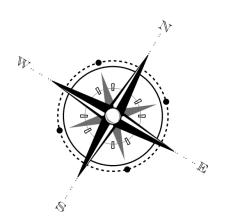




Proyecto Fin de Carrera

Desarrollo de una Aplicación Web para la Gestión de Rutas en Sistemas Georreferenciados



Saúl Rodríguez Rodríguez Las Palmas de Gran Canaria Septiembre de 2013

Universidad de Las Palmas de Gran Canaria Escuela de Ingeniería Informática Proyecto Fin de Carrera

Proyecto: Desarrollo de una Aplicación Web

para la Gestión de Rutas en

Sistemas Georreferenciados

Alumno: Saúl Rodríguez Rodríguez

Tutor: Javier Sánchez Pérez

"Un viaje es una nueva vida, con un nacimiento, un crecimiento y una muerte, que nos es ofrecida en el interior de la otra. Aprovechémoslo." Paul Morand

 $A\ mi\ familia$

Agradecimientos

Quisiera ofrecer una muestra de agradecimiento a todas las personas que me han permitido llevar a cabo este proyecto. También me gustaría destacar el papel de mi tutor, Javier Sánchez Pérez, autor de la idea que ha generado este proyecto y de haberme guiado en la realización del mismo, junto con el de los profesores y compañeros de la Facultad de Informática de la Universidad de Las Palmas de Gran Canaria que me han ayudado a ser capaz de llegar a este momento.

Por último, agradezco el apoyo recibido por parte de mis padres y hermano durante los años de carrera.

Índice general

Prefacio				XVII	
1.	Intr	oducci	ión	1	
	1.1.	Motiva	ación y Objetivos	. 2	
		1.1.1.	Ámbito	. 2	
		1.1.2.	Objetivos	. 3	
	1.2.	Estruc	etura del Documento	. 4	
2.	Esta	ado ac	tual del arte	5	
	2.1.	Georre	eferenciación	. 5	
		2.1.1.	Características y utilidades	. 5	
		2.1.2.	Métodos	. 6	
	2.2.	Sistem	nas de información geográfica	. 7	
		2.2.1.	Sistemas propietarios	. 7	
			2.2.1.1. ArcGIS	. 7	
			2.2.1.2. Bing Maps	. 8	
			2.2.1.3. Google Earth	. 9	
			2.2.1.4. Google Maps	. 12	
			2.2.1.5. SkylineGlobe	. 13	
		2.2.2.	Sistemas de código abierto	. 14	
			2.2.2.1. Capaware	. 14	
			2.2.2.2. GeoPista	. 15	
			2.2.2.3. Grass GIS	. 16	
			2.2.2.4. GvSIG y SEXTANTE	. 17	
			2.2.2.5. Marble	. 19	
			2.2.2.6. NASA World Wind	. 20	
			2.2.2.7. OpenJUMP	. 21	
			2.2.2.8. Quantum GIS	. 22	
	2.3.	Aplica	ciones de gestión de rutas	. 23	
		2.3.1.	CompeGPS Land	. 24	
		2.3.2.	Global Mapper	. 25	
		2.3.3.	Google Earth Pro		
		2.3.4.	OziExplorer3D	. 26	

XIV ÍNDICE GENERAL

3. Recursos necesarios					
	3.1.	Recurs	sos software	29	
		3.1.1.	Edición y documentación	30	
		3.1.2.	Lenguajes empleados para el desarrollo	31	
	3.2.	Recurs	sos hardware	32	
4.	Plai	nificaci	ón del proyecto	35	
	4.1.	Metod	ología de desarrollo	35	
		4.1.1.		36	
		4.1.2.		36	
		4.1.3.	Iterativo e incremental	37	
		4.1.4.	Fases del Proceso Unificado de Desarrollo	38	
			4.1.4.1. Fase de Inicio	39	
			4.1.4.2. Fase de Elaboración	40	
			4.1.4.3. Fase de Construcción	41	
			4.1.4.4. Fase de Transición	42	
	4.2.	Planifi	cación y temporización	42	
	4.3.		ouesto		
		4.3.1.	Costes de personal	47	
		4.3.2.	Costes inventariables	48	
		4.3.3.	Costes fungibles	48	
		4.3.4.	Costes indirectos	49	
		4.3.5.	Total del presupuesto	49	
5.	Des	arrollo	del proyecto	51	
				51	
		•	Modelo del dominio	51	
			5.1.1.1. Introducción	51	
			5.1.1.2. Desarrollo de un modelo del dominio	52	
				53	
		5.1.2.	Enumeración de requisitos candidatos	60	
				60	
			5.1.2.2. Leyenda	60	
			· · · · · · · · · · · · · · · · · · ·	61	
			5.1.2.4. Rasgos ligados a los Usuarios	69	
				70	
			-	71	
				74	
			•	79	
	5.2.	Requis		79	
		5.2.1.		80	
		5.2.2.		81	
			5.2.2.1. Gestión de Usuarios		

ÍNDICE GENERAL XV

			5.2.2.2.	Gestión de Elementos del Globo	3
			5.2.2.3.	Web 2.0)
		5.2.3.	Especific	cación de casos de uso)
			5.2.3.1.	Gestión de Usuarios)
			5.2.3.2.	Gestión de Elementos del Globo	3
			5.2.3.3.	Web 2.0	L
		5.2.4.	Interfaz	de usuario	3
			5.2.4.1.	Construcción de un Prototipo para la Interfaz de Usuario 143	}
			5.2.4.2.	Prototipo de la interfaz de usuario	Į
	5.3.	Model	o de análi	isis	2
		5.3.1.	Organiza	ación del sistema	}
		5.3.2.	Clases d	el análisis	Į
		5.3.3.	Realizac	iones de casos de uso	Į
			5.3.3.1.	Subsistema de Usuarios	Į
			5.3.3.2.	Subsistema de Elementos del Globo)
			5.3.3.3.	Subsistema Web 2.0)
	5.4.	Model	o de diser	ño)
		5.4.1.	Arquited	etura del sistema	}
		5.4.2.	Arquited	etura del software	Ĺ
		5.4.3.	Estereot	ipos utilizados	í
		5.4.4.	Especific	cación de casos de uso)
			5.4.4.1.	Subsistema de Usuarios)
			5.4.4.2.	Subsistema de Elementos del Globo	7
			5.4.4.3.	Subsistema Web 2.0	3
	5.5.	Implen	nentación	227	7
		5.5.1.	Captura	s de pantalla)
			5.5.1.1.	Pantalla Inicial)
			5.5.1.2.	Registro de usuarios	L
			5.5.1.3.	Pantalla Principal	L
c	Con	aluai au		abajo futuro 233	•
υ.				abajo futuro 233	
	0.2.	mabaj	o iuturo		E
Α.	Fran	neworl	ks de de	sarrollo 237	7
	A.1.	CakeP	HP		7
	A.2.	CodeIg	gniter		3
	A.3.	Prado)
	A.4.	Symfor	ny		_
	A.5.	Yii			}
	A.6.	Zend .			Ę
	A.7.	Listade	o de fram	eworks PHP	5

XVI	ÍNDICE GENERAI

B. Glosario	247
Índice de figuras	254
Índice de cuadros	255
Bibliografía	258

Prefacio

La georreferenciación es un campo que a día de hoy forma parte de nuestra vida diaria. Hemos pasado de usar mapas y globos terráqueos representando la superficie del planeta a representaciones virtuales en tres dimensiones del área que queremos visualizar con un grado de detalle nunca visto. Este progreso se ha completado en el transcurso de unas pocas décadas. El objetivo de la georreferenciación, así como de las herramientas que utilizan este concepto, es informar y visualizar el terreno de la mejor forma posible de cara al usuario, facilitándole la toma de decisiones para que éste pueda tomar la más acertada. Ello ha hecho que nos convirtamos en usuarios que hacen un gran uso de este tipo de sistemas. No sólo los usuarios individuales usan este tipo de sistemas; empresas, organizaciones y gobiernos también utilizan sistemas de información geográfica con el propósito de extraer patrones de distribución sobre el terreno. De esta forma, su toma de decisiones es más simple y proporcionará un mejor resultado en el futuro.

La evolución de los Sistemas de Información Geográfica (SIG) se ha debido a la mejora de las herramientas y programas asociados a éstos. Los últimos ahora son capaces de trabajar usando información geográfica de manera equivalente a la que emplean los usuarios: nombres de calles, lugares y similares. Sin embargo, también emplean la representación formal a través de coordenadas como latitud y longitud. Ambas representaciones se combinan usando diccionarios (denominados nomenclátores) que enlazan los nombres con sus coordenadas correspondientes.

Junto con los sistemas de información geográfica han aparecido una serie de estándares con el objetivo de facilitar la interoperabilidad entre los mismos. Estos estándares cubren varios aspectos propios de estos sistemas, desde la representación del terreno al almacenamiento de la información. Algunos de los estándares establecidos son abiertos, generalmente creados por el Open Geospatial Consortium (OGC), y otros propietarios, usualmente por parte de empresas. Algunos de los estándares actuales son:

- 1. Conversores de datos (DLG, MOSS, GIRAS)
- 2. Formatos de intercambio estándar (SDTS, DXF, GML)
- 3. Formatos de fichero abiertos (VPF, shapefiles, KML)
- 4. Interfaces de Programación de Aplicaciones (APIs) (ArcSDE API, CAD Reader, ArcS-DE CAD Client)
- 5. Características comunes en Sistemas de Gestión de Bases de Datos (SGBD) (OGC Simple Feature Specification for SQL)

XVIII PREFACIO

6. Integración de servicios web SIG estandarizados (WMS, WFS, ArcIMS)

Muchas de las aplicaciones que emplean técnicas SIG que se utilizan actualmente están disponibles vía web. Ello facilita su uso debido a que el usuario puede acceder desde diferentes puntos a la aplicación manteniendo su información centralizada y siempre disponible. Algunas de estas aplicaciones han incorporado aspectos de la denominada "Web 2.0" como la facilidad a la hora de compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración. En las aplicaciones Web 2.0 no sólo se ofrece un contenido estático dispuesto para que los usuarios lo visualicen, sino que éstos últimos participan del contenido existente, ya sea mejorándolo o ampliándolo colaborando y comunicándose entre sí. Para ello fue necesario que aparecieran tecnologías software que permitieran hacer todo esto posible como AJAX o XML.

El siguiente paso dentro de la evolución tecnológica que ha traído el uso de los sistemas SIG es el poder hacer uso de los mismos desde los nuevos teléfonos móviles o *smartphones*. La potencia de procesamiento que poseen estos dispositivos, junto con la capacidad de poder conectarse a Internet, ha dado cabida a la aparición de aplicaciones SIG en este nuevo nicho de mercado. Su principal cualidad, el poder acceder a un sistema SIG desde cualquier punto, ha supuesto una revolución dentro de este campo.

Varios de los conceptos indicados anteriormente se han tenido en cuenta a la hora de conceptuar, dar forma e implementar este proyecto.

El trabajo expuesto en este documento refleja la realización de un proyecto de Ingeniería de Software con el objetivo de desarrollar una aplicación web de gestión de rutas y que éstas se visualicen en un sistema georreferenciado como Google Earth. Un sistema georreferenciado permite obtener la posición de un objeto presente en un entorno de dos o tres dimensiones.

La aplicación tiene soporte de usuarios permitiendo que éstos puedan crear, almacenar y visualizar las rutas de manera cómoda y sencilla. Dichas rutas pueden incorporar otros elementos multimedia, como fotos y vídeos, que permiten una mejor comprensión de las mismas por parte del usuario añadiendo conocimiento.

Se ilustrará la metodología de desarrollo empleada, que en este caso es la del Proceso Unificado de Desarrollo de Software (PUD) y se presentarán los productos generados según indica la misma, utilizando técnicas y herramientas como UML entre otras. Es una metodología iterativa e incremental, dirigida por casos de uso, centrada en la arquitectura y enfocada en detectar los posibles riesgos dentro del ciclo de vida del proyecto.

La aplicación resultante hace uso de un framework en PHP denominado CakePHP encargado de operar con la base de datos MySQL, además de proporcionar otras utilidades que permiten un desarrollo de la aplicación más rápido y fácil. Para la interfaz de usuario se ha empleado la librería MochaUI que permite construir aplicaciones web con apariencia de escritorio de forma sencilla. También se ha utilizado JavaScript y JQuery de cara a proporcionar al usuario una aplicación más accesible de utilizar.

Por último se especificarán las conclusiones obtenidas tras realizar este trabajo, tanto a la hora de hablar sobre las herramientas utilizadas como por las decisiones y tareas realizadas por mi parte.

Capítulo 1

Introducción

Un sistema georreferenciado es aquel que es capaz de recoger, manipular y mostrar información sobre la posición geográfica de un objeto. Este tipo de sistemas se creó con el propósito inicial de resolver problemas de planificación y gestión geográfica. El nivel de detalle que proporcionan ha ido evolucionando desde pequeñas zonas de terreno hasta el planeta entero a día de hoy. Actualmente, los sistemas más modernos de georreferenciación recogen toda la información a través de la digitalización de imágenes provenientes de satélites y fotografías aéreas por parte de programas de CAD (Diseño Asistido por Computador). La información obtenida puede visualizarse posteriormente usando programas informáticos que la muestran usando mapas virtuales en dos o tres dimensiones. Para representar el mapa se hace uso de diversos modelos matemáticos que emplean un sistema de referencia.

La aparición de los sistemas georreferenciados ha supuesto un impulso de gran magnitud para la sociedad actual, ya sea de manera individual o colectiva. Las múltiples ventajas que ofrecen estos sistemas ha permitido un progreso evidente a varios niveles en cuanto a comodidad, facilidad y utilidad para muchísimas personas y grupos. Ya sea desde poder observar zonas del planeta sin tener que estar allí físicamente, o poder conocer cómo llegar a un sitio gracias a las herramientas informáticas asociadas a estos sistemas; sirven como ejemplos de la potencialidad que han alcanzado estos sistemas. Por ejemplo, en el caso de Google, el 30 % de las búsquedas realizadas en su buscador contienen un componente geográfico. Este porcentaje se ve incrementado hasta el 40 % en las búsquedas realizadas desde dispositivos móviles, ya que la gente quiere saber lo que le rodea¹. Esto permite a Google redirigir la búsqueda hacia Google Maps, uno de sus sistemas de información geográfica.

Como se indicaba anteriormente, para que se produjese este auge repentino de los sistemas georreferenciados, éstos han debido de apoyarse en aplicaciones y programas software que han sabido explotar las características de estos sistemas, facilitando su uso a los usuarios generalistas. Antes estas herramientas sólo estaban al alcance de los usuarios especializados o profesionales, dependiendo del enfoque que se le hubiera dado a la aplicación que usase el sistema georreferenciado.

Las aplicaciones de georreferenciación más conocidas en la actualidad, de cara al usuario

¹http://www.scienceomega.com/article/1033/uncharted-territory-google-map-maker-reaches-uk-shores

general, son Google Maps y Google Earth. El primero es un sistema de dos dimensiones y el segundo de tres dimensiones. Un equivalente a Google Maps proviene de la mano de Microsoft con Bing Maps (aunque ya dispone también de versión en tres dimensiones). Las posibles alternativas para Google Earth serían Nasa World Wind o ArcGIS. Todos estas aplicaciones son propietarias, excepto la realizada por la NASA que es código abierto, y están pensadas para ser explotadas comercialmente promocionando servicios y negocios de terceros. Nasa World Wind está más enfocado a la investigación debido a su motor de simulación atmosférica y la posibilidad de visualizar otros planetas.

En general, estas aplicaciones no están cerradas a su función principal, sino que disponen de algún SDK o API que permite a los desarrolladores ampliar las características del sistema. Entre ellas están manipular el contenido que se muestra en el mapa u ofrecer nuevo contenido al usuario en función de sus gustos. Muchas aplicaciones del mercado actualmente están construidas siguiendo este esquema.

1.1. Motivación y Objetivos

En esta sección se describirá la razón de realizar este proyecto y se resumirán los objetivos que se pretenden alcanzar con la realización del mismo.

1.1.1. Ámbito

El marco de trabajo designado para el proyecto es la gestión de rutas independientemente de su tipología (terrestres, aéreas o marítimas) u otro tipo de características. El usuario de la aplicación desarrollada tras la finalización de este proyecto podrá crear rutas, modificarlas o eliminarlas. Para su visualización se optará por un entorno 3D que mostrará el globo terráqueo de manera realista.

Además, se tendrán en cuenta aspectos de la denominada Web 2.0 para el desarrollo de la aplicación del proyecto. La Web 2.0 nace con el objetivo de permitir que los usuarios de los sitios web tengan una mayor participación en los contenidos de los mismos. Para ello fue necesario una evolución en diversos aspectos tecnológicos, como nuevas interfaces de usuario y tecnologías que permiten mostrar el nuevo contenido creado de manera dinámica. La Web 2.0 ha permitido crear aplicaciones Web, redes sociales, servicios de alojamiento de vídeos, wikis, blogs o mashups entre otros ejemplos.

Por lo tanto, se desarrollará una aplicación Web 2.0 donde los usuarios podrán crear y visualizar sus rutas. Asimismo, dichas rutas podrán ser compartidas con otros usuarios a fin de que estos también dispongan de ellas, o que puedan verlas y comentarlas. Como se indicó previamente, la tipología de las rutas puede ser muy variada, como por ejemplo: las etapas de la Vuelta Ciclista a España, la travesía de una competición náutica, el recorrido que siguió Colón en su primer viaje a América o los caminos reales de Canarias.

Así, con el objetivo de poder registrar y gestionar todos estos ejemplos, además de facilitar su difusión, se requiere desarrollar un sistema informático donde la aplicación a implementar estará enmarcada dentro de la gestión de rutas. La aplicación poseerá un enfoque general, que será válido para todos los ejemplos mencionados previamente.

En este proyecto de fin de carrera se presentará de forma pormenorizada las diversas fases por las que debe pasar la construcción de una aplicación software haciendo uso de técnicas, herramientas y metodologías propias de la ingeniería del software.

1.1.2. Objetivos

Propósito principal:

 Realizar una aplicación web que permita al usuario de la misma gestionar rutas y visualizarlas en 3D haciendo uso de elementos de la Web 2.0 como la interoperabilidad y la facilidad de uso.

Objetivos principales del proyecto:

- 1. Gestión de rutas del usuario de manera eficiente. La aplicación permitirá al usuario crear rutas desde la propia representación del terreno, así como modificar y eliminar rutas de manera sencilla.
- 2. Gestión de sitios de interés del usuario de forma cómoda. El usuario podrá incorporar localizaciones de diversa índole a su biblioteca de elementos para tener una mejor información de sus lugares predilectos.
- 3. Visualización de elementos 3D durante la visualización de las rutas. Cuando el usuario seleccione mostrar una ruta creada previamente a la que le ha añadido un elemento 3D, éste se visualizará en el navegador.
- 4. Administración de usuarios. Permitir que múltiples usuarios puedan utilizar la aplicación al mismo tiempo, además de que los usuarios puedan modificar su información personal.

Objetivos secundarios:

- 1. Compartición de rutas y otros elementos entre los usuarios de la aplicación. Un usuario podrá enviar a otro un elemento del globo (ruta, sitio de interés, etc.) propio para que el segundo también disponga de él.
- 2. Comunicación entre sí de los usuarios de la aplicación. Los usuarios de la aplicación podrán crear grupos donde podrán intercambiar comentarios y opiniones sobre rutas u otros temas de interés.

Con el fin de cumplir los objetivos descritos previamente, se realizará en la medida que sea posible, una aplicación rigurosa de la metodología PUD. Además se realizará un estudio general del estado del arte con el fin de representarlo en la memoria del proyecto. Dicha memoria de proyecto se estructurará y documentará de manera conveniente para ilustrar el proyecto de manera fidedigna.

1.2. Estructura del Documento

A lo largo del contenido de este documento se describirá todo el proceso necesario para acometer el desarrollo de la aplicación resultante, desde las etapas iniciales de investigación y análisis de requisitos hasta las etapas finales de diseño e implementación. Para cada una de las etapas se especificarán las tareas involucradas y a continuación se desglosará el contenido correspondiente.

El documento está dividido en seis capítulos centrados en el trabajo realizado en el proyecto y un conjunto de apéndices encargados de la información más técnica y específica.

El **primer capítulo** está dedicado a dar una introducción al tema de los sistemas georreferenciados y las aplicaciones asociadas a los mismos. También se describirá el problema a resolver —la gestión de rutas— y se indicarán los objetivos que se buscan con la realización del proyecto.

El **segundo capítulo**, centrado en el estado actual del tema, hará un recorrido por los diferentes sistemas de información geográfica existentes a día de hoy, así como se describirán algunas de las aplicaciones más utilizadas en la gestión de rutas.

Posteriormente, en el **tercer capítulo** se detallarán los recursos empleados para llevar a cabo este proyecto. Se indicarán los recursos hardware y software empleados a lo largo de la vida del mismo.

El capítulo número cuatro, centrado en la planificación del proyecto, englobará el plan de trabajo y el presupuesto necesario para construir la aplicación. Se expondrá y se explicará la metodología de desarrollo empleada en este proyecto para posteriormente realizar una descomposición temporal de las actividades que componen la planificación. Por último, se hará una estimación económica del coste de desarrollar la aplicación, teniendo en cuenta los diversos factores que hay que aglutinar para obtener el coste total.

El capítulo cinco, el más largo de todos los que componen este documento, descompone todas las etapas del desarrollo del proyecto siguiendo la metodología descrita en el capítulo anterior. Dará comienzo con los requisitos del sistema, se seguirá con las fases de análisis y diseño y concluirá con la fase de implementación, que estará dedicado a comentar algunos aspectos concretos de la implementación adoptada para dar forma a la aplicación web construida. También se hará hincapié en algunos problemas que surgieron durante esta fase de desarrollo y cómo se resolvieron. En cada etapa se mostrarán los productos generados de acuerdo a la metodología.

Para finalizar, el **último capítulo** contendrá las conclusiones obtenidas tras finalizar este proyecto, ya sea en cuanto al trabajo realizado como a nivel personal. Además, se hará un inciso acerca de posibles líneas de actuación de cara a posibles ampliaciones en versiones posteriores de la aplicación.

El documento se cierra con los apéndices técnicos y con la bibliografía empleada en este proyecto a modo de ayuda para llevarlo a cabo.

Capítulo 2

Estado actual del arte

Dado el ámbito en el que se encuadra este proyecto, este capítulo debe enfocarse desde dos partes bien diferenciadas. Por un lado se hablará de la georreferenciación y de los Sistemas de Información Geográfica (SIG), y por el otro, de las aplicaciones de gestión de rutas existentes.

En el primer caso se definirá el concepto de georreferenciación y se hará un recorrido por los distintos programas y aplicaciones web que hacen uso de este concepto. Por último, en el segundo apartado se listará un pequeño abanico de las principales aplicaciones de gestión de rutas, realizando una breve sinopsis de cada una de ellas.

2.1. Georreferenciación

Por georreferenciación se entiende definir la existencia de algo en un espacio físico. Es decir, el establecimiento de su ubicación en términos de proyecciones cartográficas o sistemas de coordenadas. El término se utiliza tanto la hora de establecer la relación entre imágenes ráster o vectoriales y las coordenadas; o para determinar la localización espacial entre otras características geográficas. Como ejemplos se pueden indicar el establecimiento de la posición correcta de una fotografía aérea en un mapa, encontrar las coordenadas geográficas de una ciudad o la dirección de una calle. Este procedimiento es pues, imprescindible para el modelado de datos en el campo de los Sistemas de Información Geográfica (SIG) y otros métodos cartográficos. Cuando los datos provienen de diferentes fuentes deben ser combinados y deben de disponer de un sistema de referencia común para ser usados posteriormente en una aplicación SIG. Esto se logra mediante el uso de diversas técnicas de georreferenciación. La mayoría de las tareas de georreferenciación se llevan a cabo ya sea porque el usuario desea crear un nuevo mapa o porque quiere enlazar dos o más diferentes conjuntos de datos entre sí debido a que comparten a las mismas ubicaciones geográficas.

2.1.1. Características y utilidades

La georreferenciación es muy importante para la toma de imágenes aéreas y de satélite, por lo general en forma de imágenes ráster. Ello es muy útil para la cartografía, ya que explica cómo otros datos, como los anteriores puntos GPS, se refieren a las imágenes.

- Puede haber información muy valiosa contenida en los datos o imágenes que se produjeron en un momento de tiempo diferente. Puede ser necesario o deseable combinar o comparar estos datos con los disponibles en la actualidad. Estos últimos se pueden utilizar para analizar los cambios en las características de un estudio durante un periodo de tiempo.
- Mapas diferentes pueden utilizar sistemas de proyección diferentes. Las herramientas de georreferenciación contienen métodos para combinar y superponer estos mapas con una distorsión mínima.
- Utilizando métodos de georreferenciación, los datos obtenidos a partir de herramientas topográficas como estaciones totales pueden usarse como punto de referencia a partir de mapas topográficos ya disponibles.
- Puede ser necesario establecer la relación entre los resultados de una encuesta de población que han sido codificados con códigos postales o direcciones de calles y otras áreas geográficas, como las zonas censales u otras áreas utilizadas en la administración pública o la planificación de servicios.
- Puede ser necesario establecer una relación entre zonas con códigos postales o direcciones de calles que no concuerden con otras áreas geográficas como zonas censales u otras áreas usadas en la administración pública o planificación de servicios.

2.1.2. Métodos

Hay varias herramientas SIG disponibles que pueden transformar los datos provenientes de las imágenes a algún framework de control geográfico, como el comercial ArcMap¹, PCI Geomática², o ERDAS IMAGINE³. Pueden georreferenciarse un conjunto de puntos, líneas, polígonos, imágenes o estructuras 3D. Por ejemplo, un dispositivo GPS puede registrar las coordenadas de latitud y longitud de un punto de interés determinado, georreferenciándolo de manera efectiva. Una georreferencia debe ser un identificador único. En otras palabras, debe haber solamente una ubicación para cada georreferencia.

Las imágenes pueden codificarse utilizando formatos especiales de archivo para SIG o ir acompañadas de un archivo mundial.

Para georreferenciar una imagen, primero hay que establecer puntos de control, indicar las coordenadas geográficas conocidas de estos puntos de control, elegir el sistema de coordenadas y otros parámetros de proyección y minimizar los residuos. Los residuos son la diferencia entre las coordenadas reales de los puntos de control y las coordenadas predichas por el modelo geográfico creado usando los puntos de control. Proporcionan un método para determinar el nivel de precisión del proceso de georreferenciación.

En situaciones en las que los datos se han recogido y asignado a códigos postales o de área, por lo general es necesario convertir éstos a coordenadas geográficas mediante el uso de

¹http://www.esri.com/software/arcgis

²http://www.pcigeomatics.com/products/geomatica2013

 $^{^3}$ http://geospatial.intergraph.com/products/ERDASIMAGINE/ERDASIMAGINE/Details.aspx

un nomenclátor, un catálogo de nombres geográficos. Estos catálogos se crean normalmente a través de agencias censales, organizaciones cartográficas nacionales (una de ellas en España es el IGN⁴) o proveedores de servicios postales. En su forma más simple, un catálogo de este tipo puede estar formado simplemente por una lista de códigos de área o nombres de lugares y otra lista de los códigos correspondientes, nombres o localizaciones en coordenadas. Evidentemente, el alcance y la finalidad de los códigos disponibles son específicas de cada país. Los ficheros con estructura de nomenclátor soportan muchos sistemas de mapeo vía web donde a través de un entorno gráfico y de un cierto conjunto de elementos permiten al usuario mostrar un lugar en el mapa o descubrir rutas usando entradas como códigos postales, direcciones o nombres de lugares.

2.2. Sistemas de información geográfica

Un Sistema de Información Geográfica (SIG) es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión geográfica.

En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

La tecnología de los Sistemas de Información Geográfica puede ser utilizada para una gran variedad de necesidades.

A continuación se presenta y describe una pequeña comparativa de diversos sistemas SIG, divididos según el tipo de licencia que empleen, así como algunas aplicaciones que hacen un importante uso de estos sistemas en su funcionamiento habitual.

2.2.1. Sistemas propietarios

En esta sección se muestran algunos de los sistemas de información geográfica propietarios más empleados. Que sean propietarios no implica que algunos sean gratis, simplemente hay que aceptar una licencia propietaria para hacer uso de la aplicación.

2.2.1.1. ArcGIS

ArcGIS⁵ es el nombre de un conjunto de productos de software en el campo de los Sistemas de Información Geográfica o SIG. Producido y comercializado por ESRI, bajo el nombre genérico ArcGIS se agrupan varias aplicaciones para la captura, edición, análisis, tratamiento, diseño, publicación e impresión de información geográfica. Estas aplicaciones se engloban en familias temáticas como ArcGIS Server, para la publicación y gestión web, o ArcGIS Móvil para la captura y gestión de información en campo.

⁴http://www.ign.es/ign/main/index.do

⁵http://www.arcgis.com/about/

ArcGIS Desktop, la familia de aplicaciones SIG de escritorio, es una de las más ampliamente utilizadas, incluyendo en sus últimas ediciones las herramientas ArcReader, ArcMap, ArcCatalog, ArcToolbox, ArcScene y ArcGlobe, además de diversas extensiones. ArcGIS for Desktop se distribuye comercialmente bajo tres niveles de licencias que son, en orden creciente de funcionalidades (y coste): ArcView, ArcEditor y ArcInfo.

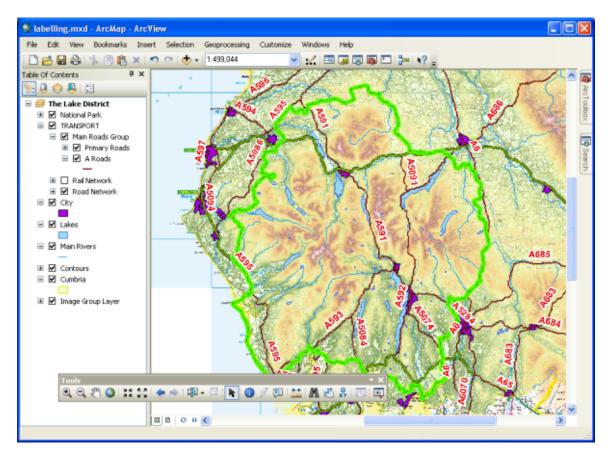


Figura 2.1: ArcGIS

2.2.1.2. Bing Maps

Bing Maps⁶ (anteriormente Live Search Maps, Windows Live Maps, Windows Live Local y MSN Virtual Earth) es un servicio de mapas para la web creado por Microsoft para su buscador Bing.

Bing Maps se basó en tecnologías existentes de Microsoft, como Microsoft MapPoint y TerraServer. La versión original no disponía de varias de las características que podemos distinguir ahora; incluyendo la Vista de Pájaro, mapas en 3D y las "colecciones" de puntos de interés. Su lanzamiento fue en diciembre de 2005, pasando a ser Windows Live Local la cara pública de la plataforma de Virtual Earth. El 6 de noviembre de 2006, Microsoft añadió la capacidad de ver los mapas en 3D mediante un control de .NET administrado usando Direct3D. Microsoft posteriormente oficialmente denomina este producto "Live Search maps", integrándolo como parte de su servicio Live Search. El 3 de junio de 2009, Microsoft renombra oficialmente a Live Search Maps como Bing Maps y a la plataforma de

⁶http://www.bing.com/maps/

Virtual Earth como Bing Maps for Enterprise.

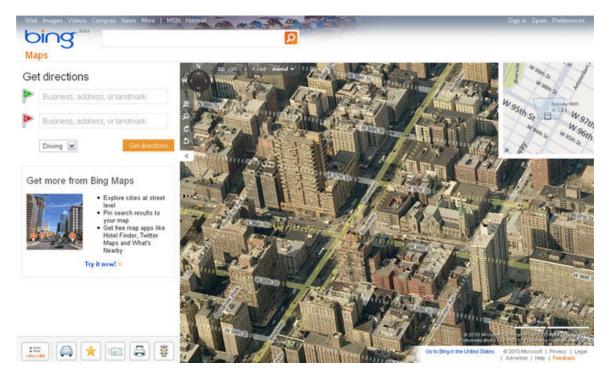


Figura 2.2: Bing Maps

2.2.1.3. Google Earth

Google Earth⁷ es un globo virtual, mapa y programa de información geográfica denominado originalmente EarthViewer 3D, creado originalmente por Keyhole, Inc., una empresa fundada en sus comienzos por la Agencia Central de Inteligencia (CIA), y luego adquirida por Google en 2004. Mapea la Tierra por la superposición de imágenes obtenidas a partir de imágenes de satélite, fotografías aéreas y datos SIG 3D. Estaba disponible bajo tres licencias diferentes, dos actualmente: Google Earth, una versión gratuita con funciones limitadas, Google Earth Plus (descatalogado), que incluía funciones adicionales, y Google Earth Pro, para uso comercial.

El producto, re-editado como Google Earth en junio de 2005 para el público general, está actualmente disponible para ordenadores que ejecuten Windows, Mac OS X, Linux y FreeBSD. Google Earth también está disponible como un plugin de navegador desde el 28 de mayo de 2008. También se puso a disposición de los usuarios de iPhone el 28 de octubre de 2008 como una descarga gratuita desde la App Store, y está disponible para los usuarios de Android como una aplicación gratuita en Google Play Store. Además de la liberación de un software cliente actualizado basado en el de Keyhole, Inc., Google también agregó imágenes de la base de datos de Earth a su software de mapas en Internet, Google Maps.

Para varias partes de la superficie de la Tierra hay disponibles imágenes 3D de terrenos y edificios. Google Earth utiliza un Modelo Digital del Terreno (MDT) a partir de datos recogidos por la misión topográfica Radar Shuttle (SRTM) de la NASA. Esto significa que se puede ver toda la Tierra en tres dimensiones. Desde noviembre de 2006 las vistas en 3D

⁷http://earth.google.com/

de muchas montañas, incluyendo el Monte Everest, se han mejorado mediante el uso de datos complementarios MDT para llenar los huecos dejados por la cobertura SRTM.

Muchas personas usan la aplicación para añadir sus propios datos, poniéndolos a disposición a través de diversas fuentes, como por ejemplo blogs. Google Earth es capaz de mostrar todo tipo de imágenes superpuestas en la superficie de la tierra y también funciona como cliente de un Servicio de Mapas Web. Google Earth también soporta la gestión de datos geoespaciales tridimensionales a través de Keyhole Markup Language (KML).

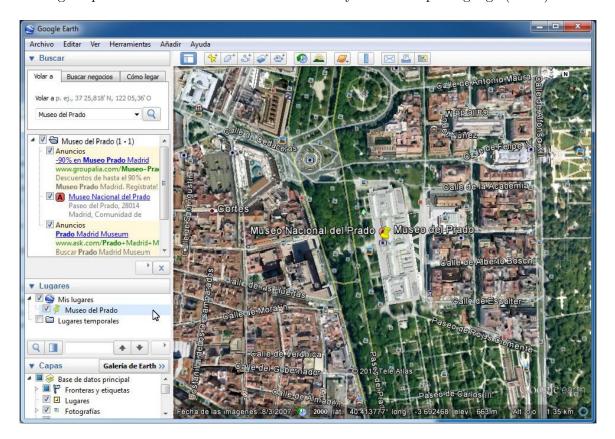


Figura 2.3: Google Earth

KML

KML (del acrónimo en inglés Keyhole Markup Language) es un lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones. Fue desarrollado para ser usado con Keyhole LT, precursor de Google Earth (Google adquirió Keyhole LT en octubre de 2004 tras lanzar su versión LT 2). Su gramática contiene muchas similitudes con la de GML (Geography Markup Language).

Los ficheros KML a menudo suelen distribuirse comprimidos como ficheros KMZ. Un fichero KML especifica una característica (un lugar, una imagen o un polígono) para Google Earth. Contiene título, una descripción básica del lugar, sus coordenadas (latitud y longitud) y alguna otra información. Un documento KML de ejemplo podría ser:

Algoritmo 2.1 Ejemplo de fichero KML

Collada

Collada⁸ (del inglés **Colla**borative **D**esign **A**ctivity) es un formato de fichero que permite la colaboración entre diferentes aplicaciones 3D interactivas. Es administrado por el Grupo Khronos, un consorcio tecnológico sin ánimo de lucro, y ha sido adoptado por la ISO como una especificación de disposición pública: ISO/PAS 17506 en 2012. COLLADA define un estándar abierto de esquema XML para el intercambio de recursos digitales entre varias aplicaciones software de gráficos que de otra manera podrían almacenar sus recursos en formatos de archivo incompatibles. Los documentos COLLADA que describen los recursos digitales son archivos XML, por lo general identificados con la extensión de fichero .Dae (**D**igital **A**sset **E**xchange). Un fichero Collada tiene una estructura similar a la siguiente:

⁸https://collada.org/

Algoritmo 2.2 Estructura de un fichero Collada

```
<?xml version="1.0" encoding="utf-8"?>
<COLLADA xmlns="http://www.collada.org/2005/11/COLLADASchema" version="</pre>
   1.4.1">
  \langle asset \rangle \dots \langle /asset \rangle
  <library_animations>.../library_animations>
  <library_physics_scenes>...</library_physics_scenes>
  <library_lights>...</library_lights>
  <library_images>...</library_images>
  <library_materials>...</library_materials>
 <library_effects>...</library_effects>
 <library_geometries>...</library_geometries>
  <library_controllers>...</library_controllers>
  <library_visual_scenes>...</library_visual_scenes>
  <scene> . . . </scene>
</COLLADA>
<source id="vertices_source" name="Vertices">
 <float_array id="values" count="6"> 0.3 0.5 0.7 0.2 0.4 0.6 </
     float_array>
    <technique_common>
      <accessor source="#values" count="2" stride="3">
        <param name="X" type="float"/>
        <param name="Y" type="float"/>
        <param name="Z" type="float"/>
      </accessor>
  </technique_common>
</source>
<mesh>
 <source id="position"/>
 <source id="normal"/>
  <source id="textureCoords"/>
  <vertices id="verts">
    <input semantic="POSITION" source="#position"/>
  </re>
  <triangles count="2" material="Bricks">
    <input semantic="VERTEX" source="#verts" offset="0"/>
    <input semantic="NORMAL" source="#normal" offset="1"/>
    <input semantic="TEXCOORD" source="#textureCoords" offset="2" set="1</pre>
    0 \ 0 \ 1 \ 3 \ 2 \ 1
      0 0 2 1 3 2
    </ triangles>
</mesh>
```

2.2.1.4. Google Maps

Google Maps⁹ es un servidor de aplicaciones de mapas en la web. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle con Google Street View.

⁹https://www.google.com/maps

Google Maps ofrece la posibilidad de que cualquier usuario poseedor de una página Web pueda integrarlo a la hora de hablar de cualquier tema donde exista una localización. Ello se consigue mediante la opción de Google Maps "enlace a esta página", se inserta una cadena larga de URL la cual contiene la latitud y la longitud del punto del mapa seleccionado. El usuario puede controlar el mapa para moverse a la ubicación que desee, introducir una dirección, una intersección o un área en general para buscar en el mapa.

Al igual que otros servicios de mapas, Google Maps permite la creación de pasos para llegar a alguna dirección. Esto permite al usuario crear una lista paso a paso para saber cómo llegar a su destino, calculando el tiempo necesario y la distancia recorrida entre las ubicaciones.

Tanto Google Maps como Google Earth ofrecen APIs para permitir a los desarrolladores realizar aplicaciones aprovechando la potencialidad de estas herramientas pudiendo ampliar sus funcionalidades.

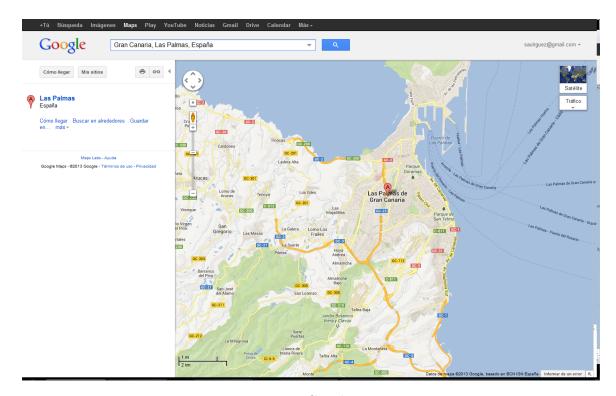


Figura 2.4: Google Maps

2.2.1.5. SkylineGlobe

La plataforma de software SkylineGlobe proporciona a los usuarios un acceso rápido a los datos geoespaciales 3D a través de *streaming*. Está implementado en ASP.NET y usa bases de datos Oracle o MS SQL Server. Posee una arquitectura abierta y una API que ofrece a los desarrolladores un conjunto de funcionalidades para utilizar en una amplia gama de aplicaciones web 3D.

El paquete web SkylineGlobe contiene la aplicación principal SkylineGlobe 3D y todas las herramientas asociadas, como Administrador de Capas, herramienta de dibujo y una herramienta de medición. El paquete web SkylineGlobe también incluye las herramientas

avanzadas disponibles para los usuarios de SkylineGlobe Pro.

Los usuarios del sitio web pueden navegar a través de un entorno intuitivo virtual donde ver, analizar y anotar los datos en su contexto geográfico. Dispone de herramientas potentes capaces de publicar en línea, promover la colaboración y el intercambio de datos. Entre otras capacidades se encuentran el soporte de capas, potentes herramientas de dibujo, herramientas de análisis avanzadas y funcionalidad de búsqueda.



Figura 2.5: SkylineGlobe

2.2.2. Sistemas de código abierto

A continuación se expone una pequeña selección de sistemas de información geográfica de código abierto. La diferencia con los sistemas propietarios es que estos se pueden utilizar y modificar de forma gratuita para cualquier fin.

2.2.2.1. Capaware

Capaware¹⁰ es una iniciativa del Gobierno de Canarias junto con la Universidad de Las Palmas de Gran Canaria (ULPGC) para su uso como visor 3D propio en temas de gestión de emergencias. Este proyecto ha sido liberado con el propósito de fomentar el desarrollo del software libre en Canarias. Capaware permite la interacción con terrenos virtuales 3D con precisión cartográfica y se distribuye bajo licencia GNU GPL. Permite acceder a información que cumpla las especificaciones del OGC.

Está desarrollado en lenguaje de programación C++, con lo que la suavidad en el movimiento es increíblemente realista mejorando a otras implementaciones en lenguajes de más alto nivel pero más "lentos". En la actualidad funciona en Microsoft Windows, aunque está planificada la capacidad de que sea multiplataforma (de momento hay una versión beta para Linux). Capaware utiliza OpenSceneGraph como motor gráfico, otra iniciativa de software libre logrando tasas de frames por segundo elevadas. Capaware posee además una

¹⁰http://www.capaware.org/

arquitectura de plugins que le permite crecer en funcionalidades a medida que se le añadan nuevos plugins.

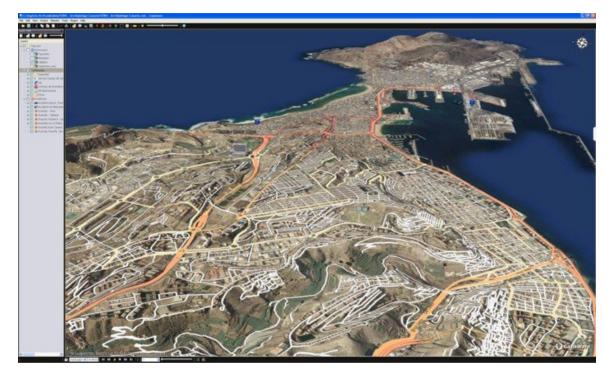


Figura 2.6: Capaware

2.2.2.2. GeoPista

GeoPista¹¹ es un Sistema de Información Territorial para entidades locales en España (diputaciones, mancomunidades, ayuntamientos, etc.) que facilita realizar la gestión municipal de forma georreferenciada y ofrecer servicios de información online a los ciudadanos utilizando la cartografía del municipio.

GeoPista es una iniciativa del Ministerio de Industria, Turismo y Comercio que cuenta con el respaldo de importantes organismos a nivel nacional, como la FEMP -Federación Española de Municipios y Provincias-, el Ministerio de Administraciones Públicas, Catastro, INE -Instituto Nacional de Estadística-, IGN -Instituto Geográfico Nacional-, etc.

Se basa en tecnologías SIG que permiten acceder y gestionar el alto volumen de datos asociado a la gestión municipal mediante una interfaz muy intuitiva: un mapa.

GeoPista es un sistema multiplataforma, Open Source, libre, escalable y que cumple con los estándares internacionales más relevantes relativos a la gestión de la información geográfica, como son la utilización de una base de datos compatible Simple Features, servidor de mapas compatible WMS, formato de intercambio GML, metadatos según la norma ISO 19115, directiva europea Inspire, etc.

GeoPista cubre las necesidades de las entidades locales de disponer de un software libre de gestión cartográfica que favorece la accesibilidad rápida y efectiva a la información a un coste menor, aumentando por lo tanto la eficiencia municipal, tanto en aspectos relativos a la gestión interna como de cara a los servicios que se van a poder ofrecer a los ciudadanos.

¹¹http://www.geopista.com/

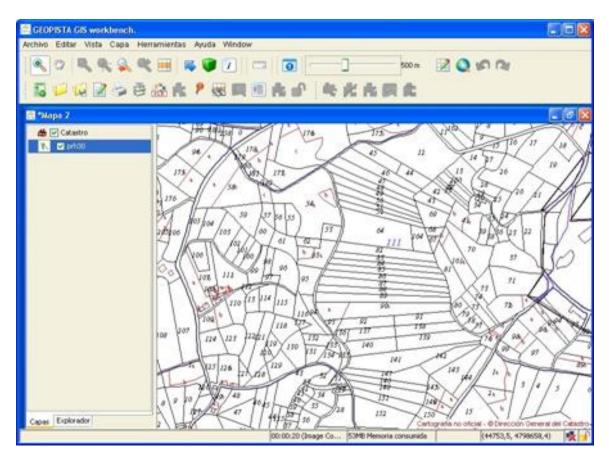


Figura 2.7: GeoPista

2.2.2.3. Grass GIS

GRASS¹² (acrónimo inglés de Geographic Resources Analysis Support System) es un software SIG (Sistema de Información Geográfica) bajo licencia GPL (software libre). Puede soportar información tanto ráster como vectorial y posee herramientas de procesado digital de imágenes.

En sus inicios, en 1982, el software fue desarrollado por el Cuerpo de Ingenieros del Laboratorio de Investigación de Ingeniería de la Construcción del Ejército de los Estados Unidos (USA-CERL) como herramienta para la supervisión y gestión medioambiental de los territorios bajo administración del Departamento de Defensa al no encontrar ningún SIG en el mercado que cubriera estas necesidades. En 1991 se pone a disposición pública a través de Internet. Su popularidad se incrementa en universidades, empresas y agencias gubernamentales. En 1997, ante el anuncio de USA-CERL de que dejaría de dar soporte al programa, la Universidad de Baylor se hace cargo de su desarrollo. A partir de esta fecha aumenta su aceptación dentro del mundo académico. El 26 de octubre de 1999 con la versión 5.0 se libera el código del programa bajo licencia GNU GPL. GRASS era uno de los primeros ocho proyectos de la Fundación OSGeo. En 2008 finalmente se trasladó toda la infraestructura de Grass bajo el mando de la fundación.

GRASS está disponible principalmente para plataformas *NIX (GNU/Linux), aunque existe un proyecto paralelo denominado winGRASS GIS que ha portado el programa a

¹²http://grass.itc.it/

versiones basadas en la tecnología NT del Sistema Operativo Microsoft Windows (Windows NT, Windows 2000, Windows XP, etc.) usando las librerías Cygwin. Todo ello con un código idéntico al de la versión UNIX y GNU/Linux.

La versión 6.x ha mejorado sensiblemente la experiencia del usuario respecto a la versión 5.x, ya que ofrece un entorno gráfico más amigable. Existen tutoriales y datos de ejemplo para la versión 6.x con los cuales es posible dar los primeros pasos con GRASS.

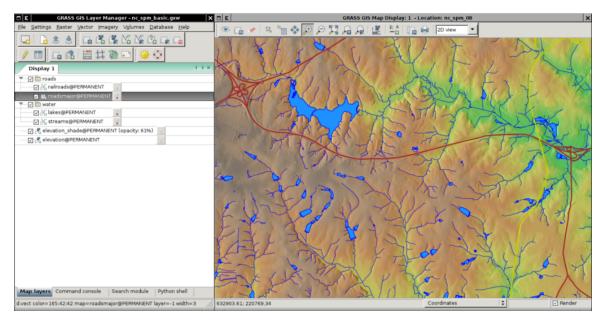


Figura 2.8: Grass SIG

2.2.2.4. GvSIG y SEXTANTE

gvSIG¹³ es un proyecto de desarrollo de Sistemas de Información Geográfica usando software libre, que incluye principalmente las aplicaciones gvSIG Desktop y gvSIG Mobile. La aplicación gvSIG Desktop fue la primera que se desarrolló dentro del proyecto gvSIG, por lo que también se conoce abreviadamente como gvSIG. Este proyecto fue desarrollado por el gobierno local de la Comunidad Valenciana (Generalidad Valenciana) de España, con el objetivo inicial de realizar la gestión de datos geográficos de esa colectividad; precisamente la sigla gvSIG abrevia la denominación Generalitat Valenciana Sistema de Información Geográfica.

gvSIG Desktop es un programa informático para el manejo de información geográfica con precisión cartográfica que se distribuye bajo licencia GNU GPL v2. Permite acceder a información vectorial y rasterizada así como a servidores de mapas que cumplan la especificaciones del OGC. Esta es una de las principales características de gvSIG respecto a otros Sistema de Información Geográfica, la importante implementación de servicios OGC: WMS (Web Map Service), WFS (Web Feature Service), WCS (Web Coverage Service), Servicio de Catálogo y Servicio de Nomenclátor.

Está desarrollado en el lenguaje de programación Java y funciona con los sistemas operativos Microsoft Windows, Linux y Mac OS X. Utiliza bibliotecas estándar de SIG reco-

¹³http://www.gvsig.org/

nocidas, como Geotools o Java Topology Suite (JTS). Asimismo, gvSIG posee un lenguaje de *scripting* basado en Jython y también se pueden crear extensiones en Java utilizando las clases de gvSIG.

Entre los formatos gráficos de fichero más habituales cuenta entre otros con acceso a formatos vectoriales GML, SHP, DXF, DWG, DGN, KML y formatos de imagen rasterizada como MrSID, GeoTIFF, ENVI o ECW.

Iniciado en el año 2004, es un proyecto de desarrollo informático impulsado inicialmente por la Conselleria de Infraestructuras y Transportes de la Generalidad Valenciana y la Unión Europea mediante el Fondo Europeo de Desarrollo Regional (FEDER). Actualmente está impulsado por un conjunto de entidades (empresas, administraciones, universidades) englobadas bajo la Asociación gvSIG.

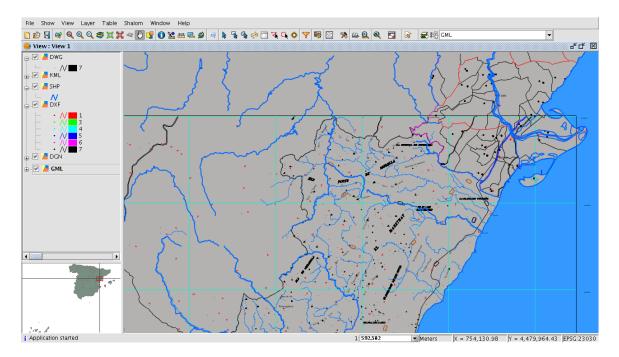


Figura 2.9: gvSIG

El Sistema EXTremeño de ANálisis TErritorial (SEXTANTE¹⁴) es una biblioteca de algoritmos de análisis espacial de código libre disponible para varios softwares de Sistemas de Información Geográfica. Su objetivo principal es crear una plataforma que facilite tanto el uso como la implementación de estos algoritmos.

Actualmente SEXTANTE contiene más de 240 herramientas de análisis geográfico.

En un principio SEXTANTE estaba basado en el SIG SAGA, para posteriormente pasarse a gvSIG. Inicialmente se centraba principalmente en el modelado y análisis de la información mediante imágenes ráster aunque en la actualidad son más de 240 extensiones tanto ráster como vectorial.

Tras el desarrollo de una gran colección de algoritmos de análisis geoespacial desarrollados para gvSIG, los desarrolladores entendieron que muchos otros proyectos requerían de análisis geoespacial, pero no existía una biblioteca que pudiera proporcionarles los algorit-

¹⁴http://www.sextantegis.com/

mos correspondientes. Ante este hecho, en el año 2008 tomaron la decisión de independizar SEXTANTE de cualquier software SIG, creando una biblioteca de tal manera que otros programas diferentes de procesamiento de información geográfica pudieran hacer uso de sus algoritmos de forma igual de sencilla que se venía haciendo hasta ahora.

A partir de la versión 0.6 SEXTANTE abandonó la licencia GNU GPL y pasó a utilizar una licencia MIT. La razón principal del cambio fue evitar posibles problemas de licencia de software que no permitiesen integrar SEXTANTE en otras aplicaciones.

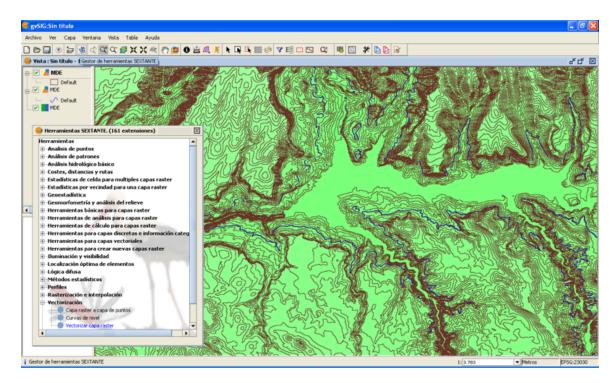


Figura 2.10: SEXTANTE

2.2.2.5. Marble

Marble¹⁵ es una aplicación geográfica a modo de globo virtual, desarrollada por KDE y licenciado bajo los términos de la licencia GNU LGPL 2, siendo software libre. Se puede ejecutar en cualquier PC que tenga un sistema operativo compatible con Qt 4, tales como Linux, Windows o Mac OS X, entre otros. La propuesta de Marble es ser flexible. Por ello puede funcionar sin necesidad de aceleración por hardware, comenzando rápidamente y viene ya con unos mínimos datos (5-10 MB) que lo hacen funcional sin necesidad de conexión a Internet. Se puede usar con OpenGL para dibujar los mapas y éstos se pueden obtener de fuentes online como OpenStreetMap. Marble puede asimismo emplear los archivos KML que actualmente usan Google Earth /Google Maps. Actualmente permite escoger entre mapas de la Tierra, Marte, Venus y mapas históricos.

¹⁵http://marble-globe.org/

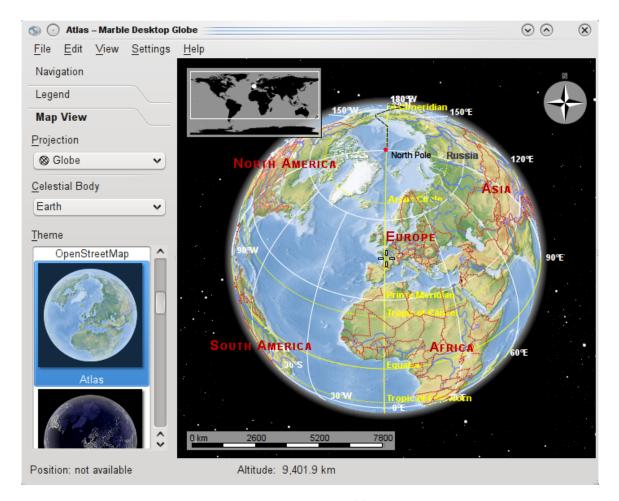


Figura 2.11: Marble

2.2.2.6. NASA World Wind

El NASA World Wind¹⁶ es un programa que actúa como un globo terráqueo virtual, o globo virtual desarrollado por la NASA para ser usado en ordenadores personales con Microsoft Windows, MacOS y Linux. Superpone imágenes de satélites de la NASA y fotografías aéreas del United States Geological Survey (USGS) sobre modelos tridimensionales de la Tierra, y en las últimas versiones, Marte y la Luna.

El usuario puede interactuar con el planeta seleccionado, rotándolo y ampliando zonas. Además se pueden superponer topónimos y fronteras, entre otros datos, a las imágenes. El programa también contiene un módulo para visualizar imágenes de otras fuentes en Internet que usen el protocolo del Open Geospatial Consortium Web Map Service. Adicionalmente, existen multitud de extensiones para World Wind que aumentan su funcionalidad, como por ejemplo, poder medir distancias u obtener datos de posición desde un GPS, gracias a que está desarrollado en Java y ha evolucionado hasta tener un SDK que permite ampliar su funcionalidad con trabajos de terceros escritos en otros lenguajes de programación.

¹⁶http://worldwind.arc.nasa.gov/

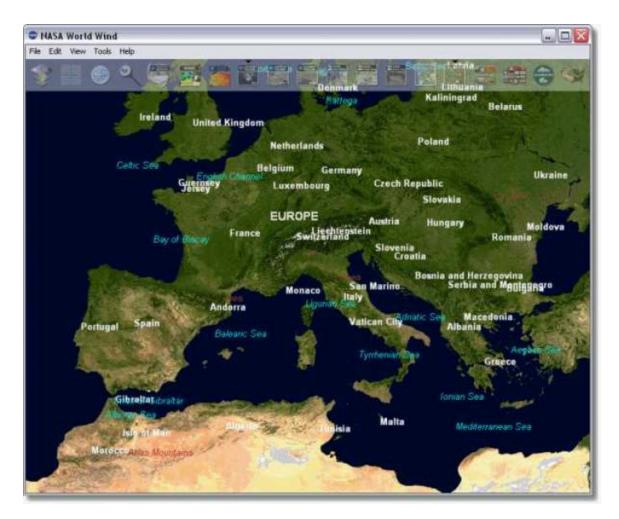


Figura 2.12: NASA World Wind

2.2.2.7. OpenJUMP

OpenJUMP¹⁷ es una aplicación SIG modular de código libre que permite la consulta y la creación o modificación de datos geográficos vectoriales almacenados bajo distintos formatos incluidos como GML, DXF o ESRI SHP. El programa permite también la explotación de servicios WMS. Inicialmente su nombre era JUMP.

Este Sistema de Información Geográfica está programado en Java y es multiplataforma. Su arquitectura modular facilita la creación de numeroso plugins que añaden funcionalidades específicas tales como: comprobación de topología, generación de Modelos Digitales del Terreno, lectura de formatos ráster, métodos de interpolación (*kriging*, triangulación de Delaunay, polígonos de Voronoi), *tracing*, creación de metadatos, etc.

JUMP fue desarrollada inicialmente en 2002 por la empresa Vivid Solutions a raíz de un concurso público convocado por el Ministerio de Recursos Naturales de la Columbia Británica (Canadá). Actualmente el desarrollo regular de este SIG por parte de la empresa que lo creó es discontinuo. Debido a ello y al constante crecimiento de la comunidad de usuarios en torno a JUMP surgieron diferentes grupos independientes de desarrolladores que han ido ampliando las capacidades de este Sistema de Información Geográfica lo que

¹⁷http://openjump.org/

hizo aparecer diferentes forks de la aplicación original.

En esta situación comenzaron a presentarse problemas de compatibilidad para la comunidad de usuarios entre los diferentes proyectos que se estaban desarrollando. Aunque la ayuda técnica todavía era facilitada por Vivid Solutions, no había un grupo unificado para coordinar el desarrollo y esfuerzos que aseguraran una continuidad de futuro de JUMP. Llegados a este punto diferentes usuarios y programadores, incluyendo a antiguos empleados de Vivid Solutions que desarrollaron la aplicación inicial, decidieron fundar JUMP Pilot Project (o JPP) con el objetivo de mejorar y coordinar los esfuerzos llevados a cabo en todo el mundo en torno a JUMP. Esta plataforma unificada, mantenida por un núcleo duro de administradores y supervisores, permitió eliminar incompatibilidades entre las diferentes ramas existentes. El nombre elegido para este nuevo SIG de código abierto fue OpenJUMP.

En la actualidad OpenJUMP está apostando por la interoperabilidad con otros Sistemas de Información Geográfica a través de uno de sus principales esfuerzos; impulsar el estándar de datos SIG GML. Una de las metas que se ha marcado el proyecto es hacer de OpenJUMP el primer programa capaz de trabajar con GML.

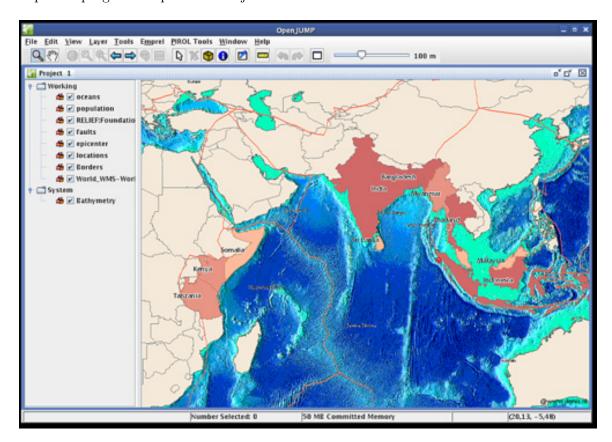


Figura 2.13: OpenJUMP

2.2.2.8. Quantum GIS

Quantum GIS¹⁸ (o QGIS) es un Sistema de Información Geográfica de código libre para plataformas GNU/Linux, Unix, Mac OS y Microsoft Windows. Era uno de los primeros ocho proyectos de la Fundación OSGeo y en 2009 vio la luz su primera versión. Permite

¹⁸ http://www.qgis.org/

manejar formatos ráster y vectoriales a través de las bibliotecas GDAL y OGR, así como bases de datos. Algunas de sus características son: soporte para la extensión espacial de PostgreSQL, PostGIS; manejo de archivos vectoriales como Shapefile, ArcInfo coverages, Mapinfo, GRASS GIS, etc. y soporte para un importante número de tipos de archivos ráster (GRASS GIS, GeoTIFF, TIFF, JPG, etc.).

Una de sus mayores ventajas es la posibilidad de usar Quantum GIS como GUI del SIG GRASS, utilizando toda la potencia de análisis de este último en un entorno de trabajo más amigable. QGIS está desarrollado en C++, usando la biblioteca Qt para su interfaz gráfica de usuario. Quantum GIS permite la integración de plugins desarrollados tanto en C++ como Python.

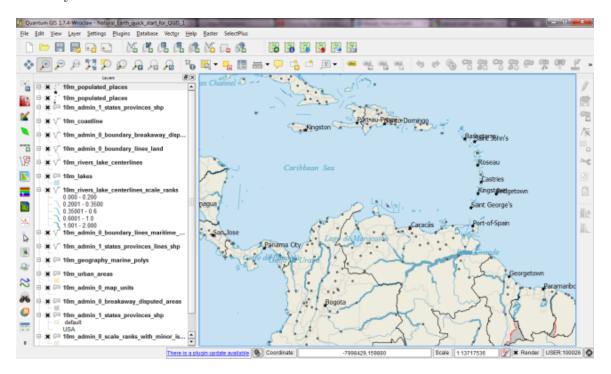


Figura 2.14: Quantum GIS

2.3. Aplicaciones de gestión de rutas

Actualmente hay una gran variedad de aplicaciones comerciales disponibles al gran público para la gestión de rutas, sobre todo dentro del ámbito de las rutas para propietarios de vehículos. En general, estas aplicaciones se basan en disponer de soporte para dispositivos GPS como los navegadores presentes en los coches o teléfonos móviles. Las aplicaciones de estas características se centran en otros aspectos que no son compartidos por la aplicación desarrollada en este proyecto y por lo tanto no se describirán aquí.

De entre las aplicaciones restantes que no se corresponden con lo indicado anteriormente resultan dos grupos: aquellas que visualizan los mapas en dos dimensiones o en tres dimensiones. Dado que la aplicación web construida utiliza un sistema de información geográfica en tres dimensiones, se optará por escoger a los programas del segundo grupo.

Dentro de las aplicaciones de gestión de rutas en tres dimensiones se han seleccionado aquellas que son de mayor interés de cara a ilustrar las posibilidades de las aplicaciones

de este tipo. Las aplicaciones seleccionadas son: CompeGPS Land, Global Mapper, Google Earth Pro y OziExplorer3D. En las siguientes secciones se hará una descripción de cada una de ellas.

2.3.1. CompeGPS Land

CompeGPS Land¹⁹ puede abrir varios tipos de mapas, ya sean topográficos, vectoriales, mapas escaneados o incluso fotografías de satélite. Puede pasar de uno a otro rápidamente, combinar la información de diferentes mapas mediante un avanzado sistema de transparencia o usar múltiples ventanas para mostrarlos por separado. También permite planificar rutas con pocos "clics", usando una interfaz muy intuitiva y totalmente personalizable.

Con CompeGPS Land el usuario puede crear recorridos, rutas y waypoints, modificarlos y añadir etapas para viajes de varios días. A su vez, podrá analizarlos usando información diversa (distancia, altura máxima, metros acumulados, pendiente máxima, etc.), permitiendo generar gráficos con un sistema incorporado para ello; además de poder guardar y clasificar los recorridos por actividad, distancia, pendiente, etc.

Los recorridos también pueden organizarse según diversos periodos de calendario. Por último, se puede imprimir un informe completo de la actividad del usuario.

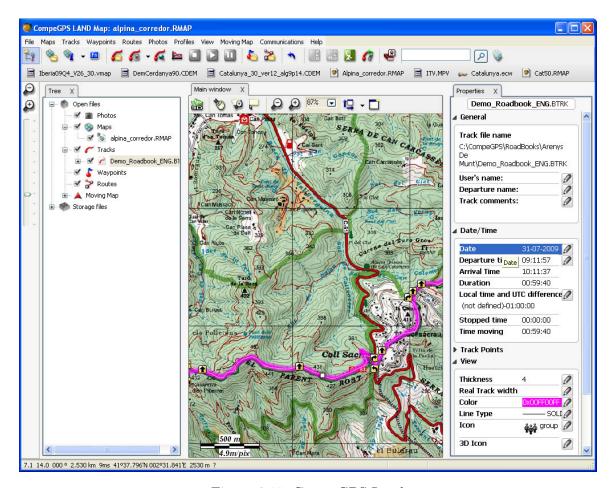


Figura 2.15: CompeGPS Land

¹⁹http://www.compegps.es/productos/software/land/

2.3.2. Global Mapper

Global Mapper²⁰ es una aplicación de procesamiento de datos SIG, capaz de obtenerlos desde múltiples fuentes incluyendo dispositivos GPS. Posee un conjunto de herramientas que permite el cálculo de distancias, áreas y volúmenes; fusión de capas y suavizado de bordes (técnicas traídas desde los programas de edición de imagen); análisis espectral y puede generar mapas de Contour y triangulación entre otras características.

Por el listado de utilidades indicado antes, se observa como el programa tiene un enfoque muy específico, el tratamiento de información geográfica, y no está totalmente centrado en la gestión de rutas. Aunque la posibilidad de gestionar rutas está presente en la aplicación, tanto la creación como el posterior trabajo de documentación de las mismas son muy básicos.

Posee un amplio soporte en cuanto a importación y exportación de una gran variedad de formatos de datos SIG, posiblemente sea la mejor aplicación SIG en este apartado.

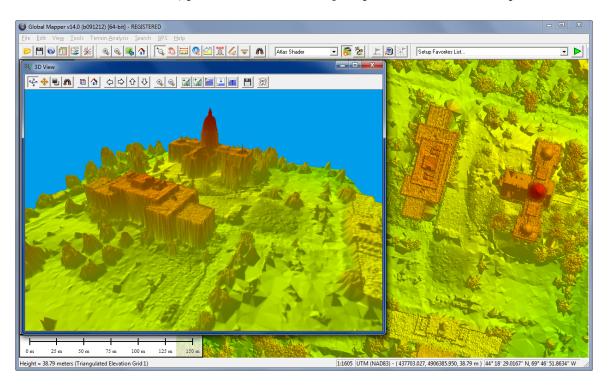


Figura 2.16: Global Mapper

2.3.3. Google Earth Pro

Google Earth Pro²¹ ofrece acceso a los datos geográficos de Google, lo que incluye imágenes de satélite, datos sobre calles, imágenes panorámicas de Street View, imágenes históricas y puntos de interés. Google Earth Pro parte de Google Earth y ofrece todas las características y funcionalidades de las que dispone éste. Además incorpora herramientas y funciones diseñadas específicamente para los usuarios empresariales, entre las que se incluyen: imprimir capturas de pantalla en alta resolución, crear mapas a partir de grandes conjuntos de datos SIG, crear mapas de direcciones rápidamente, crear películas personalizadas para ser

²⁰http://www.bluemarblegeo.com/products/global-mapper.php

²¹http://www.google.com/enterprise/mapsearth/products/earthpro.html

compartidas, medir áreas mediante polígonos o círculos, consultar datos demográficos, de parcelación y de tráfico usando varias capas.

La posibilidad de crear rutas se sustenta principalmente en las herramientas vistas en la versión básica que ya es bastante eficiente y cómoda de por sí, e incluye algunas posibilidades más (por ejemplo transferir las rutas a dispositivos GPS).

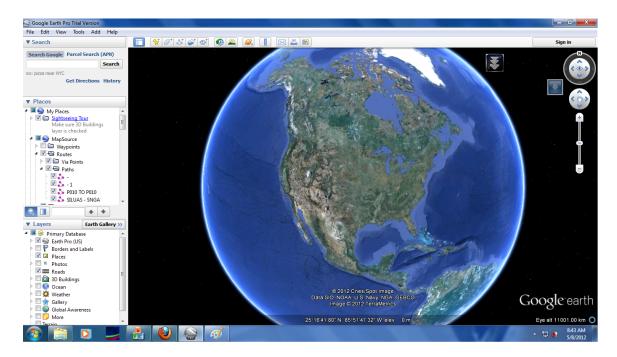


Figura 2.17: Google Earth Pro

2.3.4. OziExplorer3D

OziExplorer3D²²es un visor de mapas en tres dimensiones que representa los mapas construidos con la aplicación OziExplorer GPS Mapping Software. Este último trabaja con mapas en dos dimensiones, pero si se dispone de los datos referentes a la altura se pueden ver en 3D. Por lo tanto es un complemento que necesita del segundo para funcionar. Hablando específicamente del visor dispone de las siguientes funcionalidades: soporte para varios tipos de mapas, importación de datos desde dispositivos GPS, medición de áreas y distancias, visualización de varios mapas al mismo tiempo, inserción de comentarios y notas en puntos del mapa y poder imprimir los mapas.

Los mapas se puede rotar, variar la perspectiva y aumentar o disminuir el tamaño (nivel de zoom) cuando se visualizan.

El programa permite añadir rutas, waypoints y puntos al mapa 3D, y modificarlos de manera fácil a través de listas. La gestión de estos elementos por parte del programa es muy sencilla, simplificando la operatoria al usuario.

²²http://www.oziexplorer3.com/ozi3d/oziexplorer3d.html

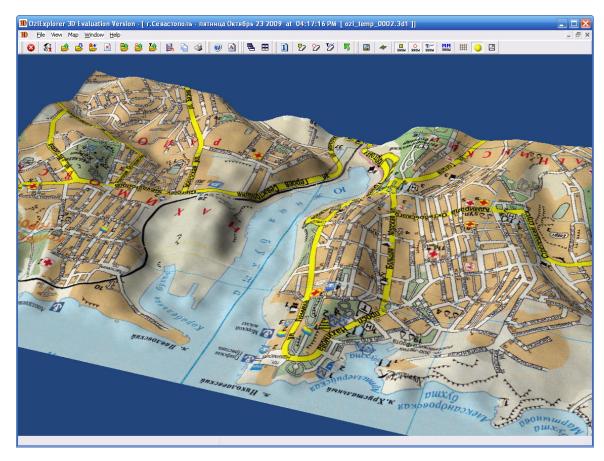


Figura 2.18: OziExplorer3D

Capítulo 3

Recursos necesarios

En este capítulo se indicarán los recursos hardware y software que han sido necesarios para la realización de este proyecto. Para cada uno de ellos se realizará una descripción somera donde se indicará el porqué de su inclusión en este trabajo, así como sus principales características.

3.1. Recursos software

Los principales recursos software utilizados para la consecución de este proyecto son:

- Microsoft Windows Ha sido el sistema operativo en el que se ha desarrollado el proyecto. Concretamente las versiones Vista y 7.
- **Apache** Es un servidor web HTTP de código abierto y multiplataforma que implementa el protocolo HTTP 1.1. Está bastante extendido en Internet debido a su soporte y a la cantidad de módulos disponibles que presenta.
- MySQL Es el SGBD (Sistema Gestor de Bases de Datos) encargado de otorgar la capa de persistencia a la aplicación. Su amplia difusión y su soporte para funciones SIG¹ siguiendo la especificación OpenGIS² lo hacen adecuado para este proyecto. También posee una estrecha colaboración con el lenguaje PHP facilitando el desarrollo de aplicaciones web.
- Mercury Es un servidor de correo compatible con los estándares actuales en cuanto a correo electrónico, tales como SMTP, POP3 e IMAP. Es altamente modular y no es excesivamente complejo de configurar.
- **StarUML** Es una herramienta CASE (Computer-aided Software Engineering) para UML (Unified Modeling Language). Se integra fácilmente con la metodología PUD permitiendo un desarrollo más ágil e integrado a lo largo de todas las fases del desarrollo del proyecto.

¹http://www.opengeospatial.org/standards/sfs

²http://www.opengeospatial.org/standards

- Complemento Google Earth El complemento de Google Earth permite al usuario visualizar y explorar datos geográficos sobre un globo terráqueo en 3D desde un navegador web. Proporciona herramientas que permiten ampliar su funcionalidad básica.
- CakePHP Es un framework para el desarrollo de aplicaciones web en PHP. Es código abierto y se basa fundamentalmente en el patrón de diseño MVC. Permite que las aplicaciones se puedan traducir de manera sencilla, proporciona acceso a bases de datos y proporciona caching, validación y autentificación entre otras características. Además posee una baja barrera de entrada para empezar a trabajar con él.
- **MochaUI** Es una biblioteca en JavaScript diseñada para crear interfaces de usuario en aplicaciones web basada en el framework de JavaScript MooTools. Proporciona herramientas gráficas como ventanas, paneles y listas para crear aplicaciones que tengan un "look & feel" parecido al de aplicaciones de escritorio.
- JQuery Es una biblioteca en JavaScript creada para simplificar la realización de ciertas operaciones en páginas web. Entre sus funcionalidades se encuentran: operar con el árbol DOM del documento web, crear animaciones, manejar eventos y facilitar el desarrollo de aplicaciones AJAX.
- Opera DragonFly Depurador de código JavaScript integrado dentro del propio navegador web Opera. Permite visualizar la estructura del documento web, monitorizar el tráfico de red; seleccionar un fichero JavaScript y poder ejecutarlo paso a paso, viendo los valores de las variables, entre otras características.

3.1.1. Edición y documentación

- Komodo Edit Es un editor de texto gratuito especializado en la escritura de código para diferentes lenguajes de programación. Entre ellos están HTML, CSS, JavaScript y PHP permitiendo el desarrollo de la aplicación desde una sola herramienta pudiendo trabajar en paralelo con varios lenguajes al mismo tiempo. Posee un explorador de ficheros y permite pestañas simplificando el poder trabajar con múltiples archivos al mismo tiempo. También permite autocompletado y la posibilidad de usar *snippets* de código.
- Notepad++ Es otro editor de texto con herramientas específicas para la escritura de código. Se usó como editor secundario para modificaciones pequeñas y puntuales. También la opción de poder buscar texto en varios ficheros al mismo tiempo fue de utilidad en ocasiones.
- LyX Es el procesador de textos utilizado para la redacción de esta memoria. Es un editor visual para el lenguaje de edición Latex. Éste hace que el usuario se centre en el contenido del documento, no en la apariencia final del mismo, tarea encargada para el sistema Latex, que maqueta el documento para dar el resultado final.

Producto Descripción Evolución Costes Estándares Adaptación Windows Sistema En Desarrollo < 100 € Media Operativo Apache Servidor Web En Desarrollo 0 HTTP Sencilla MvSQL En Desarrollo 0 SQLSistema Sencilla Gestor de Bases de Datos SMTP Mercury Servidor de En Desarrollo 0 / Media Correo 75-695 €) StarUML Herramienta Parado 0 UML Media CASE CakePHP En Desarrollo 0 PHP Sencilla Framework Web MySQL Herramienta En Desarrollo 0 SQLSencilla Workbench para crear Bases de Datos MochaUI Librería UI Parado 0 Media JavaScript Librería UI Media **JQuery** En Desarrollo 0 JavaScript Komodo Editor de En Desarrollo 0 HTML, Sencilla JavaScript Edit Código Web CSS Editor de En Desarrollo 0 Latex Media Lyx Latex

Cuadro 3.1: Herramientas Software

3.1.2. Lenguajes empleados para el desarrollo

HTML HyperText Markup Language (Lenguaje de Marcado de HiperTexto). Es un lenguaje de marcado utilizado para la realización de páginas web y que éstas se muestren en un navegador. Permite estructurar el contenido del documento web, el formateado de textos, así como incluir imágenes y otros tipos de objetos.

CSS Cascading Style Sheets (Hojas de Estilo en Cascada). Es un lenguaje creado para describir la semántica de la presentación del documento web. En la práctica se utiliza para dar formato a los elementos del documento web, dejando la estructura para el HTML o XHTML. En general, se trata de uno o varios ficheros que indican mediante una serie de reglas el formato de los diversos elementos que componen el documento web.

JavaScript Es un lenguaje interpretado que implementan los navegadores web y que permite realizar un cierto conjunto de operaciones en las páginas web desde el lado del cliente tales como: interacción con el usuario, control del navegador, comunicación asíncrona y posibilidad de modificar el documento web una vez mostrado. Se define por ser basado en prototipos, imperativo, débilmente tipado y dinámico.

PHP PHP Hypertext Pre-processor (PHP Preprocesador de Hipertexto). Es un lenguaje de programación de propósito general que trabaja en el lado del servidor. Fue diseñado para el desarrollo web de contenido dinámico. Se puede incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.

3.2. Recursos hardware

El proyecto se ha realizado con un ordenador portátil Dell XPS M1530 con las siguientes características:

- Procesador Intel Core 2 Duo T8100 a 2,1 Ghz.
- Memoria RAM 4 GB DDR2
- Tarjeta Gráfica NVIDIA GeForce 8600M GT
- Pantalla de 15,4 pulgadas
- Conexión a Internet vía Wi-Fi

Además se utilizó almacenamiento secundario (en forma de discos duros externos) para copias de seguridad, además de servicios a través de Internet que también permitían la realización de copias de respaldo para una mayor protección ante imprevistos.

Para poder ejecutar la aplicación realizada los requisitos necesarios son los siguientes:

- Hardware Cliente
 - Sistema operativo: Windows XP, Windows Vista o Windows 7 / Apple Mac OS X 10.5 ó superior (Intel)
 - CPU: Pentium 4, 2,4 GHz o versiones posteriores o AMD 2400 xp o versiones posteriores
 - Memoria del sistema (RAM): 512 MB
 - Disco duro: 2 GB de espacio libre
 - Velocidad de red: 768 Kbps
 - Tarjeta gráfica: DirectX9 compatible con 3D con 256 MB de RAM de vídeo
 - Pantalla: 1.280 x 1.024 píxeles en color real de 32 bits
- Hardware Servidor
 - Procesador: Pentium IV o compatible
 - Memoria RAM: 1 Gb
 - Disco Duro: 2 Gb libres
 - Placa de Red: Ethernet compatible

- Otros: Tarjeta Gráfica, Monitor recomendable
- Configuración de la Red
 - $\bullet\,$ Red LAN que soporte TCP/IP (en general, Internet)

Capítulo 4

Planificación del proyecto

En este capítulo se describirá la metodología aplicada para la realización de este proyecto, se desglosarán las tareas a realizar y se elaborará una estimación temporal del desarrollo del mismo. También se procederá a hacer una estimación de los costes de mano de obra y material necesarios para realizar este trabajo.

4.1. Metodología de desarrollo

Para la realización del proyecto se ha optado por la metodología PUD (Proceso Unificado de Desarrollo de Software). El proceso de desarrollo de software puede definirse como el conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema software. Sin embargo, PUD no es una única metodología, es un marco genérico que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y diferentes tamaños de proyectos. El Proceso Unificado de Software está basado en componentes, haciendo que el software resultante esté formado por componentes software interconectados a través de interfaces bien definidas.

El Proceso Unificado de Desarrollo utiliza el Lenguaje Unificado de Modelado (UML) para definir y especificar las diversas partes de un sistema. Es un lenguaje con un fuerte bagaje debido a su amplio uso. De hecho, en esta metodología, el uso de UML está entroncado a lo largo de todo el proceso.

Esta metodología está encuadrada dentro de un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura, además de ser iterativo e incremental. La utilización de esta metodología permite realizar software de calidad cumpliendo con los objetivos propuestos.

El ciclo de vida del proceso unificado consta de cuatro fases:

- 1. Inicio
- 2. Elaboración
- 3. Construcción
- 4. Transición

Cada fase se subdivide en iteraciones. En cada iteración se desarrolla en secuencia un conjunto de disciplinas o flujos de trabajos. Las más importantes son: Requisitos, Análisis, Diseño, Codificación, y Prueba. Estas disciplinas se realizan para cada una de las cuatro fases

Cada ciclo constituye una versión del sistema.

4.1.1. Dirigido por casos de uso

Un sistema software ve la luz para dar servicio a sus usuarios. Por lo tanto, para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean.

El término usuario no sólo referencia a usuarios humanos sino también a otros sistemas, es decir, todo aquello que interactúe con el sistema que estamos desarrollando. A esta interacción la llamamos caso de uso. Un caso de uso es un grafo de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema. Una especificación funcional define lo que debe hacer el sistema y al añadir los casos de uso nos fuerza a pensar en términos para el usuario. Sin embargo, los casos de uso no son sólo una herramienta para especificar los requisitos de un sistema. También guían su diseño, implementación y prueba; esto es, guían el proceso de desarrollo. Basándose en el modelo de casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. Los desarrolladores revisan cada uno de los sucesivos modelos para que sean conformes al modelo de casos de uso. Los ingenieros de prueba testean la implementación para garantizar que los componentes del modelo de implementación implementan correctamente los casos de uso. De este modo, los casos de uso no sólo inician el proceso de desarrollo sino que le proporcionan un hilo conductor. Dirigido por casos de uso quiere decir que el proceso de desarrollo sigue un hilo, es decir, avanza a través de una serie de flujos de trabajo que parten de los caso de uso. Los casos de uso se especifican, diseñan, y los casos de uso finales son las fuentes a partir de la cual los ingenieros de prueba construyen sus casos de prueba.

Aunque es cierto que los casos de uso guían el proceso, no se desarrollan aisladamente. Se desarrollan a la vez que la arquitectura del sistema. Es decir, los casos de uso guían la arquitectura del sistema y éste influye en la selección de los casos de uso. Por lo cual, tanto la arquitectura del sistema como los casos de uso maduran según avanza el ciclo de desarrollo.

4.1.2. Centrado en la arquitectura

La arquitectura de un sistema software se describe mediante diferentes vistas del sistema en construcción. El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, como las que perciben los usuarios y los inversores, y se refleja en los casos de uso. Sin embargo, también se ve influida por muchos otros factores, como la plataforma en la que tiene que funcionar el software, los bloques de construcción reutilizables de que se dispone,

consideraciones de implantación, sistemas heredados, y requisitos no funcionales. La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado. Debido a que lo que es significativo depende en parte de una valoración, que a su vez, se adquiere con la experiencia, el valor de una arquitectura depende de las personas que se hayan responsabilizado de su creación. No obstante, el proceso ayuda al arquitecto a centrarse en los objetivos adecuados, como la comprensibilidad, la capacidad de adaptación al cambio y la reutilización.

Debe existir interacción entre los casos de uso y la arquitectura. Por un lado, los casos de uso deben encajar en la arquitectura cuando se lleva a cabo y por otro lado, la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, ahora y en el futuro. En realidad, tanto la arquitectura como los casos de uso deben evolucionar en paralelo. Por lo tanto, los arquitectos modelan el sistema para darle forma. Es esta forma, la arquitectura, la que debe diseñarse para permitir que el sistema evolucione, no sólo en su desarrollo inicial, sino también a lo largo de futuras generaciones. Este diseño se esboza a partir de la comprensión general de las operaciones básicas del sistema, es decir, se debe trabajar sobre los casos de uso claves del sistema. Estos casos de uso clave pueden suponer solamente entre el cinco y el diez por ciento de todos los casos de uso, pero son los significativos, los que constituyen las funciones fundamentales del sistema.

4.1.3. Iterativo e incremental

El desarrollo de un producto software comercial supone un gran esfuerzo que puede durar entre varios meses y hasta posiblemente un año o más. Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar controladas; esto es, deben seleccionarse y ejecutarse de una forma planificada. Es por esto por lo que son miniproyectos.

Los desarrolladores basan la selección de lo que se implementará en una iteración en dos factores. En primer lugar, la iteración trata de un grupo de casos de uso que juntos amplían la utilidad del producto desarrollado hasta ahora. En segundo lugar, la iteración trata los riesgos más importantes. Las iteraciones sucesivas se construyen sobre los artefactos de desarrollo tal como quedaron al final de la última iteración. Al ser miniproyectos, comienzan con los casos de uso y continúan a través del trabajo de desarrollo subsiguiente —análisis, diseño, implementación y prueba-, que termina convirtiendo en código ejecutable los casos de uso que se desarrollaban en la iteración. Por supuesto, un incremento no necesariamente es aditivo. Especialmente en las primeras fases del ciclo de vida, los desarrolladores pueden tener que reemplazar un diseño superficial por uno más detallado o sofisticado. En fases posteriores, los incrementos son típicamente aditivos. En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con sus objetivos, el desarrollo continúa con la siguiente iteración. Cuando una iteración no cumple sus objetivos,

los desarrolladores deben revisar sus decisiones previas y probar con un nuevo enfoque.

Para alcanzar el mayor grado de economía en el desarrollo, un equipo de proyecto intentará seleccionar sólo las iteraciones requeridas para lograr el objetivo del proyecto. Intentará secuenciar las iteraciones en un orden lógico. Un proyecto con éxito se ejecutará de una forma directa, sólo con pequeñas desviaciones del curso que los desarrolladores planificaron inicialmente. Por supuesto, en la medida en que se añadan iteraciones o se altere el orden de las mismas por problemas inesperados, el proceso de desarrollo consumirá más esfuerzo y tiempo. Uno de los objetivos de la reducción del riesgo es minimizar los problemas inesperados. Son muchos los beneficios de un proceso iterativo controlado:

- La iteración controlada reduce el coste del riesgo a los costes de un solo incremento. Si los desarrolladores tienen que repetir la iteración, las organización sólo pierde el esfuerzo mal empleado de la iteración, no el valor del producto entero.
- La iteración controlada reduce el riesgo de no sacar al mercado el proyecto en el calendario previsto. Mediante la identificación de riesgos en fases tempranas del desarrollo, el tiempo que se gasta en resolverlo se emplea al principio de la planificación, cuando la gente está menos presionada para cumplir los plazos. En el método tradicional, en el cual los problemas complicados se revelan por primera vez en la prueba del sistema, el tiempo necesario para resolverlo normalmente es mayor que el tiempo que queda en la planificación, y casi siempre obliga a retrasar la entrega.
- La iteración controlada acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan de manera más eficiente para obtener resultados claros a corto plazo, en lugar de tener un calendario largo, que se prolonga eternamente.
- La iteración controlada reconoce una realidad que a menudo se ignora —que las necesidades del usuario y sus correspondientes requisitos no puedan definirse completamente al principio. Típicamente, se refinan en iteraciones sucesivas. Esta forma de operar hace más fácil la adaptación a los requisitos cambiantes.

4.1.4. Fases del Proceso Unificado de Desarrollo

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Al final de cada uno de ellos se obtiene una versión final del producto, que no sólo satisface ciertos casos de uso, sino que está lista para ser entregada y puesta en producción. En caso de que fuese necesario publicar otra versión, deberían repetirse los mismos pasos a lo largo de otro ciclo.

Como se ha comentado en el apartado anterior, cada ciclo se compone de varias fases, y dentro de cada una de ellas, los directores o los desarrolladores pueden descomponer adicionalmente el trabajo en iteraciones, con sus incrementos resultantes. Cada fase termina con un hito, determinado por la disponibilidad de un conjunto de artefactos, modelos o documentos.

Las iteraciones de cada fase se desarrollan a través de las actividades de identificación de requisitos, análisis, diseño, implementación, pruebas e integración.

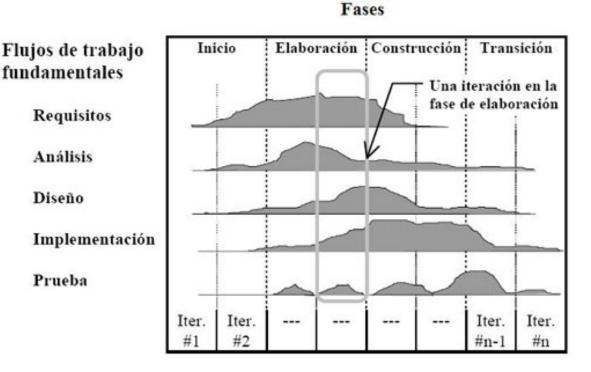


Figura 4.1: Fases PUD

4.1.4.1. Fase de Inicio

Durante la fase de inicio se desarrolla una descripción del producto final, y se presenta el análisis del negocio. Esta fase responde las siguientes preguntas:

- ¿Cuáles son las principales funciones del sistema para los usuarios más importantes?
- ¿Cómo podría ser la mejor arquitectura del sistema?
- ¿Cuál es el plan del proyecto y cuánto costará desarrollar el producto?

En esta fase se identifican y priorizan los riesgos más importantes. El objetivo de esta fase es ayudar al equipo de proyecto a decidir cuáles son los verdaderos objetivos del proyecto. Las iteraciones exploran diferentes soluciones posibles, y diferentes arquitecturas posibles. Puede que todo el trabajo físico realizado en esta fase sea descartado. Lo único que normalmente sobrevive a la fase de inicio es el incremento del conocimiento en el equipo.

Los artefactos que típicamente sobreviven a esta fase son:

- Un enunciado de los mayores requerimientos planteados generalmente como casos de uso.
- Un boceto inicial de la arquitectura.
- Una descripción de los objetivos del proyecto.

- Una versión muy preliminar del plan del proyecto.
- Un modelo del negocio.

La fase de inicio finaliza con el Hito de Objetivos del Ciclo de Vida. Este hito es alcanzado cuando el equipo de proyectos y las partes interesadas llegan a un acuerdo sobre:

- Cuál es el conjunto de necesidades del negocio, y qué conjunto de funciones satisfacen estas necesidades.
- Una planificación preliminar de iteraciones.
- Una arquitectura preliminar.

Debe poder responderse las siguientes cuestiones:

- ¿Se ha determinado con claridad el ámbito del sistema? ¿Se ha determinado lo que va a estar dentro del sistema y fuera de el sistema?
- ¿Se ha llegado a un acuerdo con todas las personas involucradas (las partes interesadas) sobre los requisitos funcionales del sistema?
- ¿Se vislumbra una arquitectura que pueda soportar estas características?
- ¿Se identifican los riesgos críticos? ¿Se prevé una forma de mitigarlos?
- ¿El uso del producto justifica la relación costo-beneficio?
- ¿Es factible para la empresa llevar adelante el proyecto?
- ¿Están los inversores de acuerdo con los objetivos?

4.1.4.2. Fase de Elaboración

Durante la fase de elaboración se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura.

Las iteraciones en la fase de elaboración:

- Establecen una firme comprensión del problema a solucionar.
- Establece la fundación arquitectural para el software.
- Establece un plan detallado para las siguientes iteraciones.
- Elimina los mayores riesgos.

El resultado de esta fase es la línea base de la arquitectura. En esta fase se construyen típicamente los siguientes artefactos:

- El cuerpo básico del software en la forma de un prototipo arquitectural.
- Casos de prueba.

- La mayoría de los casos de uso (80 %) que describen la funcionalidad del sistema.
- Un plan detallado para las siguientes iteraciones.

La fase de elaboración finaliza con el hito de la Arquitectura del Ciclo de Vida. Este hito se alcanza cuando el equipo de desarrollo y las partes interesadas llegan a un acuerdo sobre:

- Los casos de uso que describen la funcionalidad del sistema.
- La línea base de la arquitectura.
- Los mayores riesgos han sido mitigados.
- El plan del proyecto.

Al alcanzar este hito debe poder responderse a preguntas como:

- ¿Se ha creado una línea base de la arquitectura? ¿Es adaptable y robusta? ¿Puede evolucionar?
- ¿Se han identificado y mitigado los riesgos más graves?
- ¿Se ha desarrollado un plan del proyecto hasta el nivel necesario para respaldar una agenda, costes y calidad realistas?
- ¿Proporciona el proyecto una adecuada recuperación de la inversión?
- ¿Se ha obtenido la aprobación de los inversores?

4.1.4.3. Fase de Construcción

Durante la fase de construcción se crea el producto. La línea base de la arquitectura crece hasta convertirse en el sistema completo. Al final de esta fase el producto contiene todos los casos de uso implementados, sin embargo puede que no este libre de defectos. Los artefactos producidos durante esta fase son:

- El sistema software.
- Los casos de prueba.
- Los manuales de usuario.

La fase de construcción finaliza con el hito de Capacidad Operativa Inicial. Este hito se alcanza cuando el equipo de desarrollo y las partes interesadas llegan a un acuerdo sobre:

- El producto es estable para ser usado.
- El producto provee alguna funcionalidad de valor.
- Todas las partes están listas para comenzar la transición.

4.1.4.4. Fase de Transición

La fase de transición cubre el período durante el cual el producto se convierte en la versión beta.

Las iteraciones en esta fase continúan agregando características al software. Sin embargo las características se agregan a un sistema que el usuario se encuentra utilizando activamente.

Los artefactos construidos en esta fase son los mismos que en la fase de construcción. El equipo se encuentra ocupado fundamentalmente en corregir y extender la funcionalidad del sistema desarrollado en la fase anterior.

La fase de transición finaliza con el hito de Lanzamiento del Producto. Este hito se alcanza cuando el equipo de desarrollo y las partes interesadas llegan a un acuerdo sobre:

- Se han alcanzado los objetivos fijados en la fase de Inicio.
- El usuario está satisfecho.

4.2. Planificación y temporización

En este apartado se ilustran las diferentes secciones y tareas de las que se compone este proyecto, y que han sido necesario cumplimentar para llegar a la conclusión del mismo. La planificación y temporización se han hecho en base a las directrices que marca la metodología PUD, por lo tanto se incluyen las cuatro fases de las que se compone: inicio, elaboración, construcción y transición, con sus disciplinas correspondientes. Las siguientes tablas muestran las diferentes tareas y su coste temporal, desglosadas por las distintas secciones que componen el proyecto:

Cuadro 4.1: Gestión del PFC

GESTIÓN DEL PFC	Horas
Diseño de la planificación del PFC	6
Realización de la planificación	5
Mantenimiento y actualización de la planificación	5
Presentación en Secretaría de la propuesta de PFC	1
Solicitar en Secretaria la asignación del PFC	1
Matriculación en Secretaría del PFC	1
Presentación en Secretaría de la documentación final del PFC	1
Presentación en Secretaría de la solicitud de nombramiento del tribunal	1
TOTAL	21

Cuadro 4.2: Realización y tramitación de la propuesta

REALIZACIÓN Y TRAMITACIÓN DE LA PROPUESTA DEL PFC	Horas
Búsqueda de un tema	5
Búsqueda en la Biblioteca de información sobre temas para el PFC	1,5
Búsqueda en Internet de información sobre temas para el PFC	2,5
Búsqueda de un tutor	1
Consultar a profesores de la EII sobre disponibilidad de ser tutor	1
Preparación de una propuesta de PFC	2
Confeccionar un borrador de propuesta de PFC	3
Discusión de propuesta del PFC con el tutor	1
TOTAL	17

Cuadro 4.3: Cuestiones previas

CUESTIONES PREVIAS A LA REALIZACIÓN DEL PFC	Horas
Establecer el entorno de trabajo	6
Establecer las necesidades y recursos del PFC	2
Instalación del hardware y software necesario para el PFC	2
Consulta al tutor sobre el entorno de trabajo	2
Elección del procesador de texto para el PFC	4
Aprender a utilizar latex y el entorno LYX	8
Aprender a confeccionar una bibliografía BibTex con JabRef	2
Estudio de la metodología PUD	21
TOTAL	47

Cuadro 4.4: Desarrollo del PFC

DESARROLLO DEL PFC	Horas
Módulos de la aplicación:	
■ Gestión de Usuarios	
■ Gestión de Elementos del Globo	
■ Aspectos Web 2.0	
■ Interfaz de la aplicación web	
Inicio	
Requerimientos	25
Análisis	15
Diseño	5
Codificación	5
Prueba	5
SUBTOTAL	55
Elaboración	ı
Requerimientos	50
Análisis	40
Diseño	50
Codificación	50
Prueba	10
SUBTOTAL	200
Construcción	
Requerimientos	20
Análisis	40
Diseño	60
Codificación	130
Prueba	50
SUBTOTAL	300
Transición	
Requerimientos	0
Análisis	5
Diseño	10
Codificación	10
Prueba	20
SUBTOTAL	45
TOTAL	600

Cuadro 4.5: Validación y publicidad

VALIDACIÓN Y PUBLICIDAD DEL PFC	Horas
Definición de los test de validación	14
Construcción de los test de validación	3
Aplicación de los test de validación	5
Análisis de resultados de los test de validación	4
Generación de documentación de los test de validación	2
Actualizar la bibliografía de los test de validación	1
Consulta al tutor sobre los test de validación	2
Confección de manuales de usuario	15
TOTAL	46

Cuadro 4.6: Presentación y defensa

PRESENTACIÓN Y DEFENSA DEL PFC	Horas
Memoria del PFC	53
Supervisión del tutor sobre la memoria	4
Realización de copias de la memoria encuadernadas	1
Preparación de la presentación oral del PFC	10
Defensa oral del PFC	1
TOTAL	69

Cuadro 4.7: Resumen de la planificación

Concepto	Horas
GESTIÓN DEL PFC	21
REALIZACIÓN Y TRAMITACIÓN DE LA PROPUESTA DEL PFC	17
CUESTIONES PREVIAS A LA REALIZACIÓN DEL PFC	47
DESARROLLO DEL PFC	600
VALIDACIÓN Y PUBLICIDAD DEL PFC	46
PRESENTACIÓN Y DEFENSA DEL PFC	69
TOTAL	800

A continuación se muestra la planificación temporal por secciones del proyecto:



Figura 4.2: Planificación temporal

La planificación temporal desglosada queda como sigue:

ID	0	Task Name	Horas	Start	Finish	Predecessors
1	111	PFC	800	Mon 03/09/12	Wed 01/05/13	
2	_	GESTIÓN DEL PFC	21	Thu 13/09/12	Tue 16/04/13	
3	112	Diseño de la planificación del PFC	6	Thu 13/09/12	Thu 13/09/12	
4	_	Realización de la planificación	5	Fri 14/09/12	Fri 14/09/12	3
5	H	Mantenimiento y Actualizacion de la planificación	5	Mon 17/09/12	Fri 29/03/13	
6	_	Presentación Secretaria de la propuesta de PFC	1	Mon 17/09/12	Mon 17/09/12	4
7	111	Solicitar Secretaria Asignación del PFC	1	Mon 17/09/12	Mon 17/09/12	
8	111	Matriculación Secretaría PFC	1	Mon 17/09/12	Mon 17/09/12	
9	111	Presentación secretaría documentación final del PFC	1	Tue 16/04/13	Tue 16/04/13	
10	111	Presentación Secretaría solicitud nombramiento tribunal	1	Tue 16/04/13	Tue 16/04/13	
11		REALIZACIÓN Y TRAMITACIÓN PROPUESTA DE PFC	17	Mon 03/09/12	Tue 11/09/12	
12	112	Busqueda de un tutor	1	Mon 03/09/12	Mon 03/09/12	
13		Consultar profesores FI disponibilidad tutor	1	Mon 03/09/12	Mon 03/09/12	
14		Búsqueda de un tema	5	Tue 04/09/12	Tue 04/09/12	13
15		Busqueda Biblioteca Información temas PFC	1,5	Wed 05/09/12	Wed 05/09/12	14
16		Busqueda Internet Información sobre temas PFC	2,5	Thu 06/09/12	Thu 06/09/12	15
17	112	Preparación de una propuesta de PFC	2	Fri 07/09/12	Fri 07/09/12	
18		Confeccionar un borrador de propuesta de PFC	3	Mon 10/09/12	Mon 10/09/12	17
19		Discusion propuesta PFC con el tutor	1	Tue 11/09/12	Tue 11/09/12	18
20		CUESTIONES PREVIAS A LA REALIZACIÓN DEL PFC	47	Wed 12/09/12	Tue 25/09/12	
21	H	Establecer el entorno de trabajo	6	Wed 12/09/12	Wed 12/09/12	
22		Establecer necesitades recursos PFC	2	Thu 13/09/12	Thu 13/09/12	21
23	111	Instalación del hardware y software necesario PFC	2	Thu 13/09/12	Thu 13/09/12	
24	111	Consulta tutor sobre entorno trabajo	2	Thu 13/09/12	Thu 13/09/12	
25		Elección del procesador de texto para el PFC	4	Fri 14/09/12	Fri 14/09/12	24
26	112	Aprender a utilizar latex y el entorno LYX	8	Tue 18/09/12	Tue 18/09/12	25
27	111	Aprender a confeccionar una bibliografía BibTex con JabRef	2	Tue 18/09/12	Tue 18/09/12	
28	111	Estudio de la metodología PUD	21	Tue 25/09/12	Tue 25/09/12	27
29		DESARROLLO DEL PFC	600	Fri 28/09/12	Mon 11/03/13	
30		Inicio	55	Fri 28/09/12	Wed 17/10/12	
31	111	Requerimientos	25	Fri 28/09/12	Mon 08/10/12	
32	HE	Análisis	15	Tue 09/10/12	Fri 12/10/12	31
33		Diseño	5	Mon 15/10/12	Mon 15/10/12	32
34	111	Codificación	5	Tue 16/10/12	Tue 16/10/12	33
35	111	Prueba	5	Wed 17/10/12	Wed 17/10/12	34
36		Elaboración	200	Thu 18/10/12	Wed 12/12/12	

ID	0	Task Name	Horas	Start	Finish	Predecessors
37	111	Requerimientos	50	Thu 18/10/12	Thu 01/11/12	
38	***	Análisis	40	Fri 02/11/12	Tue 13/11/12	37
39	112	Diseño	50	Wed 14/11/12	Mon 26/11/12	38
40	111	Codificación	50	Tue 27/11/12	Fri 07/12/12	39
41	111	Prueba	10	Mon 10/12/12	Wed 12/12/12	40
42		Construcción	300	Thu 13/12/12	Fri 22/02/13	
43	112	Requerimientos	20	Thu 13/12/12	Tue 18/12/12	
44	112	Análisis	40	Wed 19/12/12	Fri 28/12/12	43
45	111	Diseño	60	Mon 31/12/12	Tue 08/01/13	44
46	111	Codificación	130	Wed 09/01/13	Fri 08/02/13	45
47	111	Prueba	50	Mon 11/02/13	Fri 22/02/13	46
48		Transición	45	Mon 25/02/13	Mon 11/03/13	
49	111	Requerimientos	0	Mon 25/02/13	Mon 25/02/13	
50	111	Análisis	5	Tue 26/02/13	Tue 26/02/13	49
51	111	Diseño	10	Wed 27/02/13	Thu 28/02/13	50
52	111	Codificación	10	Fri 01/03/13	Mon 04/03/13	51
53	111	Prueba	20	Tue 05/03/13	Mon 11/03/13	52
54		VALIDACIÓN Y PUBLICIDAD DEL PFC	46	Wed 13/03/13	Fri 29/03/13	
55	***	Definición de los test de validación	14	Wed 13/03/13	Fri 15/03/13	
56	111	Consulta tutor test de validación	2	Mon 18/03/13	Mon 18/03/13	55
57	111	Construcción de los test de validación	3	Tue 19/03/13	Tue 19/03/13	56
58	***	Aplicación de los test de validación	5	Wed 20/03/13	Wed 20/03/13	57
59		Análisis de resultados de los test de validación	4	Thu 21/03/13	Thu 21/03/13	58
60	112	Generación de documentación test de validación	2	Fri 22/03/13	Fri 22/03/13	59
61	111	Actualizar bibliografía test de validación	1	Mon 25/03/13	Mon 25/03/13	60
62	111	Confección de manuales de usuario	15	Tue 26/03/13	Fri 29/03/13	61
63		PRESENTACIÓN Y DEFENSA DEL PFC	69	Mon 01/04/13	Wed 01/05/13	
64	HE	Memoria del PFC	53	Mon 01/04/13	Thu 11/04/13	
65	118	Supervisión tutor memoria	4	Fri 12/04/13	Fri 12/04/13	
66	111	Realización copias memoria encuadernadas	1	Mon 15/04/13	Mon 15/04/13	
67	111	Preparación presentación oral del PFC	10	Thu 18/04/13	Fri 19/04/13	
68	111	Defensa oral del PFC	1	Wed 01/05/13	Wed 01/05/13	

Figura 4.3: Planificación desglosada

4.3. Presupuesto

El presupuesto necesario para la realización de este proyecto se descompone en cuatro partes. Para cada una se detalla por separado el coste de la misma y al final se expone el coste completo de la realización del proyecto.

4.3.1. Costes de personal

Para calcular los costes laborales lo primero sería determinar el coste por hora trabajada. En este caso se han determinado 16 euros cada hora trabajada por el alumno (suponiendo unos ingresos de 1280 euros brutos mensuales) y 30 euros cada hora trabajada por el profesor (2504 euros brutos mensuales). El reparto de horas dedicadas al proyecto por los distintos participantes quedaría de la siguiente manera:

- Tutor 15 horas
- Alumno 800 horas

Teniendo en cuenta los datos anteriores, tendríamos los siguientes costes laborales por participante:

■ Total Tutor: 15 horas * 30 euros = 450 euros

■ Total Alumno: 800 horas * 16 euros = 12.800 euros

Cuadro 4.8: Costes de personal

Concepto	Cantidad	Precio por Unidad	Precio Total
Coste Laboral Tutor	15	30	450
Coste Laboral Alumno	800	16	12.800

4.3.2. Costes inventariables

El material necesario para el proyecto será de un ordenador portátil y un disco duro externo. El software necesario para la elaboración del proyecto no supondrá ningún coste debido al uso de software libre y licencias de estudiante. Para el portátil su coste es de 1.000 $\mbox{\ensuremath{\mathfrak{C}}}$ y para el disco duro 120 $\mbox{\ensuremath{\mathfrak{C}}}$. El coste total asciende a 1.120 $\mbox{\ensuremath{\mathfrak{C}}}$. Teniendo en cuenta que el tiempo de amortización es de 48 meses, y que el equipo se usará durante 7 meses, el coste para ese periodo es de 1.120 $\mbox{\ensuremath{\mathfrak{C}}}$ 7 / 48 = 163,33 $\mbox{\ensuremath{\mathfrak{C}}}$.

Cuadro 4.9: Costes inventariables

Concepto	Cantidad	Precio por Unidad	Precio Total
Ordenador Portátil	1	1.000	1000
Disco Duro Externo	1	120	120

4.3.3. Costes fungibles

Para la entrega del proyecto es necesario realizar una copia de la memoria del mismo, debidamente encuadernada, la cual habrá que entregar en la administración. Además, hay que añadir dos copias digitales de la misma en CD que también han de entregarse. Por lo tanto el coste es:

- Tomo: 0.04 euros/página * 2 (ambas caras) * 278 páginas + 6 euros (encuadernación) = 28,24 euros
- CDs: 0.30 euros * 2 discos = 0.60 euros

Cuadro 4.10: Costes fungibles

Concepto	Cantidad	Precio por Unidad	Precio Total
Tomo de la Memoria	1	28,24	28,24
CD	2	0,30	0,60

4.3. PRESUPUESTO 49

4.3.4. Costes indirectos

Son aquellos que no dependen directamente de la realización del proyecto, esto es: trabajo del personal de la administración, gastos en los servicios de alumbrado y de red del edificio de la EII, etc. Estos costes son difíciles de calcular por su complejidad y variedad, así que para simplificar se ha presupuestado un $5\,\%$ del total del presupuesto.

■ (Coste de personal + Costes inventariables + Costes fungibles) * 5 %= 14.398,84 * $5\% = 719.94 \ \odot$

Cuadro 4.11: Costes indirectos

Concepto	Total	Porcentaje	Precio Total
C.P. + C.I. + C.F.	14.398,84	5%	719,94

4.3.5. Total del presupuesto

A continuación se muestra una tabla resumen con los costes asociados al proyecto:

Cuadro 4.12: Coste total

Concepto	Coste
Costes laborales del alumno	12.800 €
Costes laborales del tutor	450 €
Costes materiales	1.120 €
Costes de documentación	28,84 €
Costes indirectos	719,94 €
PRESUPUESTO TOTAL	15.118,78 €

El presupuesto total del proyecto es de 15.118,78 euros.

Capítulo 5

Desarrollo del proyecto

En este capítulo se ilustrarán varios de los diferentes artefactos que se han realizado hasta llegar a obtener el producto final, desde la recopilación de requisitos hasta la etapa de diseño del software. La primera parte del desarrollo del proyecto consistirá en identificar los requisitos del sistema y los requisitos del software. Este estudio servirá como base para definir y delimitar el ámbito del proyecto, y para utilizarlo como guía de desarrollo. Las herramientas que se utilizarán serán el modelo del dominio, la lista de características y los casos de uso. La segunda permite definir cualquier característica que fuese deseable incluir en la aplicación y la última perfila de forma algo más concreta las funciones que se esperan del mismo.

5.1. Requisitos del sistema

En este apartado se describirán dos artefactos muy importantes a la hora de establecer el ámbito en el que se moverá la aplicación: el modelo del dominio y la lista de características. También ayudan a los desarrolladores a comprender el contexto del sistema, además de poder recopilar requisitos funcionales y no funcionales.

5.1.1. Modelo del dominio

5.1.1.1. Introducción

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema.

Muchos de los objetos del dominio o clases (para emplear una terminología más precisa) pueden obtenerse de una especificación de requisitos o mediante una entrevista con los expertos del dominio. Las clases del dominio aparecen en tres formas típicas:

- Objetos del negocio que representan cosas que se manipulan en el negocio, como pedidos, cuentas y contratos.
- Objetos del mundo real y conceptos de los que el sistema debe hacer un seguimiento, como la aviación enemiga, misiles y trayectorias.

 Sucesos que ocurrirán o han ocurrido, como la llegada de un avión, su salida y la hora de la comida.

El modelo del dominio se describe mediante diagramas de UML (especialmente diagramas de clases).

Estos diagramas muestran a los clientes, usuarios, revisores y a otros desarrolladores las clases del dominio y cómo se relacionan unas con otras mediante asociaciones.

5.1.1.2. Desarrollo de un modelo del dominio

El modelado del dominio se realiza habitualmente en reuniones organizadas por los analistas del dominio, que utilizan UML y otros lenguajes de modelado para documentar los resultados. Para formar un equipo eficaz, estas reuniones deberían incluir tanto a expertos del dominio como a gente con experiencia en modelado.

El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema. Los dominios de tamaño moderado normalmente requieren entre 10 y 50 de esas clases. Los dominios más grandes pueden requerir muchas más.

Los restantes cientos de clases candidatas que los analistas pueden extraer del dominio se guardan como definiciones en un glosario de términos; de otra manera, el modelo del dominio se haría demasiado grande y requeriría más esfuerzo del necesario para esta parte del proceso.

Algunas veces, como en los dominios de negocio muy pequeños, no es necesario desarrollar un modelo de objetos para el dominio; en su lugar puede ser suficiente un glosario de términos.

El glosario y el modelo del dominio ayudan a los usuarios, clientes, desarrolladores y otros interesados a utilizar un vocabulario común. La terminología común es necesaria para compartir el conocimiento con los otros. Cuando abunda la confusión, el proceso de ingeniería se hace difícil, si no imposible. Para construir un sistema software de cualquier tamaño, los ingenieros de hoy en día deben "fundir" el lenguaje de todos los participantes en uno solo consistente.

Por último, es necesaria una llamada de atención sobre el modelado del dominio. Puede ser bastante fácil el comenzar modelando las partes internas de un sistema y no su contexto. Por ejemplo, algunos objetos del dominio podrían tener una representación inmediata en el sistema, y algunos analistas del dominio podrían a su vez caer en la trampa de especificar los detalles relativos a esa representación. En casos como éstos, es muy importante recordar que el objetivo del modelado del dominio es contribuir a la comprensión del contexto del sistema, y por lo tanto también contribuir a la comprensión de los requisitos del sistema que se desprenden de este contexto. En otras palabras, el modelado del dominio debería contribuir a una comprensión del *problema* que se supone que el sistema resuelve en relación a su contexto. El modo interno por el cual el sistema resuelve este problema se tratará en los flujos de trabajo de análisis, diseño, e implementación.

Uso del modelo del dominio

Las clases del dominio y el glosario de términos se utilizan en el desarrollo de los modelos de casos de uso y de análisis. Se utilizan:

- Al describir los casos de uso y al diseñar la interfaz de usuario.
- Para sugerir clases internas al sistema en desarrollo durante el análisis.

5.1.1.3. Diagramas

A continuación se presenta el modelo del dominio para el contexto en el que se engloba la aplicación. Debido a las dimensiones del modelo, éste se ha dividido en varias partes para facilitar su comprensión y dar mayor claridad al mismo. El modelo del dominio se desglosa en tres diagramas principalmente: el principal, otro dedicado a las rutas y un tercero para la parte social de la aplicación. Hay otros auxiliares que se detallarán más adelante.

La identificación de los elementos o clases que componen los diversos diagramas del modelo de dominio se ha conseguido mediante varias iteraciones al proceso de recoger aquellos objetos u aspectos pertenecientes al mundo real asociado al contexto de la aplicación. En este caso el contexto de la aplicación es el de las rutas y su gestión por parte del usuario de la aplicación.

Principal

El usuario de la aplicación gestiona capas, que son contenedores de rutas y sitios de interés. También puede gestionar las rutas y los sitios de interés por separado. La representación de estos elementos se hace sobre una recreación del globo terráqueo mostrada por la aplicación. Por otro lado, el usuario dispone de un GPS para orientarse durante el recorrido de las rutas.

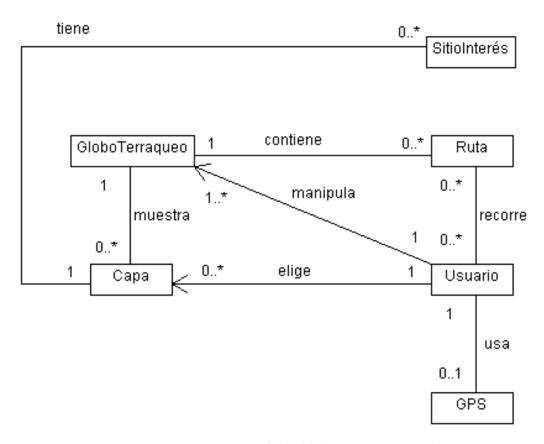


Figura 5.1: Modelo del dominio - Principal

Elementos del globo

En el siguiente diagrama se observa la relación entre los diversos elementos que se pueden mostrar en el globo. En primer lugar está Capa, que consiste principalmente en un contenedor para los otros dos elementos: Ruta y Sitio de Interés. El primero de ellos representa un trayecto del usuario a través de un conjunto de puntos dibujados en el globo terráqueo y que puede ser recorrido por el usuario. Como se ve, de Ruta surgen tres clases más: RutaMarítima, RutaAerea y RutaTerrestre. Ello cubre los tipos básicos de Ruta según el medio donde se encuentre el recorrido. Para RutaTerrestre hay dos nuevas clases en función del medio de locomoción: RutaaPie y RutaCoche. Para ésta última hay una herencia que se corresponde con los diversos tipos de carreteras que hay.

En cuanto a Sitio de Interés, se corresponde con un punto en el globo asociado a algún elemento que pueda ser interesante para el usuario durante el recorrido de sus rutas.

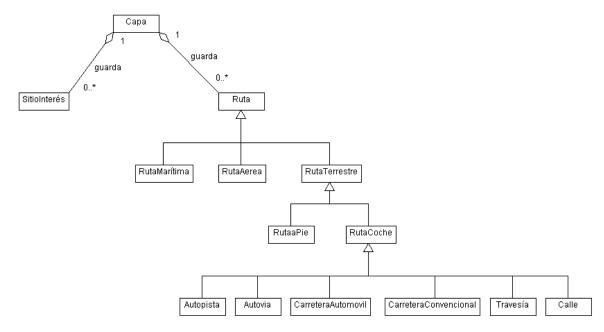


Figura 5.2: Modelo del dominio - Elementos del globo

Web 2.0

Esencialmente, la parte de Web 2.0 de cara al usuario se puede dividir en dos partes: comunicación y compartición. Para la parte de comunicación entre los usuarios existen dos clases fundamentales: Chat y E-Mail. Ambas se corresponden con métodos que permiten a los usuarios comunicarse y mantener conversaciones entre ellos. En cuanto a la compartición, ésta se realiza entre los diferentes Elementos Audiovisuales que el usuario haya añadido a la aplicación. Hay una clase base, Elemento-Audiovisual, de la cual heredan cuatro clases más: Foto, Documento, Modelo3D y Vídeo. Cada una de estas clases se corresponde con su equivalente digital, y que el usuario puede interactuar en la aplicación, así como compartir entre el resto de usuarios que tenga agregados en la misma.

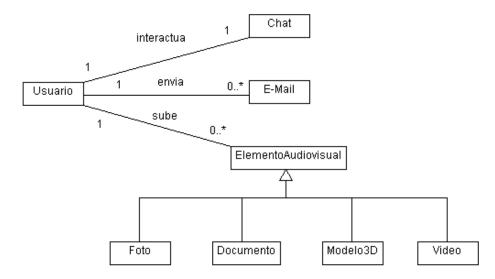


Figura 5.3: Modelo del dominio - Web 2.0

Sitios de interés

A partir de la clase **SitioInterés** aparece una jerarquía de subclases que ayudan a distinguir al usuario los diferentes tipos de sitios de interés que se puede encontrar durante el recorrido que realice sobre una ruta. Tenemos **AccidenteGeográfico**, que se describirá en la siguiente sección, y el resto de clases representan a sitios de interés naturales, culturales, comerciales o urbanísticos.

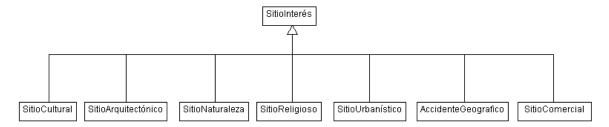


Figura 5.4: Modelo del dominio - Sitios de interés

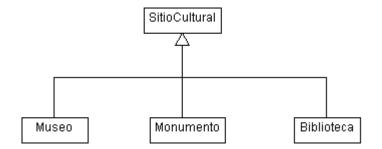


Figura 5.5: Modelo del dominio - Sitios de interés culturales

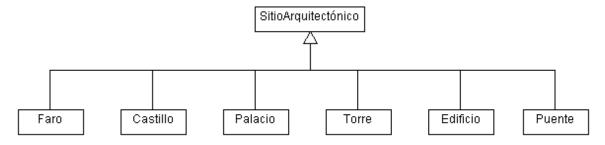


Figura 5.6: Modelo del dominio - Sitios de interés arquitectónicos

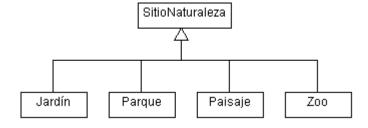


Figura 5.7: Modelo del dominio - Sitios de interés natural

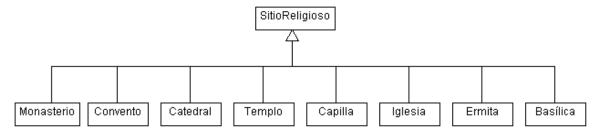


Figura 5.8: Modelo del dominio - Sitios de interés religiosos

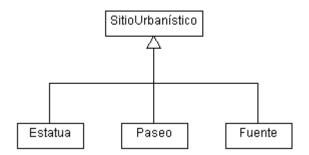


Figura 5.9: Modelo del dominio - Sitios de interés urbanísticos

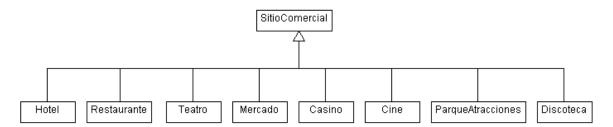


Figura 5.10: Modelo del dominio - Sitios de interés comercial

Accidentes geográficos

Los accidentes geográficos son cada uno de los elementos que componen la superficie del planeta. Se categorizan en función de aspectos tales como la altura o las características del terreno donde se encuentre. La clasificación que se presenta a continuación no procede de ningún estándar, pero sí se ha construido a partir de las informaciones relativas al tema para conseguir el mayor grado de fiabilidad posible. Al igual que en el caso anterior con los sitios de interés, el usuario podrá encontrarse con alguno de los accidentes geográficos indicados aquí durante el recorrido de una de sus rutas.

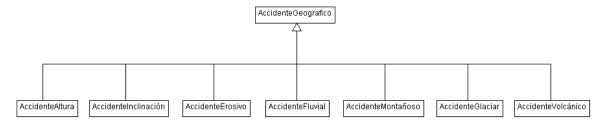


Figura 5.11: Modelo del dominio - Accidentes geográficos

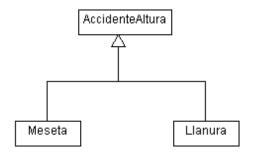


Figura 5.12: Modelo del dominio - Accidentes geográficos (altura)

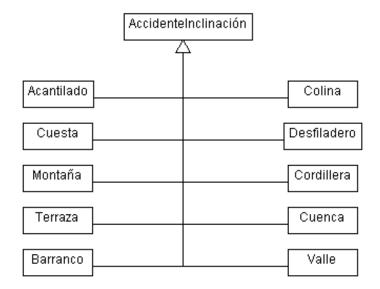


Figura 5.13: Modelo del dominio - Accidentes geográficos (inclinación)

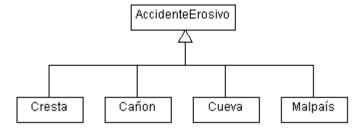


Figura 5.14: Modelo del dominio - Accidentes geográficos (erosivos)

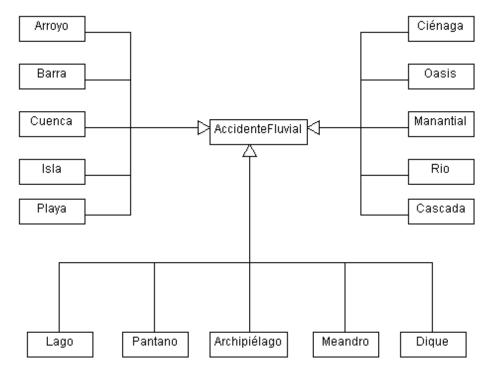


Figura 5.15: Modelo del dominio - Accidentes geográficos (fluviales)

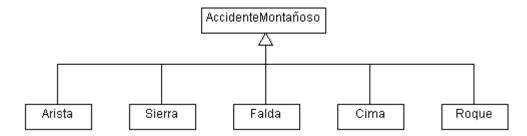


Figura 5.16: Modelo del dominio - Accidentes geográficos (montañosos)

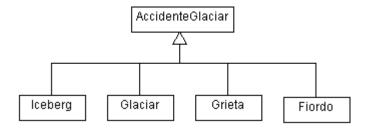


Figura 5.17: Modelo del dominio - Accidentes geográficos (glaciares)

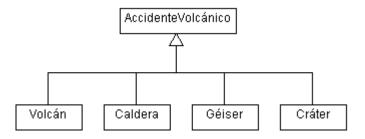


Figura 5.18: Modelo del dominio - Accidentes geográficos (volcánicos)

5.1.2. Enumeración de requisitos candidatos

5.1.2.1. Introducción

La lista de características es un artefacto que se obtiene después de aplicar la tarea de "Enumerar los requisitos candidatos" que se propone en la metodología PUD (Proceso Unificado de Desarrollo), para la captura de requisitos. Por lo tanto, se encuentra englobada dentro de la fase de inicio. Esta lista sirve para contener las ideas de clientes, usuarios, analistas y desarrolladores a modo de fichas sobre los posibles aspectos que se podrían incluir en la aplicación, y que, posteriormente, se podrán traducir en requisitos del software. Estas ideas se consideran requisitos candidatos que se podrán desarrollar en la versión actual del sistema o se podrán postergar a versiones futuras. Este artefacto sirve para gestionar el proyecto y sólo se utiliza para la planificación del trabajo. Podrá ir variando a medida que avance el proyecto, pudiéndose añadir y modificar las características que se crean oportunas, en cualquier momento del desarrollo.[JBR99]

5.1.2.2. Leyenda

Cada ficha que representa a una característica se compone de diversos campos que pasamos a describir:

Código: Es el identificador de la característica. Su formato es el siguiente: "LC-Categoría.Número".

Nombre: Nombre de la característica.

Descripción: Se describe la funcionalidad de la característica a través de un texto explicativo.

Coste: Coste de desarrollar la característica.

Prioridad: Indica el orden a la hora de desarrollar dicha característica.

Estado: Indica como va evolucionando el desarrollo de la característica. Sus posibles valores son: Propuesto, Aprobado, Incluido, En desarrollo, Finalizado.

Nivel de riesgo: Especifica la complejidad para conseguir realizar la característica de manera correcta. Sus posibles valores son: Crítico, Significativo, Rutinario.

Estos valores se utilizan para estimar el tamaño del proyecto y decidir cómo dividirlo en una secuencia de iteraciones. La prioridad y nivel de riesgo asociados, por ejemplo, se emplea para decidir en qué iteración se implementará la característica.

5.1.2.3. Aspectos involucrados con las Rutas

LC-A.1	Gestión de rutas
Descripción:	Los usuarios de la aplicación podrán crear, guardar,
	visualizar, renombrar, modificar y eliminar las rutas
	creadas de manera cómoda y sencilla a través de la
	aplicación. La creación de las rutas se hará marcando
	con el ratón los puntos en el globo terráqueo. Para
	modificar la ruta se arrastrarán dichos puntos usando
	también el ratón.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-A.2	Ver Tour
Descripción:	Cuando el usuario seleccione la visualización de un
	tour para una ruta determinada, la cámara
	comenzará un recorrido virtual empezando desde el
	punto inicial y siguiendo por el resto de puntos de la
	ruta hasta llegar al último. El recorrido se hará usando
	una perspectiva cenital. Si hay puntos de interés en la
	ruta, la cámara se detendrá en ellos y se
	mostrará información asociada al punto de interés (de
	manera opcional). El recorrido utilizará controles de
	reproducción para iniciar, parar y salir del mismo.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.3	Ver sitios de interés
Descripción:	El usuario durante la visualización de una ruta debe
	poder ver aquellos lugares de interés que posea la
	misma. Al hacer clic en uno de ellos con el ratón se
	mostrará información asociada y contextual sobre el
	sitio de interés seleccionado. Dicha información
	también podrá ocultarse una vez mostrada.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-A.4	Gestionar tipos de sitios de interés
Descripción:	El usuario deberá poder crear, modificar y borrar
	tipos de sitios de interés (p.e. geográficos, turísticos,
	poblaciones, etc.) que desee visualizar durante el
	recorrido de las rutas. Cada tipo de sitio de interés
	tendrá sus propias propiedades en función del tipo al
	que pertenezca. El usuario también podrá clasificarlos
	en diversas categorías de su elección.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riego:	Rutinario

LC-A.5	Propiedades de un sitio de interés
Descripción:	El usuario durante la visualización de una ruta que
	contenga sitios de interés podrá visualizar información
	característica de un sitio de interés concreto haciendo
	clic con el ratón en el mismo. Dichas propiedades
	variarán según la tipología del sitio de interés
	seleccionado.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-A.6	Ver modelos 3D
Descripción:	El usuario durante la visualización de una ruta
	podrá ver elementos característicos de la misma en 3D
	como estatuas u otros de diversa índole (monumentos,
	efigies, etc.).
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-A.7	Obtener información de un punto 3D
Descripción:	El usuario durante la visualización de una ruta puede
	hacer que la aplicación le muestre, para un punto
	determinado de dicha ruta, diversa información
	geográfica, como las coordenadas (longitud y latitud)
	del punto y su altitud, entre otras.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-A.8	Obtener información de una ruta
Descripción:	El usuario durante la visualización de una ruta puede
	hacer que la aplicación le muestre, para dicha ruta,
	información geográfica como la altura mínima, máxima,
	desnivel, distancia total, etc. Dicha información se
	mostrará mediante un cuadro contextual.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Significativo

LC-A.9	Medidor de distancias
Descripción:	El usuario durante la visualización de una ruta
	podrá medir distancias entre diversos puntos de la
	misma. El usuario hará clic con el ratón en el primer
	punto y en el segundo. Tras ello, el sistema le
	mostrará la distancia entre ellos. La medición puede
	ser en línea recta o siguiendo la orografía del terreno.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.10	Visualizar objetos 3D
Descripción:	El usuario podrá elegir diversas texturas con las que se
	podrán reconstruir algunos objetos 3D como edificios,
	carreteras, plazas, etc. a la hora de mostrar el globo
	terráqueo. Esta elección puede ser cambiada en
	cualquier momento por parte del usuario.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-A.11	Insertar elemento 3D
Descripción:	El usuario puede añadir modelos 3D de elementos
	característicos de rutas a los existentes en la aplicación
	para documentar mejor las mismas. La aplicación lo
	almacenará dentro de su almacén de elementos 3D.
	Tras ello, el usuario puede usarlo para colocarlo donde
	crea más conveniente durante la visualización de una
	ruta.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-A.12	Almacén de elementos 3D
Descripción:	La aplicación mantendrá un almacén de elementos 3D
	genéricos para ser utilizado por todos los usuarios en
	la creación de sus rutas. Además el usuario
	podrá añadir más elementos 3D a su propio almacén si
	así lo quisiera. Dichos elementos incorporados podrán
	ser borrados por el usuario si no los necesita.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.13	Visualizar Brújula
Descripción:	El usuario podrá activar o desactivar una brújula en el
	navegador que indicará el norte geográfico. Esta opción
	puede ser cambiada por parte del usuario en cualquier
	momento.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.14	Visualizar localización global
Descripción:	Existirá un panel en el que se mostrará la posición
	global del usuario en términos de longitud, latitud y
	altura. El usuario podrá habilitar o deshabilitar este
	panel.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.15	Indicar localización
Descripción:	El usuario podrá introducir el nombre de un lugar y la
	aplicación mostrará el lugar en el globo a partir de la
	dirección dada. Para ello, desplazará la cámara desde
	el punto actual hasta la nueva localización. Además se
	mostrará información adicional sobre la localización
	especificada.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-A.16	Gestión de eventos
Descripción:	La aplicación permitirá crear, borrar y modificar
	eventos. Los eventos son sucesos que ocurren cuando el
	usuario está visualizando una ruta, y durante su
	recorrido se cumplen ciertas condiciones (p.e. se
	aproxima a un sitio, sale de los límites de una región),
	o asociados a modelos 3D existentes en la ruta.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Significativo

LC-A.17	Buscar información
Descripción:	La aplicación permitirá a los usuarios buscar
	información sobre lugares de interés: ya sea a través de
	un buscador de Internet o Wikipedia. Durante la
	visualización de una ruta, cuando el usuario haga clic
	en un sitio de interés, podrá buscar la información
	directamente de manera cómoda.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-A.18	Crear ruta óptima
Descripción:	La aplicación permitirá a los usuarios mostrarle la ruta
	óptima entre dos puntos dados tras calcularla. Para
	ello, el usuario elegirá los dos puntos usando el ratón y
	el sistema calculará dicha ruta en función de diversos
	parámetros: medio de transporte, tiempo necesario
	para recorrer la ruta, etc.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Significativo

LC-A.19	Termómetro
Descripción:	La aplicación permitirá mostrar la temperatura del
	lugar donde se encuentre situado el usuario en ese
	momento.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.20	Barómetro
Descripción:	La aplicación permitirá mostrar la presión atmosférica
	del lugar donde se encuentre situado el usuario en ese
	momento.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.21	Guía
Descripción:	La aplicación mostrará guías sobre rutas (ya sean en
	audio o vídeo). Por ejemplo, durante la visualización
	de un tour, el usuario puede estar escuchando al
	mismo tiempo una pista de audio describiendo la ruta
	que se está visualizando. El usuario puede controlar la
	reproducción de la guia. Por otro lado, el usuario
	también podría ver guias en formato vídeo
	describiendo las características de la ruta seleccionada.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.22	Importar rutas
Descripción:	La aplicación permitirá importar al usuario rutas en
	otros formatos creadas con otras aplicaciones. Tras
	importarlas, se añadirán a las rutas ya existentes que
	tuviera el usuario.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.23	Exportar rutas
Descripción:	La aplicación permitirá exportar al usuario sus rutas
	en otros formatos para que pueda usarlas en otras
	aplicaciones.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.24	Cambiar unidad de distancia
Descripción:	La aplicación permitirá mostrar las distancias entre
	puntos usando el sistema métrico o anglosajón. Esta
	opción podrá ser cambiada por el usuario en cualquier
	momento.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.25	Alquiler de coches
Descripción:	La aplicación permitirá al usuario realizar la gestión de
	alquilar un coche (usando otros servicios). Para ello,
	debería indicar la ruta que desea realizar de entre las
	que disponga y el sistema le mostraría una lista de
	empresas que trabajen en la zona.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Significativo

LC-A.26	Reservas de hotel
Descripción:	La aplicación permitirá realizar la gestión de reservas
	de hotel al usuario (usando otros servicios). Para ello,
	debería indicar el sitio al que desea ir y el sistema le
	mostraría una lista de hoteles que estén situados en
	dicha localización.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Significativo

LC-A.27	Planificación temporal
Descripción:	La aplicación permitirá al usuario a través de un
	calendario especificar fechas para eventos concretos a
	realizar en la ruta. Además de crear eventos,
	podrá modificarlos o eliminarlos. También
	podrá indicar la hora concreta de los mismos.
	Asimismo, el usuario podrá añadir tareas por realizar
	a la planificación.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-A.28	Permitir múltiples instancias del plugin
Descripción:	La aplicación mostrará al usuario si lo desea varios
	globos para visualizar diferentes contenidos en
	paralelo. Se podría usar por ejemplo para mostrar
	diferentes rutas, una en cada instancia del plugin, para
	comparar diversas características de las rutas.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

5.1.2.4. Rasgos ligados a los Usuarios

LC-B.1	Gestión de usuarios
Descripción:	La aplicación permitirá dar de alta a usuarios,
	así como que éstos se den de baja de la misma y
	permitirá modificar sus perfiles. En principio, el
	usuario no tendrá por qué dar todos sus datos para
	registrarse, bastará con una información mínima que
	permita el alta y la posibilidad de recuperar sus datos.
	Posteriormente podrá rellenar el resto de la
	información si así lo desea. También podrá desactivar
	su perfil y eliminar sus datos de la aplicación.
	Habrá un usuario gestor de la aplicación que actúe
	como administrador de la misma y será el encargado de
	las tareas propias del mantenimiento de la aplicación.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-B.2	Avatares
Descripción:	Los usuarios podrán usar avatares en sus perfiles a
	modo de imagen personal. Ésta será usada cuando
	varios usuarios estén conectados entre sí siendo lo que
	visualizarán en los navegadores el resto de los usuarios.
	Cuando un usuario se registre en la aplicación, su
	perfil dispondrá de un avatar por defecto, que
	podrá modificar a voluntad.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

5.1.2.5. Características relacionadas con las Capas

LC-C.1	Seleccionar capas a visualizar del globo
Descripción:	El usuario tendrá la opción de seleccionar qué capas de
	todas las que haya creado en la aplicación se
	visualizarán en su navegador. Ello hace que pueda
	mostrarse más de una al mismo tiempo.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-C.2	Elección de capas predefinidas
Descripción:	La aplicación permitirá elegir al usuario las capas que
	desea visualizar (edificios, carreteras, ciudades, etc.),
	de un grupo predefinido. También será posible mostrar
	más de una al mismo tiempo.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-C.3	Gestionar capas del globo
Descripción:	Se podrán añadir, borrar y modificar las capas del
	globo terráqueo. El usuario podrá crear capas
	incorporando rutas o sitios de interés existentes si lo
	desea o creándola vacía. Dichas capas, una vez creadas,
	podrán ser modificadas por el usuario añadiendo o
	quitando elementos (rutas, sitios de interés, etc.) sin
	dificultad. Por último, el usuario también
	podrá eliminar capas de la aplicación si ya no le son
	necesarias, teniendo en cuenta que el resto de
	elementos pertenecientes a la capa serán borrados
	igualmente.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Significativo

LC-C.4	Compartir capas del globo
Descripción:	Las capas creadas con la aplicación por parte de un
	usuario podrán ser compartidas con el resto de
	usuarios. Al compartir la capa, lo harán asimismo el
	resto de elementos que conforman la misma. Los
	cambios que haga un usuario en una capa compartida
	también se verán vistos por los demás usuarios que
	tengan dicha capa.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

5.1.2.6. Facetas de la Web social 2.0

LC-D.1	Rutas favoritas
Descripción:	La aplicación mostrará al usuario las rutas favoritas
	por parte de otros usuarios que tenga agregados. Si un
	usuario marca una de sus rutas como favorita, ésta
	aparecerá en una lista separada obteniendo una mayor
	visibilidad.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-D.2	Presentar otras rutas
Descripción:	La aplicación mostrará al usuario si lo desea otras
	rutas ya creadas por parte de otros usuarios de
	acuerdo a sus gustos similares u otros criterios:
	número de usuarios que tienen esa ruta, criterios
	geográficos, características de la ruta, etc.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-D.3	Chat
Descripción:	Los usuarios podrán utilizar una aplicación de chat
	para conversar entre ellos. Los usuarios podrán
	enviarse mensajes de texto de manera rápida cuando
	estén conectados al mismo tiempo. También se
	mostrarán los avatares de los usuarios durante la
	conversación.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Significativo

LC-D.4	Comentarios
Descripción:	Los usuarios podrán enviar comentarios sobre las
	rutas. Los comentarios serán visibles por el resto de
	usuarios que tengan dicha ruta. También se mostrarán
	los avatares de los usuarios al dejar un comentario.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-D.5	Etiquetas
Descripción:	La aplicación permitirá que el usuario pueda añadir
	etiquetas a las rutas creadas. Ello permite que el
	usuario pueda buscar y clasificar sus rutas usando las
	etiquetas, incluso facilitar la compartición de rutas
	entre usuarios basándose en las etiquetas asociadas a
	las rutas.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-D.6	Compartir rutas
Descripción:	Los usuarios podrán enviarse rutas ya creadas para su
	posterior visualización por parte de otros usuarios,
	siempre y cuando estén agregados. Si la ruta posee
	sitios de interés, también se visualizarán por parte del
	resto de usuarios.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-D.7	E-Mail
Descripción:	El usuario podrá enviar un correo electrónico a otro u
	otros de los que tenga agregados. Esta opción
	estará siempre disponible y se presentará como opción
	si los otros usuarios no están disponibles para
	conversar.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-D.8	Subir ficheros
Descripción:	Los usuarios podrán subir ficheros (fotos, vídeos,
	modelos 3D, etc.) para documentar las rutas
	existentes. Los ficheros estarán disponibles para el
	resto de rutas que quieran crear o modificar, así como
	de eliminarlos si ya no los necesitan. Dichos ficheros
	también podrán ser compartidos con el resto de
	usuarios.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-D.9	Crear comunidades
Descripción:	Los usuarios con afinidades comunes podrán agrupase
	en comunidades para compartir sus aficiones. Las
	comunidades permitirán facilitar la compartición de
	contenidos entre los usuarios que la componen, ya sean
	rutas, capas u otros elementos; así como el intercambio
	de información en grupo. El usuario que cree la
	comunidad será el gestor de la misma.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-D.10	Conexión con otras redes sociales
Descripción:	Los usuarios podrán enviar mensajes desde la
	aplicación para que aparezcan en otras redes sociales y
	aplicaciones web tales como FaceBook, Twitter,
	Tuenti, etc. También podrán compartir otros
	elementos como rutas o fotos por ejemplo.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-D.11	Suscripción
Descripción:	Los usuarios podrán suscribirse a otros usuarios y
	recibir aquellas rutas que éstos creen. De esta manera,
	cualquier ruta o capa que cree el usuario,
	aparecerá disponible para el seguidor, de forma que
	pueda decidir si desea añadir dicho elemento como
	suyo propio.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

5.1.2.7. Utilidades que ofrece la Aplicación

LC-E.1	Soporte para multilenguaje
Descripción:	La interfaz de usuario de la aplicación se adaptará a
	distintos idiomas (inglés, francés, alemán, español). El
	sistema detectará el idioma del navegador del usuario
	y se configurará automáticamente con el idioma
	detectado la primera vez. El usuario puede cambiar el
	idioma seleccionado a otro diferente posteriormente.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Significativo

LC-E.2	Seleccionar nivel de detalle de los edificios
Descripción:	El usuario podrá seleccionar entre distintos niveles de
	detalle para los edificios. Esto influirá sobre la
	velocidad de navegación sobre el globo terráqueo. El
	sistema podría detectar la capacidad gráfica del
	ordenador del usuario y elegir la opción más adecuada.
	Esta opción puede ser modificada por el usuario en
	cualquier momento.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-E.3	Seleccionar la velocidad de navegación
Descripción:	El usuario podrá seleccionar la velocidad de
	navegación. Ello modifica la velocidad a la que se
	mueve la cámara. Permite al usuario recorrer partes de
	la ruta a una velocidad menor por ser de mayor interés
	por ejemplo, y otras más rápido. Esta opción puede ser
	modificada por el usuario en cualquier momento.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-E.4	Modo pantalla completa
Descripción:	El usuario podrá cambiar la interfaz de la aplicación
	cuando active el modo de pantalla completa. La
	aplicación adaptará la interfaz para suministrar la
	misma funcionalidad pero ocultando ciertos menús y
	elementos de interfaz.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-E.5	Actualizar las texturas
Descripción:	La aplicación permitirá actualizar las texturas con
	nuevas versiones de las mismas. En el navegador deben
	aparecer las texturas más actuales posibles, sin obligar
	al usuario a reinstalar o sincronizar su navegador.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-E.6	Actualización automática
Descripción:	La aplicación se podrá actualizar, de forma automática,
	con los nuevos cambios que hayan ocurrido en la API.
	Esto se realizará de forma transparente al usuario.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-E.7	Carga continua de datos remotos
Descripción:	Se irán visualizando en el terreno los datos que vayan
	llegando a través de Internet de forma continua
	durante la visualización del terreno por parte del
	usuario.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-E.8	Histórico de operaciones
Descripción:	La aplicación puede guardar un fichero de registro con
	todas las operaciones que haya realizado el usuario, o
	con sólo aquellas que él elija. El usuario puede ver
	dicho fichero, así como borrarlo. La opción de crear el
	fichero es opcional y es el usuario quién decide si
	crearlo o no.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-E.9	Visualizar gran número de texturas en tiempo real
Descripción:	A medida que se navega se debe ir cargando las
	texturas, a distintas resoluciones, del terreno que se
	está visualizando. En función de la distancia a la que
	se encuentre el usuario como observador variará la
	calidad de las texturas para optimizar la visualización.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Significativo

LC-E.10	Multinavegador
Descripción:	La aplicación deberá soportar estándares que hagan
	que se pueda ejecutar en varios navegadores sin
	modificar la experiencia del usuario
	independientemente del navegador que esté usando el
	usuario en cualquier momento. El conjunto de
	estándares debe abarcar desde la interfaz gráfica hasta
	los aspectos internos de la aplicación.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-E.11	Multiplataforma
Descripción:	La aplicación debe poder compilarse y ejecutarse en
	distintos sistemas operativos como Windows, Linux y
	Mac.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Significativo

LC-E.12	Interfaz de usuario simple
Descripción:	La interfaz de la aplicación será muy sencilla e
	intuitiva, evitando ocultar demasiada información en
	menús anidados y formularios demasiado complicados.
	La funcionalidad principal estará disponible en la
	propia pantalla del navegador.
Coste:	
Prioridad:	
Estado:	Aprobado
Nivel de riesgo:	Rutinario

LC-E.13	Atajos de teclado
Descripción:	La interfaz de la aplicación permitirá al usuario activar
	las funciones más comunes usando su teclado a través
	de pulsaciones de teclas o con combinaciones de las
	mismas. Seguirán un patrón estándar y coherente.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-E.14	Software libre
Descripción:	Para la construcción de la aplicación, tanto las
	herramientas internas como la interfaz gráfica, se
	utilizarán herramientas de software libre. Se
	realizará un estudio de cuales son las mejores opciones
	indicando los pros y contras de cada una y se
	tomará una decisión en consecuencia.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

5.1.2.8. Propiedades de la versión para móviles (PDA)

LC-F.1	Uso PDA
Descripción:	Los usuarios se podrán conectar a la aplicación a
	través de PDAs y teléfonos móviles. La información
	del usuario estará sincronizada entre la versión de
	escritorio y la de PDA, y cualquier cambio se
	verá reflejado en ambas versiones.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Rutinario

LC-F.2	Interfaz para móviles
Descripción:	La interfaz de la aplicación se adaptará para mostrarse
	de la mejor manera posible en móviles y PDAs.
	Tendrá que tener en cuenta las limitaciones de espacio
	en cuanto a la pantalla y de capacidad de cómputo para
	ofrecer el mejor resultado posible de cara al usuario.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Significativo

LC-F.3	Gestión del tráfico
Descripción:	La aplicación permitirá mostrar la información del
	tráfico para las rutas en coche al usuario. Dada la ruta
	seleccionada por el usuario, la aplicación le indicará al
	usuario qué carreteras están congestionadas y le
	ofrecerá alternativas (si las hubiera) para llegar al
	destino.
Coste:	
Prioridad:	
Estado:	Propuesto
Nivel de riesgo:	Crítico

5.2. Requisitos del software

La especificación de requisitos del software es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software, deducidos a partir de la información obtenida con el modelo de dominio y la lista de características. Los casos de uso también son conocidos como requisitos funcionales.

Para cada usuario se crean "Casos de uso", los cuales detallan y describen las operaciones que podrá realizar cada uno, en un lenguaje de alto nivel, en términos más coloquiales que técnicos. A partir de los casos de uso se refinan las operaciones, definiendo los pasos concretos que incluye cada una. Estos pasos se describen mediante las "Tablas de flujo de sucesos". En general se realiza una tabla por cada caso de uso que exista.

Además de los casos de uso, también contiene requisitos no funcionales, que son aquellos que realiza el sistema sin la interacción con un usuario de forma directa y que son necesarios para poder llevar a cabo acciones solicitadas por los mismos.

Todo este trabajo servirá como base para desarrollar la etapa de análisis, la cual se explicará más adelante.

5.2.1. Jerarquía de actores

Los actores son una parte relevante del modelado del sistema, pues forman parte del entorno del mismo. Normalmente, un sistema tiene muchos tipos de usuarios. Cada tipo de usuario se representa por un actor. Los actores utilizan el sistema interactuando con los casos de uso.

Podemos encontrar y especificar todos los actores examinando a los usuarios que utilizarán el sistema y a otros sistemas que deben interactuar con él. Cada categoría de usuarios o sistemas que interactúan se representan por tanto como actores.

Partiendo del modelo del dominio, el analista del sistema, junto con el cliente, identifican los usuarios e intentan organizarlos en categorías representadas por actores. Debemos identificar los actores que representan sistemas externos y los actores para el mantenimiento y operación del sistema.

Para el caso particular de nuestro sistema se han detectado cuatro actores básicos: Administrador, Usuario Registrado, Usuario No Registrado y Empresa. A continuación describiremos brevemente las particularidades de cada uno de ellos:

- Administrador Es el actor encargado del buen funcionamiento del sistema. Por lo tanto es el encargado de realizar las tareas de mantenimiento que hagan que el sistema se mantenga operativo.
- Usuario No Registrado Es el actor que no está identificado en el sistema. Ello implica que no puede acceder a la totalidad de casos de uso que el sistema presenta. Para ello el usuario debe cambiar de rol y pasar a ser otro actor.
- Usuario Registrado Es el actor principal del sistema. La mayoría de usuarios se corresponderán con este actor. La mayor parte de la funcionalidad del sistema se concentra en los casos de uso para este actor.
- **Empresa** Es una especialización del actor Usuario Registrado. Sus posibles particularidades con respecto a éste último nos recomienda que le asignemos un actor específico para

él. Se diferencia con respecto al actor Usuario Registrado en la interacción con el resto de usuarios del sistema y en la posibilidad de dar soporte a la publicidad.

La jerarquía de actores se muestra en la figura que se presenta a continuación. Como raíz del árbol está el actor Usuario, que sin embargo no es un actor real del sistema, es abstracto. El resto de actores heredan de él todas sus propiedades.

Del actor Usuario nacen tres actores: Administrador, Usuario Registrado y Usuario No Registrado. Estos tres actores fueron descritos anteriormente. Del actor Usuario Registrado aparece el actor Empresa, para diferenciar las particularidades de los actores.

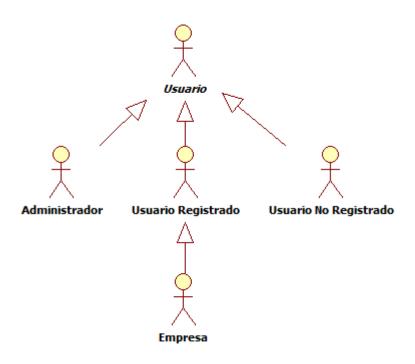


Figura 5.19: Actores del sistema

5.2.2. Casos de uso

5.2.2.1. Gestión de Usuarios

La gestión de Usuarios engloba los casos de uso correspondientes a dos actores: Usuario No Registrado y Usuario Registrado. Los casos de uso para ambos actores están separados en dos secciones que pasan a describirse a continuación:

El Usuario No Registrado no puede acceder al sistema, luego debe pasar primero por un registro para poder ser Usuario Registrado. Por lo tanto, dispone de un primer caso de uso llamado Registrarse, que permite a la aplicación identificarle. Al completar este caso de uso, automáticamente se ejecuta el de Iniciar Sesión, que le autentifica en el sistema. Este caso se puede realizar por separado, una vez el usuario esté registrado, que sólo es necesario hacerlo una vez. Por último, está el caso de uso Recordar Contraseña, que permite a un usuario existente en el sistema realizar el ingreso, pero que no puede identificarse en el sistema por no recordar los datos para realizarlo. En resumen, el actor Usuario No Registrado es el paso

previo del Usuario Registrado.

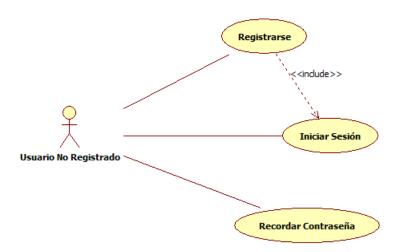


Figura 5.20: Usuario No Registrado

El Usuario Registrado puede ejecutar el caso de uso Cerrar Sesión que concluye la actividad del usuario en el sistema. Tras ello, el usuario toma el rol del actor Usuario No Registrado. También puede Darse de Baja lo que elimina la cuenta del usuario y los datos existentes del mismo en el sistema. Como en el caso anterior, su papel se ve modificado, pues se convierte también en Usuario No Registrado. Además puede modificar sus datos personales para actualizar la información propia en la aplicación, así como cambiar la contraseña de acceso al sistema, caso de uso que está incluido en el anterior. Estos dos casos de uso no alteran el actor implicado.

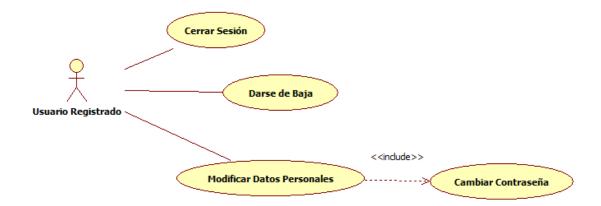


Figura 5.21: Usuario Registrado

5.2.2.2. Gestión de Elementos del Globo

En este apartado se describirán los casos de uso relativos a los diferentes componentes con los que puede interactuar el actor Usuario Registrado y que pueden ser visualizados en el globo terráqueo de la aplicación. Están divididos por categorías, las cuales se describen en las secciones siguientes. Las categorías son: Capas, Rutas, Sitios de Interés y Tours. Cada categoría se compone de los casos de uso correspondientes siguiendo una filosofía CRUD (Create-Read-Update-Delete), añadiendo algunos particulares para cada caso, que se detallarán con más profundidad en las secciones correspondientes.

La gestión de capas que puede realizar el actor Usuario Registrado a través de la aplicación se compone de los siguientes casos de uso: Crear Capa, Visualizar Capa, Modificar Capa, Renombrar Capa y Eliminar Capa. Visualizar Capa se descompone en Visualizar Información, Desplegar Capa, Activar Capa e Ir a. Sintetizan las diferentes formas de ver el contenido de una capa. Para poder modificar una capa, es necesario que la capa se esté visualizando en ese momento.

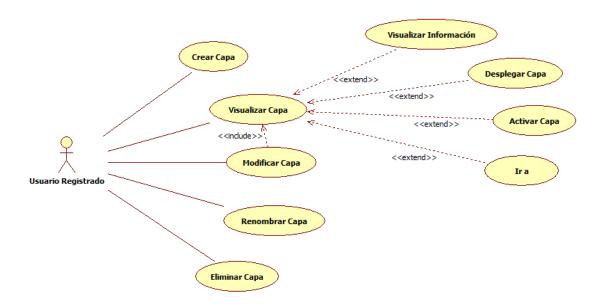


Figura 5.22: Gestión de Capas

El actor Usuario Registrado puede realizar los siguientes casos de uso: Ver Ruta, Ver Tour, Ver Sitios de Interés y Seleccionar Capas. Las diferentes opciones que cambie el usuario en el caso de uso Seleccionar Capas tendrán influencia en el caso de uso Ver Sitios de Interés.

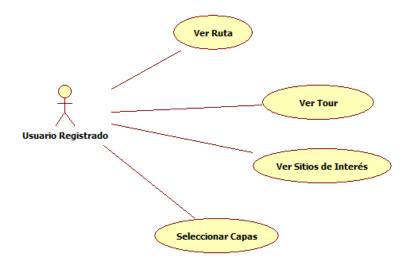


Figura 5.23: Visualizar Ruta

El actor Usuario Registrado dispone de los siguientes casos de uso para gestionar sus rutas: Crear Ruta, Guardar Ruta, Visualizar Ruta, Modificar Ruta, Renombrar Ruta y Eliminar Ruta. En el caso de Guardar Ruta es obligatorio haber utilizado el caso de uso Crear Ruta y para ejecutar el caso de uso Modificar Ruta es necesario haber ejecutado previamente el caso de uso Visualizar Ruta. El caso de uso Crear Ruta se divide en dos casos de uso más: Insertar Puntos e Insertar Elementos. El primero es elemental y no se puede descomponer más; pero el segundo es una abstracción de otros cuatro casos de uso más: Insertar Modelo 3D, Insertar Foto, Insertar Vídeo e Insertar Documento, que están agrupados bajo Insertar Elementos.

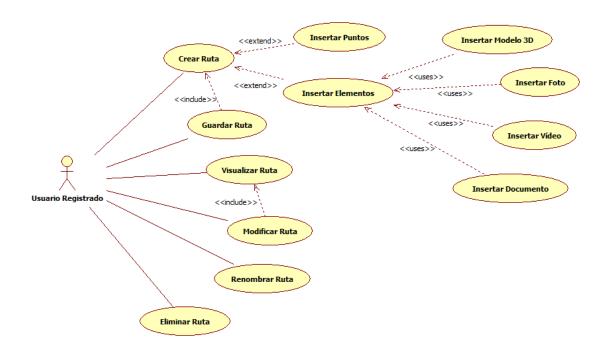


Figura 5.24: Gestionar Rutas

La gestión de sitios de interés por parte del actor Usuario Registrado se resume en los siguientes casos de uso: Añadir Sitio de Interés, Visualizar Sitio de Interés, Modificar Sitio de Interés, Renombrar Sitio de Interés y Eliminar Sitio de Interés. Recordar que los sitios de interés pueden presentar una gran diversidad, por lo que los casos de uso deben adecuarse a tal situación. En general, todos los casos de uso interactuarán con el usuario de manera contextual debido a la tipología de estos elementos a la hora de disponerlos sobre el globo terráqueo.

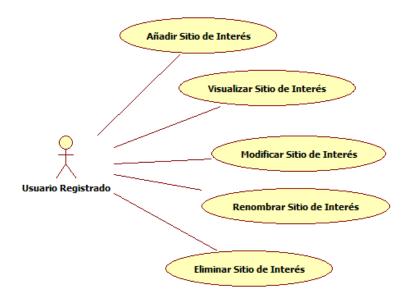


Figura 5.25: Gestionar Sitios de Interés

La gestión de los tours o recorridos por parte del actor Usuario Registrado se concentra en los siguientes seis casos de uso: Crear Tour Automático, Crear Tour Manual, Visualizar Tour, Modificar Tour, Renombrar Tour y Eliminar Tour. Se ha optado por poner dos casos de uso diferentes para crear un tour para otorgar mayor autonomía al usuario. En el caso automático la aplicación generará los tours, lo cual es más cómodo para el usuario, pero no es tan flexible. Al permitir el caso manual, el usuario es libre de crear el tour de la forma que desee.

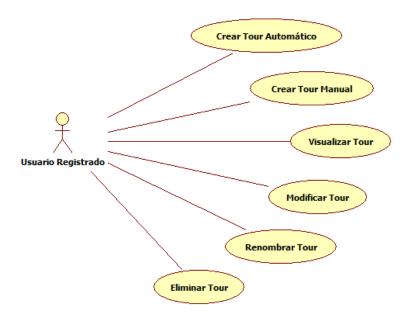


Figura 5.26: Gestionar Tours

5.2.2.3. Web 2.0

La gestión de los aspectos Web 2.0 se ha dividido en tres partes para una mejor clasificación de los casos de uso. Las tres partes se denominan: Compartir, Comunicación y Usuarios. En el primer apartado tenemos la gestión de los diversos elementos que pueden compartirse entre los usuarios de la aplicación, así como formas para facilitar el intercambio de rutas. En cuanto a la comunicación, se presentan las diferentes maneras que tienen los usuarios de comunicarse entre ellos a través de la aplicación. Además hay casos de uso relativos en cuanto a la interacción con otros servicios web que facilitan la comunicación con otros usuarios que no pertenecen a la aplicación. Por último, el apartado concerniente a los usuarios trata sobre cómo el usuario de la aplicación administra los contactos que posee en la misma.

Los casos de uso relacionados con la compartición para el actor Usuario Registrado son los siguientes: Compartir Archivos, Compartir Rutas, Rutas Favoritas y Presentar Otras Rutas. Los archivos que se pueden compartir son: fotos, vídeos, modelos 3D o documentos. Al compartir el archivo una copia del mismo pasa del usuario que lo envía a otro usuario que actúa como destinatario. Lo mismo para las rutas. Las Rutas Favoritas permiten al usuario gestionar de manera más eficiente las rutas que posee. El caso de uso Presentar Otras Rutas favorece la interacción entre las rutas del usuario y las de sus contactos.

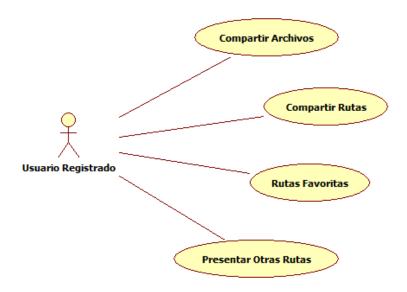


Figura 5.27: Compartición

La comunicación entre usuarios a través de la aplicación se puede realizar con los casos de uso que se indican a continuación: Chatear, Enviar Email, Escribir Comentarios, Crear Comunidades y Comunicarse con otras Redes Sociales. En cuanto a la comunicación interna tenemos los cuatro primeros casos de uso indicados anteriormente. El último caso de uso queda reservado para la comunicación externa con otros usuarios de otras aplicaciones. Habrá que limitar el conjunto de servicios con los que se operará, pues puede ser muy grande.

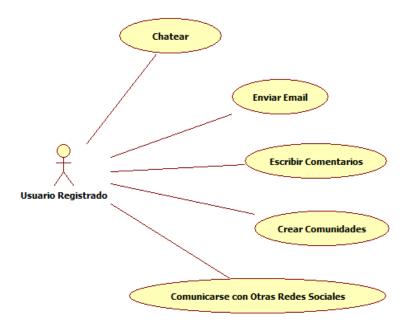


Figura 5.28: Comunicación

Los casos de uso correspondientes a la gestión de usuarios de los que dispone el usuario de la aplicación son: Añadir Usuario, Ver Usuario, Modificar Usuario y Eliminar Usuario. Para añadir usuarios, se hará uso del concepto de solicitud, por la cual el usuario de la aplicación debe enviar una al usuario que desea añadir a su lista de contactos. Este último debe aceptarla para convertirse en contacto del usuario de la aplicación. Los casos de uso Ver Usuario y Modificar Usuario son prácticamente una extensión uno del otro. Por último está el caso de uso Eliminar Usuario, para poder borrar contactos de la lista del usuario.

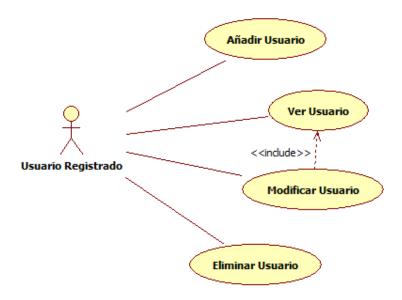


Figura 5.29: Gestión de Contactos

5.2.3. Especificación de casos de uso

Para completar el apartado anterior, en cuanto a la funcionalidad del sistema, es necesario añadir información referente a:

- Descripción del escenario, es decir, cómo un actor interactúa con el sistema, y cuál es la respuesta obtenida.
- Precondiciones y postcondiciones.
- Condiciones de fallo que afectan al escenario, así como la respuesta del sistema (escenarios secundarios).

Por lo tanto, para cada caso de uso identificado en el punto anterior, se detallan los puntos mencionados previamente:

5.2.3.1. Gestión de Usuarios

Nombre:	Registrarse
Descripción:	Con este caso de uso una persona puede convertirse en usuario
	de la aplicación. Simplemente tiene que rellenar un formulario
	estándar de registro dando una serie de datos para hacerlo. La
	aplicación le enviará un correo de bienvenida y le dará acceso
	al sistema.
Precondición:	
Parámetros:	Nombre
	Contraseña
	E-Mail

Flujo de Ejecución:

- 1. El usuario accede al formulario de registrar usuario
- 2. El usuario proporciona sus datos personales:
 - El usuario introduce su nombre
 - El usuario introduce una contraseña
 - El usuario vuelve a introducir la contraseña
 - El usuario introduce su correo electrónico
- 3. El usuario hace clic en el botón de aceptar del formulario
- 4. El sistema realiza una serie de verificaciones:
 - El sistema comprueba que el nombre utilizado no está en uso
 - El sistema comprueba que las dos contraseñas coinciden
 - El sistema comprueba que la contraseña tiene una longitud mínima determinada
 - El sistema comprueba que la contraseña tiene números
 - El sistema comprueba que la contraseña tiene caracteres especiales
 - El sistema comprueba que la dirección de correo es válida
- 5. Si todas las comprobaciones tienen éxito, el usuario se añade a la base de datos del sistema
- 6. El sistema envía un correo a la dirección indicada por el usuario con información relevante acerca de los datos de identificación, cómo darse de baja y una bienvenida
- 7. El usuario recién creado se autentifica en el sistema
- 8. El sistema muestra la pantalla inicial al usuario

Caminos Alternativos:	Nombre en uso (4)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
	 Además se le ofrecerán sugerencias de nombres disponibles
	Las contraseñas no coinciden (4)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
	Las contraseñas no tiene la longitud mínima (4)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
	La contraseña no tiene números (4)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
	La contraseña no tiene caracteres especiales (4)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
	La dirección de correo no es correcta (4)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
Postcondición:	El usuario estará registrado en el sistema y con la sesión iniciada.

Nombre:	Iniciar Sesión
Descripción:	Un usuario registrado de la aplicación para entrar en la misma
	debe autentificarse en el sistema para obtener sus datos y
	acceder a la pantalla inicial. Si no está autentificado y desea
	acceder a un recurso propio de la aplicación, ésta
	deberá cancelar la petición del usuario y llevarlo hasta la
	pantalla de identificación.
Precondición:	Usuario no autentificado
Parámetros:	Nombre
	Contraseña
Flujo de Ejecución:	
	1. El usuario accede al formulario de registrar usuario
	2. El usuario proporciona sus datos personales:
	■ El usuario introduce su nombre
	■ El usuario introduce la contraseña
	3. El usuario hace clic en el botón de identificarse del
	formulario
	4. El sistema realiza una serie de verificaciones:
	■ El sistema comprueba que el nombre existe en la base de datos
	■ El sistema comprueba que la contraseña es correcta
	5. Si todas las comprobaciones tienen éxito el usuario queda autentificado en el sistema
	6. El sistema muestra la pantalla inicial al usuario
Caminos Alternativos:	Usuario no existente (2)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
	Contraseña incorrecta (2)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
Postcondición:	El usuario habrá iniciado sesión en la aplicación.

Nombre:	Recordar Contraseña
Descripción:	Si un usuario de la aplicación no puede iniciar sesión por no
	recordar sus datos de acceso, puede solicitar a la aplicación
	especificando su correo electrónico los datos de acceso. La
	aplicación enviará un correo a la dirección indicada con los
	datos de acceso. Recordar también que cuando el usuario se
	registra, la aplicación envió en su momento al usuario otro
	correo con los datos de acceso.
Precondición:	Usuario no autentificado
Parámetros:	E-Mail
Flujo de Ejecución:	
	1. El usuario hace clic en Recordar Contraseña
	2. El usuario introduce su dirección de correo
	3. El sistema verifica si la dirección de correo está registrada en la base de datos del mismo
	4. Si se encuentra la contraseña, se envía un correo electrónico a dicha dirección indicando la contraseña
	5. Se informa al usuario de que su solicitud se ha completado y se le solicita que mire su correo
Caminos Alternativos:	E-Mail incorrecto (2)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
Postcondición:	Se habrá enviado un e-mail al correo del usuario con la contraseña del usuario.

Nombre:	Cerrar Sesión
Descripción:	El usuario cuando haya finalizado con todas las tareas que
	necesite hacer en la aplicación, puede ejecutar el caso de uso
	Cerrar Sesión para abandonar la aplicación. Ésta devolverá al
	usuario a la pantalla inicial y deberá volver a iniciar sesión
	para acceder.
Precondición:	Usuario autentificado
Parámetros:	
Flujo de Ejecución:	
	1. El usuario hace clic en el botón de cerrar sesión
	2. El sistema cierra la sesión del usuario
	3. El sistema devuelve al usuario a la pantalla inicial de la aplicación
Caminos Alternativos:	
Postcondición:	El usuario habrá cerrado sesión en la aplicación.

Nombre:	Darse de baja
Descripción:	El usuario que no desee continuar haciendo uso de la
	aplicación, puede darse de baja de la misma. El usuario que lo
	haga pierde todos los datos que haya subido a la aplicación. Se
	podría añadir algún formulario para que el usuario expresara
	que le ha parecido la aplicación y qué cosas mejoraría.
Precondición:	Usuario autentificado
Parámetros:	Contraseña
Flujo de Ejecución:	
	1. El usuario accede al formulario de darse de baja
	2. El usuario introduce su contraseña
	3. La aplicación solicita al usuario confirmar la operación
	4. Si el usuario acepta:
	■ El sistema cierra la sesión del usuario
	■ El sistema elimina la cuenta del usuario y todos sus datos asociados
	■ El sistema devuelve al usuario a la pantalla inicial de la aplicación
Caminos Alternativos:	Contraseña incorrecta (2)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
	Operación cancelada (3)
	■ Se interrumpe la operación y se vuelve al punto de partida
Postcondición:	El usuario se habrá borrado de la aplicación.

Nombre:	Modificar Datos Personales
Descripción:	El usuario puede completar o cambiar ciertos aspectos de su
	perfil personal. Cuando un usuario se da de alta solamente
	tiene el nombre y el correo electrónico como datos personales.
	Con este caso de uso el usuario puede añadir otros elementos
	tales como: foto personal, descripción, estudios, etc.
Precondición:	Usuario autentificado
Parámetros:	Datos del usuario nuevos
Flujo de Ejecución:	
	1. El usuario accede al formulario de modificar datos
	2. El usuario cambia los datos de su perfil
	3. El sistema verifica los datos
	4. El sistema actualiza la información del usuario
Caminos Alternativos:	Datos incorrectos (2)
	• Se emite un mensaje informando al usuario y se vuelve al
	punto de partida
D 4 1: ''	
Postcondición:	Los datos del usuario se habrán actualizado en la aplicación.

Nombre:	Cambiar Contraseña
Descripción:	Si el usuario quiere cambiar la contraseña que usa, puedo
	hacerlo desde la aplicación. Simplemente por seguridad debe
	teclear la contraseña actual y dos veces la nueva contraseña. La
	nueva contraseña debe seguir el mismo formato que la antigua
	para asegurarse de que sea segura.
Precondición:	Usuario autentificado
Parámetros:	Contraseña Actual
	Contraseña Nueva
Flujo de Ejecución:	
	1. El usuario accede al formulario de cambiar contraseña
	2. El usuario proporciona sus datos personales:
	■ El usuario introduce su contraseña actual
	■ El usuario introduce la nueva contraseña
	■ El usuario introduce otra vez la nueva contraseña
	3. El sistema realiza una serie de verificaciones:
	■ El sistema comprueba que la contraseña es correcta
	■ El sistema comprueba que las contraseñas nuevas coinciden
	4. Si todas las comprobaciones tienen éxito la contraseña del usuario quedará actualizada
Caminos Alternativos:	Contraseña incorrecta (2)
	■ Se emite un mensaje informando al usuario y se vuelve al
	punto de partida
	Contraseñas diferentes (2)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
Postcondición:	La contraseña del usuario se habrá actualizado en la aplicación.

5.2.3.2. Gestión de Elementos del Globo

Nombre:	Crear Capa
Descripción:	El usuario puede crear una capa en la aplicación para gestionar
	de manera más eficiente el conjunto de elementos que puede
	visualizar en el globo terráqueo. Una capa es un contenedor que
	puede almacenar rutas, sitios de interés e incluso otras capas.
Precondición:	Usuario autentificado
Parámetros:	Nombre
	Padre
Flujo de Ejecución:	
	1. El usuario hace clic en Crear Capa
	2. El usuario introduce un nombre para la capa
	3. El usuario elige una capa padre para la nueva capa a crear
	4. La capa se habrá creado (sin elementos)
C : A14 .:	
Caminos Alternativos:	
Postcondición:	El usuario dispondrá de una capa más en su lista de elementos.

Nombre:	Visualizar Capa
Descripción:	Cuando el usuario seleccione una capa, todo el contenido de la
	misma se mostrará en el globo: rutas y sitios de interés. Si la
	capa tuviera otras capas dentro, también se mostraría su
	contenido. Posiblemente si los elementos son muchos, la carga
	de los elementos sería muy lenta, lo que obligaría a utilizar un
	sistema que los filtre usando algún criterio (altura, zona que se
	está viendo en ese momento, etc.)
Precondición:	Usuario autentificado
Parámetros:	Capa seleccionada
Flujo de Ejecución:	
	1. El usuario hace clic en una capa
	2. La aplicación carga desde la base de datos la información
	de la capa a partir del identificador
	3. La aplicación mostrará en el globo terráqueo todos los elementos de la capa:
	Cicircinos de la capa.
	 Las rutas almacenadas
	■ Los sitios de interés guardados
	 Si los hubiera, también se mostrarán el resto de elementos descriptivos de la ruta (modelos 3D, imágenes, vídeos, etc.)
Caminos Alternativos:	
Postcondición:	La capa seleccionada y sus elementos se mostrarán en el globo de la aplicación.

Nombre:	Modificar Capa
Descripción:	El usuario puede cambiar el contenido de las capas usando el
	ratón para arrastrar elementos de la lista incorporándolos o
	eliminándolos a la capa seleccionada por ejemplo. El contenido
	de la capa se verá siguiendo una estructura de árbol.
Precondición:	Usuario autentificado
Parámetros:	Capa seleccionada
Flujo de Ejecución:	
	1. El usuario hace clic en una capa
	2. El usuario puede alterar el contenido de la capa de diversas formas:
	■ Puede añadir, modificar o quitar rutas
	■ Puede añadir, modificar o quitar sitios de interés
	■ Puede añadir, modificar o quitar otras subcapas
	3. Una vez hecha la operación, la capa quedará actualizada con los cambios realizados
Caminos Alternativos:	
Postcondición:	La capa habrá visto como el número de sus elementos ha sido
	alterado o alguna de las propiedades de la misma ha cambiado.

Nombre:	Renombrar Capa
Descripción:	El usuario puede cambiar el nombre de la capa en cualquier
	momento. Con pulsar en el botón correspondiente la aplicación
	deja que el usuario puede escribir un nuevo nombre. La
	aplicación seleccionará completamente el viejo nombre para
	que el usuario puede escribirlo directamente.
Precondición:	Usuario autentificado
Parámetros:	Capa seleccionada
	Nuevo nombre
Flujo de Ejecución:	
	1. El usuario hace clic en una capa
	2. El usuario hace clic en el botón de Renombrar Capa
	3. El sistema queda en estado de espera hasta que el usuario
	introduce el nuevo nombre
	4. Tras introducir el nombre, el sistema comprueba si no
	existe otra capa del usuario con el mismo nombre
	existe otra capa dei usuario con ei mismo nombre
	5. Si el nombre es correcto se actualiza la capa en la base de
	datos del sistema
Caminos Alternativos:	Nombre en uso (4)
	■ Se emite un mensaje informando al usuario y se vuelve al
	punto de partida
	pariso do partido
Postcondición:	La capa se habrá actualizado con el nuevo nombre.

Nombre:	Eliminar Capa
Descripción:	Cuando un usuario decide borrar una capa existente, la
	aplicación se lo permitirá realizar teniendo en cuenta que los
	elementos que contenga la capa también se eliminarán. Al ser
	una acción destructiva, la aplicación solicitará al usuario que la
	confirme antes de realizarla.
Precondición:	Usuario autentificado
Parámetros:	Capa seleccionada
Flujo de Ejecución:	
	1. El usuario selecciona una capa de entre las disponibles
	2. El usuario hace clic en el botón de Eliminar Capa
	3. El sistema emite un mensaje preguntando al usuario si quiere continuar
	 En caso de respuesta afirmativa, el sistema eliminará dicha capa de la base de datos junto con el resto de elementos que la conforman Se informa al usuario de que la capa se ha eliminado
Caminos Alternativos:	Operación cancelada (3)
	■ Se interrumpe la operación y se vuelve al punto de partida
Postcondición:	La capa se habrá borrado de la lista de capas del usuario así como el resto de elementos que conforman la capa.

Nombre:	Visualizar Información
Descripción:	Cuando el usuario desee ver cierta información adicional de
	una capa, la aplicación se lo permite. Puede ver el nombre, la
	descripción que haya escrito, así como el número de elementos
	de cada tipo que hay en la capa.
Precondición:	Usuario autentificado
Parámetros:	Capa seleccionada
Flujo de Ejecución:	
	1. El usuario hace clic en una capa
	2. El usuario hace clic en Visualizar Información
	3. En el centro de la pantalla aparecerá un cuadro mostrando diversa información sobre la capa seleccionada:
	Nombre
	 Descripción
	■ Comentario
	 Número de elementos que almacena desglosado por tipos
Caminos Alternativos:	
Postcondición:	La aplicación habrá mostrado el cuadro de información al
	usuario y se quedará a la espera de que éste cierre dicho cuadro.

Nombre:	Desplegar Capa
Descripción:	El usuario haciendo clic en el botón "+" al lado del nombre de
	la capa podrá ver el contenido de la misma: otras capas, rutas
	y sitios de interés. El botón cambiará para mostrar un "-". Si el
	usuario vuelve a hacer clic se dejará de ver el contenido de la
	capa. Si la capa posee subcapas, también tendrán el símbolo
	"+".
Precondición:	Usuario autentificado
Parámetros:	Capa seleccionada
Flujo de Ejecución:	
	1. El usuario hace clic en el botón de desplegar capa
	2. La aplicación carga desde la base de datos la información
	de la capa a partir del identificador
	3. La aplicación mostrará los diversos elementos de la capa
	siguiendo una estructura de árbol.
Caminos Alternativos:	
Postcondición:	La aplicación habrá mostrado los elementos que contiene la
	capa.

Nombre:	Activar Capa
Descripción:	Cuando el usuario seleccione una capa, todo el contenido de la
	misma se mostrará en el globo: rutas y sitios de interés. Se
	hará de una sola vez. Si la capa tuviera otras capas dentro,
	también se mostraría su contenido.
Precondición:	Usuario autentificado
Parámetros:	Capa seleccionada
Flujo de Ejecución:	
	1. El usuario hace clic en Activar Capa
	2. Si el botón estaba desactivado, se hará lo siguiente:
	■ La aplicación cargará desde la base de datos la información de la capa a partir del identificador
	■ La aplicación mostrará los diversos elementos de la capa sobre el globo
	3. Si el botón estaba activado, se hará lo siguiente:
	• Se quitarán de la visualización del globo todos los elementos que pertenecen a la capa
Caminos Alternativos:	
Postcondición:	Los elementos de la capa se mostrarán u ocultarán en el globo
	terráqueo.

Nombre:	Ir a
Descripción:	Cuando el usuario haga uso de Ir a en una capa, la cámara que
	muestra el globo terráqueo, se desplazará desde la situación en
	la que se encuentre hasta la posición de alguno de los elementos
	que tenga la capa. El criterio para ello podría variar. Si no
	tuviera alguno, entonces la cámara no se movería.
Precondición:	Usuario autentificado
Parámetros:	Capa seleccionada
Flujo de Ejecución:	
	1. El usuario hace clic en una capa
	2. El usuario hace clic en "Ir a"
	3. La cámara se desplazará desde el punto actual hasta un
	punto concreto perteneciente a la capa (ya sea el punto
	del primer punto de interés u otro preestablecido)
Caminos Alternativos:	
Postcondición:	El globo terráqueo quedará centrado en un punto concreto de
	la capa.

Nombre:	Ver Ruta
Descripción:	Cuando el usuario seleccione una ruta de las que posee, ésta se mostrara en el globo terráqueo a través de una línea dibujada sobre el mismo. El usuario podrá cambiar la apariencia de la línea a través de varias características: color, grosor y modo de altura.
Precondición:	Usuario autentificado
Parámetros:	ID de la Ruta
Flujo de Ejecución:	 El usuario selecciona una ruta de entre las disponibles El sistema carga desde la base de datos los datos de la ruta a partir del identificador El sistema muestra las coordenadas de la ruta en el globo junto al resto de la información en los paneles correspondientes Si los hubiera, también se mostrarán el resto de elementos descriptivos de la ruta (modelos 3D, imágenes, vídeos, etc.)
Caminos Alternativos:	
Postcondición:	La ruta seleccionada se mostrará en el globo.

Nombre:	Ver Tour
Descripción:	El usuario puede visualizar una ruta de manera automática con
	la opción de ver tour. La cámara seguirá la trayectoria de la
	ruta de principio a fin simulando una perspectiva aérea y
	deteniéndose en los sitios de interés de la ruta si los hubiera.
Precondición:	El usuario debe estar identificado
	Ruta visualizada
Parámetros:	ID de la Ruta
Flujo de Ejecución:	
	1. El usuario selecciona una ruta de entre las disponibles
	2. El usuario hace clic en Ver Tour
	3. El sistema activará la opción de Ver Tour
	4. El sistema mostrará los controles de reproducción del Tour (Play, Pause, Stop)
	5. El usuario hace clic en Iniciar Tour
	6. El sistema comienza con la reproducción del Tour
Caminos Alternativos:	
Postcondición:	La ruta seleccionada se mostrará en el globo.

Nombre:	Ver Sitios de Interés
Descripción:	Cuando una ruta tenga sitios de interés y el usuario
	esté visualizando una ruta, éste podrá hacer clic en los sitios de
	interés que desee y la aplicación le mostrará al usuario
	información relevante sobre el sitio de interés seleccionado.
Precondición:	El usuario debe estar identificado
	Ruta mostrada
Parámetros:	ID de la Ruta
	IDs de los tipos de sitios de interés
Flujo de Ejecución:	
	1. El usuario selecciona una ruta de entre las disponibles
	2. El sistema muestra la ruta seleccionada en el globo
	3. El usuario elige de entre las diversas capas disponibles
	(monumentos, restaurantes, etc.) cuales quiere ver
	4. El sistema mostrará los sitios de interés correspondientes
	a las capas elegidas más cercanos a la ruta seleccionada
Caminos Alternativos:	
Postcondición:	Los sitios de interés de la ruta se mostrarán en el globo
	siguiendo el recorrido.

Nombre:	Selectionar Capas
Descripción:	El usuario puede cambiar qué tipos de sitios de interés quiere
	visualizar en cada momento. Ello hace que varíe cuales se
	muestran durante la visualización de las rutas.
Precondición:	El usuario debe estar identificado
Parámetros:	
Flujo de Ejecución:	
	1. El sistema muestra al usuario una lista con las capas disponibles
	2. El usuario elige una o varias de entre las que se muestran
Caminos Alternativos:	
Postcondición:	Las capas seleccionadas quedan asignadas en el sistema.

Nombre:	Crear Ruta
Descripción:	El usuario puede crear una ruta usando el ratón para dibujar
	una línea sobre el globo terráqueo de la aplicación. El estilo de
	la línea puede cambiarse. Cuando el usuario desee terminar de
	crear la línea, hará doble clic en globo terráqueo para indicar el
	punto final de la ruta. La forma de crear la línea con el ratón
	puede variar.
Precondición:	El usuario debe estar identificado
	Debe haber una capa seleccionada (si no, se usará la capa por
	defecto)
Parámetros:	
Flujo de Ejecución:	El usuario a través del ratón irá usando el ratón en el globo
	para ir dibujando la ruta que desee.
	1. El usuario hace clic en Crear Ruta
	1. El usuario nace che en Crear Ruta
	2. El usuario a través del ratón irá creando la ruta sobre el
	globo
	Haciendo clics sobre el globo generando segmentos
	de recta
	■ Manteniendo el botón del ratón pulsado
	3. Para finalizar la ruta hará doble clic en el punto final de
	la misma
	4. La ruta creada se añadirá a la capa seleccionada
Caminos Alternativos:	
Postcondición:	La ruta creada se muestra en el globo terráqueo.

Nombre:	Guardar Ruta
Descripción:	Cuando el usuario haya terminado de crear la ruta, la
	aplicación mostrará al usuario un cuadro para que especifique
	el nombre de la ruta y otros datos asociados.
Precondición:	El usuario debe estar identificado
	Ruta creada
Parámetros:	Nombre de la Ruta
	Puntos de la Ruta
	Capa
	Descripción
Flujo de Ejecución:	
	1. Tras crear la ruta, el sistema mostrará al usuario un
	formulario donde deberá introducir los datos necesarios
	para almacenar la ruta:
	■ Nombre
	■ Capa
	■ Descripción
	2. El usuario hará clic en el botón de Guardar Ruta
	3. Si los datos son correctos, la ruta creada se
	almacenará en el sistema
Caminos Alternativos:	Nombre en uso (1)
	` '
	Se emite un mensaje informando al usuario y se vuelve al
	punto de partida
Postcondición:	La ruta se habrá añadido a la lista de rutas del usuario.

Nombre:	Visualizar Ruta
Descripción:	Cuando el usuario seleccione una ruta de las que posee, ésta se
	mostrara en el globo terráqueo a través de una línea dibujada
	sobre el mismo. El usuario podrá cambiar la apariencia de la
	línea a través de varias características: color, grosor y modo de
	altura.
Precondición:	El usuario debe estar identificado
Parámetros:	Ruta seleccionada
Flujo de Ejecución:	
	1. El usuario selecciona una ruta de entre las disponibles
	2. El sistema carga desde la base de datos los datos de la ruta a partir del identificador
	3. El sistema muestra las coordenadas de la ruta en el globo junto al resto de la información en los paneles correspondientes
	4. Si los hubiera, también se mostrarán el resto de elementos descriptivos de la ruta (modelos 3D, imágenes, vídeos, etc.)
Caminos Alternativos:	
Postcondición:	La ruta seleccionada se mostrará en el globo.

Nombre:	Modificar Ruta
Descripción:	Si el usuario desea poder cambiar el recorrido de una ruta lo
	puede hacer arrastrando con el ratón los puntos de la ruta. La
	aplicación actualizará el conjunto de puntos de la ruta.
	También pueden modificarse otros metadatos de la ruta como
	la descripción, el tipo de altura, etc.
Precondición:	El usuario debe estar identificado
Parámetros:	Ruta seleccionada
	Puntos de la Ruta
Flujo de Ejecución:	
	1. El usuario selecciona una ruta de entre las disponibles
	2. El sistema carga desde la base de datos los datos de la ruta a partir del identificador
	3. El sistema muestra las coordenadas de la ruta en el globo junto al resto de la información en los paneles correspondientes
	4. Si los hubiera, también se mostrarán el resto de elementos descriptivos de la ruta (modelos 3D, imágenes, vídeos, etc.)
	5. El usuario puede ahora cambiar diferentes aspectos de la ruta:
	El usuario puede ahora con el ratón arrastrar los puntos que forman la ruta para modificarla
	■ El usuario puede añadir o eliminar los elementos que documentan la ruta
	6. En cualquiera de los casos el sistema guardará en la base de datos la ruta actualizada
Caminos Alternativos:	
Postcondición:	Los datos de la ruta se habrán actualizado.

Nombre:	Renombrar Ruta
Descripción:	El usuario puede cambiar el nombre de la ruta en cualquier
	momento. Con pulsar en el botón correspondiente, la aplicación
	deja que el usuario pueda escribir un nuevo nombre. La
	aplicación seleccionará completamente el viejo nombre para
	que el usuario puede escribirlo directamente.
Precondición:	El usuario debe estar identificado
Parámetros:	Ruta seleccionada
Flujo de Ejecución:	
	1. El usuario selecciona una ruta de entre las disponibles
	2. El usuario hace clic en el botón de Renombrar Ruta
	3. El sistema queda en estado de espera hasta que el usuario introduce el nuevo nombre
	4. Tras introducir el nombre, el sistema comprueba si no existe otra ruta del usuario con el mismo nombre
	5. Si el nombre es correcto se actualiza la ruta en la base de datos del sistema
Caminos Alternativos:	Nombre en uso (4)
	■ Se emite un mensaje informando al usuario y se vuelve al punto de partida
Postcondición:	La ruta estará actualizada con el nuevo nombre.

Nombre:	Eliminar Ruta
Descripción:	Cuando el usuario no desee utilizar más una ruta puede
	borrarla de la aplicación desde la opción destinada a tal efecto.
	La aplicación le solicitará al usuario confirmar la operación. La
	ruta será eliminada de la base de datos y no aparecerá más en
	la lista de rutas del usuario.
Precondición:	El usuario debe estar identificado
Parámetros:	Ruta seleccionada
Flujo de Ejecución:	
	1. El usuario selecciona una ruta de entre las disponibles
	2. El usuario hace clic en el botón de Eliminar Ruta
	3. El sistema emite un mensaje preguntando al usuario si quiere continuar
	■ En caso de respuesta afirmativa, el sistema
	eliminará dicha ruta de la base de datos
	■ Se informa al usuario de que la ruta se ha eliminado
Caminos Alternativos:	Operación cancelada (3)
	■ Se interrumpe la operación
Postcondición:	La ruta se habrá borrado de la lista de rutas del usuario.

Nombre:	Insertar Puntos
Descripción:	La inserción de puntos de a la hora de crear una ruta se
	hará con el ratón haciendo clic sobre el globo terráqueo. Con
	cada clic la aplicación irá pintando segmentos de recta
	simulando la ruta. También podrán añadirse puntos
	arrastrando el ratón sobre el globo terráqueo. Hay que
	seleccionar el método a usar o incluso usar los dos al mismo
	tiempo.
Precondición:	El usuario debe estar identificado
	El sistema debe estar en Crear Ruta
Parámetros:	
Flujo de Ejecución:	
	 El usuario haciendo uso del ratón irá haciendo clic sobre el globo para indicar los puntos que conforman la ruta A medida que el usuario va clicando sobre el globo, el sistema va guardando los puntos en una estructura de datos Cuando el usuario haga doble clic sobre el globo terminará la edición de la ruta
Caminos Alternativos:	
Postcondición:	Los puntos se habrán añadido a la ruta que se está creando.

Nombre:	Insertar Elementos
Descripción:	El usuario podrá añadir elementos con los que documentar las
	rutas de manera que se muestren a lo largo de la ruta por parte
	de la aplicación. El usuario hace clic en el botón
	correspondiente para insertar el elemento que desea y luego
	indica con el ratón donde colocarlo. Tras ello, la aplicación
	mostrará un cuadro para el usuario rellene los datos del
	elemento. Una vez hecho, el elemento se mostrará en el globo
	terráqueo.
Precondición:	El usuario debe estar identificado
Parámetros:	Ruta seleccionada
	Punto de la ruta
Flujo de Ejecución:	
	1. El usuario hace clic en Insertar Elemento
	2. El usuario hace clic con el ratón en el punto donde quiere
	colocar el elemento
	colocal el elemento
	3. El sistema muestra un formulario para insertar el
	elemento
	4. Se solicitará que indique la dirección del mismo
	■ Ya sea a través de un fichero local
	■ O esté almacenado en otro servicio web
	5. El elemento se mostrará en el globo
	6. El sistema actualizará la ruta en la base de datos con el
	elemento recién añadido
Caminos Alternativos:	Ruta Inválida (4)
	• Se emite un mensaje informando al usuario y se vuelve al
	punto de partida
Postcondición:	El elemento se habrá añadido a la ruta seleccionada.

Nombre:	Insertar Modelo 3D
Descripción:	El usuario podrá añadir modelos 3D con los que documentar
	las rutas de manera que se muestren a lo largo de la ruta por
	parte de la aplicación. El formato de los ficheros debe ser
	COLLADA. El usuario hace clic en el botón para insertar el
	modelo 3D que desea y luego indica con el ratón donde
	colocarlo. Tras ello, la aplicación mostrará un cuadro para el
	usuario rellene los datos del modelo 3D. Una vez hecho, el
	modelo 3D se mostrará en el globo terráqueo.
Precondición:	El usuario debe estar identificado
Parámetros:	Ruta seleccionada
	Punto de la ruta
Flujo de Ejecución:	
	1. Ver caso de uso "Insertar Elementos"
Caminos Alternativos:	Ruta Inválida (4)
	• Se emite un mensaje informando al usuario y se vuelve al
	punto de partida
Dogt oan diritir	El madala 2D sa babué añadida a la muta calcariar - J-
Postcondición:	El modelo 3D se habrá añadido a la ruta seleccionada.

Nombre:	Insertar Foto
Descripción:	El usuario podrá añadir fotos con los que documentar las rutas
	de manera que se muestren a lo largo de la ruta por parte de la
	aplicación. La aplicación reconocerá los formatos más usuales
	de imagen. El usuario hace clic en el botón para insertar la foto
	que desea y luego indica con el ratón donde colocarla. Tras ello,
	la aplicación mostrará un cuadro para el usuario rellene los
	datos de la foto. Una vez hecho, la foto se mostrará en el globo
	terráqueo.
Precondición:	El usuario debe estar identificado
Parámetros:	Ruta seleccionada
	Punto de la ruta
Flujo de Ejecución:	
	1. Ver caso de uso "Insertar Elementos"
Caminos Alternativos:	Ruta Inválida (4)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
Postcondición:	La foto se habrá añadido a la ruta seleccionada.

Nombre:	Insertar Vídeo
Descripción:	El usuario podrá añadir vídeos con los que documentar las
	rutas de manera que se muestren a lo largo de la ruta por parte
	de la aplicación. Los vídeos deben estar online. El usuario hace
	clic en el botón para insertar el vídeo que desea y luego indica
	con el ratón donde colocarlo. Tras ello, la aplicación
	mostrará un cuadro para el usuario rellene los datos del vídeo.
	Una vez hecho, el vídeo se mostrará en el globo terráqueo.
Precondición:	El usuario debe estar identificado
Parámetros:	Ruta seleccionada
	Punto de la ruta
Flujo de Ejecución:	
	1. Ver caso de uso "Insertar Elementos"
Caminos Alternativos:	Ruta Inválida (4)
	• Se emite un mensaje informando al usuario y se vuelve al
	punto de partida
Postcondición:	El vídeo se habrá añadido a la ruta seleccionada.

Nombre:	Insertar Documento
Descripción:	El usuario podrá añadir documentos con los que documentar
	las rutas de manera que se muestren a lo largo de la ruta por
	parte de la aplicación. En general será documentos de tipo
	texto (Word, PDF y otros) El usuario hace clic en el botón
	para insertar el documento que desea y luego indica con el
	ratón donde colocarlo. Tras ello, la aplicación mostrará un
	cuadro para el usuario rellene los datos del documento. Una
	vez hecho, el documento se mostrará en el globo terráqueo.
Precondición:	El usuario debe estar identificado
Parámetros:	Ruta seleccionada
	Punto de la ruta
Flujo de Ejecución:	
	1. Ver caso de uso "Insertar Elementos"
Caminos Alternativos:	Ruta Inválida (4)
	• Se emite un mensaje informando al usuario y se vuelve al punto de partida
Postcondición:	El documento se habrá añadido a la ruta seleccionada.

Nombre:	Añadir Sitio de Interés
Descripción:	El usuario puede añadir sitios de interés a sus rutas para
	disponer de mejor información sobre las rutas. Tras hacer clic
	en el botón de Añadir Sitio de Interés, el usuario
	especificará donde irá el sitio de interés haciendo clic con el
	ratón sobre el globo terráqueo. La aplicación mostrará un
	cuadro donde el usuario podrá rellenar los datos del sitio de
	interés, que se añadirá a la capa seleccionada.
Precondición:	Usuario autentificado
Parámetros:	Capa seleccionada
	Ruta seleccionada
	Punto del Sitio de Interés
	Datos del Sitio de Interés
Flujo de Ejecución:	
	1. El usuario hace clic en Añadir Sitio de Interés
	2. El usuario hace clic en el globo para indicar dónde
	estará el sitio de interés
	3. El sistema mostrará al usuario un formulario donde
	introducir los datos del sitio de interés:
	■ Nombre
	■ Descripción
	■ Tipo
	4. Tras aceptar, el sistema añadirá el sitio de interés a la
	base de datos, y aparecerá en la capa que tenga
	seleccionada el usuario
Caminos Alternativos:	
Postcondición:	El sitio de interés se habrá añadido a la lista de elementos de la
	capa actual del usuario.

Nombre:	Visualizar Sitio de Interés
Descripción:	Durante la visualización de una ruta o capa, si el usuario ha
	añadido previamente sitios de interés para estos elementos,
	dichos sitios de interés se mostrarán en el globo a través de un
	icono representativo. Si el usuario hace clic en uno de estos
	iconos, la aplicación cargará la información disponible para el
	sitio de interés seleccionado y la mostrará junto al icono.
Precondición:	Usuario autentificado
Parámetros:	Sitio de interés seleccionado
Flujo de Ejecución:	
	1. El usuario hace clic en un sitio de interés
	2. El sistema carga desde la base de datos toda la
	información almacenada sobre el sitio de interés indicado
	3. La información se muestra al usuario a través de un
	cuadro asociado al punto donde está el sitio de interés
Caminos Alternativos:	
Postcondición:	La aplicación habrá mostrado información relacionada con el
	sitio de interés indicado por el usuario.

Nombre:	Modificar Sitio de Interés
Descripción:	Si el usuario selecciona un sitio de interés de los que posee en
	su lista de elementos, y hace clic en Modificar Sitio de Interés,
	la aplicación le muestra un cuadro donde puede actualizar los
	datos referentes al sitio de interés seleccionado, o añadir la
	información faltante.
Precondición:	Usuario autentificado
Parámetros:	Sitio de interés seleccionado
Flujo de Ejecución:	
	1. El usuario hace clic en un sitio de interés
	2. El usuario hace clic en Modificar Sitio de Interés
	3. La aplicación mostrará un formulario con la información relativa al sitio de interés de la que dispone
	4. El usuario modificará la información que necesite y pulsará en Aceptar
	5. La aplicación actualizará la información del sitio de interés en la base de datos
Caminos Alternativos:	
Postcondición:	La aplicación habrá actualizado la información del sitio de
	interés especificado.

Nombre:	Renombrar Sitio de Interés
Descripción:	El usuario puede cambiar el nombre del sitio de interés a
	voluntad. Solo tiene que hace clic en el botón encargado de ello.
	Entonces la aplicación espera a que el usuario pueda escribir un
	nuevo nombre. La aplicación seleccionará completamente el
	viejo nombre para que el usuario puede escribirlo directamente.
Precondición:	Usuario autentificado
Parámetros:	Sitio de interés seleccionado
Flujo de Ejecución:	
	1. El usuario hace clic en un sitio de interés
	2. El usuario hace clic en el botón de Renombrar Sitio de
	Interés
	3. El sistema queda en estado de espera hasta que el usuario
	introduce el nuevo nombre
	4. Tras introducir el nombre, el sistema comprueba si no
	existe otro sitio de interés en la capa seleccionada del
	usuario con el mismo nombre
	5. Si el nombre es correcto se actualiza el sitio de interés en
	la base de datos del sistema
Caminos Alternativos:	Nombre en uso (4)
	■ Se emite un mensaje informando al usuario y se vuelve al
	punto de partida
Postcondición:	El sitio de interés se habrá actualizado con el nuevo nombre.

Nombre:	Eliminar Sitio de Interés
Descripción:	Si el usuario considera que un sitio de interés ya no es útil,
	puede eliminarlo de la aplicación seleccionándolo en la lista de
	elementos y haciendo clic en Eliminar Sitio de Interés. La
	aplicación le solicitará al usuario confirmar la operación. Tras
	ello, el sitio de interés se habrá borrado de la aplicación.
Precondición:	Usuario autentificado
Parámetros:	Sitio de interés seleccionado
Flujo de Ejecución:	
	1. El usuario selecciona un sitio de interés de entre los
	disponibles para la capa seleccionada
	2. El usuario hace clic en el botón de Eliminar Sitio de
	Interés
	3. El sistema emite un mensaje preguntando al usuario si quiere continuar
	■ En caso de respuesta afirmativa, el sistema eliminará dicho sitio de interés de la base de datos y de la capa seleccionada
	■ Se informa al usuario de que el sitio de interés se ha eliminado
Caminos Alternativos:	Operación cancelada (3)
	■ Se interrumpe la operación y se vuelve al punto de partida
Postcondición:	El sitio de interés se habrá borrado de la lista de elementos de la capa seleccionada del usuario.

Nombre:	Crear Tour Automático
Descripción:	El usuario puede dejar que la aplicación cree un tour de
	manera automática para la ruta seleccionada. Tras pulsar en
	Crear Tour Automático, la aplicación movería la cámara hasta
	el punto inicial de la ruta e iría pasando por el resto de puntos
	y sitios de interés hasta llegar hasta el final.
Precondición:	Usuario autentificado
Parámetros:	Ruta seleccionada
Flujo de Ejecución:	
	1. El usuario selecciona una ruta de entre las disponibles
	2. El usuario hace clic en Crear Tour Automático
	3. La aplicación comienza a recorrer el tour de manera automática:
	 Hace que la cámara pase por todos los puntos de la ruta desde el comienzo hasta el final Detiene la cámara en los sitios de interés durante un intervalo corto de tiempo y continua con el recorrido
Caminos Alternativos:	
Postcondición:	La ruta se muestra sobre el globo terráqueo.

Nombre:	Crear Tour Manual
Descripción:	Si el usuario decide crear un tour de forma manual, tras pulsar
	el botón para ello, la aplicación le mostrará los controles para
	ello. Uno de los botones dará comienzo a la grabación del tour,
	momento en el cual el usuario realizará las acciones que desee:
	movimientos de cámara o visualización de sitios de interés por
	ejemplo. La aplicación almacenará todos los movimientos
	realizados para dar forma al tour.
Precondición:	Usuario autentificado
Parámetros:	Movimientos de cámara
Flujo de Ejecución:	
	1. El usuario hace clic en Crear Tour Manual
	2. La aplicación muestra los controles dedicados para ello
	3. El usuario hace clic en Iniciar Grabación
	4. Mientras el usuario recorre el globo de manera manual, la aplicación irá recogiendo todas las operaciones realizadas
	5. Cuando el usuario termine el tour hará clic en Finalizar Grabación, donde la aplicación pedirá al usuario un nombre para guardar el tour
	6. Tras especificar el nombre y aceptar, el tour creado se añade a la lista de tours del usuario
Caminos Alternativos:	Operación cancelada (4)
	■ Se interrumpe la operación y se vuelve al punto de partida
	Operación cancelada (6)
	 Se interrumpe la operación y se vuelve al punto de partida
Postcondición:	El tour creado se añadirá a la lista de tours del usuario.

Nombre:	Visualizar Tour
Descripción:	Cuando el usuario haya seleccionado uno de los tours
	disponibles, y pulse en el botón de visualización de tour, la
	aplicación le mostrará los controles de reproducción dedicados
	para ello. El usuario hará clic en "Play" y empezará a verse el
	tour. Además hay controles de "Pause", "Rewind", "Forward"
	y "Stop" que dará por finalizado al tour.
Precondición:	Usuario autentificado
Parámetros:	Tour seleccionado
Flujo de Ejecución:	
	1. El usuario selecciona un tour de los que tiene disponibles
	en su lista
	9. Eli. h li Vili T
	2. El usuario hace clic en Visualizar Tour
	3. La aplicación muestra los controles de reproducción del
	tour
	4. El usuario hace clic en Iniciar Reproducción
	5. La aplicación comenzará a desplazar la cámara por los puntos del tour hasta llegar al último
	a) Opcionalmente, y si el tour dispone de dicha
	funcionalidad, empezará a sonar una guía
	describiendo las características del tour
	6. La aplicación sale del modo de Reproducir Tour
Caminos Alternativos:	Reproducción cancelada (5)
	■ El usuario hace clic en Detener Reproducción y se vuelve
	al punto de partida
	T arrest are P arrestage
Postcondición:	La reproducción del tour habrá finalizado, y la aplicación
	estará disponible para otras operaciones.

Nombre:	Modificar Tour
Descripción:	La modificación del tour permite modificar los metadatos
	asociados al tour seleccionado. Para ello, el usuario debe hacer
	clic en el botón Modificar Tour. La aplicación mostrará un
	cuadro donde el usuario puede cambiar diversos aspectos del
	tour. Sin embargo, no puede cambiar el recorrido del tour, para
	ello debería crear uno nuevo.
Precondición:	Usuario autentificado
Parámetros:	Tour seleccionado
Flujo de Ejecución:	
	1. El usuario selecciona un tour de los que tiene disponibles
	en su lista
	2. El usuario hace clic en Modificar Tour
	3. La aplicación muestra un formulario donde el usuario
	puede añadir diversa información sobre el tour:
	descripción, audioguía, etc.
	4. El usuario hace clic en Aceptar y el sistema guardará en
	la base de datos el tour actualizado
Caminos Alternativos:	
Postcondición:	El tour se habrá actualizado con los nuevos datos añadidos de
	la aplicación.

Nombre:	Renombrar Tour
Descripción:	El usuario puede cambiar el nombre del tour en cualquier
	momento. Con pulsar en el botón correspondiente la aplicación
	deja que el usuario puede escribir un nuevo nombre. La
	aplicación seleccionará completamente el viejo nombre para
	que el usuario puede escribirlo directamente.
Precondición:	Usuario autentificado
Parámetros:	Tour seleccionado
Flujo de Ejecución:	
	1. El usuario hace clic en un tour
	2. El usuario hace clic en el botón de Renombrar Tour
	3. El sistema queda en estado de espera hasta que el usuario
	introduce el nuevo nombre
	4. Tras introducir el nombre, el sistema comprueba si no existe otro tour del usuario con el mismo nombre
	5. Si el nombre es correcto se actualiza el tour en la base de
	datos del sistema
Caminos Alternativos:	Nombre en uso (4)
	■ Se emite un mensaje informando al usuario y se vuelve al
	punto de partida
Postcondición:	El tour se habrá actualizado con el nuevo nombre.

Nombre:	Eliminar Tour
Descripción:	Si un usuario no desea utilizar un tour creado con anterioridad,
	puede eliminarlo de la aplicación usando el botón destinado
	para ello tras seleccionar el tour deseado. La aplicación le
	solicitará al usuario confirmar la operación. Tras ello, el tour
	eliminado no aparecerá en la lista de tours del usuario.
Precondición:	Usuario autentificado
Parámetros:	Tour seleccionado
Flujo de Ejecución:	
	1. El usuario selecciona un tour de entre los disponibles
	2. El usuario hace clic en el botón de Eliminar Tour
	3. El sistema emite un mensaje preguntando al usuario si quiere continuar
	■ En caso de respuesta afirmativa, el sistema
	eliminará dicho tour de la base de datos
	• Se informa al usuario de que el tour se ha eliminado
Caminos Alternativos:	Operación cancelada (3)
	■ Se interrumpe la operación y se vuelve al punto de partida
Postcondición:	El tour se habrá borrado de la lista de tours del usuario.

5.2.3.3. Web 2.0

Nombre:	Compartir Rutas
Descripción:	El usuario puede compartir las rutas que posea con otros
	usuarios que tenga agregados a través de la aplicación. Para
	ello, el usuario debe seleccionar la ruta que desee y pulsar en el
	botón Compartir Ruta. La aplicación le pedirá al usuario el
	destinatario de la ruta. Una vez elegido, la ruta se enviará al
	usuario de destino.
Precondición:	Usuario autentificado
	Que el usuario tenga al menos un usuario agregado
	Que el usuario tenga al menos una ruta guardada
Parámetros:	Usuario de destino
	Ruta seleccionada
Flujo de Ejecución:	
	1. El usuario hace clic en el nombre de una las rutas que
	posee
	2. El usuario hace clic en Compartir Ruta
	3. La aplicación muestra al usuario la lista de usuarios disponibles
	4. El usuario selecciona uno de la lista y hace clic en Enviar
	5. La ruta seleccionada aparecerá en la lista de rutas compartidas del otro usuario
Caminos Alternativos:	Operación cancelada (4)
	■ Se interrumpe la operación y se vuelve al punto de partida
Postcondición:	El usuario de destino habrá recibido la ruta enviada por el
	usuario de la aplicación.

Nombre:	Compartir Archivos
Descripción:	De forma similar al caso de uso anterior, el usuario puede
	enviar archivos a otros usuarios que tenga agregados. Para
	conseguirlo debe seleccionar el fichero que quiere compartir
	–fotos, vídeos, modelos 3D o documentos– y pulsar en
	Compartir Archivo. La aplicación le pedirá al usuario el
	destinatario del archivo. Una vez elegido, el archivo se
	enviará al usuario de destino.
Precondición:	Usuario autentificado
	Que el usuario tenga al menos un usuario agregado
	Que el usuario tenga al menos un archivo guardado
Parámetros:	Usuario de destino
	Archivo seleccionado
Flujo de Ejecución:	
	1. El usuario hace clic en alguno de los archivos que posee:
	fotos, vídeos, modelos 3D o documentos; dentro de sus
	listas respectivas
	2. El usuario hace clic en Compartir
	3. La aplicación muestra al usuario la lista de usuarios
	disponibles
	4. El usuario selecciona uno de la lista y hace clic en Enviar
	5. El archivo seleccionado aparecerá en la lista de rutas
	compartidas del otro usuario
Caminos Alternativos:	Operación cancelada (4)
	■ Se interrumpe la operación y se vuelve al punto de partida
Postcondición:	El usuario de destino habrá recibido el archivo enviado por el
	usuario de la aplicación.

Nombre:	Rutas Favoritas
Descripción:	El usuario puede categorizar las rutas que posea, ya sean propias o compartidas, utilizando una marca de "Ruta Favorita". Ello le permite acceder de manera más rápida y
	accesible a las rutas que use con más frecuencia. Estas rutas aparecerán en una lista aparte.
Precondición:	Usuario autentificado
Parámetros:	Ruta seleccionada
Flujo de Ejecución:	 El usuario selecciona una de las rutas de las que tiene disponibles, ya sean propias o compartidas por parte de otros usuarios El hace clic en Marcar como Ruta Favorita La ruta seleccionada se añadirá a la lista de rutas favoritas
Caminos Alternativos:	
Postcondición:	La ruta seleccionada se habrá añadido a la lista de rutas favoritas.

Nombre:	Presentar Otras Rutas
Descripción:	Cuando el usuario se identifique en la aplicación y acceda a la
	pantalla inicial, si así lo tiene configurado, la aplicación le
	mostrará rutas de los usuarios que tenga agregados en base a
	diferentes criterios, ya sean temporales, categorías u otros. El
	usuario podrá visualizar las rutas presentadas, y si así lo desea,
	puede añadirlas a su lista de rutas.
Precondición:	Usuario autentificado
Parámetros:	
Flujo de Ejecución:	
	1. El usuario se autentifica en la aplicación
	2. La aplicación muestra en una lista de la pantalla inicial
	del usuario un conjunto de rutas de otros usuarios que
	tenga éste agregados. Para elegir las rutas se pueden
	elegir diversos criterios
	3. El usuario puede visualizar las rutas mostradas desde
	dicha lista, así como añadirlas a la lista de rutas
	compartidas
Caminos Alternativos:	
Postcondición:	La aplicación habrá mostrado el conjunto de rutas de otros
	usuarios al usuario de la aplicación.

Nombre:	Chatear
Descripción:	Para iniciar una conversación con otro usuario, el usuario de la
	aplicación debe seleccionarlo de la lista de usuarios agregados y
	pulsar en la opción de chat. Para ello, la aplicación mostrará el
	estado de dichos usuarios indicando si están conectados o no. Si
	el usuario seleccionado está conectado la aplicación abrirá otra
	ventana donde se podrá seguir el curso de la conversación. La
	ventana estará dividida en dos partes: una superior donde se
	mostrarán los mensajes que hayan escrito los usuarios junto
	con el avatar del autor del mensaje y una inferior donde
	habrá un cuadro de texto donde el usuario podrá escribir un
	nuevo mensaje.
Precondición:	Usuario autentificado
	Usuario de destino conectado
Parámetros:	Usuario de destino seleccionado
Flujo de Ejecución:	
	1. El usuario selecciona un usuario de los que tiene
	disponibles (conectados) en la lista
	2. El usuario hace clic en Chat
	3. La aplicación muestra en una ventana aparte la
	conversación de los dos usuarios
	4. El usuario teclea un mensaje y tras pulsar Intro, el
	mensaje aparece en la ventana
	5. El mensaje llegará al otro usuario que también lo verá en
	su propia ventana
Caminos Alternativos:	Conversación cancelada
	■ En cualquier momento cualquiera de los dos usuarios
	puede finalizar la conversación y ambos vuelven al punto
	de partida
Postcondición:	La conversación habrá finalizado y la aplicación
	estará disponible para realizar otras tareas.

Nombre:	Enviar Email
Descripción:	Para enviar un correo electrónico a otro usuario, el usuario de
	la aplicación necesitará seleccionarlo de la lista y pulsar en el
	botón de Enviar Email. Tras ello, la aplicación mostrará un
	cuadro donde se podrá indicar los elementos del correo,
	principalmente asunto y cuerpo del mensaje. Tras rellenarlos,
	la aplicación enviará el correo al usuario de destino.
Precondición:	Usuario autentificado
	Que el usuario tenga al menos un usuario agregado
Parámetros:	Usuario de destino
	Asunto del mensaje
	Cuerpo del mensaje
Flujo de Ejecución:	
	1. El usuario selecciona uno de los usuarios que tiene
	agregados de la lista
	2. El usuario hace clic en Enviar Correo
	3. La aplicación muestra un formulario donde el usuario
	puede indicar el asunto y escribir el contenido del correo
	4. El usuario hace clic en Enviar
	5. El correo escrito se manda al destinatario
Caminos Alternativos:	Operación cancelada (3)
	■ Se interrumpe la operación y se vuelve al punto de partida
Postcondición:	La aplicación habrá enviado el correo al usuario indicado.

Nombre:	Comentarios
Descripción:	Los usuarios de la aplicación pueden ver los comentarios que
	han dejado otros usuarios agregados sobre rutas compartidas.
	Para ello, deben seleccionar la ruta y pulsar en Comentarios.
	La aplicación mostrará la lista de comentarios que han dejado
	los usuarios, junto con sus avatares. Además en la parte
	inferior habrá un cuadro de texto para que el usuario pueda
	añadir el suyo.
Precondición:	Usuario autentificado
Parámetros:	Ruta seleccionada
Flujo de Ejecución:	
	1. El usuario selecciona una de las rutas compartidas que
	posee
	2. El usuario hace clic en Comentarios
	3. La aplicación muestra la lista de comentarios existente
	para la ruta seleccionada por parte de otros usuarios que
	tienen también la ruta seleccionada
	4. El usuario escribe su comentario en el cuadro de texto
	destinado a tal efecto y pulsa en Aceptar
	describado a um crecuo y parsa en riceptar
	5. El usuario verá como su comentario se ha añadido a los
	existentes
Caminos Alternativos:	Operación cancelada (4)
	■ Se interrumpe la operación y se vuelve al punto de
	partida
	F 3-2 3-3-3-3
Postcondición:	La ruta compartida tendrá un comentario más por parte del
	usuario de la aplicación.

Nombre:	Crear Comunidades
Descripción:	El usuario de la aplicación puede crear una comunidad donde
	añadir los usuarios que él decida tras petición. Los usuarios de
	la comunidad podrán acceder a los recursos compartidos con
	mayor facilidad a través de páginas destinadas a ello
	específicamente y compartir información.
Precondición:	Usuario autentificado
Parámetros:	Nombre
	Descripción
Flujo de Ejecución:	
	1. El usuario hace clic en Crear Comunidad desde la interfaz
	de la aplicación
	2. La aplicación le pedirá al usuario una serie de datos sobre
	la comunidad:
	Nombre
	 Descripción
	3. Tras rellenar los datos y pulsar en Aceptar, la comunidad
	estará creada
Caminos Alternativos:	Operación cancelada (3)
	■ Se interrumpe la operación y se vuelve al punto de
	partida
Postcondición:	El usuario habrá creado una comunidad de usuarios dentro de
	la aplicación.

Nombre:	Comunicarse con Otras Redes Sociales
Descripción:	El usuario de la aplicación podrá dejar mensajes en otros
	servicios web usando las facilidades que proporciona el sistema
	para ello. Tras pulsar en el botón del servicio web deseado, la
	aplicación mostrará una ventana donde el usuario
	podrá escribir el mensaje que quiere publicar. Tras hacer clic
	en Aceptar, la aplicación publicará el mensaje. El usuario
	tendrá que aportar previamente los datos de su cuenta para el
	servicio seleccionado.
Precondición:	Usuario autentificado
Parámetros:	Mensaje
	Datos de identificación de la red social
Flujo de Ejecución:	
	1. El usuario hace clic en el botón del servicio web o red
	social
	2. El vavorio escribo el mengo io que desse y pulso en
	2. El usuario escribe el mensaje que desee y pulsa en
	Aceptar
	3. La aplicación envía el mensaje para que sea publicado
	por el servicio elegido
	4. El navegador mostrará el servicio web en otra pestaña
	4. Di navegadoi mostrara el servicio web en otra pestana
Caminos Alternativos:	Operación cancelada (2)
	()
	■ Se interrumpe la operación y se vuelve al punto de
	partida
	Usuario no autentificado (3)
	■ Si el usuario no está autentificado en el servicio, se le
	pedirá que lo haga
	I
Postcondición:	El usuario habrá publicado el contenido en el servicio web
	escogido.
	TET 10 TET

Nombre:	Añadir Usuario
Descripción:	Para agregar un usuario a su lista de contactos, el usuario de la aplicación deberá buscarlo primero en la aplicación. La búsqueda puede hacerse usando diversos campos: nombre, email u otros. Una vez localizado, deberá enviarle una petición al usuario, que si éste acepta pasará a ser un contacto del usuario en la aplicación.
Precondición:	Usuario autentificado
Parámetros:	Nombre
T ditalifector.	Email
Flujo de Ejecución:	 El usuario hace clic en Añadir Usuario La aplicación muestra al usuario un formulario para que el usuario pueda buscar al usuario que desea añadir usando diversas formas: Por Nombre Por Email Si se encuentra al usuario, la aplicación mandará una solicitud a éste indicando que el usuario de la aplicación desea añadirlo a su lista de contactos Si el usuario buscado acepta la petición, será añadido a la lista de contactos del usuario de la aplicación
Caminos Alternativos:	Operación cancelada (2) Se interrumpe la operación y se vuelve al punto de partida Usuario no encontrado (3) Se interrumpe la operación y se vuelve al punto de partida Respuesta negativa (4) Se interrumpe la operación y se vuelve al punto de partida
Postcondición:	El usuario habrá añadido a otro usuario de la aplicación a su
	lista.

Nombre:	Ver Usuario
Descripción:	El usuario de la aplicación selecciona un usuario de su lista de
	contactos y pulsa en Ver Usuario. La aplicación le mostrará los
	datos del usuario seleccionado en una ventana aparte.
Precondición:	Usuario autentificado
Parámetros:	Usuario seleccionado
Flujo de Ejecución:	
	 El usuario selecciona uno de los usuarios que tiene agregados de la lista El usuario hace clic en Ver Usuario La aplicación muestra los datos del usuario seleccionado
Caminos Alternativos:	
Postcondición:	La aplicación habrá mostrado los datos del usuario
	seleccionado.

Nombre:	Modificar Usuario
Descripción:	El usuario de la aplicación selecciona un usuario de su lista de
	contactos y pulsa en Ver Usuario. La aplicación le mostrará los
	datos del usuario seleccionado en una ventana aparte. En dicha
	ventana el usuario puede modificar ciertos datos que aparecen
	para actualizarlos.
Precondición:	Usuario autentificado
Parámetros:	Usuario seleccionado
	Datos del usuario
Flujo de Ejecución:	
	1. El usuario selecciona uno de los usuarios que tiene
	agregados de la lista
	2. El usuario hace clic en Ver Usuario
	3. La aplicación muestra los datos del usuario seleccionado
	4. El usuario modifica los datos que desee y pulsa en Aceptar
	5. La aplicación actualiza los datos del usuario seleccionado.
Caminos Alternativos:	Operación cancelada (4)
	■ Se interrumpe la operación y se vuelve al punto de
	partida
Postcondición:	El usuario habrá modificado los datos que posee del usuario
	seleccionado en la aplicación.

Nombre:	Eliminar Usuario
Descripción:	Para eliminar a un contacto, basta con que el usuario de la
	aplicación lo seleccione de su lista de usuarios y pulse en
	Eliminar Usuario. La aplicación le solicitará al usuario que
	confirme la acción. Si el usuario acepta, el contacto
	desaparecerá de la lista de contactos.
Precondición:	Usuario autentificado
Parámetros:	Usuario seleccionado
Flujo de Ejecución:	
	1. El usuario selecciona un usuario de entre los disponibles
	2. El usuario hace clic en el botón de Eliminar Usuario
	3. El sistema emite un mensaje preguntando al usuario si
	quiere continuar
	■ En caso de respuesta afirmativa, el sistema
	eliminará dicho usuario de la base de datos
	■ Se informa al usuario de que el contacto se ha
	eliminado
Caminos Alternativos:	Operación cancelada (3)
	• Se interrumpe la operación y se vuelve al punto de
	partida
Postcondición:	El usuario se habrá borrado de la lista de contactos del usuario.

5.2.4. Interfaz de usuario

En esta sección se describirá la futura interfaz de usuario de la aplicación, tanto a nivel lógico como físico, además de los elementos que la componen. Esta parte del desarrollo del software es de vital importancia, pues es la responsable de la manera en que el usuario interactúa con la aplicación: ya sea durante la entrada o salida de datos, la forma de operar con los mismos, la generación de flujos de trabajo u otros.

5.2.4.1. Construcción de un Prototipo para la Interfaz de Usuario

El prototipado de la interfaz de usuario es otro elemento a considerar en la etapa de captura de requisitos. Ayuda a comprender y esbozar como el usuario interactuará con el sistema manejando los elementos que le son comunes del modelo de negocio.

Este proceso se realiza partiendo de los casos de uso para cada usuario del sistema. Para cada caso de uso se identifican los elementos del modelo de negocio básicos y cómo deben representarse éstos en la aplicación. Esta etapa se divide en dos partes: primero se realiza un diseño lógico del prototipo conteniendo los objetos necesarios para la interacción y a partir

de él se crea su equivalente físico, normalmente en forma de ventanas de aplicación.

Al finalizar este proceso por parte de los diseñadores de la interfaz, se dispone de un conjunto de esquemas y prototipos de interfaces de usuario que describen la apariencia de la futura aplicación de cara a los usuarios de la misma. Deben asegurarse que cada caso de uso es accesible a través de los elementos de la interfaz y de que el conjunto de casos de uso accesibles para cada usuario tenga una interfaz de usuario bien integrada, fácil de utilizar y consistente.

5.2.4.2. Prototipo de la interfaz de usuario

Para este proyecto tenemos dos tipos de usuario, tal y como se indica en la especificación de casos de uso, y para cada uno de ellos se muestra el prototipo de la interfaz correspondiente:

Usuario No Registrado

Iniciar Sesión (Pantalla Inicial)

Diseño Lógico Los elementos de la interfaz de usuario necesarios en este caso son:

- Nombre de Usuario: Nombre del usuario registrado en la aplicación (TextBox)
- Contraseña: Contraseña con la que el usuario se autentifica en el sistema (TextBox)

Diseño Físico La pantalla inicial de la aplicación estará dividida en tres áreas: una superior con el logotipo de la aplicación a la izquierda y la información necesaria para identificarse a la derecha. La parte central, que será la más grande, contendrá un texto explicativo describiendo la funcionalidad de la aplicación y a la derecha una imagen representativa del ámbito de la misma. Debajo de ésta habrá un enlace hacia la página de registro para el nuevo usuario. Por último, en la parte inferior estará un enlace hacia la información de contacto.

En cuanto a la información para identificarse, los dos cuadros de texto dedicados al nombre de usuario y la contraseña se colocarán de manera horizontal, alineados con el botón encargado de realizar el proceso. Debajo de los cuadros de texto estará colocado un enlace hacia la página de recordar contraseña.

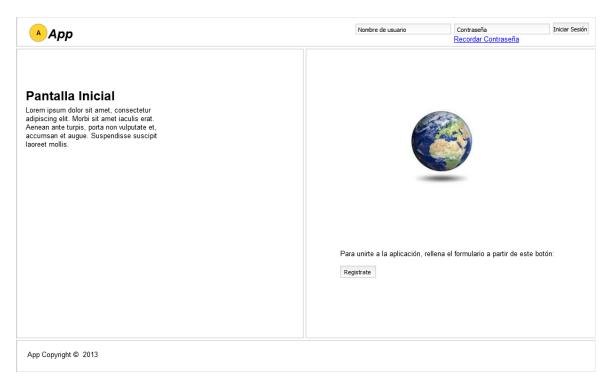


Figura 5.30: Pantalla Inicial

Registro

Diseño Lógico Los elementos de la interfaz de usuario necesarios en este caso son:

- Nombre de Usuario: Nombre del usuario registrado en la aplicación (TextBox)
- Contraseña 1: Contraseña con la que el usuario se autentificará en el sistema (TextBox)
- Contraseña 2: Debe ser igual que Contraseña 1 (TextBox)
- Correo Electrónico: El sistema se comunicará con el usuario a través de la dirección dada por éste. (TextBox)

Diseño Físico Esta página es similar a la anterior, solo cambia la parte central, donde a la izquierda se colocará un texto explicativo sobre la información de registro y a la derecha los campos del formulario a rellenar por parte del usuario. Éstos se dispondrán verticalmente, empezando por el nombre, a continuación los dos para la contraseña, después el del correo electrónico, y por último un cuadro para mostrar la licencia y el botón encargado de realizar el proceso.

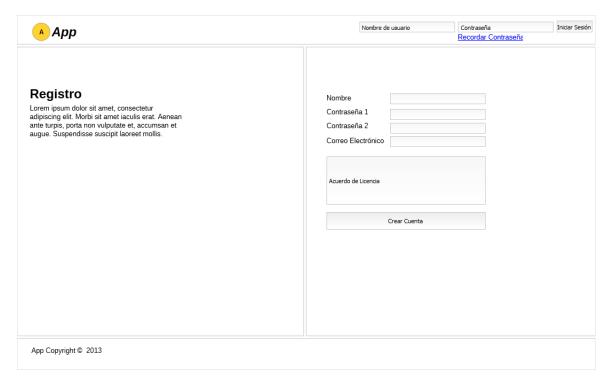


Figura 5.31: Registro

Recordar Contraseña

Diseño Lógico Los elementos de la interfaz de usuario necesarios en este caso son:

 Correo Electrónico: El sistema se comunicará con el usuario a través de la dirección dada por éste. (TextBox)

Diseño Físico Esta página es similar a la anterior, solo cambia la parte central, donde a la izquierda se colocará un texto explicativo sobre la información relativa al proceso de recordar contraseña y a la derecha los campos del formulario a rellenar por parte del usuario. Éstos se dispondrán verticalmente, empezando por un texto explicativo, a continuación el del correo electrónico, y por último un botón encargado de realizar el proceso.

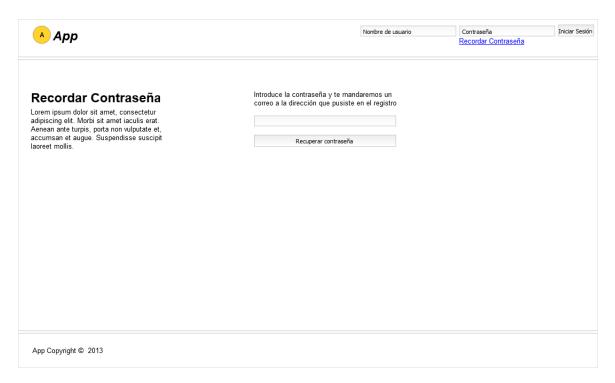


Figura 5.32: Recordar Contraseña

Usuario Registrado

Pantalla Inicial

- Ruta: Conjunto de puntos que forman un recorrido sobre el globo terráqueo (Icono, Línea)
- Sitio de Interés: Punto sobre el globo terráqueo que representa un lugar que puede ser útil o de provecho al usuario (Icono, Modelo 3D)
- Capa: Contenedor que almacena rutas y sitios de interés (Icono, Lista de Elementos)
- Tour: Animación pregrabada que permite al usuario recorrer una ruta sin su intervención directa por ejemplo (Icono)
- Contacto: Persona con la que el usuario mantiene una conexión y con la que puede interactuar (Icono)
- Archivo (Foto, Vídeo, Documento, Modelo 3D): Fichero que permite documentar las rutas (Icono, Objeto 3D)
- Globo terráqueo: Representación del mundo y donde el usuario podrá ver sus rutas y otros elementos (Objeto 3D)

Diseño Físico Ésta es la pantalla inicial que se encontrará el usuario una vez identificado en el sistema. Se divide en dos partes: una superior y otra central. En la superior estará el logotipo de la aplicación a la izquierda y en la parte derecha una zona dedicada al usuario con tres elementos. El primero será un enlace con el nombre del usuario que le llevará a su página personal; el segundo, un botón que al pulsarlo desplegará una lista con tres opciones: Modificar Datos Personales, Cambiar Contraseña y Cerrar Cuenta. El tercero será un enlace para cerrar la sesión del usuario. Estos tres elementos estarán dispuestos de manera horizontal y alineados entre sí.

En la parte central habrán dos áreas principales. La izquierda estará segmentada en tres partes: Elementos del Globo, Contactos y Archivos. Estas partes se pueden solapar para mostrar más contenido de alguna de las partes ocultando las otras. En Elementos del Globo habrá dos pestañas, una para las rutas favoritas, y otra para el resto de elementos (rutas, capas, sitios de interés y tours). Dichos elementos estarán organizados jerárquicamente en forma de árbol. En Contactos estará la lista de amigos que habrá agregado el usuario junto con un icono al lado del nombre para que el usuario pueda saber si están conectados o no. Además estarán los botones que permitirán interactuar con otras redes sociales y servicios web. En Archivos estarán los ficheros que el usuario haya subido a la aplicación, en forma de lista y disponibles para su visualización y manipulación. Tanto la parte de Elementos del Globo como la de Archivos tendrán un icono situado en la esquina superior derecha que permitirá compartir dichos objetos con otros contactos del usuario. El área de la derecha tendrá una barra superior con botones para crear nuevos elementos del globo, como para el tour hay dos opciones el botón designado para ello dejándolo pulsado. Debajo de la barra estará el plugin del Google Earth con la representación del globo terráqueo.

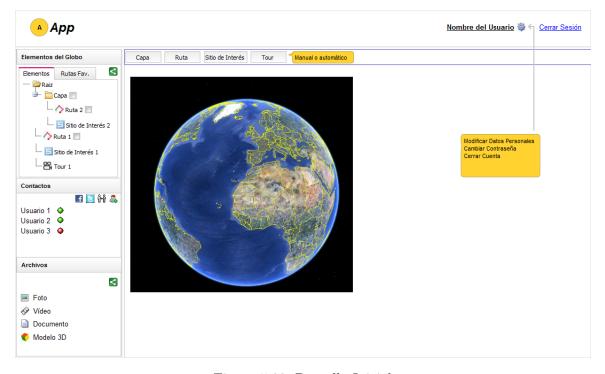


Figura 5.33: Pantalla Inicial

Darse de baja

Diseño Lógico Este caso de uso no requiere de atributos.

Diseño Físico Esta página se divide en tres partes: una superior, otra central y una última inferior. En la superior estará el logotipo de la aplicación a la izquierda y en la parte derecha la zona dedicada al usuario con los mismos tres elementos explicados en la sección anterior. La inferior tampoco se ve modificada con respecto a la pantalla de registro por ejemplo. La parte central se verá modificada de manera homogénea: a la izquierda se colocará un texto explicativo sobre la información relativa al proceso de darse de baja y a la derecha los campos del formulario a rellenar por parte del usuario. Éstos se dispondrán verticalmente, empezando por una imagen alegórica, y por último los botones de cancelar y el encargado de realizar el proceso.

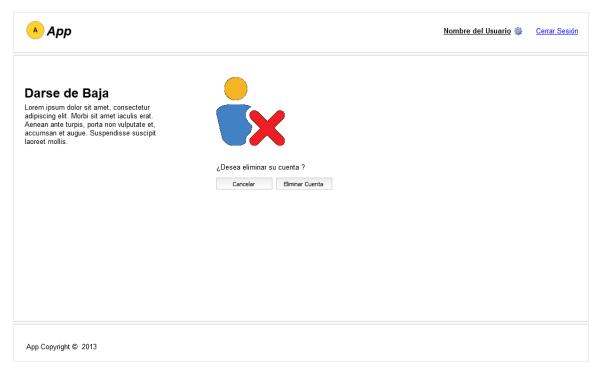


Figura 5.34: Darse de Baja

Modificar Datos Personales

- Descripción: Descripción personal del usuario (RichTextBox)
- URL: Dirección Web que el usuario disponga para su uso propio (Textbox)
- Nombre: Nombre del usuario (Textbox)
- Apellidos: Apellidos del usuario (Textbox)
- Estudios: Estudios del usuario (Textbox)

Trabajo: Trabajo del usuario (Textbox)

■ Intereses: Aficiones del usuario (Textbox)

Diseño Físico Esta página se divide en tres partes: una superior, otra central y una última inferior. En la superior estará el logotipo de la aplicación a la izquierda y en la parte derecha la zona dedicada al usuario con los mismos tres elementos explicados previamente. La inferior tampoco se ve modificada con respecto a la pantalla de registro por ejemplo. La parte central se verá modificada de manera homogénea: a la izquierda se colocará un texto explicativo sobre la información relativa al proceso de configuración de perfil y a la derecha los campos del formulario a rellenar por parte del usuario. Éstos se dispondrán verticalmente en dos columnas, empezando por una imagen del usuario a modo de avatar (que podrá cambiar con un enlace colocado al lado), y siguiendo con todos los campos relativos al usuario.

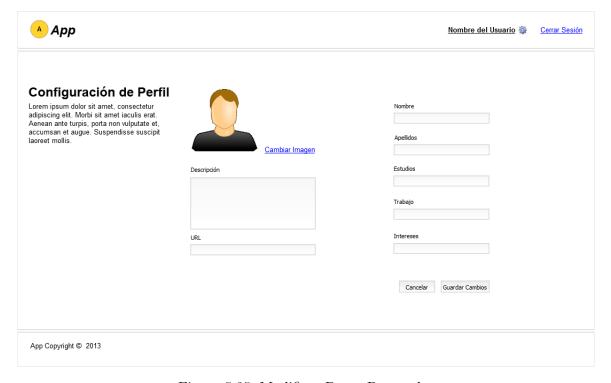


Figura 5.35: Modificar Datos Personales

Modificar Contraseña

- Contraseña Antigua: Contraseña con la que el usuario se autentifica en el sistema hasta el momento (Textbox)
- Contraseña Nueva 1: Contraseña con la que el usuario se autentificará en el sistema (Textbox)
- Contraseña Nueva 2: Debe ser igual que Contraseña Nueva 1 (Textbox)

Diseño Físico Esta página se divide en tres partes: una superior, otra central y una última inferior. En la superior estará el logotipo de la aplicación a la izquierda y en la parte derecha la zona dedicada al usuario con los mismos tres elementos explicados previamente. La inferior tampoco se ve modificada con respecto a la pantalla de registro por ejemplo. La parte central se verá modificada de manera homogénea: a la izquierda se colocará un texto explicativo sobre la información relativa al proceso de cambiar contraseña y a la derecha los campos del formulario a rellenar por parte del usuario. Éstos se dispondrán verticalmente, consistiendo en tres campos de texto, uno para la contraseña antigua y dos para la nueva, además del botón encargado de realizar el proceso.

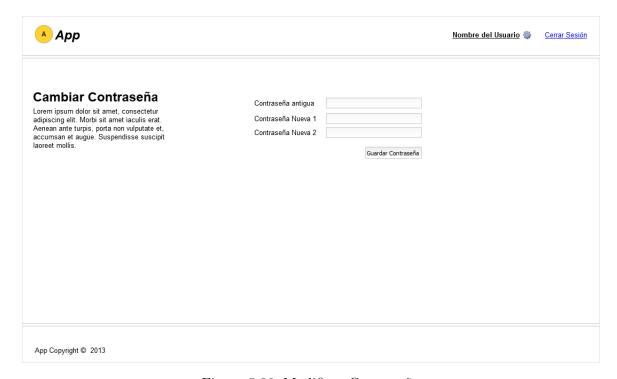


Figura 5.36: Modificar Contraseña

Información del Usuario

- Descripción: Descripción personal del usuario (RichTextBox)
- URL: Dirección Web que el usuario disponga para su uso propio (Textbox)
- Nombre: Nombre del usuario (Textbox)
- Apellidos: Apellidos del usuario (Textbox)
- Estudios: Estudios del usuario (Textbox)
- Trabajo: Trabajo del usuario (Textbox)
- Intereses: Aficiones del usuario (Textbox)

Diseño Físico Esta página se divide en tres partes: una superior, otra central y una última inferior. En la superior estará el logotipo de la aplicación a la izquierda y en la parte derecha la zona dedicada al usuario con los mismos tres elementos explicados previamente. La inferior tampoco se ve modificada con respecto a la pantalla de registro por ejemplo. La parte central se verá modificada de manera homogénea: a la izquierda se colocará un texto explicativo y a la derecha la información del usuario. Ésta se dispondrá verticalmente en dos columnas, empezando por una imagen del usuario a modo de avatar, y siguiendo con todos los campos relativos al usuario.

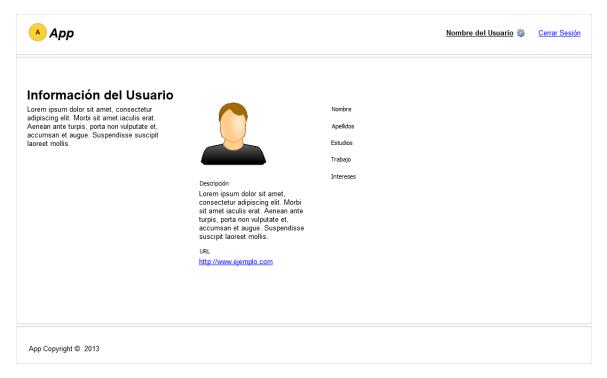


Figura 5.37: Información del Usuario

Los prototipos de todas las pantallas de la aplicación se han realizado con la aplicación Pencil¹, que es software libre.

5.3. Modelo de análisis

El análisis se desarrollará sobre las tablas de flujo de sucesos descritas durante los requisitos. Primero se presentará una visión general de la organización de la aplicación, con los diferentes paquetes identificados. A continuación se realizarán los diagramas de clase para identificar los elementos estructurales que intervienen en el caso de uso especificado. Posteriormente se obtendrá el diagrama de colaboración, en el que quedará reflejado cómo interactúan estas entidades, tanto en la forma de comunicarse (mensajes que se envían), como en la temporización de la interacción (orden en el que se realiza cada paso de la comunicación).

¹http://pencil.evolus.vn/

5.3.1. Organización del sistema

La organización de los paquetes del sistema y las dependencias entre ellos se muestra a continuación:

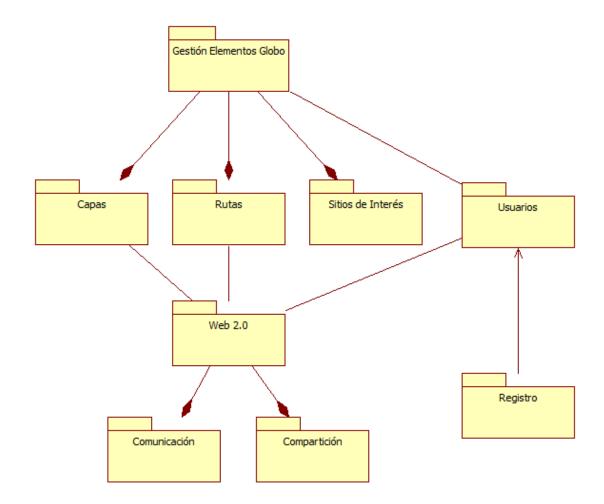


Figura 5.38: Arquitectura de paquetes

El paquete Gestión Elementos Globo engloba las operaciones a nivel general sobre los objetos que el usuario puede manejar en el globo terráqueo. Este paquete se descompone a su vez en otros tres: Capas, Rutas y Sitios de Interés. Cada uno de ellos es el encargado de realizar las operaciones básicas sobre cada elemento del globo en concreto; ya sea creación, modificación u otras.

El paquete Web 2.0 se compone de dos paquetes: Comunicación y Compartición. El primero es el encargado de posibilitar que los usuarios de la aplicación puedan mantener diferentes vías de conversación y traspaso de información. El paquete Compartición es el encargado de que los usuarios puedan intercambiarse diferentes elementos del globo.

Por último está el paquete Usuarios, que contiene las clases propias relativas a las operaciones típicas sobre la gestión de usuarios en una aplicación, como por ejemplo el Registro.

5.3.2. Clases del análisis

A la hora de modelar las clases de análisis se utilizan dos tipos de diagramas: estáticos o de colaboración. Para elaborar estos diagramas se utilizan tres tipos de clases:

- Interfaces: son clases que actúan de intermediarias en la comunicación entre usuario y sistema, o entre sistemas.
- Control: son las clases que realizan la lógica del modelo de negocio. Sirven para coordinar y comunicar clases (interfaces, datos y otras clases de control).
- Datos: representan un objeto de datos estático, normalmente una entidad del modelo de negocio. Generalmente son manipuladas por clases de control.

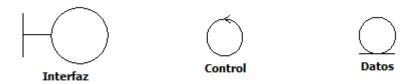


Figura 5.39: Clases del análisis

5.3.3. Realizaciones de casos de uso

El objetivo de este apartado es identificar las clases cuyos objetos son necesarios para realizar un caso de uso y describir su comportamiento mediante la interacción dichos objetos.

5.3.3.1. Subsistema de Usuarios

Usuario No Registrado

Registrarse

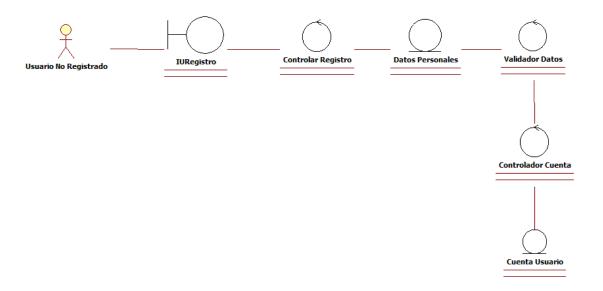


Figura 5.40: Clases de Análisis: Registrarse

Iniciar Sesión

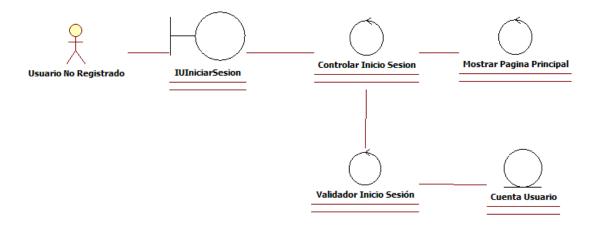


Figura 5.41: Clases de Análisis: Iniciar Sesión

• Recordar Contraseña

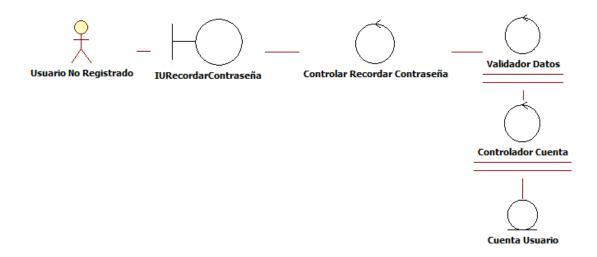


Figura 5.42: Clases de Análisis: Recordar Contraseña

Usuario Registrado

Modificar Datos Personales

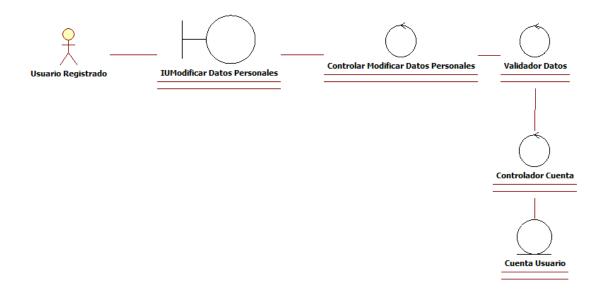


Figura 5.43: Clases de Análisis: Modificar Datos Personales

\blacksquare Cambiar Contraseña

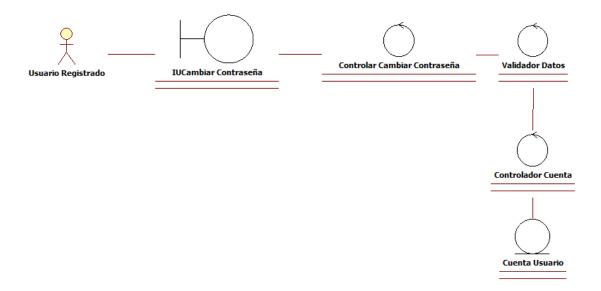


Figura 5.44: Clases de Análisis: Cambiar Contraseña

■ Darse de Baja

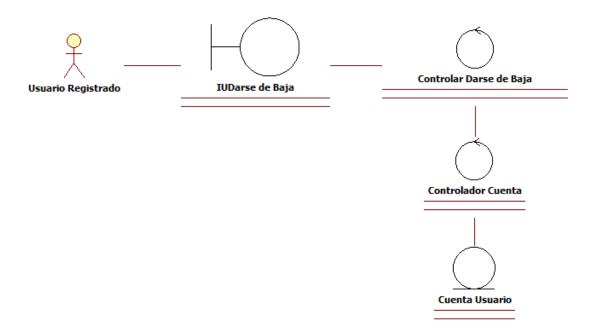


Figura 5.45: Clases de Análisis: Darse de Baja

■ Cerrar Sesión



Figura 5.46: Clases de Análisis: Cerrar Sesión

Contactos

Añadir Usuario

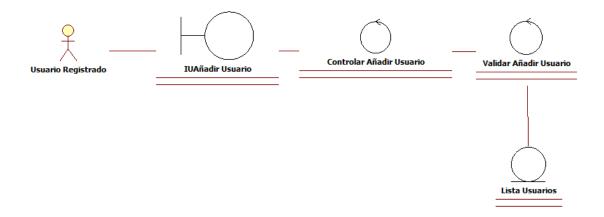


Figura 5.47: Clases de Análisis: Añadir Usuario

■ Eliminar Usuario

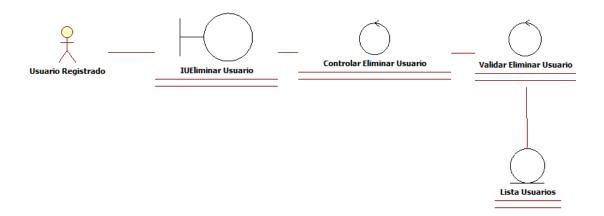


Figura 5.48: Clases de Análisis: Eliminar Usuario

■ Ver Usuario

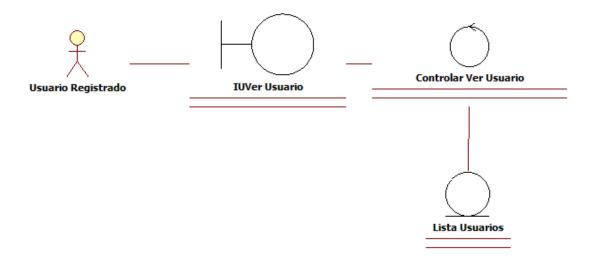


Figura 5.49: Clases de Análisis: Ver Usuario

■ Modificar Usuario

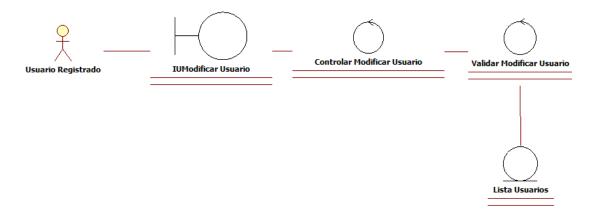


Figura 5.50: Clases de Análisis: Modificar Usuario

5.3.3.2. Subsistema de Elementos del Globo

Capas

Crear Capa

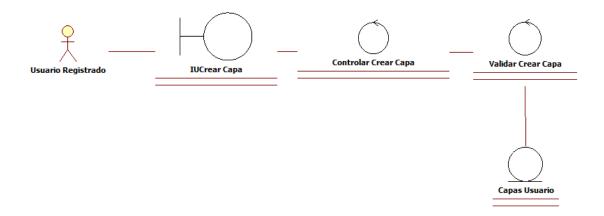


Figura 5.51: Clases de Análisis: Crear Capa

■ Modificar Capa

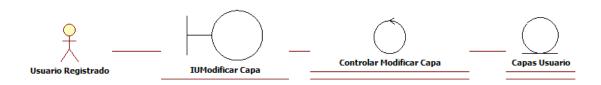


Figura 5.52: Clases de Análisis: Modificar Capa

• Renombrar Capa

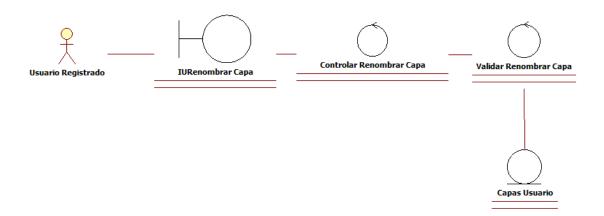


Figura 5.53: Clases de Análisis: Renombrar Capa

■ Eliminar Capa



Figura 5.54: Clases de Análisis: Eliminar Capa

Crear Ruta

Insertar Puntos

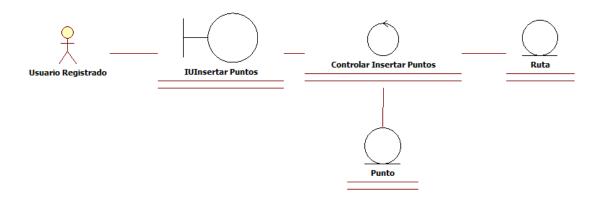


Figura 5.55: Clases de Análisis: Insertar Puntos

- Insertar Modelo 3D
- Insertar Documento
- Insertar Foto
- Insertar Vídeo

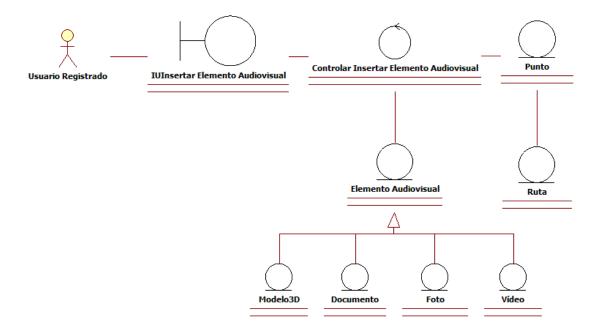


Figura 5.56: Clases de Análisis: Insertar Elemento Audiovisual

Gestionar Rutas

• Guardar Ruta



Figura 5.57: Clases de Análisis: Guardar Ruta

■ Modificar Ruta



Figura 5.58: Clases de Análisis: Modificar Ruta

• Visualizar Ruta (contenedor)



Figura 5.59: Clases de Análisis: Visualizar Ruta (contenedor)

■ Eliminar Ruta



Figura 5.60: Clases de Análisis: Eliminar Ruta

• Renombrar Ruta



Figura 5.61: Clases de Análisis: Renombrar Ruta

Sitios de Interés

■ Añadir Sitio de Interés

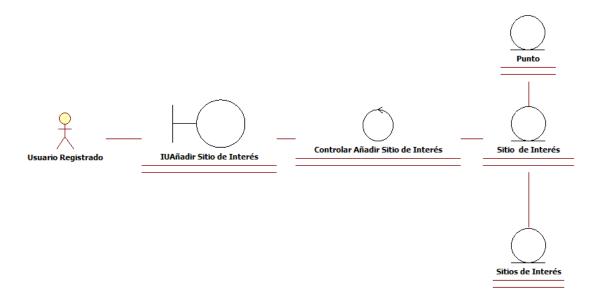


Figura 5.62: Clases de Análisis: Añadir Sitio de Interés

Visualizar Sitio de Interés



Figura 5.63: Clases de Análisis: Visualizar Sitio de Interés

■ Eliminar Sitio de Interés



Figura 5.64: Clases de Análisis: Eliminar Sitio de Interés

Modificar Sitio de Interés



Figura 5.65: Clases de Análisis: Modificar Sitio de Interés

• Renombrar Sitio de Interés

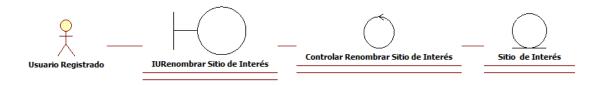


Figura 5.66: Clases de Análisis: Renombrar Sitio de Interés

Tours

■ Crear Tour Automático

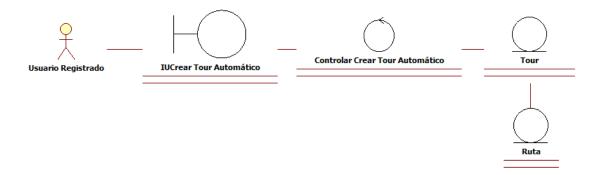


Figura 5.67: Clases de Análisis: Crear Tour Automático

Visualizar Tour



Figura 5.68: Clases de Análisis: Visualizar Tour

• Eliminar Tour



Figura 5.69: Clases de Análisis: Eliminar Tour

■ Modificar Tour



Figura 5.70: Clases de Análisis: Modificar Tour

• Renombrar Tour



Figura 5.71: Clases de Análisis: Renombrar Tour

■ Crear Tour Manual

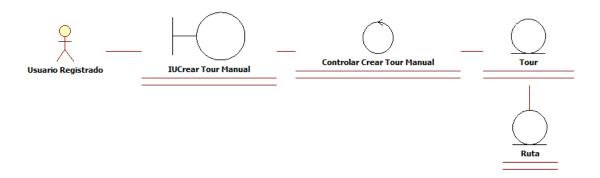


Figura 5.72: Clases de Análisis: Crear Tour Manual

Visualizar Capa

Activar Capa (Crear)



Figura 5.73: Clases de Análisis: Activar Capa (Crear)

■ Desplegar Capa



Figura 5.74: Clases de Análisis: Desplegar Capa

Visualizar Información



Figura 5.75: Clases de Análisis: Visualizar Información

■ Ir a



Figura 5.76: Clases de Análisis: Ir a

Visualizar Rutas

• Ver Ruta

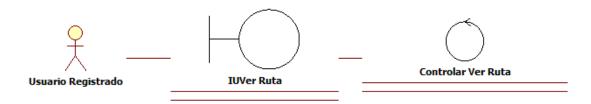


Figura 5.77: Clases de Análisis: Ver Ruta

Ver Tour

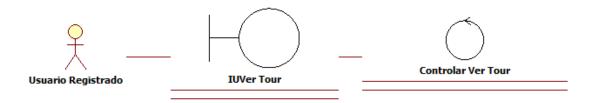


Figura 5.78: Clases de Análisis: Ver Tour

• Ver Sitios de Interés



Figura 5.79: Clases de Análisis: Ver Sitios de Interés

Seleccionar Capas



Figura 5.80: Clases de Análisis: Seleccionar Capas

5.3.3.3. Subsistema Web 2.0

Compartir

■ Compartir Rutas

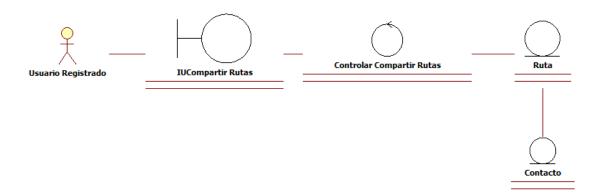


Figura 5.81: Clases de Análisis: Compartir Rutas

■ Compartir Archivos

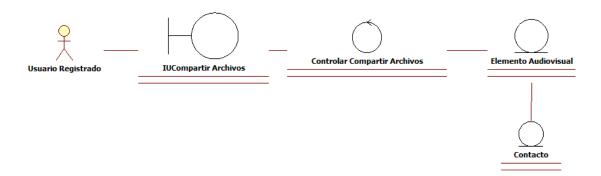


Figura 5.82: Clases de Análisis: Compartir Archivos

Rutas Favoritas



Figura 5.83: Clases de Análisis: Rutas Favoritas

• Presentar Otras Rutas

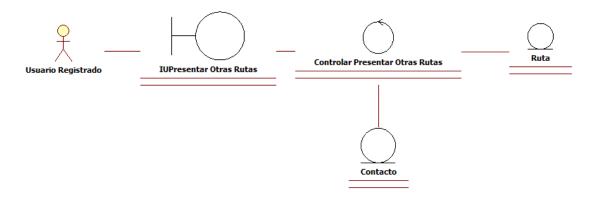


Figura 5.84: Clases de Análisis: Presentar Otras Rutas

Comunicación

■ Comunicarse con otras Redes Sociales

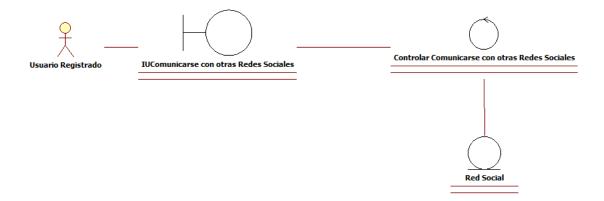


Figura 5.85: Clases de Análisis: Comunicarse con otras Redes Sociales

\bullet Escribir Comentarios

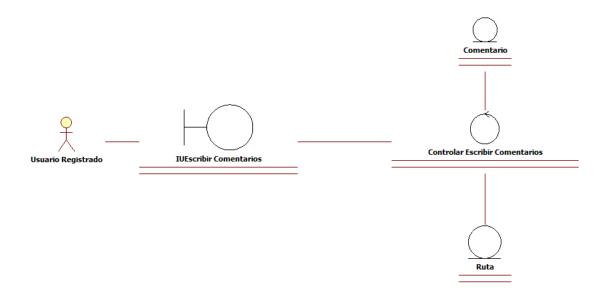


Figura 5.86: Clases de Análisis: Escribir Comentarios

lacktriangle Chatear

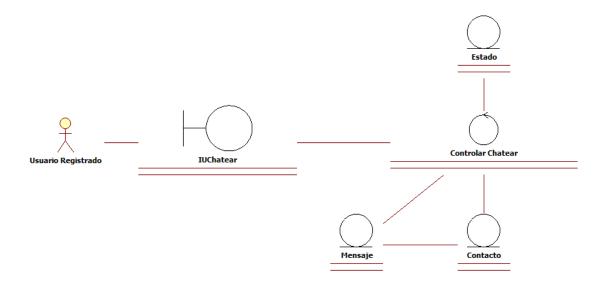


Figura 5.87: Clases de Análisis: Chatear

• Enviar Email

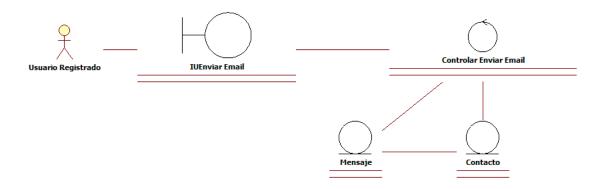


Figura 5.88: Clases de Análisis: Enviar Email

■ Crear Comunidades

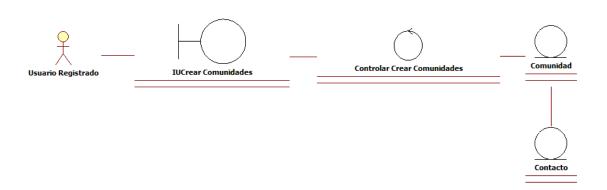


Figura 5.89: Clases de Análisis: Crear Comunidades

5.4. Modelo de diseño

Durante el diseño modelamos el sistema y su arquitectura para que soporte los requisitos funcionales y no funcionales que tendrá la aplicación. Una entrada esencial al diseño es el modelo de análisis. En esta fase se realizará, por un lado, un diagrama detallado de las clases finales que existirán en el modelo de implementación. Estas clases en muchos casos serán similares a las obtenidas durante el análisis, aunque también aparecerán nuevas clases que no fueron identificadas en la anterior etapa. Además, para cada diagrama de clases se adjuntará un diagrama de secuencia, que mostrará la evolución de las clases a lo largo del tiempo y la interacción entre ellas. Antes de ello, se mostrará una visión general de la organización de los elementos de procesamiento que intervendrán en el sistema mediante un diagrama de despliegue, así como una breve descripción de cada uno de los elementos que compondrán dicho diagrama para un correcto entendimiento de éste.

173

5.4.1. Arquitectura del sistema

En primer lugar se describirán los diferentes nodos que compondrán el diagrama de despliegue para posteriormente indicar cómo interactúan entre ellos. Los nodos de los cuales consta el sistema son los siguientes:

Usuarios del sistema: son quienes harán uso de los servicios que presta la aplicación, mediante la aportación de datos, vídeos, información, etc. y la realización de diferentes operaciones que están disponibles para los usuarios. Al tratarse de una aplicación web, los clientes usarán navegadores que intercambiarán peticiones HTTP con el servidor.

Administradores del sistema: son aquellos encargados de gestionar todos los recursos de la aplicación, tanto los aportados por los usuarios como los subidos por los propios administradores. Además son los que han de administrar los usuarios que se registran en el sistema, así como responder a las dudas, sugerencias y reclamaciones enviadas por estos.

Servidor Web: el servidor de este sistema es una combinación de servidor web que atiende las peticiones de los clientes, y un servidor de aplicaciones que realiza el procesamiento de las operaciones requeridas, comunicándose con el servidor de bases de datos para la recuperación de información necesaria en las operaciones, y devolviendo los resultados a los clientes que hayan hecho la petición.

Servidor de BD: es el encargado de almacenar los datos del sistema. Debe tener la capacidad de responder a un gran número de peticiones que pueden ser realizadas por el servidor web. También ha de tener opciones de configuración de seguridad bastante amplias y detalladas.

Servidor de copias de seguridad: es el sistema provisto para el alojamiento del servicio de copias de seguridad de la base de datos de nuestro sistema. Este servicio nos permite recuperar las distintas bases de datos en caso de desastre del servidor de bases de datos del servicio de Hosting.

Servidor de Google Earth: es el encargado de proporcionar el servicio mediante el cual se pueden visualizar los mapas de la aplicación. Además como todos los vídeos subidos al sistema tendrán unas coordenadas geográficas asociadas, éstas tendrán que ser pasadas al servidor de Google Earth para poder cargar los mapas con la información recibida. Es externo a la infraestructura de la aplicación.

Una vez descritos los diferentes nodos del sistema, mostramos como quedará el diagrama de despliegue:

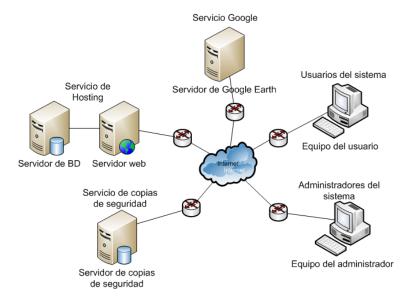


Figura 5.90: Diagrama de despliegue

5.4.2. Arquitectura del software

El desglose por capas se puede ver en el siguiente diagrama:

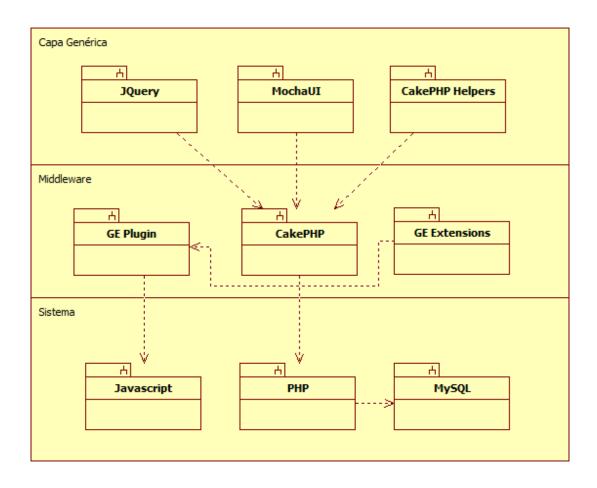


Figura 5.91: Arquitectura del software

La arquitectura del software se puede dividir en tres capas según su nivel de abstracción y están interconectadas de arriba a abajo.

■ Capa del Sistema

- PHP: lenguaje y conjunto de funciones que sirven como soporte a las capas superiores. También hace de interfaz con la base de datos. Opera desde el lado del servidor.
- MySQL: SGBD para almacenar la información de la aplicación.
- JavaScript: lenguaje empleado para aumentar la funcionalidad del globo terráqueo y la interfaz de usuario.

Middleware

- CakePHP: conjunto de clases y métodos para dar soporte a las capas anteriores. Proporciona un nuevo nivel de abstracción para acceder a la base de datos,
 además un núcleo organizado mediante patrones de diseño que facilita la extensión de la funcionalidad y el mantenimiento.
- Google Earth Plugin: extensión del navegador web que permite visualizar el globo terráqueo en tres dimensiones. Ofrece una API para desarrollar otras funciones a las ya existentes.

■ Capa Genérica

- JQuery: librería en JavaScript para realizar las funcionalidades comunes de manejo de DOM, eventos, efectos y AJAX.
- MochaUI: librería para interfaces de usuario en aplicaciones web basada en el framework de JavaScript MooTools.
- CakePHP helpers: ayudantes que extienden la funcionalidad de CakePHP, como por ejemplo, utilidades de cara a mejorar las vistas.

5.4.3. Estereotipos utilizados

Los estereotipos que se van a utilizar en la especificación de casos de uso son los siguientes:

- << Formulario >> Formulario HTML que permite al usuario introducir datos en el sistema.
- <<Vista>> Elemento de procesamiento asociado a un elemento formulario. Se encarga de preparar la información que se va a mostrar en el mismo.
- <<Modelo>> Representa tanto el estado de la aplicación como los datos que se utilizan.
- << Controlador>> Se encarga de coordinar el funcionamiento de todos los elementos del caso. Recibe la entrada del usuario y realiza las tareas adecuadas sobre el modelo de datos. Esta tarea se hará del lado del servidor.

5.4.4. Especificación de casos de uso

5.4.4.1. Subsistema de Usuarios

Usuario No Registrado

Registrarse

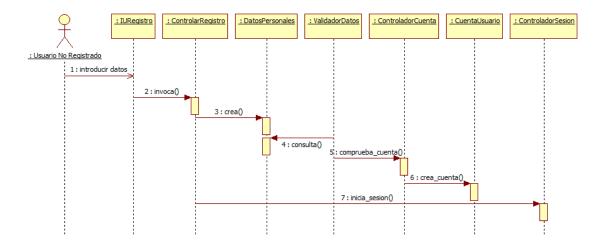


Figura 5.92: Interacción de Objetos: Registrarse

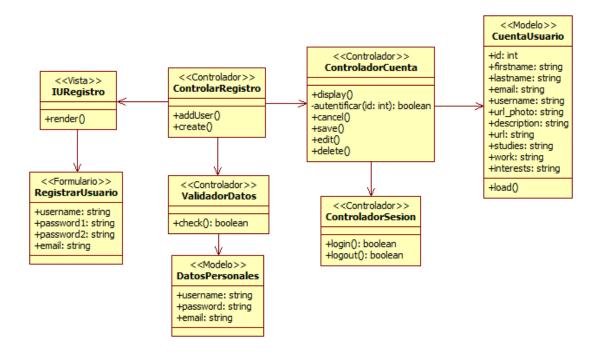


Figura 5.93: Diagrama de clases: Registrarse

Iniciar Sesión

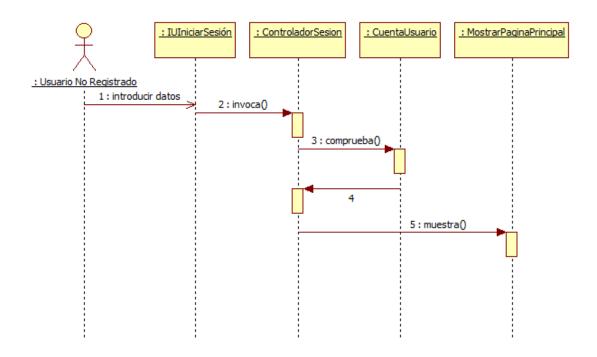


Figura 5.94: Interacción de Objetos: Iniciar Sesión

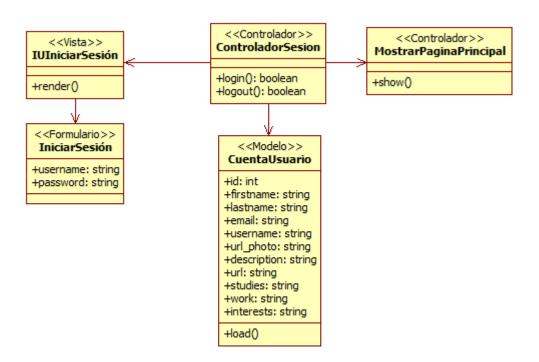


Figura 5.95: Diagrama de clases: Iniciar Sesión

• Recordar Contraseña

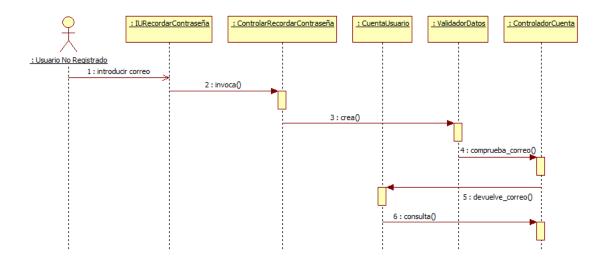


Figura 5.96: Interacción de Objetos: Recordar Contraseña

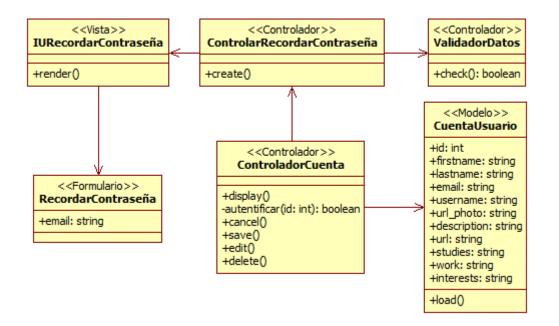


Figura 5.97: Diagrama de clases: Recordar Contraseña

Usuario Registrado

Modificar Datos Personales

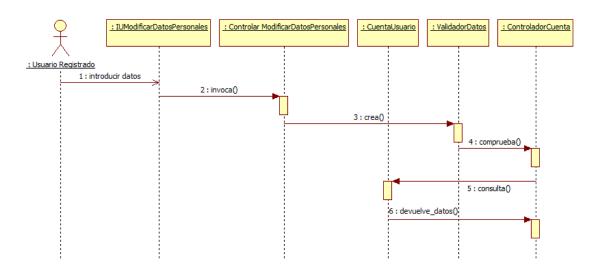


Figura 5.98: Interacción de Objetos: Modificar Datos Personales

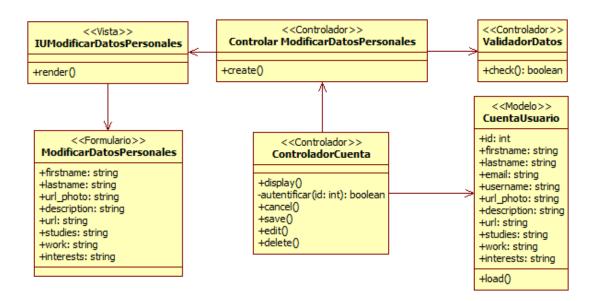


Figura 5.99: Diagrama de clases: Modificar Datos Personales

■ Cambiar Contraseña

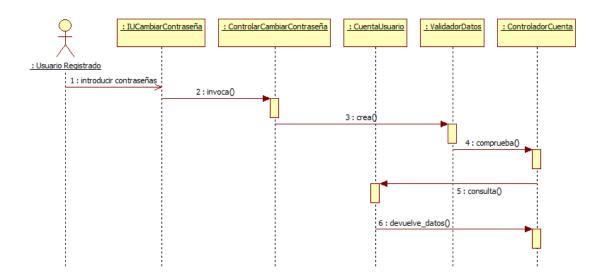


Figura 5.100: Interacción de Objetos: Cambiar Contraseña

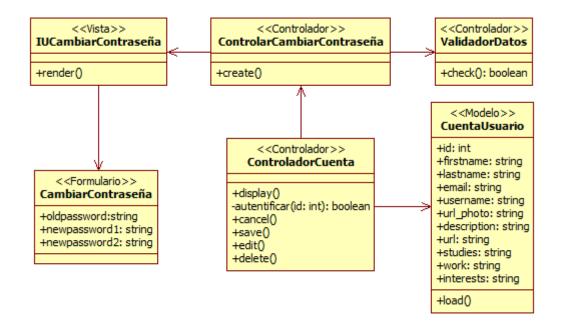


Figura 5.101: Diagrama de clases: Cambiar Contraseña

■ Darse de Baja

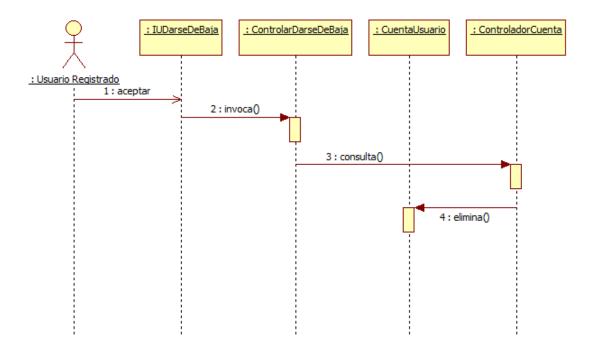


Figura 5.102: Interacción de Objetos: Darse de Baja

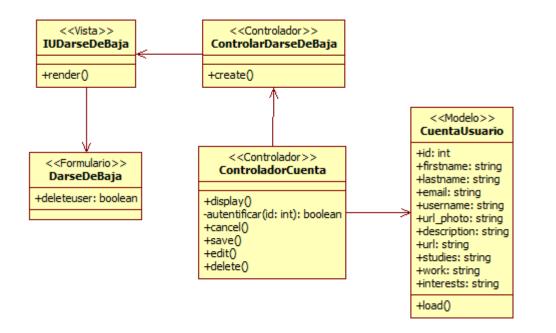


Figura 5.103: Diagrama de clases: Darse de Baja

■ Cerrar Sesión

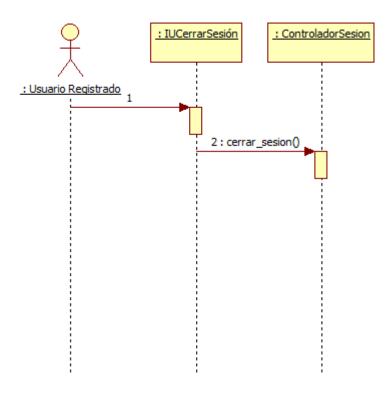


Figura 5.104: Interacción de Objetos: Cerrar Sesión

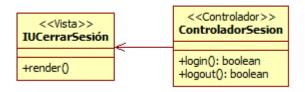


Figura 5.105: Diagrama de clases: Cerrar Sesión

Contactos

■ Añadir Usuario

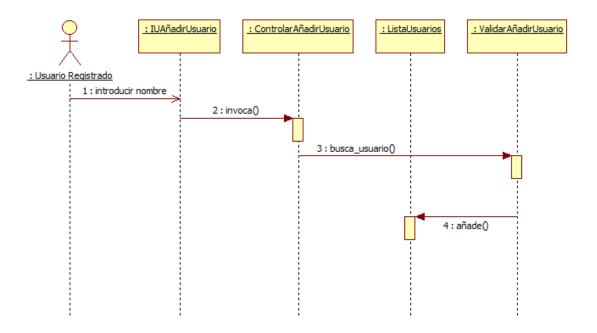


Figura 5.106: Interacción de Objetos: Añadir Usuario

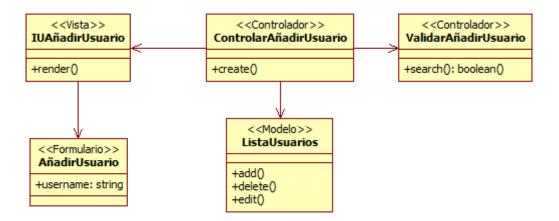


Figura 5.107: Diagrama de clases: Añadir Usuario

• Eliminar Usuario

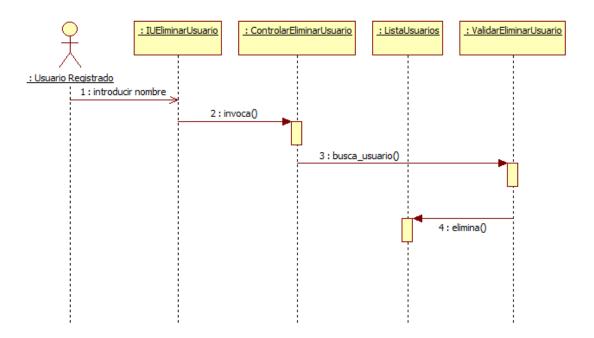


Figura 5.108: Interacción de Objetos: Eliminar Usuario

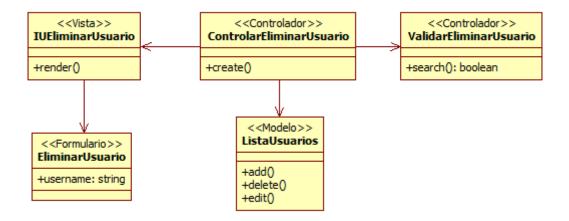


Figura 5.109: Diagrama de clases: Eliminar Usuario

Ver Usuario

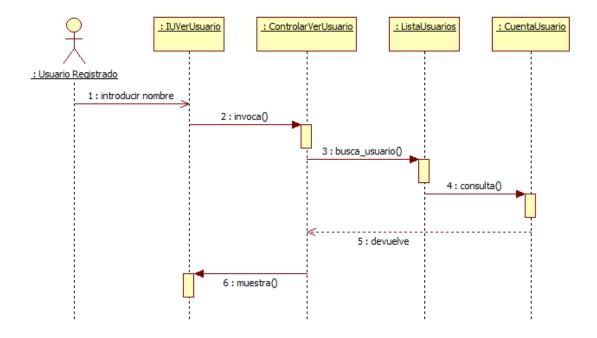


Figura 5.110: Interacción de Objetos: Ver Usuario

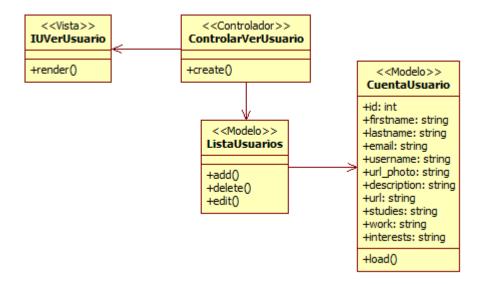


Figura 5.111: Diagrama de clases: Ver Usuario

■ Modificar Usuario

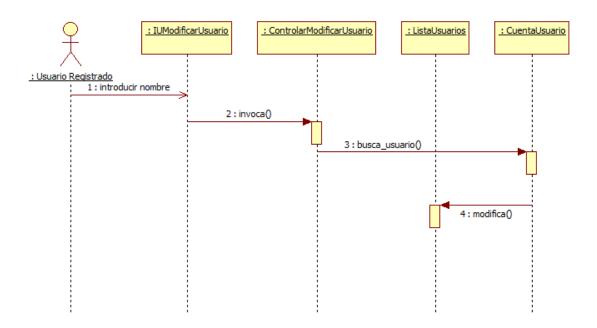


Figura 5.112: Interacción de Objetos: Modificar Usuario

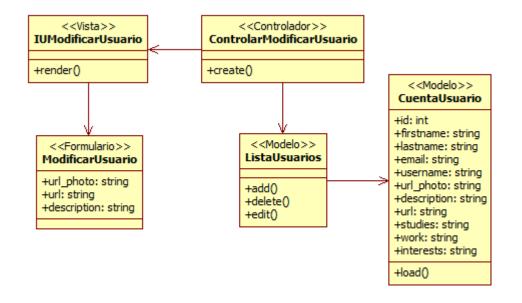


Figura 5.113: Diagrama de clases: Modificar Usuario

5.4.4.2. Subsistema de Elementos del Globo

Capas

• Crear Capa

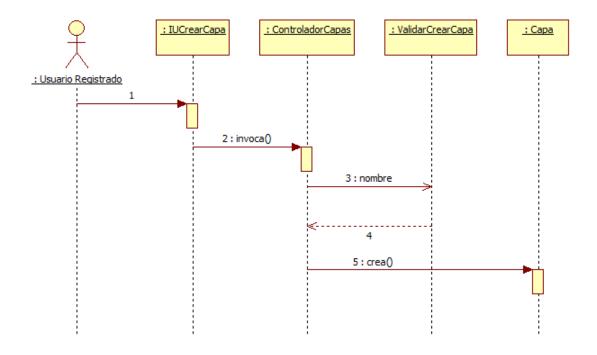


Figura 5.114: Interacción de Objetos: Crear Capa

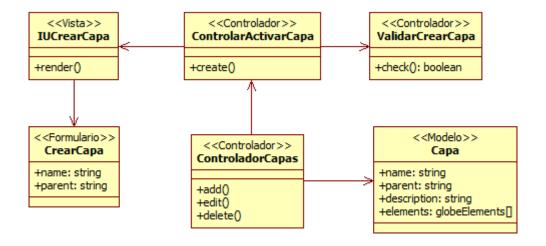


Figura 5.115: Diagrama de clases: Crear Capa

■ Modificar Capa

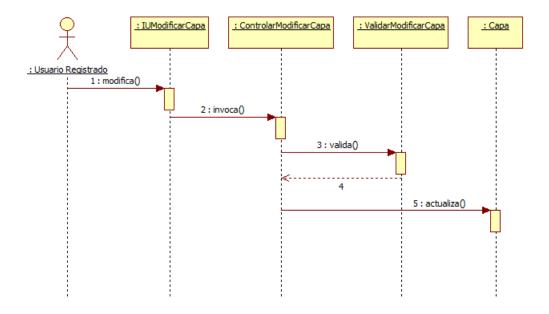


Figura 5.116: Interacción de Objetos: Modificar Capa

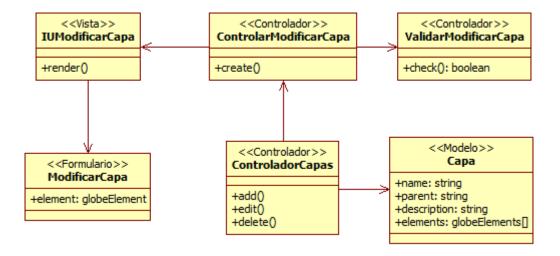


Figura 5.117: Diagrama de clases: Modificar Capa

Renombrar Capa

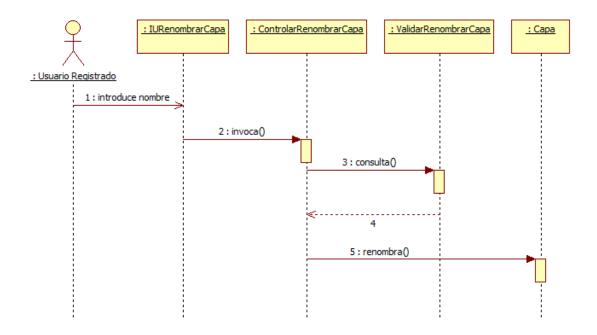


Figura 5.118: Interacción de Objetos: Renombrar Capa

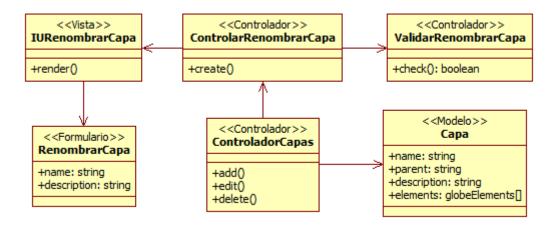


Figura 5.119: Diagrama de clases: Renombrar Capa

■ Eliminar Capa

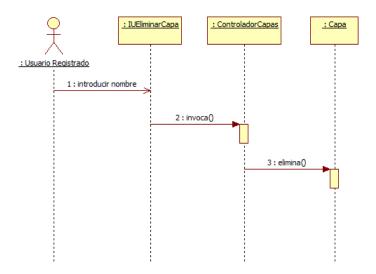


Figura 5.120: Interacción de Objetos: Eliminar Capa

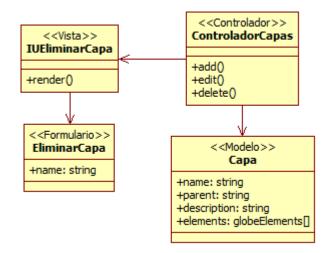


Figura 5.121: Diagrama de clases: Eliminar Capa

Crear Ruta

■ Insertar Puntos

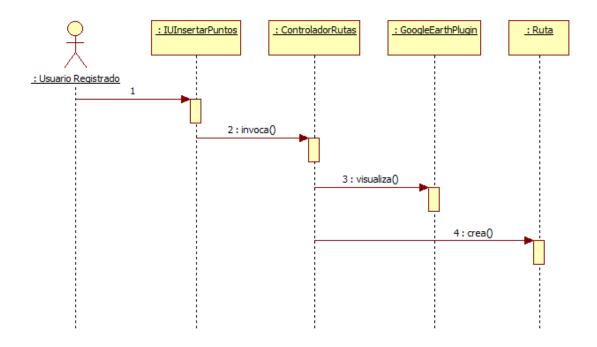


Figura 5.122: Interacción de Objetos: Insertar Puntos

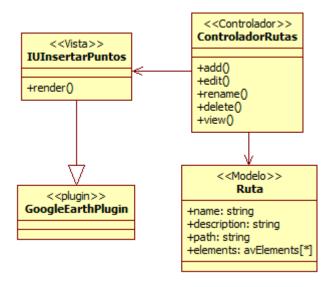


Figura 5.123: Diagrama de clases: Insertar Puntos

- Insertar Modelo 3D
- Insertar Foto
- Insertar Documento
- Insertar Vídeo

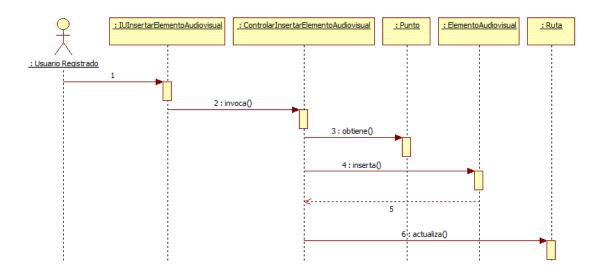


Figura 5.124: Interacción de Objetos: Insertar Insertar Elemento Audiovisual

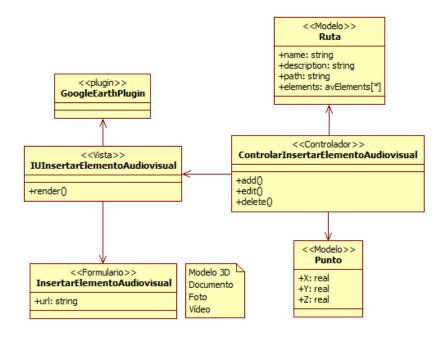


Figura 5.125: Diagrama de clases: Insertar Insertar Elemento Audiovisual

Gestionar Rutas

• Guardar Ruta

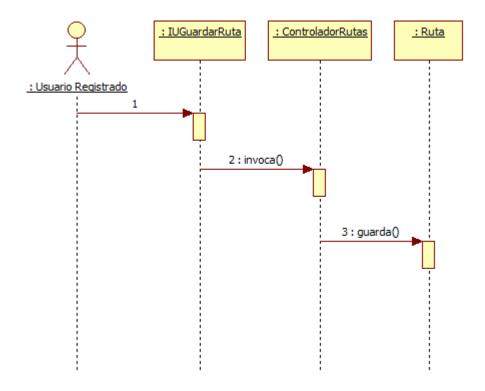


Figura 5.126: Interacción de Objetos: Guardar Ruta

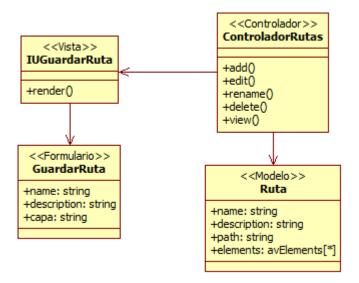


Figura 5.127: Diagrama de clases: Guardar Ruta

■ Modificar Ruta

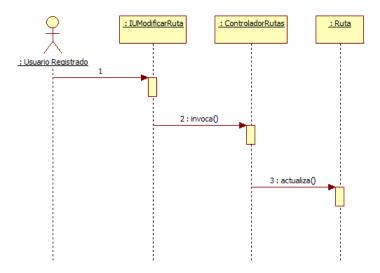


Figura 5.128: Interacción de Objetos: Modificar Ruta

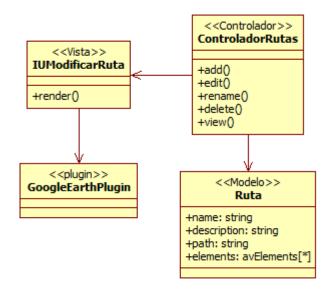


Figura 5.129: Diagrama de clases: Modificar Ruta

• Visualizar Ruta (contenedor)

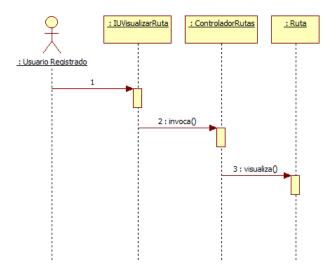


Figura 5.130: Interacción de Objetos: Visualizar Ruta (contenedor)

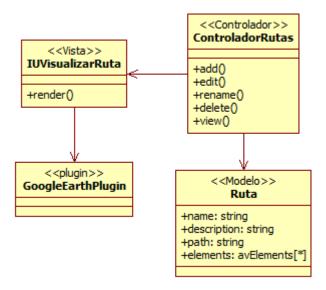


Figura 5.131: Diagrama de clases: Visualizar Ruta (contenedor)

■ Eliminar Ruta

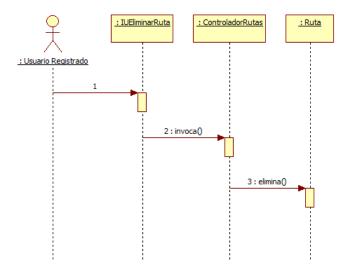


Figura 5.132: Interacción de Objetos: Eliminar Ruta

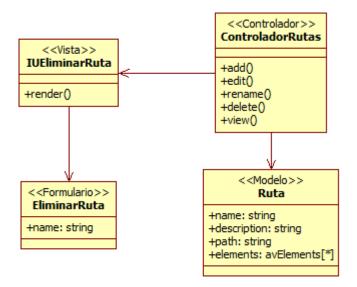


Figura 5.133: Diagrama de clases: Eliminar Ruta

• Renombrar Ruta

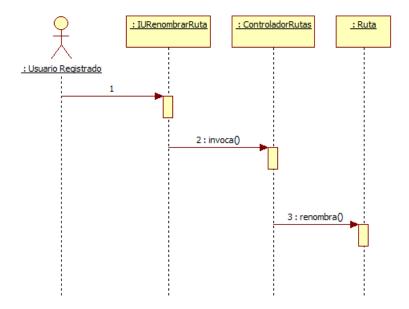


Figura 5.134: Interacción de Objetos: Renombrar Ruta

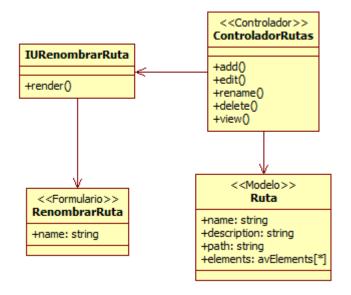


Figura 5.135: Diagrama de clases: Renombrar Ruta

Sitios de Interés

• Añadir Sitio de Interés

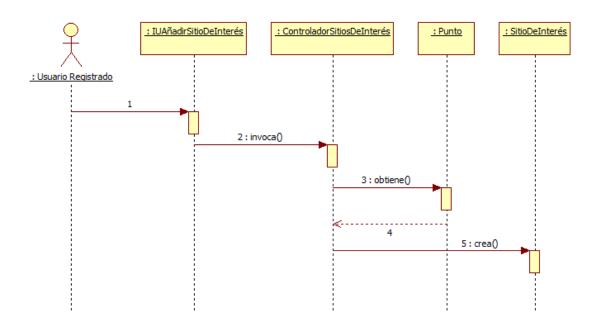


Figura 5.136: Interacción de Objetos: Añadir Sitio de Interés

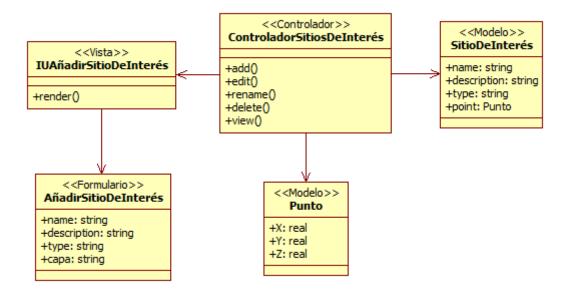


Figura 5.137: Diagrama de clases: Añadir Sitio de Interés

Visualizar Sitio de Interés

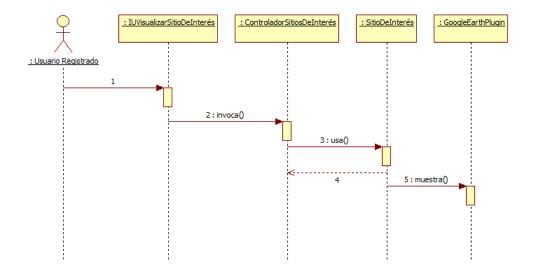


Figura 5.138: Interacción de Objetos: Visualizar Sitio de Interés

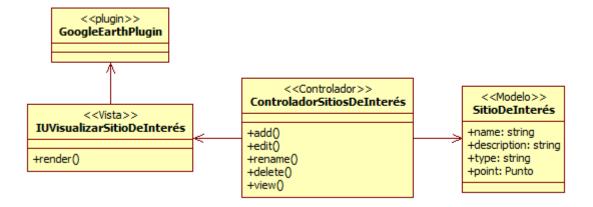


Figura 5.139: Diagrama de clases: Visualizar Sitio de Interés

■ Eliminar Sitio de Interés

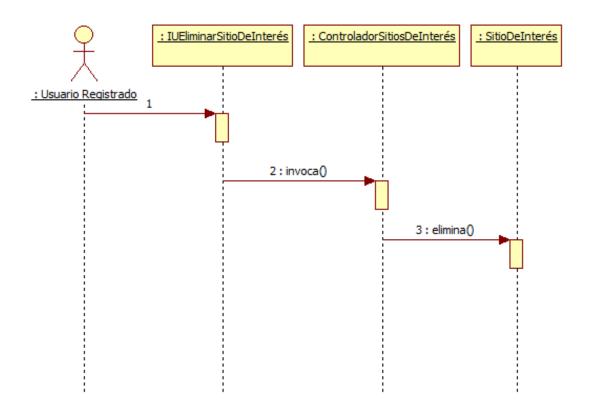


Figura 5.140: Interacción de Objetos: Eliminar Sitio de Interés

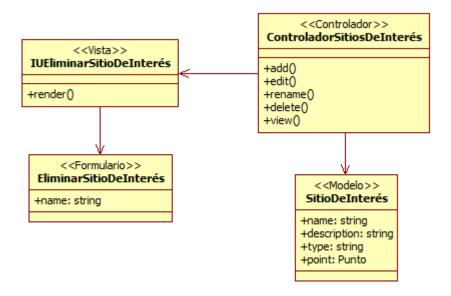


Figura 5.141: Diagrama de clases: Eliminar Sitio de Interés

■ Modificar Sitio de Interés

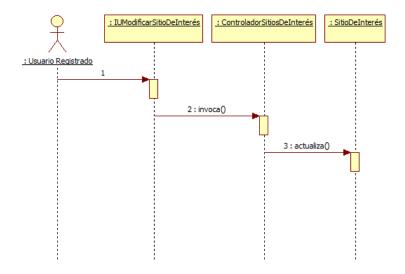


Figura 5.142: Interacción de Objetos: Modificar Sitio de Interés

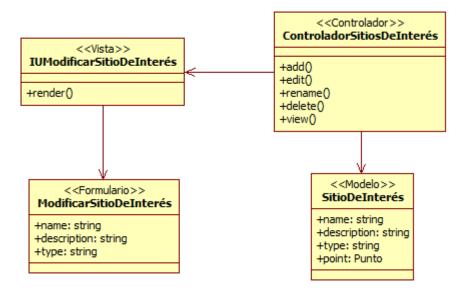


Figura 5.143: Diagrama de clases: Modificar Sitio de Interés

• Renombrar Sitio de Interés

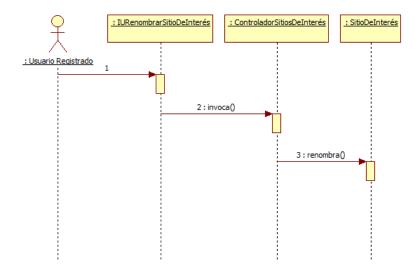


Figura 5.144: Interacción de Objetos: Renombrar Sitio de Interés

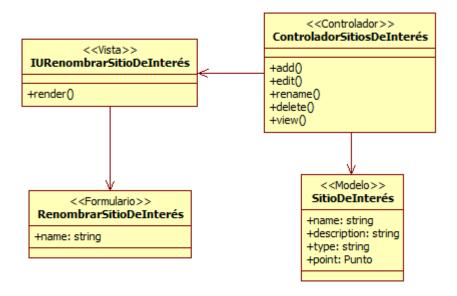


Figura 5.145: Diagrama de clases: Renombrar Sitio de Interés

Tours

Hay dos tipos de tours, ambos se muestran en el siguiente diagrama:

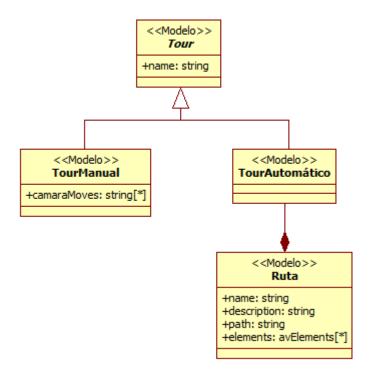


Figura 5.146: Tipos de Tours

■ Crear Tour Automático

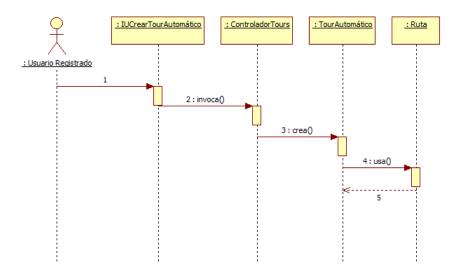


Figura 5.147: Interacción de Objetos: Crear Tour Automático

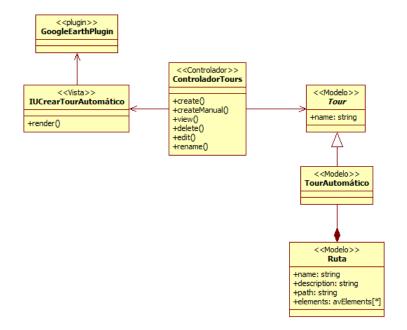


Figura 5.148: Diagrama de clases: Crear Tour Automático

\blacksquare Visualizar Tour

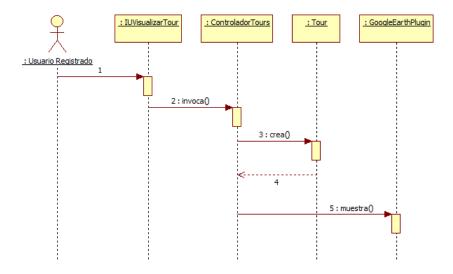


Figura 5.149: Interacción de Objetos: Visualizar Tour

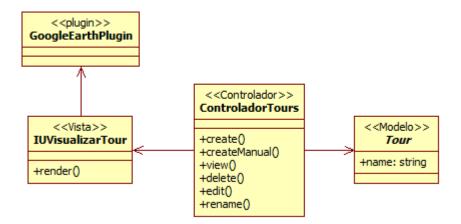


Figura 5.150: Diagrama de clases: Visualizar Tour

■ Eliminar Tour

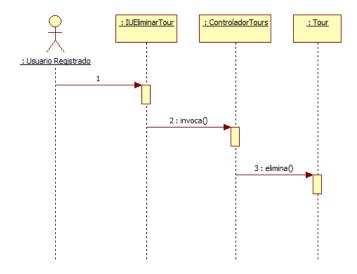


Figura 5.151: Interacción de Objetos: Eliminar Tour

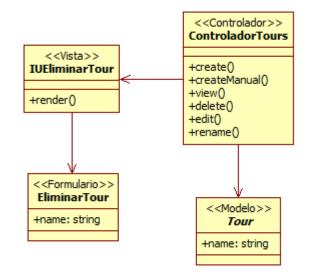


Figura 5.152: Diagrama de clases: Eliminar Tour

Modificar Tour

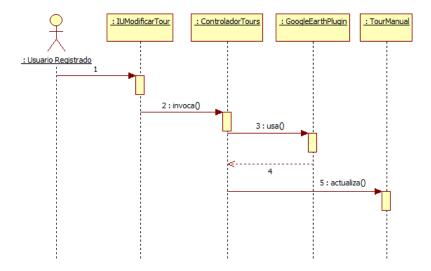


Figura 5.153: Interacción de Objetos: Modificar Tour

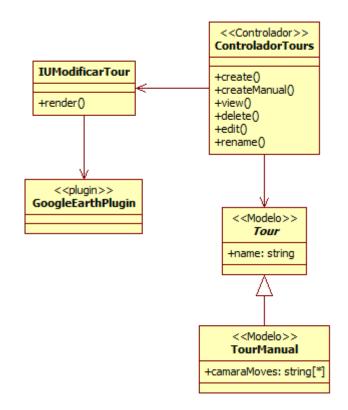


Figura 5.154: Diagrama de clases: Modificar Tour

• Renombrar Tour

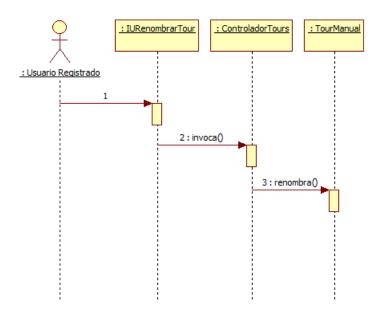


Figura 5.155: Interacción de Objetos: Renombrar Tour

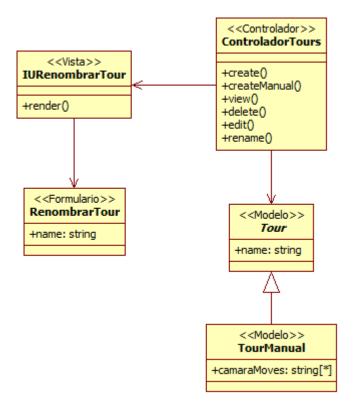


Figura 5.156: Diagrama de clases: Renombrar Tour

Crear Tour Manual

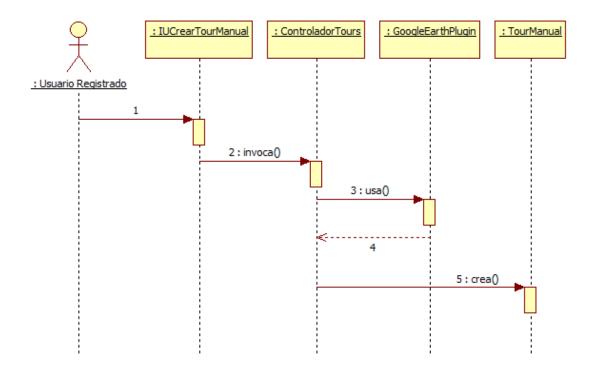


Figura 5.157: Interacción de Objetos: Crear Tour Manual

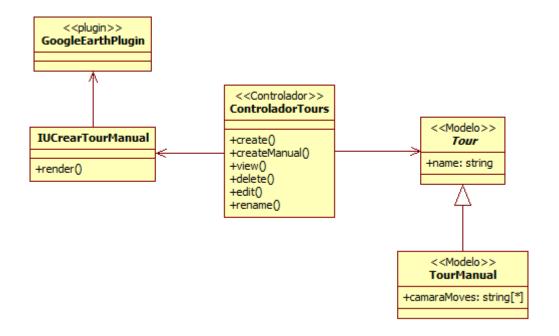


Figura 5.158: Diagrama de clases: Crear Tour Manual

Visualizar Capa

• Activar Capa (Crear)

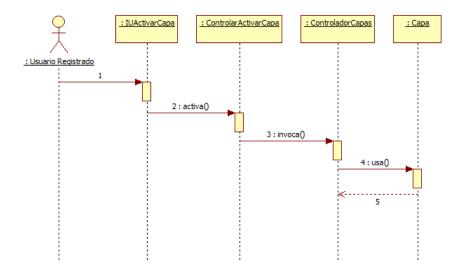


Figura 5.159: Interacción de Objetos: Activar Capa (Crear)

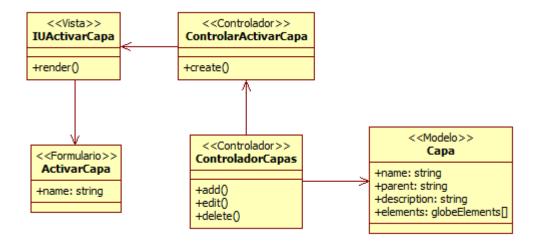


Figura 5.160: Diagrama de clases: Activar Capa (Crear)

■ Desplegar Capa

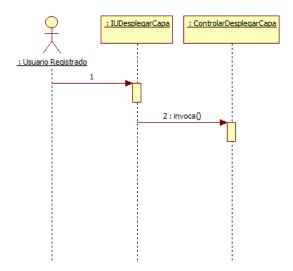


Figura 5.161: Interacción de Objetos: Desplegar Capa

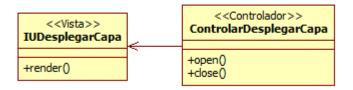


Figura 5.162: Diagrama de clases: Desplegar Capa

• Visualizar Información

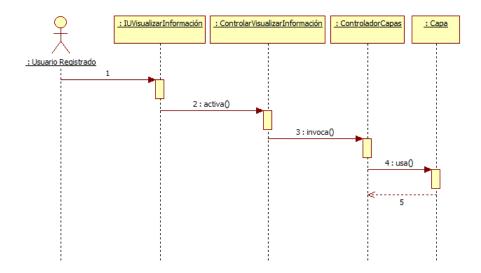


Figura 5.163: Interacción de Objetos: Visualizar Información

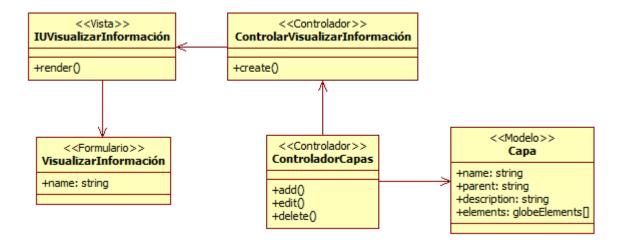


Figura 5.164: Diagrama de clases: Visualizar Información

■ Ir a

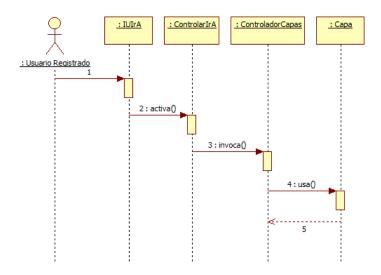


Figura 5.165: Interacción de Objetos: Ir a

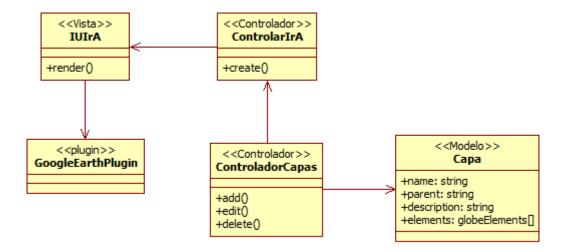


Figura 5.166: Diagrama de clases: Ir a

Visualizar Rutas

■ Ver Ruta

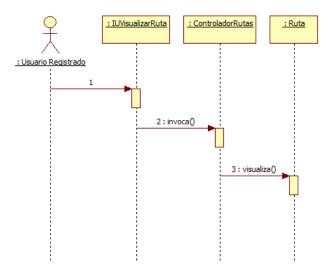


Figura 5.167: Interacción de Objetos: Ver Ruta

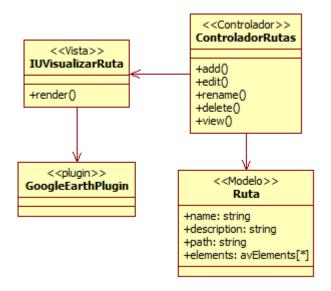


Figura 5.168: Diagrama de clases: Ver Ruta

\bullet Ver Tour

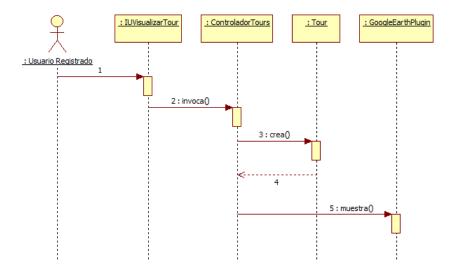


Figura 5.169: Interacción de Objetos: Ver Tour

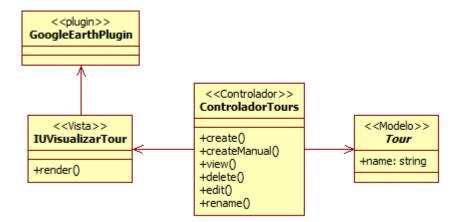


Figura 5.170: Diagrama de clases: Ver Tour

■ Ver Sitios de Interés

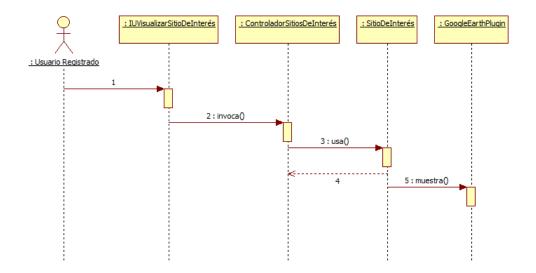


Figura 5.171: Interacción de Objetos: Ver Sitios de Interés

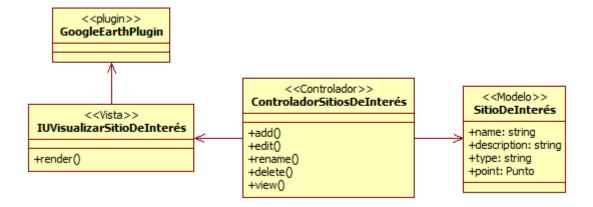


Figura 5.172: Diagrama de clases: Ver Sitios de Interés

Seleccionar Capas

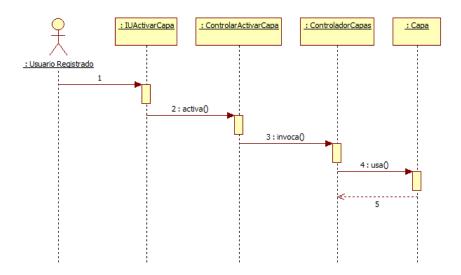


Figura 5.173: Interacción de Objetos: Seleccionar Capas

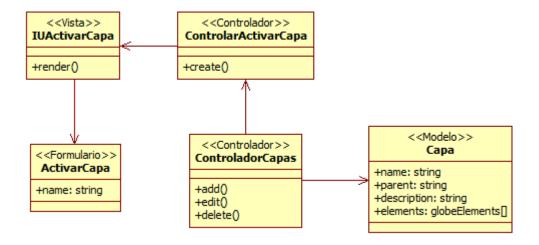


Figura 5.174: Diagrama de clases: Seleccionar Capas

5.4.4.3. Subsistema Web 2.0

Compartir

■ Compartir Rutas

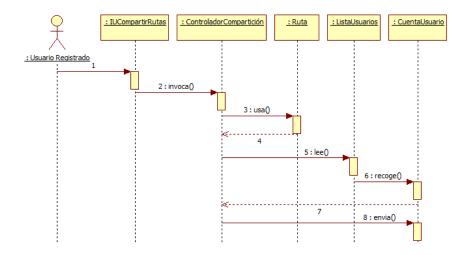


Figura 5.175: Interacción de Objetos: Compartir Rutas

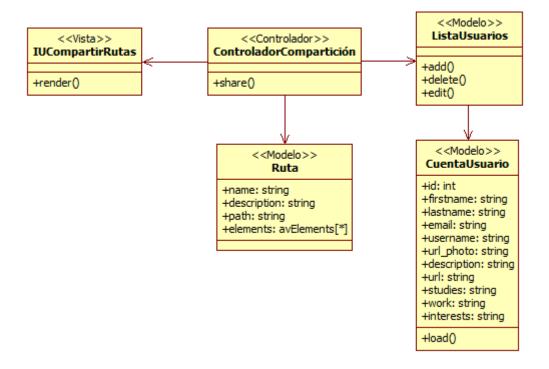


Figura 5.176: Diagrama de clases: Compartir Rutas

■ Compartir Archivos

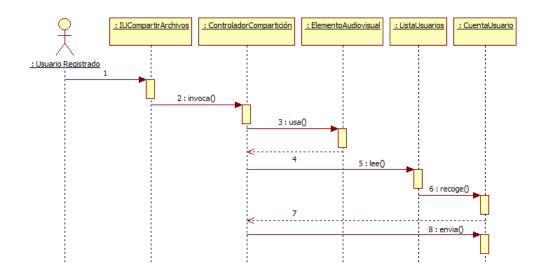


Figura 5.177: Interacción de Objetos: Compartir Archivos

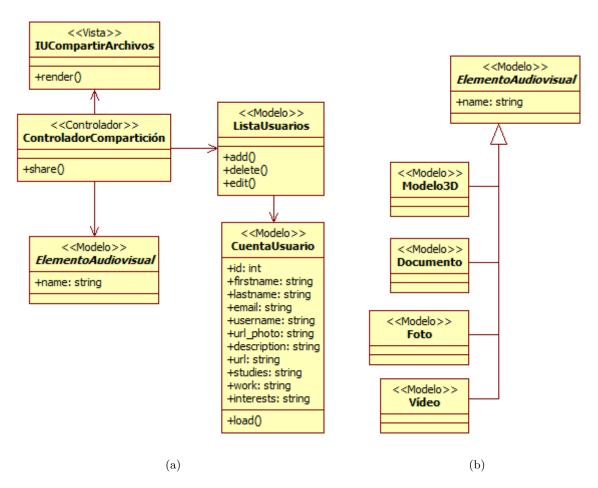


Figura 5.178: Diagrama de clases: Compartir Archivos

Rutas Favoritas

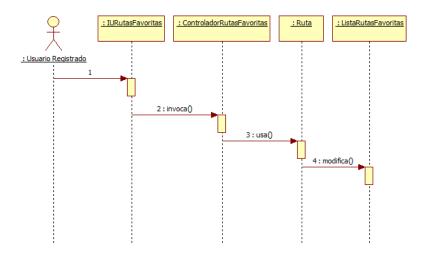


Figura 5.179: Interacción de Objetos: Rutas Favoritas

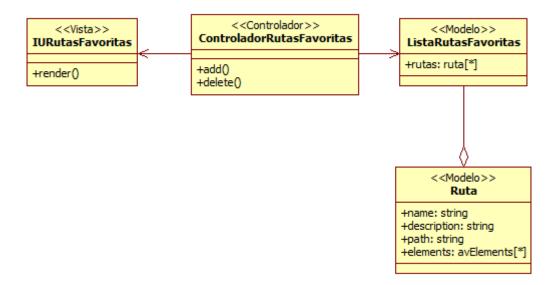


Figura 5.180: Diagrama de clases: Rutas Favoritas

• Presentar Otras Rutas

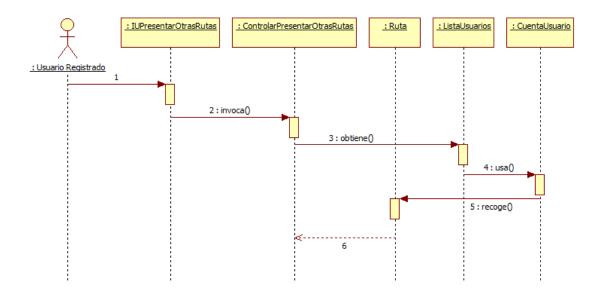


Figura 5.181: Interacción de Objetos: Presentar Otras Rutas

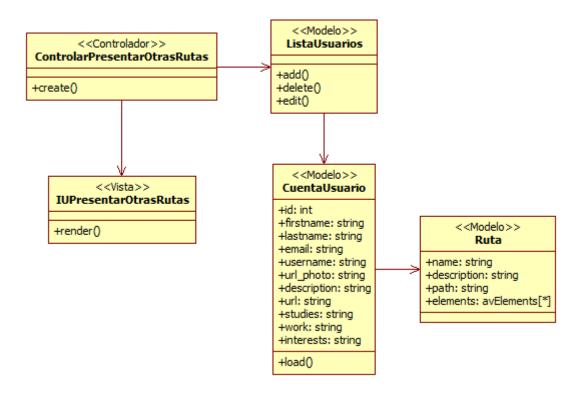


Figura 5.182: Diagrama de clases: Presentar Otras Rutas

Comunicación

■ Comunicarse con otras Redes Sociales

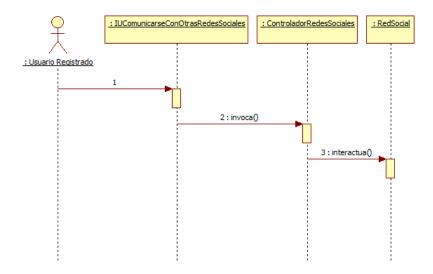


Figura 5.183: Interacción de Objetos: Comunicarse con otras Redes Sociales

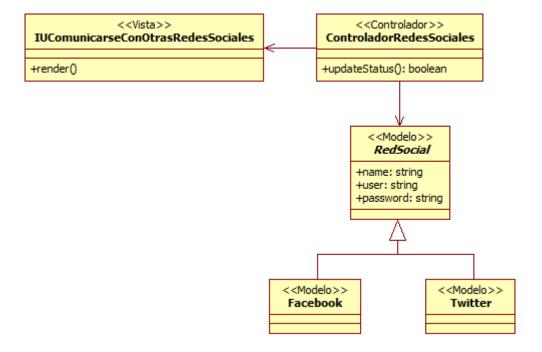


Figura 5.184: Diagrama de clases: Comunicarse con otras Redes Sociales

■ Escribir Comentarios

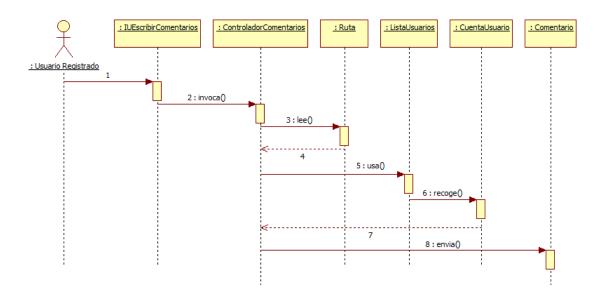


Figura 5.185: Interacción de Objetos: Escribir Comentarios

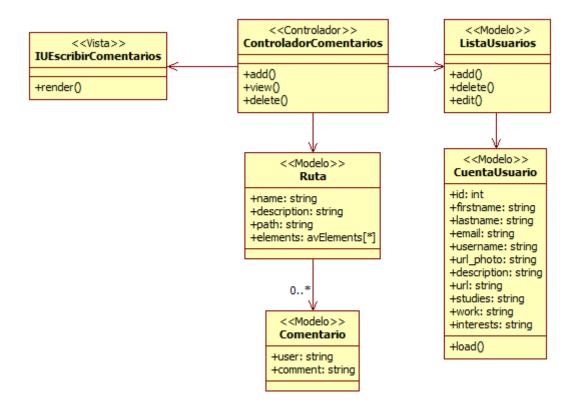


Figura 5.186: Diagrama de clases: Escribir Comentarios

Chatear

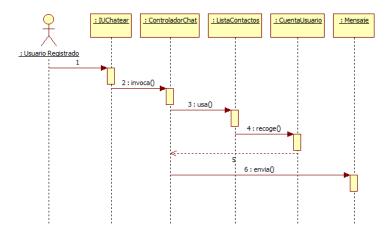


Figura 5.187: Interacción de Objetos: Chatear

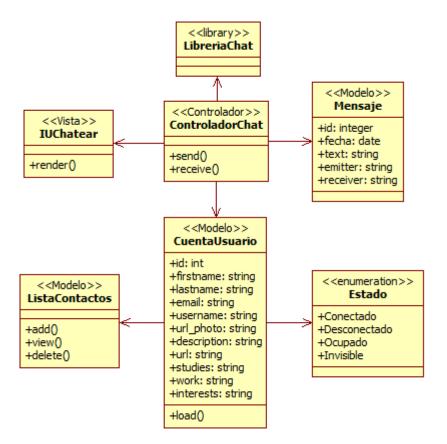


Figura 5.188: Diagrama de clases: Chatear

• Enviar Email

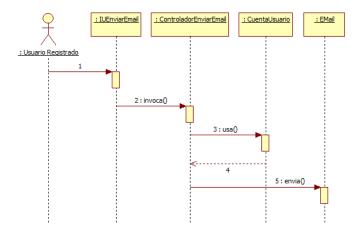


Figura 5.189: Interacción de Objetos: Enviar Email

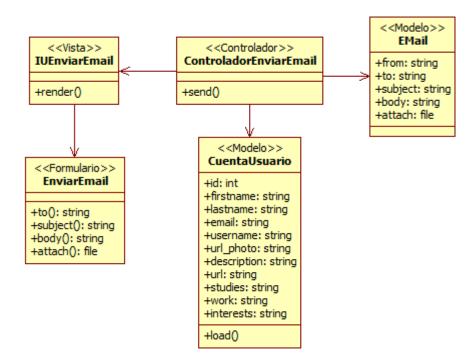


Figura 5.190: Diagrama de clases: Enviar Email

Crear Comunidades

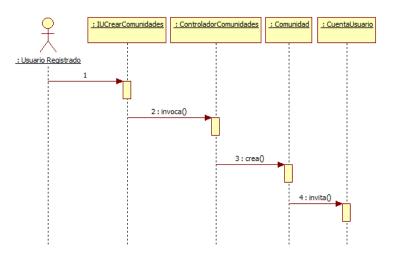


Figura 5.191: Interacción de Objetos: Crear Comunidades

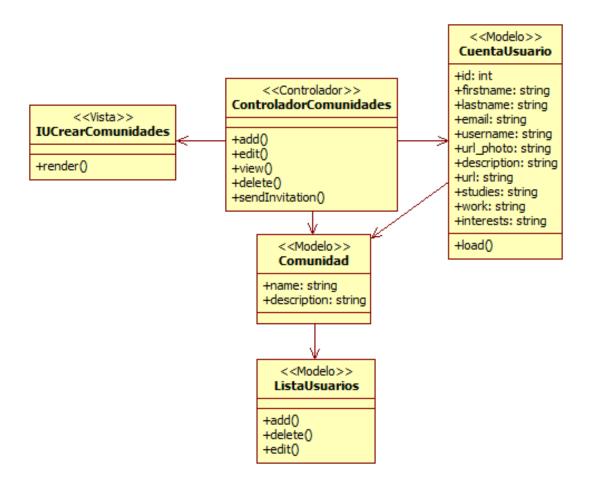


Figura 5.192: Diagrama de clases: Crear Comunidades

5.5. Implementación

En esta sección, dedicada a la implementación de la aplicación, se explicarán las decisiones técnicas tomadas y algunos de los problemas aparecidos y la solución adoptada para cada uno de ellos, teniendo en cuenta y ponderando diversos criterios de calidad y eficiencia. Evidentemente, esta fase del proyecto ha sido la que ha tomado la mayor parte de la planificación temporal del proyecto, además de ser la de mayor complejidad.

El primer paso en la implementación de este proyecto fue tomar una decisión acerca del framework a utilizar. Habiendo decidido que el lenguaje de programación a utilizar sería PHP, seleccioné un pequeño conjunto de ellos: CakePHP, CodeIgniter y Symfony. Para estos tres, realicé una batería de pruebas limitada para escoger el más adecuado para realizar el proyecto. Tras ello, se decidió optar por el framework CakePHP. Posee una menor barrera de entrada, programar para él es más fácil que para el resto y posee una mejor documentación. El que disponga de mayores facilidades a la hora de programar, dadas las herramientas de las que dispone, hace que sea un framework más lento que otros. Pero esta desventaja, aunque es importante, es asumible dadas las ventajas que supone su uso.

Tras seleccionar CakePHP como framework y viendo los prototipos de la interfaz de usuario, decidí optar por cambiar la interfaz que viene por defecto en CakePHP para usar una librería gráfica en JavaScript denominada MochaUI. Además de JavaScript, usa HTML y CSS para renderizar los diversos elementos de una página web. También permite crear ventanas modales en el navegador a partir de una página existente, dando al usuario una sensación de aplicación de escritorio en un entorno web. La segunda característica que hacía bastante interesante la elección de MochaUI como librería gráfica era la posibilidad de crear diferentes paneles para dividir el contenido de la página web, permitiendo crear una disposición para la misma de forma fácil y cómoda.

El sistema de información geográfica elegido para este proyecto es Google Earth, en su versión como complemento para navegadores web. Dispone de un conjunto de funcionalidades bastante potente como poder observar el globo terráqueo libremente en tres dimensiones, poder crear marcadores y líneas también en 3D o poder visualizar modelos COLLADA en el propio globo terráqueo. Además dispone de una API en JavaScript sobre la cual se puede desarrollar aplicaciones de terceros y que se puedan ejecutar en un navegador como en este caso.

Para poder usar el complemento de Google Earth y poder hacer uso de las funcionalidades de la API, es necesario antes de comenzar crear una clave² de API de Google Maps, la cual también es válida para Google Earth. Esta clave permite controlar cómo utiliza la aplicación la API de Google Maps/Earth y funciona como identificador de la aplicación de cara a Google para posibles notificaciones.

A la hora de comenzar con el desarrollo de la aplicación, se optó por usar XAMPP³ como solución combinada para tener de forma operativa rápidamente Apache como servidor web, MySQL como SGBD, Mercury como servidor de correo y PHP. Además de la facilidad a la hora de instalar los diferentes servicios necesarios para el correcto funcionamiento de

²https://code.google.com/apis/console

³http://www.apachefriends.org/es/xampp.html

la aplicación, gracias a XAMPP también es sencillo realizar una correcta administración de los mismos, gracias a las herramientas que incorpora, como por ejemplo PHPMyAdmin.

Tras probar las características básicas del complemento de Google Earth y comprobar su correcto funcionamiento tras incluir la clave de la API en páginas web independientes, se pasó a realizar un test completo en un sitio web de prueba. Tras realizar la prueba el complemento de Google Earth mostraba un mensaje de error y no cargaba. Tras realizar una investigación del problema, la solución adoptada fue que la página donde se encontrase el complemento del Google Earth se mostrase en un *iframe* aparte. De esta forma, el complemento volvía a funcionar de manera correcta.

Una vez asentadas las bases, se paso a la tarea principal de la aplicación: la creación de rutas. En primer lugar se crearon rutas predefinidas y se cargaron en el globo para su visualización haciendo uso de la API en JavaScript. Las rutas se crean haciendo uso del elemento "LineString" de Google Earth. Posteriormente se continuó con los eventos de ratón disponibles en Google Earth para que el usuario pudiera dibujar sobre el globo la ruta que quisiera. Una vez la creación de rutas funcionaba correctamente, se pasó al guardado de la ruta creada. En un primer momento se optó por pasar a través de un parámetro GET las coordenadas de la ruta creada, pero rápidamente se descartó esta opción debido a la limitación en caracteres de la URI (aunque en el RFC 2068⁴ no se especifica tamaño alguno, las implementaciones de los navegadores fijan un máximo -normalmente 2048 caracteresque de ser superado generaban un error "400 Bad Request"). Dado que las rutas con un cierto nivel de complejidad, no las de prueba, superaban dicho límite, sobre todo cuando se hace uso del "arrastrar" del ratón, fue necesario optar por pasar las coordenadas como un parámetro POST dentro del formulario de guardado de la ruta. Además, de esta forma, se incrementa el nivel de seguridad de la aplicación puesto que las coordenadas no son visibles ni se guarda registro de ellas.

Tras la creación de rutas solventada, había que cargarlas para que el usuario pudiera verlas cuando quisiera. Esta parte se realizó haciendo uso del lenguaje KML. Una vez leídos los datos de la ruta desde la base de datos, se construyó un documento KML "al vuelo" con dichos datos junto con otras opciones de estilo, se analizó con la función "parseKML" para generar el objeto y se insertó en el globo visualizándose tal y como fue creado. Esta técnica de crear los documentos KML "al vuelo" en vez de usar unos que existieran físicamente también se emplea en otras funcionalidades de la aplicación, como por ejemplo en los tours.

Una vez el pilar básico de la aplicación funcionaba correctamente, se cambió de ámbito y se realizó la integración entre CakePHP y MochaUI. Afortunadamente, CakePHP permite realizar esto de manera cómoda a través de su sistema de plantillas. Una plantilla es simplemente una vista ya construida. Por lo tanto, sólo es imprescindible crear un documento web incluyendo todas las librerías JavaScript y hojas de estilo CSS de las que hace uso MochaUI. También hay que generar el encabezado y pie del documento usando las etiquetas <div> necesarias. Cualquier controlador dentro del framework CakePHP permite seleccionar una plantilla concreta para cada método si así fuese necesario. Con ello, el documento de salida ya dispone de una estructura que sólo necesita ser completada con los elementos que se

⁴http://www.faqs.org/rfcs/rfc2068.html

añadan en la vista asociada al método del controlador. De esta forma, el documento web queda completo, permitiendo que el usuario pueda verlo adecuadamente.

Otro de los aspectos a tratar en esta sección es la operativa escogida a la hora de tratar con las viñetas o "balloons" en lo relativo a mostrar información al usuario. Estas viñetas de elemento, que son las viñetas básicas del componente de Google Earth, muestran el contenido de la descripción de un elemento usando una ventana de información y aparecen al hacer clic en él, pero el componente elimina cualquiera de los siguientes elementos de la viñeta de un elemento⁵:

- JavaScript
- CSS
- Etiquetas <iframe>
- Etiquetas <embed>
- Etiquetas <object>

Esto es adecuado para ventanas sencillas, donde no hay mucha información y la operatoria es sencilla (normalmente una simple visualización del contenido, generalmente un texto). Sin embargo, en este caso JavaScript y CSS son necesarios debido a que varias vistas requieren de la funcionalidad del primero y de las opciones de estilo del segundo, debido a que las ventanas de la aplicación disponen de un cierto grado de complejidad.

Para estos casos, el componente de Google Earth dispone de otros dos tipos de "balloons": el HTML y el DIV. Se decidió usar el primer tipo de ellos. Usando este tipo de "balloon" es posible usar cualquier código HTML, CSS o JavaScript. Para este tipo de "balloons" simplemente hay que indicar con la función "setContentString" el contenido HTML de lo que se quiere mostrar, ya sea la estructura HTML de un documento web o la opción que finalmente se adoptó: se mostraba un *iframe* pasando como parámetro de entrada la dirección web de la vista que se iba a mostrar. De esta manera los "balloons" se encargaban de enseñar un *iframe* embebido con funcionalidad completa para la vista deseada.

En general, con el fin de obtener los elementos que complementan y describen a una ruta determinada se utilizó JQuery, para desde JavaScript poder realizar peticiones a la base de datos. Ello se hacía usando CakePHP como intermediario, el cual se encargaba de realizar la consulta y se formateaban los datos de salida usando una vista. De esta forma, era posible operar y mostrar los elementos de una ruta en el globo.

Para acabar, me gustaría hablar de una característica bastante útil de CakePHP que viene de la clase *Tree* incorporada en este framework. Gracias a esta clase es posible almacenar una estructura de datos en forma jerárquica en una tabla de una base de datos relacional. Está implementado usando lógica MPTT⁶ y haciendo uso de un árbol binario. La tabla sólo necesita tres columnas para mantener la estructura: *parent*, *left* y *right*. Para poder operar con el árbol, CakePHP ofrece todas las funciones necesarias. Los elementos del

⁵https://developers.google.com/earth/documentation/balloons?hl=es#scrubbing

⁶http://www.sitepoint.com/hierarchical-data-database-2/

globo que puede tener un usuario se pueden estructurar gracias al uso de capas (una forma de carpetas), permitiendo que esta información pueda ser guardada utilizando esta clase.

5.5.1. Capturas de pantalla

A continuación se muestran algunas capturas de pantalla de la aplicación en funcionamiento.

5.5.1.1. Pantalla Inicial



Figura 5.193: Pantalla Inicial

5.5.1.2. Registro de usuarios

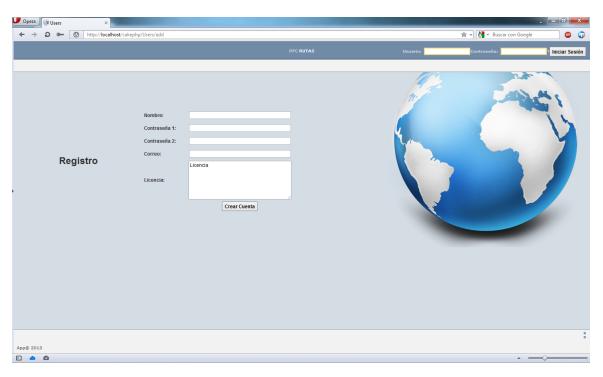


Figura 5.194: Registro de usuarios

5.5.1.3. Pantalla Principal

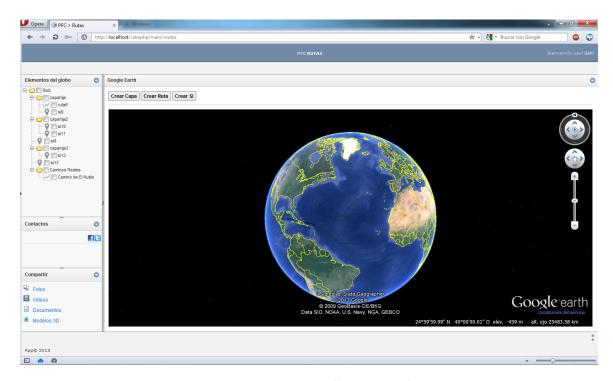


Figura 5.195: Pantalla Principal

Capítulo 6

Conclusiones y trabajo futuro

Una vez dada por concluida la fase de desarrollo y prueba del proyecto, se pueden extraer conclusiones del trabajo realizado, así como realizar un pequeño compendio de posibles ampliaciones a la aplicación creada. Estos dos puntos se tratarán por separado en los apartados que vienen a continuación.

6.1. Conclusiones

Después de haber terminado el desarrollo del proyecto y haber realizado las pruebas sobre la aplicación resultante, podemos aducir que se han cumplido los objetivos principales marcados al inicio del proyecto. Las herramientas software seleccionadas en este proyecto han sido las adecuadas, siguiendo el principio de desarrollar una aplicación con herramientas de software libre de acuerdo con la política de apoyo de la ULPGC a dicho movimiento.

Un aspecto a tener en cuenta es que se ha utilizado Google Earth como sistema georreferenciado en su versión como complemento para navegadores. Google permite el uso gratuito de este complemento bajo unas ciertas condiciones, siendo una de ellas el establecer un número máximo de peticiones a la API de geocodificación: 2.500 peticiones al día. Si está previsto superar este número, es necesario obtener la versión Google Earth Enterprise.

Cada una las herramientas empleadas en este proyecto se ha mostrado lo suficientemente capaz para ofrecer una aplicación que cumpliera con los requisitos planteados al comienzo. El complemento de Google Earth ha sido una herramienta muy fructífera y de una gran potencialidad. Es relativamente fácil programar para él y dispone de muchas utilidades que facilitan ampliar su funcionalidad, como la inclusión de elementos geométricos, informativos, visuales o la gestión de eventos. El único pero es que algunas veces se ralentiza cuando hay muchos elementos en pantalla. Por otro lado, CakePHP se ha mostrado como un framework muy capaz. El poseer una barrera de entrada inferior al resto de frameworks quizás podía entenderse como que era un framework más simple, con menos características que otros de su ámbito, pero se ha comportando como una solución muy eficaz simplificando en gran medida el desarrollo de varios módulos de la aplicación y sin duda lo utilizaría para trabajos similares en el futuro.

Hacer uso de un framework ofrece una gran cantidad de ventajas, siendo la principal simplificar el desarrollo de aplicaciones. También tiene una contrapartida en forma de re-

correr una curva de aprendizaje más o menos difícil y solventar los errores asociados al framework durante el desarrollo. Pero en general, utilizar un framework permite al desarrollador recorrer un camino que ya han pasado otros desarrolladores con más experiencia. Permite dar solución a problemas, iguales o similares, que ya se han resuelto previamente usando soluciones eficaces y de calidad demostrada.

Por el lado del cliente está JavaScript, un lenguaje más complicado a la hora de detectar el origen de los errores, haciendo que el uso de depuradores sea vital para progresar de manera fluida con el desarrollo. La librería que está basada en este lenguaje, JQuery, simplifica mucho la operatoria de ciertas operaciones sobre documentos web y el uso de AJAX facilita en gran medida la programación de la aplicación y la interacción con el usuario de la misma.

A lo largo del desarrollo de este proyecto se han utilizado diversas técnicas y herramientas propias de diferentes ramas de la titulación de Ingeniería en Informática, principalmente de asignaturas como Metodología de la Programación, Bases de Datos, Ingeniería del Software, Sistemas Multimedia y Proyectos Informáticos.

Con la realización de este proyecto, el empleo de una amplia variedad de tecnologías ha sido muy enriquecedor, permitiéndome obtener un valioso nivel de conocimiento de cada una de ellas y de cómo ser capaz de operar con todas ellas de manera conjunta en pos de lograr un producto final con la funcionalidad deseada. A título personal, la tipología de este proyecto, una aplicación web, junto con las tecnologías empleadas en la misma, son de amplia difusión en el mercado laboral en el que nos encontramos a día de hoy. Por lo tanto, puede ser una posible salida laboral dado el bagaje obtenido tras realizar este proyecto.

La realización de este trabajo ha permitido acercarme a un campo que hasta entonces era desconocido para mí como el de los sistemas georreferenciados. Este campo está experimentando un fuerte auge en los últimos años democratizándose su uso, desmarcándose de su propósito inicial, limitado a ser un mera colección de geodatos y evolucionando hacia un sistema de contenidos. Esto se ha conseguido gracias a dos vías, por un lado ha habido una gran expansión de aplicaciones web 2.0 que han aumentado las posibilidades de los sistemas de información geográfica originales; y por el otro tenemos a los usuarios de dichas aplicaciones que han aportado una importante cantidad de contenidos digitales a estas aplicaciones en forma de vídeos, noticias, etc.

6.2. Trabajo futuro

Tras dar por concluida la fase de desarrollo del proyecto y teniendo en cuenta la funcionalidad existente en la aplicación, se puede pasar a comentar algunas posibles ideas y opciones para incrementar la utilidad de la misma. A continuación se discuten algunas de las posibilidades:

- 1. Permitir la compartición de rutas entre los usuarios de la aplicación y que puedan incorporarlas a las que ya tienen.
- 2. Realizar una versión de la aplicación para dispositivos móviles Android e iOS. Ahora es posible debido a que el complemento de Google Earth ya está disponible para ellos.

- 3. Posibilitar que el usuario pueda comunicarse con redes sociales como Twitter, Facebook y otras desde la aplicación.
- 4. Sustituir la librería gráfica MochaUI por otra en el futuro. Es un posible riesgo de cara al mantenimiento de la aplicación debido a que su desarrollo no está avanzando a gran velocidad en estos momentos.
- 5. Permitir que el usuario pueda buscar rutas siguiendo diferentes criterios (fechas, distancia, lugar, etc.) y añadirlas a las suyas propias. Las rutas deberían previsualizarse o mostrar algunos datos sobre ellas para que el usuario pudiera elegir.
- 6. Añadir la posibilidad de que los usuarios de la aplicación puedan comunicarse entre sí. Ya sea a través de paso de mensajes personales, foros o chat.
- 7. Una posibilidad de cara al mantenimiento de la aplicación sería el de disponer de soporte publicitario en forma de banners o anuncios de patrocinadores.
- 8. Añadir soporte multilenguaje para la aplicación. El framework CakePHP lo soporta, de hecho el código de la aplicación está escrito con ello en mente. Faltaría hacer las traducciones y crear un selector de idioma en la aplicación (o hacerlo de manera automática en función del idioma del navegador, por ejemplo).
- 9. Se podría añadir la posibilidad de que la aplicación ofreciera rutas recomendadas a los usuarios según sus gustos personales y las rutas creadas que tuviera hasta el momento.
- 10. Por último, cuando el usuario esté visualizando una ruta la aplicación podría ofrecer estadísticas sobre la misma en una ventana aparte. Las estadísticas podrían ser geográficas como la longitud de la ruta o la diferencia máxima de altura; o de otra índole como el número de usuarios que tienen esa ruta.

Apéndice A

Frameworks de desarrollo

En este apéndice se hará un barrido por los diferentes frameworks de desarrollo para aplicaciones web en PHP. Debido a la gran cantidad de frameworks existentes se hará una descripción de aquellos más importantes y extendidos. Al final de este apéndice se ilustrarán mediante un cuadro una gran batería de frameworks existentes en PHP.

A.1. CakePHP

CakePHP¹ es un marco de desarrollo rápido para PHP, libre y de código abierto. Se trata de una estructura que sirve de base a los programadores para que éstos puedan crear aplicaciones Web. Permite al desarrollador que pueda trabajar de forma estructurada y rápida, sin pérdida de flexibilidad.

CakePHP nació originalmente como un intento de portar la arquitectura y funcionalidad de Ruby on Rails a PHP, algo que otros frameworks han intentado de varias formas. Aunque CakePHP tiene muchas similaridades con el popular framework de Ruby, ha evolucionado de manera independiente de acuerdo a las particularidades del lenguaje PHP.

Con CakePHP el desarrollo web se simplifica pues ofrece las herramientas para que el desarrollador empiece a escribir el código que realmente necesita: la lógica específica de la aplicación. Con una copia de CakePHP, ya se dispone de un importante conjunto de herramientas que hacen que el desarrollador no tenga que comenzar de cero.

CakePHP tiene un equipo de desarrolladores y una comunidad activos, lo que añade valor al proyecto. Con CakePHP, además de no tener que reinventar la rueda, el núcleo de la aplicación se mejora constantemente y está bien probado.

Esta es una lista breve con las características que posee CakePHP:

- Comunidad amigable.
- Licencia flexible.
- Compatible con las versiones de PHP 5.2.6 y superiores.
- CRUD integrado para la interacción con la base de datos.

¹http://www.cakephp.org/

- Andamiaje de código (scaffolding).
- Generación automática de código.
- Arquitectura Modelo Vista Controlador (MVC).
- Despachador de peticiones (dispatcher), con URLs y rutas personalizadas.
- Validación integrada.
- Plantillas rápidas y flexibles (sintaxis de PHP, con ayudantes -helpers-).
- Ayudantes para AJAX, JavaScript, formularios HTML y más.
- Componentes de Email, Cookies, Seguridad, Sesión y Manejo de solicitudes.
- Listas de Control de Acceso (ACL) flexibles.
- Limpieza de datos.
- Caché flexible.
- Localización.
- Funciona en cualquier subdirectorio del sitio web, con poca o ninguna configuración del servidor web (normalmente Apache).

A.2. CodeIgniter

CodeIgniter² es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas.

CodeIgniter está basado ligeramente en el popular patrón de desarrollo Modelo Vista Controlador (MVC). Aunque las clases de vista y el controlador son una parte necesaria del desarrollo en CodeIgniter, los modelos son opcionales.

Suele destacarse que CodeIgniter es más rápido que muchos otros entornos, la mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.

Estas son las principales características de CodeIgniter:

- Sistema basado en Modelo Vista Controlador (MVC).
- Es extremadamente ligero.
- Clases completas para bases de datos con soporte para varias plataformas.
- Bases de datos con soporte Active Record.

²http://ellislab.com/codeigniter

A.2. CODEIGNITER 239

- Validación de formularios y datos.
- Seguridad y Filtrado XSS.
- Manejo de sesiones.
- Clases para el envío de Emails: adjuntos, HTML/Texto, soporte de múltiples protocolos.
- Biblioteca para la manipulación de imágenes (recorte, cambio de tamaño, etc.). Soporta GD, ImageMagick y NetPBM.
- Clase para la subida de ficheros.
- Clase para FTP.
- Localización.
- Paginación.
- Cifrado de Datos.
- Benchmarking.
- Full Page Caching.
- Registro de errores.
- Profiling de la aplicación.
- Clase Calendario.
- Clase "User Agent".
- Clase Codificación.
- Clase Motor de Plantillas.
- Clase Trackback.
- Librería XML-RPC.
- Clase de pruebas unitarias.
- URLs amistosas para los motores de búsqueda.
- Enrutado de URIs flexible.
- Gran biblioteca de funciones "auxiliares".

A.3. Prado

PRADO³ es un framework basado en componentes y con una programación dirigida por eventos, para desarrollos de aplicaciones Web en PHP5. Las siglas de PRADO se corresponden con PHP Rapid Application Development Object-oriented.

El principal objetivo de PRADO es utilizar al máximo la reutilización en la programación Web. Entendiendo por reutilización, no solamente reutilizar el código propio, sino el de otros programadores de una manera fácil. Evita el esfuerzo de reinventar nuevamente la rueda y además posibilita disminuir notablemente los tiempos de desarrollo. La introducción del concepto de componentes tiene este propósito.

Para alcanzar el propósito mencionado, PRADO estipula un protocolo para escribir y usar componentes para construir una aplicación Web. Un componente es una pieza de programa autocontenida y que puede ser reutilizado con una mínima personalización del mismo. Nuevos componentes pueden ser creados por una simple composición de componentes existentes.

Para facilitar la interacción con los componentes, PRADO implementa el paradigma de la programación dirigida por eventos (event-driven) que permite la delegación de comportamientos extensibles a los componentes. Las actividades de los usuarios finales, tales como hacer clic en un botón de un formulario, son capturados como eventos en el lado del servidor. Los métodos o funciones deben ser enlazados a dichos eventos de tal manera que cuando los eventos sucedan, estos son invocados automáticamente para responder a dicho evento. Comparado con la programación Web tradicional en la cual los desarrolladores tienen que tratar directamente con las variables POST y GET, la programación dirigida por eventos ayuda a los desarrolladores enfocarse mejor en las necesidades lógicas y reducir significativamente el código de bajo nivel repetitivo.

En resumen, desarrollar aplicaciones Web con PRADO principalmente involucra instantáneamente tipos de componentes predesarrollados, configurarlos mediante sus propiedades, responder a sus eventos escribiendo funciones manipuladoras de los mismos, y agrupándolos dentro de paginas para la aplicación. Es muy similar al kit de herramientas RAD de Borland Delphi y Microsoft Visual Basic, que son utilizadas para desarrollar aplicaciones (Interfaces Gráficas de Usuario, GUI) de escritorio.

Estas son las principales características de PRADO:

- Reutilización: Los códigos que se rigen por el protocolo basado en componentes de PRADO son altamente reutilizables. Esto beneficia a los equipos de desarrollo a largo plazo, ya que pueden reutilizar sus trabajos anteriores e integrar otras partes de trabajo con facilidad.
- Programación dirigida por eventos: Las actividades del usuario final, tales como como hacer clic en un botón de enviar, son capturadas como eventos del servidor permitiendo que los desarrolladores tengan un mejor enfoque en las interacciones del usuario.
- Integración de equipo: La capa de presentación y la capa lógica son almacenados por

³http://www.pradosoft.com/

A.4. SYMFONY

separado. Las aplicaciones en PRADO pueden ser personalizadas con temas.

 Controles web potentes: PRADO viene con un conjunto de componentes que se ocupan de las interfaces de usuario Web. Se puede crear páginas web altamente interactivas con unas pocas líneas de código.

- Fuerte soporte de bases de datos: Desde la versión 3.1, PRADO ha sido equipada con soporte total de bases de datos escrito de forma nativa y, por tanto, encaja con el resto del framework PRADO. De acuerdo a la complejidad de los objetos de negocio, se puede optar por utilizar la PDO simple, basada en el acceso a los datos, el ampliamente conocido Active Record o el mapa completo de los objetos del negocio SqlMap.
- Soporte de AJAX sin fisuras: El uso de AJAX en PRADO se ha facilitado gracias a los Active Controls, introducidos desde la versión 3.1. Se puede escribir una aplicación AJAX sin escribir una sola línea de código JavaScript. De hecho, la utilización de los Active Controls no es muy diferente a la utilización de componentes no AJAX.
- Soporte de I18N y L10N: PRADO incluye soporte completo para crear aplicaciones con múltiples idiomas.
- Compatibilidad XHTML: Las páginas Web generadas por PRADO son compatibles con XHTML.
- Albergar trabajos ya existentes: PRADO es un framework genérico, con especial atención a la capa de presentación. No excluye a desarrolladores que hacen uso de la mayoría de las actuales bibliotecas de clase o kits de herramientas. Por ejemplo, uno puede usar ADOdb o Creole para tratar con base de datos en su aplicación PRADO.
- Otras características: Potente gestión de errores y/o excepciones, registro de mensajes; caché genérico y memoria caché de salida selectiva, manejo de errores personalizable y localizable, autentificación y autorización extensibles, prevención de medidas de seguridad tales como Cross-Site Script (XSS), protección de cookies, etc.

A.4. Symfony

Symfony⁴ es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador (MVC). Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente en PHP 5.3. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL

⁴http://symfony.com/

Server. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Las principales características de Symfony son:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Sencillo de usar en la mayoría de casos, aunque es preferible para el desarrollo de grandes aplicaciones Web que para pequeños proyectos.
- Aunque utiliza MVC, tiene su propia forma de trabajar en este aspecto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una potente línea de comandos que facilitan generación de código, lo cual contribuye a ahorrar tiempo de trabajo.
- Permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización.
- La presentación usa templates y layouts que pueden ser construidos por diseñadores de HTML que no posean conocimientos del framework.
- Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en las base de datos y una mejor experiencia para el usuario.
- El manejo de cache reduce el uso de banda ancha y la carga del servidor.
- La facilidad de soportar autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.

A.5. YII

 El enrutamiento y las URLs inteligentes hacen amigable las direcciones de las páginas de la aplicación.

- Las listas son más amigables, ya que permite la paginación, clasificación y filtrado automáticos.
- Los plugins proveen un alto nivel de extensibilidad.
- La interacción con AJAX es mucho más sencilla.

A.5. Yii

Yii⁵ es un framework PHP basado en componentes de alto rendimiento para desarrollar aplicaciones Web a gran escala. Permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo. Está escrito en PHP5 y es software libre liberado bajo una licencia BSD. Tiene la concepción de hacer las cosas de manera sencilla, elegante y rápidas, ayudando con esto a construir aplicaciones eficientes, que pueden ser mantenidas fácilmente y escalables.

Sigue el patrón MVC, lo que garantiza una clara separación de la lógica del negocio y la presentación. Fomenta la sinergia en equipos de desarrollo y promueve métodos de trabajo que son ideales para combinar con metodologías ágiles.

Algunas características de Yii incluyen:

- Patrón de diseño Modelo Vista Controlador (MVC).
- Database Access Objects (DAO), query builder, Active Record y migración de base de datos.
- Integración con jQuery.
- Entradas de Formulario y validación.
- Widgets de AJAX, como autocompletado de campos de texto y demás.
- Soporte de Autenticación incorporado. Además soporta autorización vía "Role-Based Access Control" (RBAC) jerárquico.
- Personalización de aspectos y temas.
- Generación compleja automática de WSDL, especificaciones y administración de peticiones Web service.
- Internacionalización y localización (I18N y L10N). Soporta traducciones, formato de fecha y hora, formato de números y localización de la vista.
- Esquema de caching por capas. Soporta el cache de datos, cache de páginas, cache por fragmentos y contenido dinámico. El medio de almacenamiento del cache puede ser cambiado.

⁵http://www.yiiframework.com/

- Manejo de errores y logging. Los errores son manejados y personalizados, y los registros de mensajes pueden ser categorizados, filtrados y movidos a diferentes destinos.
- Las medidas de seguridad incluyen la prevención Cross-Site Scripting (XSS), prevención Cross-Site Request Forgery (CSRF), prevención de la manipulación de cookies, etc.
- Herramientas para pruebas unitarias y funcionales basados en PHPUnit y Selenium.
- Generación automática de código para el esqueleto de la aplicación, aplicaciones CRUD, etc.
- Generación de código por componentes de Yii y la herramienta por linea de comandos cumple con los estándares de XHTML.
- Cuidadosamente diseñado para trabajar bien con código de terceros. Por ejemplo, es posible usar el código de PHP o Zend Framework en una aplicación Yii.

A.6. Zend

Zend⁶ Framework (ZF) es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. ZF es una implementación que usa código 100 % orientado a objetos. La estructura de los componentes de ZF es algo único, cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado.

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible framework de aplicaciones web al combinarse. ZF ofrece un gran rendimiento y una implementación del patrón MVC, una abstracción de base de datos fácil de usar, un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Otros componentes proporcionan autentificación de usuarios y autorización. También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los servicios web más populares.

Otras características a destacar son:

- Cuenta con módulos para manejar archivos PDF, canales RSS, Web Services (Amazon, Flickr, Yahoo), etc.
- El framework también incluye objetos para las diferentes bases de datos, por lo que es extremadamente simple para consultar bases de datos, sin tener que escribir ninguna consulta SQL.
- Una solución para el acceso a base de datos que balancea el ORM con eficiencia y simplicidad.

⁶http://www.framework.zend.com/

- Completa documentación y tests de alta calidad.
- Soporte avanzado para i18n (internacionalización).
- Un buscador compatible con Lucene.
- Robustas clases para autenticación y filtrado de entrada.
- Clientes para servicios web, incluidos Google Data APIs y StrikeIron.
- Muchas otras clases útiles para hacerlo tan productivo como sea posible.

A.7. Listado de frameworks PHP

A continuación se muestra una recopilación de la mayoría de frameworks en PHP para aplicaciones web existentes en el mercado, tanto libres como comerciales. Cada uno de ellos presenta una diversa lista de funcionalidades, desde las más básicas hasta las más avanzadas que poseen algunos frameworks. Resaltar que el grado de madurez de todos los frameworks aquí listados varía, lo que hace necesario un trabajo previo de análisis a la hora de seleccionar alguno de ellos cuando se vaya a realizar una aplicación. En conclusión, la tabla resultante, obtenida a partir de diversas fuentes⁷ 8 entre otras, es la siguiente:

Cuadro A.1: Listado de frameworks PHP

Akelos	ash.MVC	CakePHP	CodeIgniter
DIY	Fat-Free Framework (F3)	FuelPHP	Fusebox
Hazaar MVC	Horde	HTML5 Builder	KumbiaPHP
Lithium	Osezno PHP Framework	PHP on TRAX	PHPDevShell
PHPixie	PhpOpenbiz	PHPWork	Prado
Seagull	Symfony	TYPO3 Flow	WACT
WASP	Yii	Zend	Zeta Components

⁷http://www.phpframeworks.com/

⁸http://en.wikipedia.org/wiki/Category:PHP_frameworks

Apéndice B

Glosario

El glosario es un artefacto obtenido en la disciplina de requisitos dentro de la fase de inicio según la metodología del Proceso Unificado de Desarrollo.

Se puede utilizar un glosario para definir términos comunes importantes que los analistas (y otros desarrolladores) utilizan al describir el sistema. Un glosario es muy útil para alcanzar un consenso entre los desarrolladores relativo a la definición de los diversos conceptos y nociones, y para reducir en general el riesgo de confusiones.

Habitualmente podemos obtener un glosario a partir de un modelo del negocio o de un modelo del dominio, pero debido a que es menos formal (no incluye clases o relaciones explícitas), es más fácil de mantener y es más intuitivo para utilizarlo con terceras personas externas, como usuarios y clientes. Además, un glosario tiende a estar más centrado en el sistema que se va a construir, en lugar de en su contexto, como es el caso de los modelos del negocio o del dominio.

Definiciones

Globo Terráqueo Representación del mundo por parte de la aplicación donde se muestran las rutas y sitios de interés del usuario.

Capa Elemento contenedor a modo de carpeta que almacena rutas y sitios de interés del usuario.

Árbol de capas Las capas también pueden guardar otras capas, lo que da lugar a una jerarquía de capas en forma de árbol.

Ruta Línea formada por un conjunto de puntos y los segmentos que los unen definiendo un itinerario.

Ruta Marítima La que atraviesa el mar usando algún tipo de embarcación.

Ruta Aérea La que surca el aire usando algún tipo de aeronave.

Ruta Terrestre La que recorre algún trozo de terreno, ya sea caminando o usando algún tipo de vehículo.

Usuario Persona que usa la aplicación.

Perfil del Usuario Conjunto de datos que representan al usuario en la aplicación, así como al resto de usuarios del sistema.

GPS Sistema de posicionamiento global que permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo.

Brújula Instrumento que señala el norte magnético y que sirve para determinar las direcciones de la superficie terrestre.

Tour Visualización automática de una ruta, a modo de recorrido donde los movimientos están grabados.

Comunidad Conjunto de usuarios de la aplicación vinculadas por intereses comunes.

Autentificación (Login) Proceso por el cual el usuario se identifica en el sistema, indicando quién es.

Chat Comunicación escrita realizada de manera instantánea a través de Internet entre dos usuarios de la aplicación que estén conectados en el mismo instante.

Email Mensaje electrónico que permite a dos usuarios comunicarse entre sí. No es necesario que ambos estén conectados al mismo tiempo.

Foto Archivo digital que contiene una imagen.

Modelo 3D Archivo digital que representa un objeto en tres dimensiones.

Almacén de elementos 3D La aplicación contendrá un conjunto de modelos 3D básico que los usuarios podrán utilizar para documentar sus rutas.

Vídeo Archivo digital que contiene un vídeo que puede reproducirse en la aplicación.

Documento Archivo digital que contiene un escrito. También puede albergar imágenes, tablas, etc.

Sitio de interés Lugar o terreno determinado de valor en una ruta.

Sitio Cultural Aquel que permite al usuario aumentar su conocimiento.

Sitio Arquitectónico Es un sitio donde su arquitectura establece un estilo diferente, único e impresionante.

Sitio Naturaleza Normalmente aquel donde hay abundante vegetación.

Sitio Religioso Aquel donde hay algún edificio donde se practica un oficio religioso.

Sitio Urbanístico Aquel donde hay alguna obra escultórica o decorativa.

Sitio Comercial Aquel donde haya algún negocio útil para el usuario durante el recorrido de la ruta.

Accidente Geográfico Unidad geomorfológica o forma de la superficie terrestre.

Índice de figuras

2.1.	ArcGIS	8
2.2.	Bing Maps	9
2.3.	Google Earth	0
2.4.	Google Maps	3
2.5.	SkylineGlobe	4
2.6.	Capaware	5
2.7.	GeoPista	6
2.8.	Grass SIG	17
2.9.	gvSIG	8
2.10.	SEXTANTE	9
2.11.	Marble	20
2.12.	NASA World Wind	21
2.13.	OpenJUMP	22
2.14.	Quantum GIS	23
2.15.	CompeGPS Land	24
2.16.	Global Mapper	25
2.17.	Google Earth Pro	26
2.18.	OziExplorer3D	27
4.1.	Fases PUD	39
4.2.	Planificación temporal	
4.3.	Planificación desglosada	17
F 1	M 11 111 · · · D · · · 1	- 1
5.1.	Modelo del dominio - Principal	
5.2.	Modelo del dominio - Elementos del globo	
5.3.	Modelo del dominio - Web 2.0	
5.4.	Modelo del dominio - Sitios de interés	
5.5.	Modelo del dominio - Sitios de interés culturales	
5.6.	•	56
5.7.		56
5.8.	<u> </u>	57
5.9.		57
5.10.	Modelo del dominio - Sitios de interés comercial	57
5.11.	Modelo del dominio - Accidentes geográficos	58

5.12.	Modelo del dominio - Accidentes geográficos (altura)
5.13.	Modelo del dominio - Accidentes geográficos (inclinación)
5.14.	Modelo del dominio - Accidentes geográficos (erosivos)
5.15.	Modelo del dominio - Accidentes geográficos (fluviales) 59
5.16.	Modelo del dominio - Accidentes geográficos (montañosos) 59
5.17.	Modelo del dominio - Accidentes geográficos (glaciares) 59
5.18.	Modelo del dominio - Accidentes geográficos (volcánicos) 60
5.19.	Actores del sistema
5.20.	Usuario No Registrado
5.21.	Usuario Registrado
5.22.	Gestión de Capas
5.23.	Visualizar Ruta
5.24.	Gestionar Rutas
5.25.	Gestionar Sitios de Interés
5.26.	Gestionar Tours
5.27.	Compartición
5.28.	Comunicación
5.29.	Gestión de Contactos
5.30.	Pantalla Inicial
5.31.	Registro
5.32.	Recordar Contraseña
5.33.	Pantalla Inicial
5.34.	Darse de Baja
5.35.	Modificar Datos Personales
5.36.	Modificar Contraseña
5.37.	Información del Usuario
5.38.	Arquitectura de paquetes
5.39.	Clases del análisis
5.40.	Clases de Análisis: Registrarse
5.41.	Clases de Análisis: Iniciar Sesión
5.42.	Clases de Análisis: Recordar Contraseña
5.43.	Clases de Análisis: Modificar Datos Personales
5.44.	Clases de Análisis: Cambiar Contraseña
5.45.	Clases de Análisis: Darse de Baja
5.46.	Clases de Análisis: Cerrar Sesión
5.47.	Clases de Análisis: Añadir Usuario
5.48.	Clases de Análisis: Eliminar Usuario
5.49.	Clases de Análisis: Ver Usuario
5.50.	Clases de Análisis: Modificar Usuario
5.51.	Clases de Análisis: Crear Capa
5.52.	Clases de Análisis: Modificar Capa
5.53.	Clases de Análisis: Renombrar Capa

ÍNDICE DE FIGURAS 251

5.54.	Clases de Análisis: Eliminar Capa
5.55.	Clases de Análisis: Insertar Puntos
5.56.	Clases de Análisis: Insertar Elemento Audiovisual
5.57.	Clases de Análisis: Guardar Ruta
5.58.	Clases de Análisis: Modificar Ruta
5.59.	Clases de Análisis: Visualizar Ruta (contenedor)
5.60.	Clases de Análisis: Eliminar Ruta
5.61.	Clases de Análisis: Renombrar Ruta
5.62.	Clases de Análisis: Añadir Sitio de Interés
5.63.	Clases de Análisis: Visualizar Sitio de Interés
5.64.	Clases de Análisis: Eliminar Sitio de Interés
5.65.	Clases de Análisis: Modificar Sitio de Interés
5.66.	Clases de Análisis: Renombrar Sitio de Interés
5.67.	Clases de Análisis: Crear Tour Automático
5.68.	Clases de Análisis: Visualizar Tour
5.69.	Clases de Análisis: Eliminar Tour
5.70.	Clases de Análisis: Modificar Tour
5.71.	Clases de Análisis: Renombrar Tour
5.72.	Clases de Análisis: Crear Tour Manual
5.73.	Clases de Análisis: Activar Capa (Crear)
5.74.	Clases de Análisis: Desplegar Capa
5.75.	Clases de Análisis: Visualizar Información
5.76.	Clases de Análisis: Ir a
5.77.	Clases de Análisis: Ver Ruta
5.78.	Clases de Análisis: Ver Tour
5.79.	Clases de Análisis: Ver Sitios de Interés
5.80.	Clases de Análisis: Seleccionar Capas
5.81.	Clases de Análisis: Compartir Rutas
5.82.	Clases de Análisis: Compartir Archivos
5.83.	Clases de Análisis: Rutas Favoritas
5.84.	Clases de Análisis: Presentar Otras Rutas
5.85.	Clases de Análisis: Comunicarse con otras Redes Sociales
5.86.	Clases de Análisis: Escribir Comentarios
5.87.	Clases de Análisis: Chatear
5.88.	Clases de Análisis: Enviar Email
5.89.	Clases de Análisis: Crear Comunidades
5.90.	Diagrama de despliegue
5.91.	Arquitectura del software
5.92.	Interacción de Objetos: Registrarse
5.93.	Diagrama de clases: Registrarse
5.94.	Interacción de Objetos: Iniciar Sesión
5.95.	Diagrama de clases: Iniciar Sesión

5.96.	Interacción de Objetos: Recordar Contraseña
5.97.	Diagrama de clases: Recordar Contraseña
5.98.	Interacción de Objetos: Modificar Datos Personales
5.99.	Diagrama de clases: Modificar Datos Personales
5.100.	Interacción de Objetos: Cambiar Contraseña
5.101.	Diagrama de clases: Cambiar Contraseña
5.102.	Interacción de Objetos: Darse de Baja
5.103.	Diagrama de clases: Darse de Baja
5.104.	Interacción de Objetos: Cerrar Sesión
5.105.	Diagrama de clases: Cerrar Sesión
5.106.	Interacción de Objetos: Añadir Usuario
5.107.	Diagrama de clases: Añadir Usuario
5.108.	Interacción de Objetos: Eliminar Usuario
5.109.	Diagrama de clases: Eliminar Usuario
5.110.	Interacción de Objetos: Ver Usuario
5.111.	Diagrama de clases: Ver Usuario
5.112.	Interacción de Objetos: Modificar Usuario
5.113.	Diagrama de clases: Modificar Usuario
5.114.	Interacción de Objetos: Crear Capa
5.115.	Diagrama de clases: Crear Capa
5.116.	Interacción de Objetos: Modificar Capa
5.117.	Diagrama de clases: Modificar Capa
5.118.	Interacción de Objetos: Renombrar Capa
5.119.	Diagrama de clases: Renombrar Capa
5.120.	Interacción de Objetos: Eliminar Capa
5.121.	Diagrama de clases: Eliminar Capa
5.122.	Interacción de Objetos: Insertar Puntos
5.123.	Diagrama de clases: Insertar Puntos
5.124.	Interacción de Objetos: Insertar Elemento Audiovisual 192
5.125.	Diagrama de clases: Insertar Insertar Elemento Audiovisual 192
5.126.	Interacción de Objetos: Guardar Ruta
5.127.	Diagrama de clases: Guardar Ruta
5.128.	Interacción de Objetos: Modificar Ruta
5.129.	Diagrama de clases: Modificar Ruta
5.130.	Interacción de Objetos: Visualizar Ruta (contenedor)
5.131.	Diagrama de clases: Visualizar Ruta (contenedor)
5.132.	Interacción de Objetos: Eliminar Ruta
5.133.	Diagrama de clases: Eliminar Ruta
	Interacción de Objetos: Renombrar Ruta
5.135.	Diagrama de clases: Renombrar Ruta
5.136.	Interacción de Objetos: Añadir Sitio de Interés
5.137.	Diagrama de clases: Añadir Sitio de Interés

5.138.	Interacción de Objetos: Visualizar Sitio de Interés
5.139.	Diagrama de clases: Visualizar Sitio de Interés
5.140.	Interacción de Objetos: Eliminar Sitio de Interés
5.141.	Diagrama de clases: Eliminar Sitio de Interés
5.142.	Interacción de Objetos: Modificar Sitio de Interés
5.143.	Diagrama de clases: Modificar Sitio de Interés
5.144.	Interacción de Objetos: Renombrar Sitio de Interés
	Diagrama de clases: Renombrar Sitio de Interés
5.146.	Tipos de Tours
5.147.	Interacción de Objetos: Crear Tour Automático
5.148.	Diagrama de clases: Crear Tour Automático
5.149.	Interacción de Objetos: Visualizar Tour
5.150.	Diagrama de clases: Visualizar Tour
5.151.	Interacción de Objetos: Eliminar Tour
5.152.	Diagrama de clases: Eliminar Tour
5.153.	Interacción de Objetos: Modificar Tour
5.154.	Diagrama de clases: Modificar Tour
5.155.	Interacción de Objetos: Renombrar Tour
5.156.	Diagrama de clases: Renombrar Tour
5.157.	Interacción de Objetos: Crear Tour Manual
5.158.	Diagrama de clases: Crear Tour Manual
5.159.	Interacción de Objetos: Activar Capa (Crear)
5.160.	Diagrama de clases: Activar Capa (Crear)
5.161.	Interacción de Objetos: Desplegar Capa
5.162.	Diagrama de clases: Desplegar Capa
5.163.	Interacción de Objetos: Visualizar Información
5.164.	Diagrama de clases: Visualizar Información
	Interacción de Objetos: Ir a
5.166.	Diagrama de clases: Ir a
5.167.	Interacción de Objetos: Ver Ruta
5.168.	Diagrama de clases: Ver Ruta
5.169.	Interacción de Objetos: Ver Tour
5.170.	Diagrama de clases: Ver Tour
5.171.	Interacción de Objetos: Ver Sitios de Interés
5.172.	Diagrama de clases: Ver Sitios de Interés
5.173.	Interacción de Objetos: Seleccionar Capas
5.174.	Diagrama de clases: Seleccionar Capas
5.175.	Interacción de Objetos: Compartir Rutas
5.176.	Diagrama de clases: Compartir Rutas
	Interacción de Objetos: Compartir Archivos
5.178.	Diagrama de clases: Compartir Archivos
5.179.	Interacción de Objetos: Rutas Favoritas

5.180.	Diagrama de clases: Rutas Favoritas
5.181.	Interacción de Objetos: Presentar Otras Rutas
5.182.	Diagrama de clases: Presentar Otras Rutas
5.183.	Interacción de Objetos: Comunicarse con otras Redes Sociales $\ \ldots \ \ldots \ 222$
5.184.	Diagrama de clases: Comunicarse con otras Redes Sociales $\dots \dots 222$
5.185.	Interacción de Objetos: Escribir Comentarios
5.186.	Diagrama de clases: Escribir Comentarios
5.187.	Interacción de Objetos: Chatear
5.188.	Diagrama de clases: Chatear
5.189.	Interacción de Objetos: Enviar Email
5.190.	Diagrama de clases: Enviar Email
5.191.	Interacción de Objetos: Crear Comunidades $\dots \dots \dots$
5.192.	Diagrama de clases: Crear Comunidades
5.193.	Pantalla Inicial
5.194.	Registro de usuarios
5.195.	Pantalla Principal

Índice de cuadros

3.1.	Herramientas Software
4.1.	Gestión del PFC
4.2.	Realización y tramitación de la propuesta
4.3.	Cuestiones previas
4.4.	Desarrollo del PFC
4.5.	Validación y publicidad
4.6.	Presentación y defensa
4.7.	Resumen de la planificación
4.8.	Costes de personal
4.9.	Costes inventariables
4.10	. Costes fungibles
4.11	. Costes indirectos
4.12	. Coste total
A.1.	Listado de frameworks PHP

Bibliografía

- [BRJ99] Grady Booch, James Rumbaugh y Ivar Jacobson, *UML: El Lenguaje Unificado de Modelado*, Addison Wesley, 1999.
- [DM02] C. W. Dawson y G. Martín, El Proyecto de Fin de Carrera en Ingeniería Informática. Una Guía para el Estudiante, Prentince Hall, 2002.
- [Fou13] The CakeSoftware Foundation, *The CakePHP Cookbook*, http://book.cakephp.org/2.0/en/index.html, 2013.
- [GHJV03] Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, Patrones de Diseño: Elementos de Software Orientado a Objetos Reutilizable, Addison Wesley, 2003.
- [Goo12] Google, Google Earth API, https://developers.google.com/earth/, 2012.
- [Gro13] The PHP Group, Documentación de PHP, http://www.php.net/manual/es/, 2013.
- [Hil06] Linda L. Hill, Georeferencing: The Geographic Associations of Information, 2006.
- [JBR99] Ivar Jacobson, Grady Booch y James Rumbaugh, El Proceso Unificado de Desarrollo Software, Addison Wesley, 1999.
- [jF13] The jQuery Foundation, JQuery API, http://api.jquery.com/, 2013.
- [KK03] Per Kroll y Philippe Kruchten, The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP, Addison-Wesley Educational Publishers, 2003.
- [Lar03] Craig Larman, UML y Patrones: Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado, Prentice Hall, 2003.
- [MR04] Juan Mendez-Rodríguez, Notas sobre Producción Documental Científica y Técnica, Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería, ULPGC, 2004.
- [Ora11] Oracle, MySQL 5.5 Reference Manual, 2011.
- [Pre06] Roger S. Pressman, Ingeniería del Software: Un Enfoque Práctico, McGraw-Hill, 2006.

258 BIBLIOGRAFÍA

[RS07] Doug Rosenberg y Matt Stephens, Use Case Driven Object Modeling with UML: Theory and Practice., Apress, 2007.

- [SKS06] Abraham Silverschatz, Henry F. Korth y S. Sudarshan, Fundamentos de Bases de Datos, McGraw-Hill, 2006.
- [Var13] Varios, Utility Libraries for the Google Earth API, http://code.google.com/p/earth-api-utility-library/, 2013.
- [W3s13a] W3schools, Manual de CSS, http://www.w3schools.com/css/, 2013.
- [W3s13b] _____, Manual de HTML, http://www.w3schools.com/html/, 2013.
- [Wik13] Fundación Wikimedia, Wikipedia: The Free Encyclopedia, http://www.wikipedia.org/, 2013.
- [WM07] Paul Wilton y Jeremy McPeak, *Beginning Javascript*, Wiley Publishing, Inc., 2007.