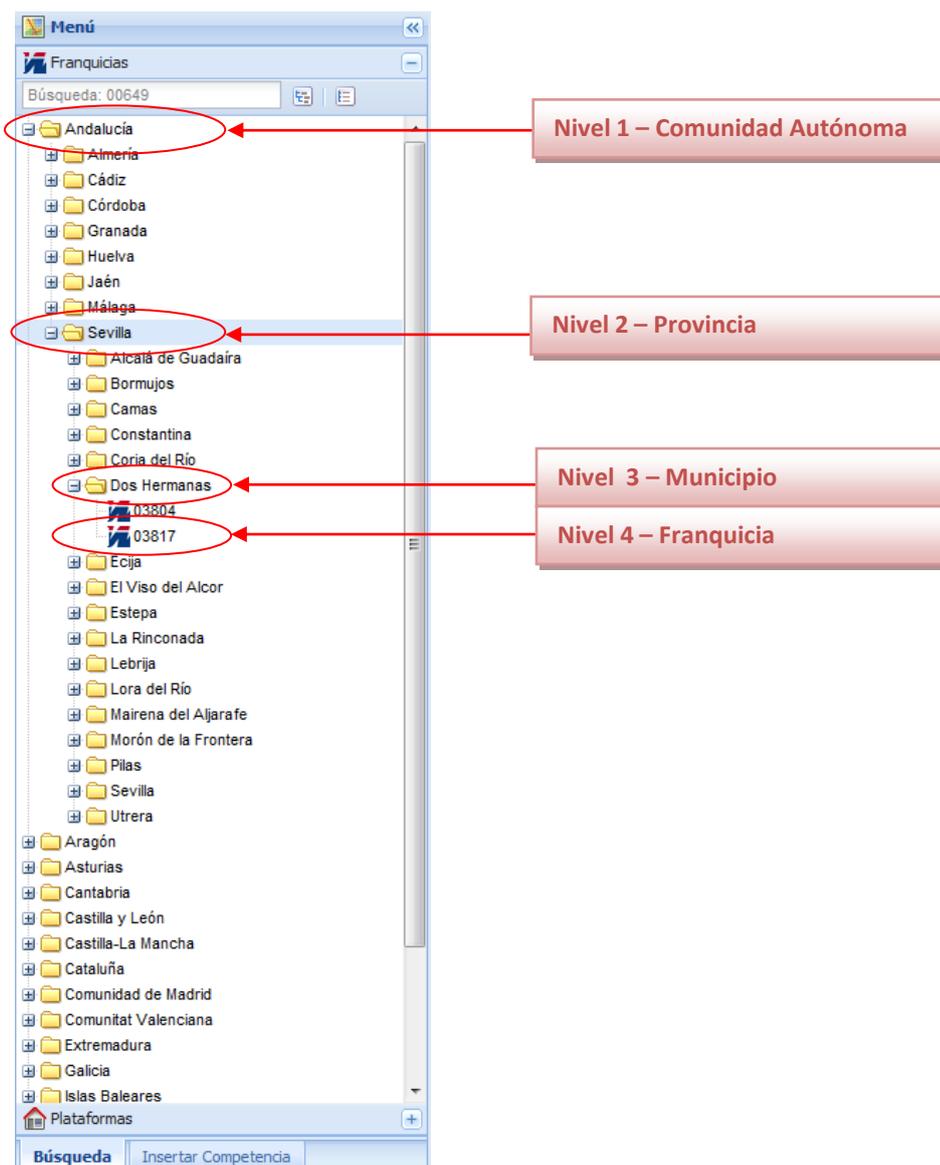
(A) – Una región central, donde se situará el mapa y los controles de zoom, movimiento y barra de navegación. Además el usuario podrá escoger el tipo de mapa: Mapa, Satélite, Híbrido (por defecto: Mapa). Mediante una función recursiva que realizará una lectura de marcas de una estructura JSON se pintarán todos aquellos puntos cuya latitud y longitud sea soportada.

El usuario podrá interactuar con el mapa mediante un evento zoom. A mayor nivel de profundidad mayor nivel de detalle de la franquicia. Todos los cambios se modificarán a medida que el usuario cambia de profundidad. Automáticamente el mapa se refrescará a cada cambio de zoom.



(B) – El panel lateral situado en la región oeste tendrá claramente definida su estructura. El objeto principal de ésta, será un menú acordeón con diferentes opciones: Franquicias y Plataformas, situadas dentro de una misma pestaña.

Protagonistas principales, las más de 600 franquicias de MRW incrustadas en el mapa se representarán mediante la lectura de datos del archivo JSON nuevamente. La manera de interactuar con el mapa y el usuario es bastante clara. Basándose en la estructura de cualquier sistema operativo, el árbol contará con cuatro niveles representados de la siguiente manera:



Además el empleado dispondrá de un buscador que filtrará su búsqueda por nombre.

Para la opción Plataformas, la metodología seguirá siendo la misma, lo que variará será la disposición de los datos para poderlos representar en una tabla o mediante una lista. El usuario deberá poder clicar y hacer que automáticamente se muestre en el mapa.

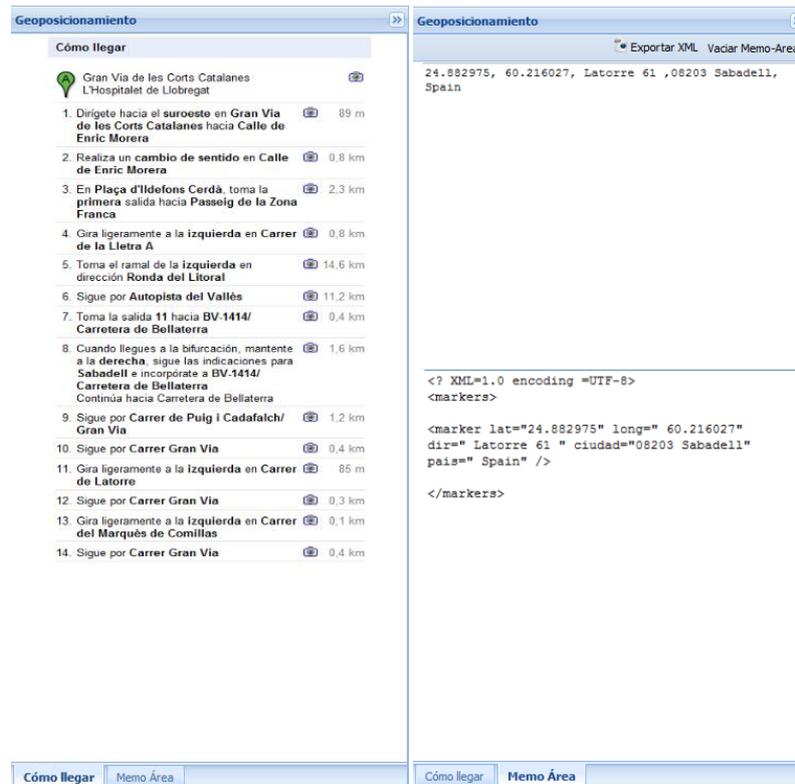
El usuario como ya se ha citado en varias ocasiones anteriores, podrá introducir puntos fuertes de competencia. Para ello dispondrá de un menú con tres opciones:



The image shows a software dialog box titled "Menú". It contains three main sections, each with a checked checkbox:

- Introduce la dirección:** Includes a text input field labeled "Dirección:" and a button labeled "Ir >>".
- Introduce las coordenadas:** Includes two text input fields labeled "Latitud:" and "Longitud:", and two buttons labeled "Crear Nueva Marca" and "Eliminar Marca".
- Geocoder:** Includes a checked checkbox for "Geocoder" and an unchecked checkbox for "Geocoder Inverso".

(C) – La función principal de este panel será informativa. Inicialmente no será visible por lo que requiere la acción del usuario. Única y exclusivamente será visible cuando el usuario necesite saber cómo llegar a una franquicia o cuando haya introducido un punto y requiera exportarlo a XML para ser utilizado en otros sistemas. En ese caso se desplegaría el panel mostrando la ruta establecida para visualizar con una ruta cómo llegar desde una dirección de origen a la franquicia o bien si su elección ha sido exportar los datos, le aparecerán las opciones para generar el fichero XML.



(D) – Una barra de herramientas con múltiples opciones para el usuario, permitirán filtrar el mapa según las necesidades de éste. Aspectos como: Cargar todas las plataformas en el mapa junto a las franquicias, visualizar el callejero fotográfico proporcionado por Google Maps, reiniciar el mapa o realizar búsquedas de negocios.

(E) – (F) – Feedback informativo: El usuario sabrá en todo momento en que nivel de zoom está, a que latitud y longitud le corresponde una determinada, además de una barra de estado en la región sud que le indica todas las acciones que va realizando.

Geocoder inverso está ahora ACTIVADO. Para desactivarlo vuelve a hacer "click" en el mismo elemento

(*) – Control de errores: En toda aplicación es necesario reportar mensajes para que el usuario sepa que hay algo que realiza de forma incorrecta. Por eso se crearán mensajes de alerta acordes también al estilo marcado en todo este desarrollo.



Después de conocer un poco más, aspectos detallados de la interfaz de usuario será necesario valorar que opciones llegarán a implementarse con éxito y, cuáles de ellas serán funcionales para la implementación de esta herramienta en una primera fase demo.

Capítulo 6. Implementación y Pruebas

6.1 Implementación

Es en esta fase del desarrollo de software donde se lleva a cabo la programación de la aplicación, plasmando todos los requisitos y especificaciones de las fases anteriores en código ejecutable.

Su implementación, como se ha visto en el capítulo anterior, estará basada principalmente en el uso de los mapas que ofrece Google. Debido a que gran parte del desarrollo de Google Maps se basa en la utilización de componentes JavaScript la librería escogida será Ext JS.

Por tanto, una vez que el diseño se ha realizado el siguiente paso, será la codificación.

Como en toda codificación de aplicaciones web es importante realizar pruebas de validación de aquellos métodos o funciones que se van implantando.

Por eso, junto a este punto se indican algunas pruebas que se han ido realizando conjuntamente con el desarrollo.

6.2 Pruebas

6.2.1 Introducción

Las pruebas son un conjunto de técnicas muy importantes en el desarrollo de software que permiten detectar los errores que no se han tenido en cuenta en la fase de codificación del proyecto y que ayudan también a determinar la calidad de éste.

Probar exhaustivamente un software es una tarea muy laboriosa y complicada por lo que ningún juego de pruebas puede garantizar que un sistema o aplicación esté totalmente libre de fallos. Por tanto, se buscará comprobar la compatibilidad del sistema detectando los fallos de funcionamiento, evaluando su usabilidad y midiendo su rendimiento en red.

Entonces, si el resultado de aplicar estas pruebas es exitoso el sistema estará preparado para alojarse en un servidor virtual de aplicaciones de la compañía.

6.2.2 Pruebas de Usabilidad

Este tipo de pruebas suponen un método de evaluación primordial en el análisis de un sitio web. Se enfocan en el diseño y en cómo los usuarios interactúan con el entorno, por eso este concepto de pruebas también es conocido como diseño de la experiencia de usuario porque gran parte de esta tarea tiene que ver con los aspectos visuales y de organización de contenidos.

Se puede considerar el tipo de evaluación más importante y una de las mejores herramientas para medir como los usuarios pueden llegar a interactuar de forma simple, cómoda y directa posible. Se puede ver como una de las pruebas más fieles de aproximarse a un uso real.

Por ello, la idea será contar con un número reducido de usuarios (4 ó 5 dependiendo de la disponibilidad) que realizarán una simulación de 20 minutos lo más ajustada a la realidad para comprobar la utilización de la aplicación. Será importante conocer previamente que conocimientos poseen y el tipo de formación que tienen. Se tratará de que las personas ejecuten tareas predefinidas como buscar una determinada franquicia o insertar datos de locales de la competencia, entre otras, con

el fin de comprobar cómo se desenvuelve y de cuan compleja puede ser para el usuario.

Otro de los aspectos necesarios para poder llevar a cabo esta prueba con éxito será realizar cuestionarios o formularios con preguntas al concluir la sesión de pruebas, en base a sus expectativas y experiencia: Saber qué opinan, qué puede aportarles o qué no les gusta permitirán adecuar los elementos con los que se esté trabajando en esos momentos, con el fin de atender a los usuarios y ofrecerles una experiencia a la altura de sus expectativas.

El resultado de esta prueba ha sido exitoso. El componente visual que acompaña a la aplicación hace que los usuarios comprendan la idea de manera fácil y vean en esta aplicación mucha capacidad de ampliación debido a su gran potencialidad.

6.2.3 Pruebas de Rendimiento

Este tipo de pruebas determinan la respuesta del sistema, la confianza y la escalabilidad bajo una carga elevada de trabajo.

Consiste por tanto en probar el software con la intención de determinar si cumple con los requerimientos predeterminados.

El tipo más sencillo de pruebas de rendimiento son las pruebas de carga. Para esta aplicación será necesario comprobar el comportamiento de los datos en la estructura árbol. Consiste en mostrar la información recuperada de un archivo JSON con más de dieciocho mil líneas de código.

El usuario aparte de expandir y contraer los datos del árbol tiene una opción que permite filtrar estos mismos datos en función de los parámetros de entrada introducidos en el campo de texto. Dependiendo de la conexión de cada usuario este proceso puede ser más pesado de lo normal, pero por norma general el proceso no suele tardar más que unos pocos segundos.

No obstante, se ha comprobado que cuando el usuario introduce un valor no válido y el árbol no se ha expandido ninguna vez, el uso de la CPU se eleva al 100% como podemos observar en la siguiente figura:

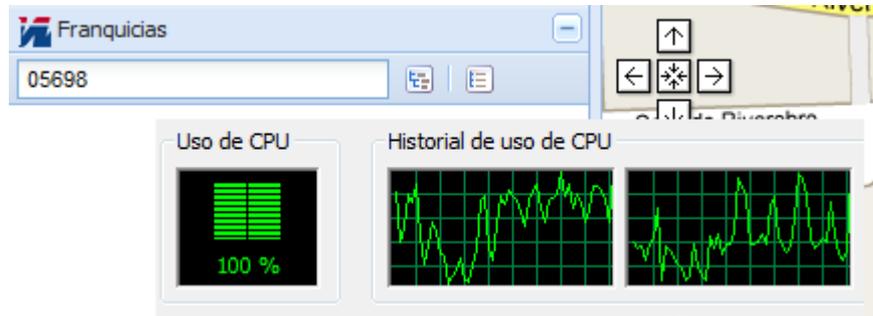


Fig. Uso de CPU al buscar información inexistente

También es importante tener claro si la aplicación tendrá un elevado volumen de carga continuada. En este caso, al tratarse de una primera fase demostración sólo estará online para un determinado número de usuarios por lo que medir su rendimiento en red a plena carga y en un plazo de corto de tiempo no será posible. Además ninguna de las aplicaciones creadas en la red de MRW tiene un acceso de diez mil personas diarias, que sería como si el total de trabajadores de la compañía se conectaran a la vez.

Además muchas de estas pruebas dependen íntegramente del funcionamiento del servicio de mapas de Google, pero el resultado también ha sido satisfactorio.

6.2.4 Pruebas de Compatibilidad

Muchas de las aplicaciones web que aparecen en Internet hoy en día son incompatibles debido a que no siguen el cumplimiento de estándares.

Siendo un portal que se desarrolle íntegramente en JavaScript la idea de poder ejecutarse desde cualquier navegador se antoja vital, ya que no se puede controlar desde que navegador será vista la aplicación.

Con el fin de intentar minimizar estas incompatibilidades, durante su desarrollo se procederá a un análisis sistemático que consistirá en seguir una serie de pasos para poder detectar de forma más rápida el error:

- Buscar errores tipográficos en el código, tales como comas, sintaxis mal definidas o escribir mal un método pueden hacer que existan problemas de este tipo. Además JavaScript no diferencia mayúsculas de minúsculas. Será el primer paso a comprobar.
- El paso siguiente consistirá en utilizar programas para la detección de errores o en el tiempo de ejecución, tales como Firebug o Companion JS, para Firefox o Internet Explorer respectivamente, ya que muchas veces existen scripts que no se ven o no se ejecutan correctamente con independencia del navegador.
- El último paso será solicitar ayuda en foros o páginas especializadas que puedan ayudar a resolver el problema o bien solicitar la experiencia de otros compañeros.



Fig. Detección de un error por medio de Companion JS para Internet Explorer

Capítulo 7. Conclusiones

La realización de esta aplicación demuestra cómo se plantea una necesidad en el marco real de una empresa y cuáles son las etapas a seguir para lograr unos objetivos determinados.

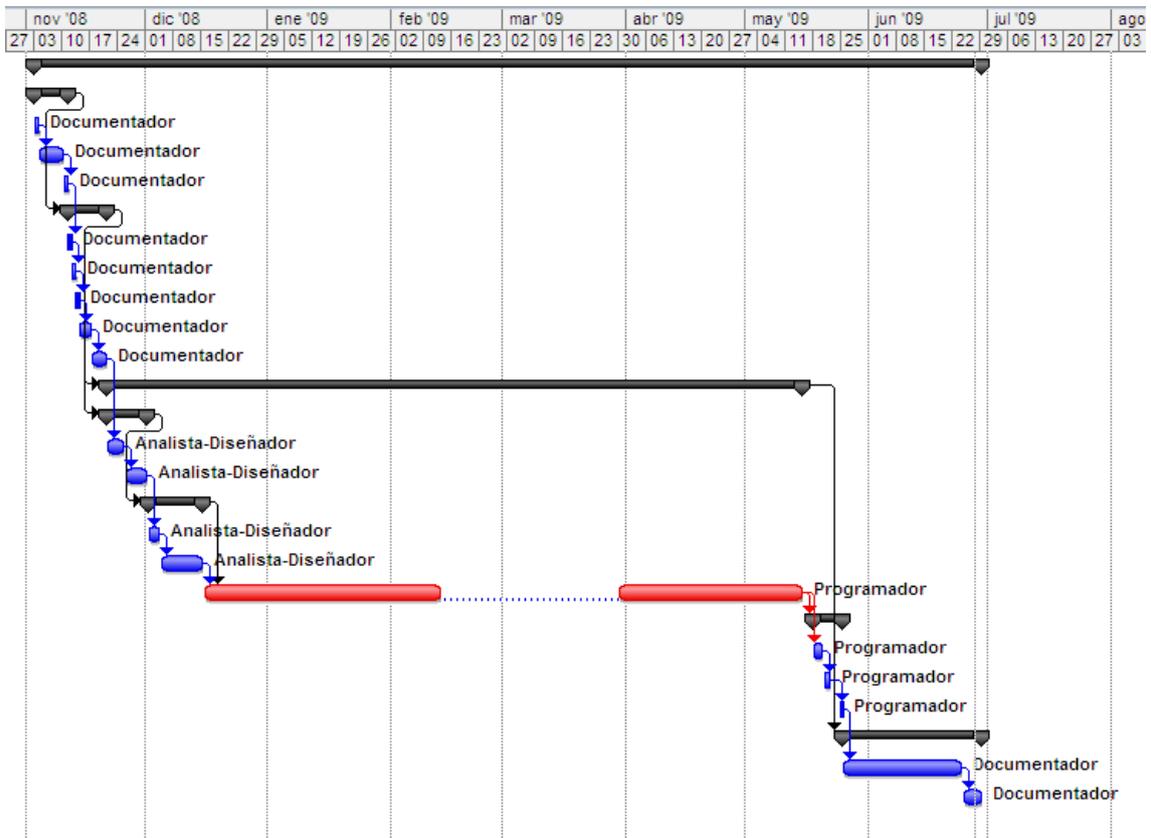
Inicialmente a nivel interno, disponer de una aplicación virtual de geoposicionamiento basado en las franquicias y su información, puede ofrecer un amplio abanico de posibilidades para MRW.

Sabiendo además que Google Maps es una tecnología en pleno desarrollo con muchas posibilidades de explotación, el trabajo realizado no es más que una primera fase de lo que realmente puede llegar a ofrecer al cliente, una herramienta que guarda un enorme potencial de creatividad, innovación y adaptabilidad.

Revisando las funcionalidades que se han descrito a lo largo de este estudio se concluye que todos los objetivos planteados al inicio del mismo se han alcanzado satisfactoriamente. El tratamiento y visualización de la información que ofrece la herramienta, permitirá apoyar la creación y difusión del conocimiento del negocio y su integración en nuevos procesos estratégicos.

Entre tanto, la planificación inicial del proyecto se ha visto alterada. En la fase de codificación el proyectista comenzó a desempañar nuevas funciones en la empresa por lo que las prioridades de trabajo cambiaron. El desarrollo se tuvo que hacer paralelamente a su nueva jornada laboral, como se puede observar en la siguiente figura. Las horas dedicadas diariamente no se vieron modificadas pero sí en la progresión continuación de la codificación.

Se observa además que la fecha de finalización se ha alargado hasta el día 29 de Junio, habiendo una diferencia de casi 59 días desde la planificación inicial.



Con todo, la duración del proyecto ha pasado a ser de 168 días, casi 6 meses, logrando de igual manera cumplir con los objetivos marcados al inicio del desarrollo tanto los de la empresa como los personales.

En conclusión, se puede considerar que la realización un proyecto ofrece la posibilidad al proyectista de aprender nuevos conceptos y afrontar nuevos retos que ayudan a alcanzar unos determinados objetivos, por lo que lo convierten en una experiencia totalmente enriquecedora.

Capítulo 8. Futuras líneas de trabajo

La idea inicial planteada a lo largo de este estudio, no ha hecho nada más que comenzar. Estas ideas plasmadas en código supondrán la base para el desarrollo de una aplicación web mucho más extensa que la actual, ya que a corto plazo el principal objetivo es disponer de un mapa actualizado de franquicias, clientes, posibles clientes, pero sobretodo tener geoposicionados todos aquellos puntos de competencia. En resumen, una aplicación retro-alimentada por las propias franquicias y empleados de la compañía, siempre supervisada por un integrante del Departamento de Tecnologías Internas.

Para ello, una de las primeras ampliaciones que se realizarán como se ha comentado en capítulos anteriores es dar acceso con privilegios de escritura y modificación a diferentes usuarios pilotos, es decir sería conveniente realizar una gestión de los usuarios que permita acceder a la aplicación introduciendo correctamente sus credenciales. No obstante se seguirá trabajando para adecuar el aspecto de la interfaz a posibles cambios y,

Otra de las mejoras que necesarias a realizar a corto plazo sería guardar todos los datos de las representaciones en una base de datos ya correctamente estructurada y adecuada para la aplicación, de esta manera si un usuario volviera a acceder a la aplicación, dispondría de todos los datos ya cargados.

Como se cita al inicio de este estudio, MRW se extiende por diferentes países por lo que promover el uso de este tipo de tecnologías permitiría globalizar la marca todavía más, por lo que podría ser una buena opción.

Existen otras posibles ampliaciones, no menos interesantes, que conllevan cambios importantes en el modelo de negocio, por lo que su implantación queda lejos, pero ayudarían a obtener ventajas competitivas importantes:

El uso de terminales PDA que registran la posición exacta de entrega de envíos, es cuestión de tiempo, por lo que proveer al sistema de las coordenadas exactas y que todos los procesos automáticos que se generan a raíz de esa entrega se muestren al instante en la aplicación, ayudarían a una franquicia a saber si el envío se ha entregado en el lugar correcto indicado en origen por la persona remitente.

Siguiendo con el tema de los terminales o dispositivos móviles, integrar toda la aplicación en un entorno de ejecución diferente como podría ser el móvil o una PDA, podría llegar a ser interesante y viable, ya que actualmente existen herramientas que facilitan la portabilidad de estos entornos a diferentes plataformas de ejecución. No obstante, primeramente sería necesario comprobar la necesidad o utilidad de aplicaciones como estas en dispositivos móviles. En este paso sería necesario además extender la aplicación web al exterior.

Otra de las funcionalidades que serían muy atractivas para su desarrollo, sería colorear el mapa de España por códigos postales. Sin ir más lejos, la superficie de EEUU ha sido coloreada de esta manera.

A continuación se muestra una captura:

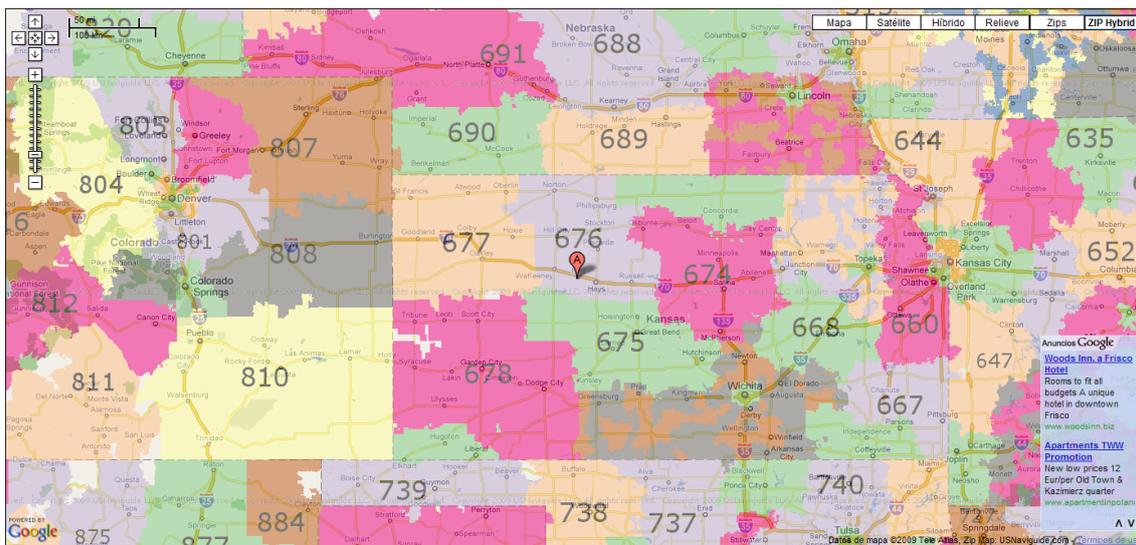


Fig. Coloración por CP del mapa de EEUU - <http://maps.huge.info/zip.htm>

En conclusión, Google Maps es una tecnología con muchas posibilidades de explotación. El API sigue estando en una fase beta debido a su constante evolución.

Por lo tanto, cualquier evolución de ésta que favorezca el crecimiento de esta aplicación será debidamente estudiada y analizada.

*Durante el proceso de desarrollo de esta aplicación han hecho aparición nuevas versiones del API de Google Maps y de la librería Ext JS, en su versión 3 y Ext JS 3.0 RC2, respectivamente. Estas nuevas versiones corrigen errores anteriores y por norma general, añaden nuevos componentes y funcionalidades, así como extensiones y mejoras de rendimiento, por lo que una de las futuras ampliaciones que se llevarán a cabo será actualizar ambas librerías.

Capítulo 9. Bibliografía

Publicaciones impresas:

- [1] Gibson, Rich; Erle, Schuyler. Google Maps Hacks. O'REILLY MEDIA, Inc. 2006
- [2] Frederick, Shea; Ramsay, Colin. Learning Ext JS. PACKT PUBLISHING LTD., 2008
- [3] C. Zakas, Nicholas. JavaScript para desarrolladores Web. EDICIONES ANAYA MULTIMEDIA / WROX, 2006

Recursos electrónicos:

- [4] <<http://code.google.com/apis/maps/>>, Documentación para el desarrollo de Google Maps en una web.
- [5] <<http://econym.org.uk/gmap/>>, Tutorial y ejemplos del API de Google Maps.
- [6] <<http://googlemapsmania.blogspot.com>>, Blog no oficial en inglés sobre mashups y herramientas basadas en Google Maps.
- [7] <<http://extjs.com/>>, Página oficial en inglés del framework de JavaScript. Disponibles ejemplos, foros de ayuda y documentación del API.
- [8] <<http://extjses.com/>>, Foro oficial en español del framework de JavaScript.
- [9] <<http://examples.extjs.eu/>>, Ejemplos de la utilización de la librería Ext JS.
- [10] <<http://www.quizpot.com/>>, Cursos y tutoriales gratuitos sobre Ext JS.
- [11] <<http://www.desarrolloweb.com/>>, Página de ayuda especializada en el desarrollo de webs.
- [12] <<http://www.wikipedia.org/>>, Información general.