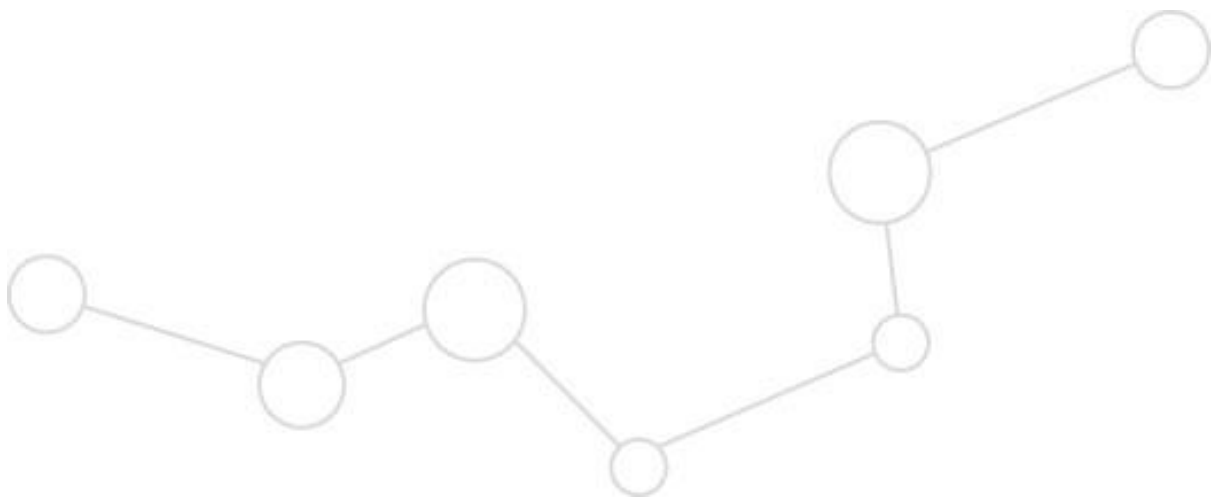




# Rocódromos - Desarrollo de una capa de Realidad Aumentada para Layar

## Memoria Trabajo Fin de Carrera



Elaborado por:

**Nicolás Luque Sáez**

Consultor:

**Roberto Ramírez Vique**

Ingeniería Técnica en Informática de Gestión

Curso 2013 - 2014

### Gestión de cambios

<b>Tipo Documento</b>	Memoria Trabajo Fin de Carrera
<b>Descripción</b>	Documentación técnica del Sistema de Información basado en desarrollar una capa de Realidad Aumentada sobre la plataforma Layar (Rocódromos).
<b>Fecha</b>	02/10/2013
<b>Versión documento</b>	1.4
<b>Estado</b>	Finalizado
<b>Distribuido a</b>	Roberto Ramírez Vique - UOC

Versión	Descripción de la versión	Fecha
1.0	Versión inicial.	02/10/2013
1.1	Modificación para añadir casos de uso y pantallas prototipo parte servidor.	11/11/2013
1.2	Modificación para añadir detalles de la implementación, implantación y pruebas.	06/12/2013
1.3	Modificación para añadir conclusiones, futuras ampliaciones y mejoras.	20/12/2013
1.4	Modificación para añadir reseña con la publicación del código fuente del servidor Web de POIs.	03/01/2014

# Índice

1.	Introducción .....	5
1.1	Términos y abreviaturas .....	6
1.2	Descripción del proyecto .....	6
1.3	Estudio de alternativas tecnológicas.....	7
1.3.1	Wikitude.....	8
1.3.2	Layar .....	9
1.4	Planificación temporal .....	11
1.5	Riesgos .....	12
2.	Requisitos iniciales.....	13
2.1	Visualización de rocódromos sobre la imagen real .....	13
2.2	Mostrar detalles del rocódromo e información ampliada .....	13
2.3	Mostar información de cómo llegar.....	14
2.4	Gestionar los datos del rocódromo .....	14
3.	Análisis del sistema.....	15
3.1	Objetivos del sistema .....	15
3.1.1	<OBJ-001> Visualizar rocódromos .....	15
3.1.2	<OBJ-002> Ampliar detalles del rocódromo .....	16
3.1.3	<OBJ-003> Mostar información de cómo llegar.....	16
3.1.4	<OBJ-004> Gestionar datos de los POIs .....	17
3.2	Catálogo de requisitos del sistema .....	17
3.2.1	Nomenclatura .....	18
3.2.2	Requisitos de información.....	19
3.2.3	Requisitos funcionales.....	19
3.2.4	Requisitos no funcionales.....	31
4.	Diseño arquitectura .....	34
4.1	Arquitectura de Layar .....	34
4.2	Servidor Web PHP .....	35
4.3	Google Maps API .....	36
4.4	Funcionamiento y conexiones entre servidores (Layar, Servidor de POI, Google Maps).....	37
5.	Prototipo de interfaz de usuario.....	39
5.1	Pantallas del servidor de POIs.....	39
5.1.1	Pantalla de acceso .....	39
5.1.2	Pantalla principal de datos de la capa .....	40
5.1.3	Pantalla detalle de capa (listado de POIs) .....	40
5.1.4	Pantalla de alta/edición/detalle de POI .....	41
5.2	Pantallas de la capa Layar .....	41
5.2.1	Pantalla búsqueda de capas.....	41
5.2.2	Pantalla ajustes de usuario.....	42
5.2.3	Pantalla de capas recientes.....	42
5.2.4	Pantalla ajustes de capa (Rango de búsqueda).....	43
5.2.5	Pantalla principal de capa.....	44
5.2.6	Pantalla de ampliar información y cómo llegar.....	45
6.	Implementación.....	46
6.1	Servidor de POIs .....	46
6.1.1	Estructura de carpetas.....	46
6.1.2	Clases modificadas.....	46
6.2	Capa Layar – Rocódromos.....	50
6.2.1	Nueva capa (Create a Layer).....	50
6.2.2	Punto de acceso (API endpoint) .....	51

6.2.3	Iconos y detalle (Listing & indexing).....	51
6.2.4	Diseño y estilos (Look & feel) .....	53
6.2.5	País (Country) .....	54
6.2.6	Filtros (Filters).....	54
6.2.7	Prueba y publicación de la capa .....	55
7.	Implantación.....	56
7.1	Instalación del servidor de POIs .....	56
7.1.1	Pasos para la instalación .....	56
7.2	Instalación navegador Layar.....	57
7.2.1	Pasos para la instalación .....	57
7.3	Prueba de capa Layar - Rocódromos .....	57
7.3.1	Pasos para probar la aplicación.....	58
8.	Conclusiones .....	59
9.	Nuevas líneas de ampliación y mejoras .....	59
10.	Bibliografía .....	60

# 1. Introducció

Este documento se elabora como resultado del proyecto de Desarrollo del Sistema de Información que constituye la Capa de Realidad Aumentada – Rocódromos, con información actualizada de datos sobre POI (*point of interest, puntos de interés*), realizado por Nicolás Luque Sáez como Trabajo Final de Carrera (en adelante TFC) para la Universitat Oberta de Catalunya.

Dicho sistema de información se denomina como “**Rocódromos** – Desarrollo de una capa de Realidad Aumentada para Layar”. Para simplificar a partir de ahora en este documento se denotará simplemente como “Sistema”.

El contenido del documento se estructura en capítulos de la siguiente manera:

- Introducción, descripción del proyecto, planificación temporal y riesgos.
- Extracción de requisitos del Sistema, como punto de partida del proceso de análisis.
- Análisis de los requisitos del Sistema.
- Diseño del Sistema: modelo tecnológico, diseño de arquitectura y prototipo del Sistema.
- Implementación.
- Implantación.

## 1.1 Términos y abreviaturas

Abreviatura / Término	Descripción
POI	Punto de interés (del inglés <i>Point Of Interest</i> ), son puntos de localización GPS. Es un punto de ubicación específica que alguien puede encontrar útil o interesante.  En español se puede encontrar como PDI.
GPS	Sistema de posicionamiento global por satélite (del inglés <i>Global Positioning System</i> ).
JSON	Acrónimo de <i>JavaScript Object Notation</i> , es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.
PHP	Lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.
XML	Siglas en inglés de <i>eXtensible Markup Language</i> ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.
HTTP	Siglas en inglés de <i>Hipertext Transfer Protocol</i> (protocolo de transferencia de hipertexto) es el protocolo utilizado en una transacción del a World Wide Web (www)
URL	Siglas en inglés de <i>Uniform Resource Locator</i> (localizador de recursos uniforme). Es una secuencia de caracteres estandarizado que se usa para nombrar recursos en Internet para su localización. Dirección de Internet.

## 1.2 Descripción del proyecto

La idea de realizar una capa de Realidad Aumentada para ubicar los rocódromos como puntos de interés (en adelante POI, *Point Of Interest*) en el mapa surge de la problemática de no tener posibilidades de escalar en roca cada vez que se quiere por lo que es necesario un lugar dedicado al entrenamiento. Es por ello que, para facilitar la ubicación de los diferentes rocódromos de una ciudad, obtener más información sobre ellos e incluso recibir indicaciones de cómo llegar hasta el mismo, nace la idea de este proyecto.

El proyecto a desarrollar está orientado a la detección y superposición de rocódromos, como POIs existentes en una ciudad, sobre la imagen transmitida en tiempo real a través de la cámara de los dispositivos móviles. Asimismo, se pretende poder ampliar la información de dichos rocódromos, así como enlazar con Google Maps para que nos indique sobre el mapa la mejor ruta para llegar haciendo uso del geo posicionamiento.

Para darle un valor añadido a la aplicación se pretende que, a través de una interfaz web, se pueda gestionar el fichero XML con los POIs por un usuario administrador.

## 1.3 Estudio de alternativas tecnológicas

Antes de comenzar exponer el estudio de alternativas tecnológicas que se ha realizado para seleccionar cuál de las distintas alternativas cumplía con las necesidades del presente TFC, es interesante acercarnos más a la realidad aumentada.

Por definición, **realidad aumentada** (RA) es el término que se usa para definir una visión directa o indirecta de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real. Consiste en un conjunto de dispositivos que añaden información virtual a la información física ya existente, es decir, añadir una parte sintética virtual a lo real. Esta es la principal diferencia con la realidad virtual, puesto que no sustituye la realidad física, sino que sobreimprime los datos informáticos al mundo real.

La realidad aumentada ofrece infinidad de nuevas posibilidades de interacción, como son la arquitectura, el entretenimiento, la educación, el arte, la medicina o las comunidades virtuales. Veamos algunos ejemplos:

**LearnAR:** ‘eLearning with Augmented Reality’ es una nueva herramienta de aprendizaje interactiva. Se trata de diez programas de estudios para maestros y estudiantes que los ayuda a explorar mediante la combinación del mundo real con contenidos virtuales, utilizando una cámara web. El paquete de recursos consiste en actividades de matemáticas, ciencias, anatomía, física, geometría, educación física e idiomas.

**WordLens:** permite traducir las palabras que aparecen en una imagen. Basta con tomar una fotografía a cualquiera texto desconocido —un anuncio, un menú, un volante, etc— y se obtiene una traducción instantánea sobre el mismo objeto. El proceso es muy sencillo: el software identifica las letras que aparecen en el objeto y busca la palabra en el diccionario. Una vez que encuentra la traducción, la dibuja en lugar de la palabra original. La aplicación es ideal para quienes viajan mucho y necesitan conocer de manera rápida el significado de alguna palabra. Por el momento, programa ofrece traducción inglés – español y español – inglés, aunque sus creadores Otavio Good y John DeWeese señalaron que el paso siguiente es la traducción en otros idiomas, como el francés, el italiano o el portugués.

Para la creación de la aplicación de visualización de rocódromos se han estudiado las dos grandes posibilidades para su diseño. Como opciones nos encontramos con, escoger una aplicación ya existente que permita desarrollar la capa deseada o, por el contrario, diseñar e implementar un nuevo sistema.

La segunda opción es alcanzable debido al hecho de que actualmente existen muchas librerías que permiten el desarrollo de aplicaciones de Realidad Aumentada. Hace unos años, dichas

librerías no estaban creadas y hubiera sido muy difícil el diseño de un aplicativo nuevo, pero hoy día nos encontramos con diferentes frameworks y SDK que facilitan la implementación.

Un ejemplo a destacar es el que barajamos en los primeros compases del proyecto que fue la posibilidad de utilizar la SDK **Vuforia** (Qualcomm), ya que es una plataforma OpenSource y que dispone de bastante soporte para el desarrollador.

Posteriormente, haciendo un análisis en profundidad de las diferentes opciones y valorando el tiempo disponible para la realización del TFC, se ha cambiado de idea y nos hemos decantado por la primera opción debido a que, crear de cero el nuevo aplicativo supone un arduo trabajo de programación para crear la base de éste. De la misma manera, nos planteamos que puede ser una buena opción el reutilizar alguna herramienta como base de nuestra aplicación.

Para apoyar la toma de decisión, se ha realizado una comparativa de las aplicaciones existentes y de la que se desprende que es posible el desarrollo de capas agilizando así el objetivo final del TFC.

Una vez decidido que el desarrollo será a partir de una aplicación que ya existe, es necesario buscar estos aplicativos y compararlos para escoger aquél que resulte más útil según nuestras necesidades. En la búsqueda realizada se han encontrado muchos aplicativos de Realidad Aumentada, no obstante, no todos eran aptos para crear la aplicación. Por poner un ejemplo, tenemos **ARViewer** que sólo permite la creación de nodos. Éste es bastante completo, ya que incluso permite la utilización de la altitud, pero no permite el desarrollo de capas, que es la característica que necesitamos para crear la superposición de rocódromos.

Asimismo, también se barajó el uso de **Aurasma** que es una de las plataformas de AR más potentes del momento, junto a Qualcomm. Con esta aplicación, es posible disfrutar de la Realidad Aumentada integrada en carteles, revistas, catálogos o periódicos donde se haga uso de esta plataforma pero en contraposición, al igual que ARViewer tampoco permite el desarrollo por capas.

Finalmente, tras la búsqueda de aplicaciones existentes se ha decidido comparar las siguientes que permiten desarrollo por capas: **Wikitude** y **Layar**. Ambas aplicaciones son muy parecidas en cuanto a funcionamiento, sin embargo, Layar está más orientado al ocio y los servicios.

### 1.3.1 Wikitude

**Wikitude** es una aplicación de realidad aumentada desarrollada por la empresa austriaca Wikitude GmbH (antes Mobilizy GmbH) y fue publicada por primera vez en octubre de 2008 como software gratuito. Su funcionalidad principal es la de mostrar información acerca de los alrededores de los usuarios a través de la cámara móvil, incluyendo el reconocimiento de imágenes y modelado 3d. **Wikitude** fue la primera aplicación disponible públicamente que utiliza realidad aumentada basada en la ubicación.



Como se ha comentado, basándose de realidad aumentada se calcula la posición de los objetos en la pantalla del dispositivo móvil utilizando la posición del usuario (mediante GPS o Wi-Fi), la dirección en la que se enfrenta el usuario (mediante el uso de la brújula) y acelerómetro. En contraste, también permite utilizar la realidad aumentada a base de un marcador (marcador de, por ejemplo, el código QR) y se utilizaría para el seguimiento.

Un elemento central en el **Wikitude World Browser** es la ubicación. Esto, en comparación con las páginas web usuales, donde la ubicación del usuario juega un papel menor, hace que Wikitude World Browser esté optimizado para el uso basado en la ubicación móvil.

El navegador hace uso de los siguientes sistemas del dispositivo:

- Acelerómetro
- Cámara incorporada
- Brújula
- GPS

Desde agosto de 2012, Wikitude también cuenta con reconocimiento de imagen, tecnología que permite utilizar imágenes para activar la tecnología de la realidad aumentada dentro de la aplicación.

### 1.3.2 Layar

**Layar** es una aplicación creada en Holanda por una empresa (SPRXmobile) con sede en Amsterdam, fundada en 2009 por Raimo van der Klein, Claire Boonstra y Maarten Lens FitzGerald. El navegador permite a los usuarios encontrar varios artículos basados en la tecnología de realidad aumentada.

El 1 de septiembre de 2010, el Foro Económico Mundial, anunció la empresa en su *Technology Pioneers Programme* para el 2011.

El software **Layar** es formado por una API que provee una arquitectura software para el desarrollo de aplicaciones de RA sin la utilización de marcadores matriz 2D. Dichas aplicaciones, denominadas capas, son incorporadas al navegador móvil de RA de Layar, el cuál es capaz de mostrar información de toda índole, muy enfocada al ámbito del entretenimiento, turismo e incluso a los negocios. Una muestra de ello es que podemos visualizar a través de su navegador las localizaciones más cercanas a nuestra posición, como museos, restaurantes, etc.

De cara al desarrollador, **Layar** está compuesto de varios sistemas encargados de proveer la información. Así nos encontramos con la aplicación cliente que se ejecuta en el dispositivo, un servidor central donde se accede para obtener los datos y otro servidor privado para el desarrollador donde gestionar los datos, que son enviados al servidor central y finalmente visualizados en la aplicación a través de la cámara incorporada. Layar también nos brinda la posibilidad de mostrar objetos 3D, utilizando un formato de representación de objetos 3D optimizado para móviles.

Al igual que Wikitude, el navegador Layar hace uso de los sistemas del dispositivo:

- Acelerómetro
- Cámara incorporada
- Brújula
- GPS

A partir de julio de 2010, Layar tenido 1.000 capas. A partir de septiembre de 2011, tenía 2.993 Layar capas.

A continuación se muestra una tabla comparativa entre ambas plataformas:

	Wikitude	Layar
<b>Número de usuarios</b>	500K – 1M	1 M – 3 M
<b>Tutoriales</b>	Sí	Sí
<b>Showcases</b>	Sí	Sí
<b>API</b>	Sí	Sí
<b>Soporte</b>	Bueno	Muy bueno
<b>Blogs</b>	Sí	Sí
<b>Madurez</b>	Refundada en 2010	5 años

En conclusión, **Layar** y **Wikitude** son el mismo tipo de aplicación pero con diferente modelo de negocio y contenido. SPRXmobile (Layar) ha sido la primera empresa en entender el concepto que intenta vender Google con Android que no es más que información de interés relacionada con el lugar que ocupan los usuarios y a la vez con las preferencias de estos (información + posición + necesidad) y basar su modelo de negocio en él. Wikitude por el contrario, se vende como una guía turística que en vez de orientar a vender contenido se orienta a vender los negocios que en ella se muestran.

Las dos aplicaciones están en constante evolución y crecimiento, esto se puede observar de la continua actualización de sus páginas web. Ambas aplicaciones permiten el desarrollo y la creación de capas, no obstante, tras el análisis de ambas por separado se ha decidido trabajar con Layar basándonos en su madurez.

Esta mayor madurez es porque lleva mayor tiempo en el mercado, lo que hace que haya un mayor número de usuarios y de desarrolladores trabajando con esta aplicación. Asimismo, el soporte a priori parece que es algo mejor ya que los desarrolladores son más expertos.

Finalmente, tras el estudio que se ha expuesto anteriormente, la aplicación escogida para el diseño es **Layar** basándonos en su madurez y modelo de negocio.

## 1.4 Planificación temporal

A continuación se muestra una tabla con la planificación temporal del proyecto, igualmente se muestra la planificación como un diagrama de Gantt el cual nos servirá para el seguimiento del proyecto a lo largo de todo el ciclo de desarrollo.

Tarea	Días	Inicio	Fin
<b>Análisis previo</b>	13	18/09/2013	01/10/2013
Lectura documentación y plan docente	2	18/09/2013	20/09/2013
Elección de proyecto	3	21/09/2013	23/09/2013
Creación del plan de trabajo	8	24/09/2013	01/10/2013
Entrega del plan de trabajo – PEC 1	0	01/10/2013	<b>01/10/2013</b>
<b>Análisis de requisitos</b>	10	02/10/2013	11/10/2013
Análisis de requisitos funcionales	5	02/10/2013	06/10/2013
Análisis de requisitos no funcionales	5	07/10/2013	11/10/2013
<b>Diseño</b>	18	12/10/2013	29/10/2013
Requisitos de diseño	2	12/10/2013	13/10/2013
Diseño de arquitectura	3	14/09/2013	16/10/2013
Estudio de alternativas tecnológicas	4	17/10/2013	20/10/2013
Diseño del prototipo	6	21/10/2013	26/10/2013
Elaborar documento de diseño	3	27/10/2013	29/10/2013
Entrega PEC 2	0	29/10/2013	<b>29/10/2013</b>
<b>Implementación</b>	44	30/10/2013	10/12/2013
Codificación	35	30/10/2013	01/12/2013
Pruebas	5	02/12/2013	06/12/2013
Elaboración de documentación usuario	4	07/12/2013	10/12/2013
Entrega PEC 3	0	10/12/2013	<b>10/12/2013</b>
<b>Memoria y presentación final</b>	29	11/12/2013	08/01/2014
Elaboración de memoria	20	11/12/2013	30/12/2013
Preparación presentación	9	31/12/2013	08/01/2014
Entrega final	0	08/01/2014	<b>08/01/2013</b>

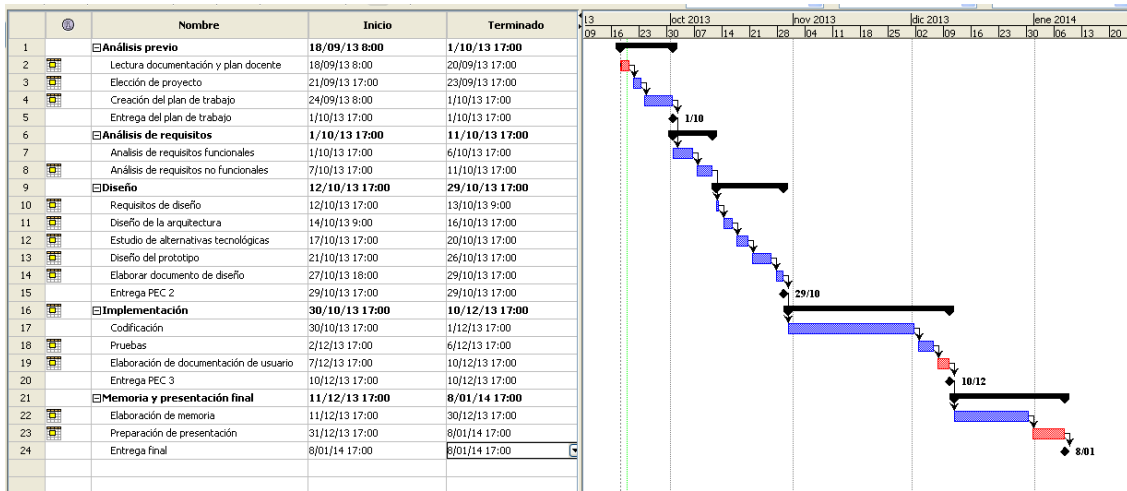


Fig. 1. Diagrama de Gantt

## 1.5 Riesgos

El mayor riesgo del proyecto al que nos enfrentamos es el bajo conocimiento acerca del desarrollo sobre la plataforma Layar y esto, con la planificación temporal tan ajustada puede causar desajustes temporales y problemas en el éxito del proyecto.

Por otra parte, nos enfrentamos a la implantación de una aplicación web PHP que será nuestro servidor de POIs. Además de la implementación y gestión del fichero XML contenedor de los datos. Asimismo, hay que tener en cuenta que dicha implantación se realizará sobre un servidor externo y esto es algo sobre lo que podemos encontrar problemas a la hora de encontrar un servidor gratuito que permita una buena interconexión entre los sistemas.

Otro de los riesgos a tener en cuenta para el éxito del proyecto es que, aunque de la comparativa se desprende que Layar es una plataforma bastante madura, tiene una madurez de unos 5 años por lo que, a la hora de buscar documentación y/o soporte puede ser que encontremos dificultades en el momento de implementar la capa de rocódromos.

## 2. Requisitos iniciales

En el presente apartado se exponen los requisitos iniciales que deberá cumplir el sistema a desarrollar en el Trabajo Final de Carrera.

### 2.1 Visualización de rocódromos sobre la imagen real

La aplicación debe mostrar los rocódromos de la ciudad de Sevilla sobre la imagen real transmitida por la cámara del dispositivo móvil. Dichos rocódromos serán visualizados en la pantalla como POIs y se mostrarán en su ubicación real al orientar el objetivo de la cámara hacia el lugar donde se encuentran coincidiendo con su posición original.

Para ubicar los POIs, el sistema utilizará los sensores del dispositivo GPS, brújula digital y acelerómetro.

A la hora de visualizar los diferentes POIs, la aplicación deberá permitir seleccionar el radio de reconocimiento medido desde la posición actual. Para ello, el usuario lo seleccionará al arrancar la capa o posteriormente accediendo al menú de opciones y seleccionando entre el rango mínimo de 100 metros y el máximo 10000 metros alrededor de la posición. Cabe destacar que, si el usuario no selecciona ningún radio, se usará un radio por defecto de 200m.

Pongamos un ejemplo de funcionamiento, si tenemos seleccionado como radio de visualización 100 m, la aplicación mostrará todos los rocódromos que estén dentro de 100 m a la redonda o menos pero no aquellos que se encuentren a una distancia superior a ésta.

Finalmente, en la esquina superior derecha de la pantalla, la capa Layar mostrará un radar donde se posicionarán los diferentes POIs para facilitar al usuario la orientación del dispositivo hacia el POI.

Los POIs que representarán los rocódromos sobre la pantalla, aparecerán en diferentes tamaños. Este tamaño, estará determinado según la distancia a la que se encuentren los POIs del dispositivo que está manejando el usuario.

### 2.2 Mostrar detalles del rocódromo e información ampliada

La aplicación debe permitir poder ampliar información acerca del rocódromo, para ello, el usuario tendrá que seleccionar pulsando encima del icono del POI, o sobre el panel informativo del mismo, para que se fije la información en la zona baja de la pantalla.

Además de mostrar el nombre del rocódromo, la aplicación debe mostrar una descripción de éste. Asimismo, desde esta pantalla de detalle se debe permitir realizar las diferentes acciones que se detallan en las siguientes funcionalidades.

Finalmente, la aplicación debe permitir enlazar con una Web externa que muestre información adicional acerca del rocódromo siempre y cuando éste disponga de ella.

## 2.3 Mostar información de cómo llegar

La aplicación debe permitir poder enlazar con la aplicación externa Google Maps, para ello debe contener un enlace dentro del detalle de un rocódromo sobre el que el usuario tendrá que actuar para acceder a dicha aplicación.

La aplicación, al realizar la llamada a Google Maps, enviará los datos necesarios para geo posicionar los datos y que se muestre en el mapa cómo llegar desde el punto donde se encuentra el usuario al lugar exacto donde se ubica el POI.

## 2.4 Gestionar los datos del rocódromo

Los datos de los rocódromos deben poder gestionarse para mantener actualizado los POIs sin necesidad de modificar el código fuente. Es por ello que se requiere una aplicación Web que permita añadir, modificar y/o eliminar los diferentes rocódromos del sistema.

### 3. Análisis del sistema

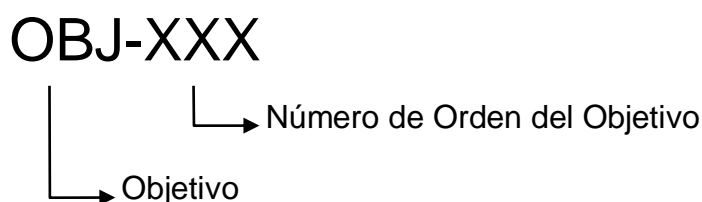
#### 3.1 Objetivos del sistema

El objetivo principal y fundamental del proyecto para la creación de una Capa sobre el navegador de Realidad Aumentada Layar es el desarrollo de un sistema que permita mantener geo localizados los rocódromos de la ciudad de Sevilla.

Con el proyecto en curso, se crea una Capa Layar de Realidad Aumentada, denominada “Rocódromos”, la cual permite a la comunidad escaladora ubicar los rocódromos dentro de una ciudad.

Partiendo de estos objetivos y basándonos en ellos se desglosan a continuación los siguientes objetivos específicos del Sistema.

Los objetivos se han codificado siguiendo la siguiente nomenclatura:



##### 3.1.1 <OBJ-001> Visualizar rocódromos

INFORMACIÓN DE OBJETIVO	
Identificación	OBJ-001
Nombre	Visualizar rocódromos
Descripción	El sistema deberá mostrar los rocódromos de la ciudad de Sevilla a través de una capa desarrollada para ser visualizada con el navegador de Realidad Aumentada Layar.
Subobjetivos	<ul style="list-style-type: none"> <li>▪ Se deberá permitir ver los rocódromos visualizados en pantalla como puntos de interés (POIs).</li> </ul>
Prioridad	Alta
Estado	Validado
Comentarios	Ninguno

### 3.1.2 <OBJ-002> Ampliar detalles del rocódromo

INFORMACIÓN DE OBJETIVO	
<b>Identificación</b>	<b>OBJ-002</b>
<b>Nombre</b>	<b>Ampliar detalles del rocódromo</b>
<b>Descripción</b>	El sistema deberá permitir ampliar la información acerca del rocódromo.
<b>Subobjetivos</b>	<ul style="list-style-type: none"> <li>▪ El sistema deberá mostrar en tiempo real la siguiente información del rocódromo:               <ul style="list-style-type: none"> <li>○ Descripción</li> <li>○ Pie de página</li> <li>○ Distancia a la que se encuentra de nuestra posición</li> <li>○ Enlace para ampliar información externa</li> <li>○ Enlace para conectar con Google Maps para ver como llegar.</li> </ul> </li> <li>▪ Se deberá disponer de un mecanismo para poder redireccionar a una página web externa que permita mostrar información adicional.</li> </ul>
<b>Prioridad</b>	<b>Alta</b>
<b>Estado</b>	<b>Validado</b>
<b>Comentarios</b>	<b>Ninguno</b>

### 3.1.3 <OBJ-003> Mostar información de cómo llegar

INFORMACIÓN DE OBJETIVO	
<b>Identificación</b>	<b>OBJ-003</b>
<b>Nombre</b>	<b>Mostrar información de como llegar</b>
<b>Descripción</b>	El sistema deberá permitir poder enlazar con la aplicación externa Google Maps, para ello debe contener un enlace dentro del detalle de un rocódromo sobre el que el usuario tendrá que actuar para acceder a dicha aplicación.
<b>Subobjetivos</b>	<ul style="list-style-type: none"> <li>▪ Se deberá disponer de un mecanismo para poder conectar con la aplicación Google Maps.</li> </ul>
<b>Prioridad</b>	<b>Alta</b>
<b>Estado</b>	<b>Validado</b>
<b>Comentarios</b>	<b>Ninguno</b>



### 3.1.4 <OBJ-004> Gestionar datos de los POIs

INFORMACIÓN DE OBJETIVO	
Identificación	OBJ-004
Nombre	Gestionar datos de los POI's
Descripción	El sistema debe permitir el almacenamiento de los datos de los POI que podrán gestionarse a través de una interfaz web accesible desde internet.
Subobjetivos	<ul style="list-style-type: none"> <li>Se deberá disponer de un mecanismo de acceso al Sistema basado en la autenticación de la persona como administrador del Sistema.</li> </ul>
Prioridad	Alta
Estado	Validado
Comentarios	Ninguno

## 3.2 Catálogo de requisitos del sistema

En este apartado se relacionan y definen los requisitos que debe satisfacer el Sistema, de tal forma que respondan a los objetivos generales y específicos del mismo.

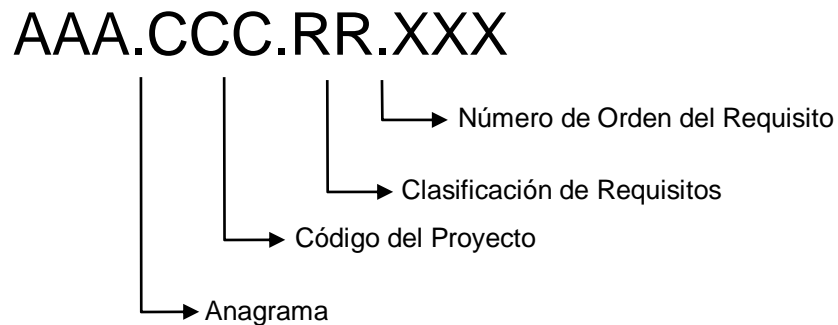
Se describen los requisitos incluyendo el tipo de requisito, estado y prioridad (alta, baja o media) de cada uno de ellos.

Los requisitos se clasifican en:

- **Requisitos de información.**
- **Requisitos funcionales:** este tipo de requisitos se especifican mediante diagramas de casos de uso, detallando cada uno de los casos de uso, y agrupando funcionalidades del Sistema en subsistemas.
- **Requisitos no funcionales:** engloban los requisitos de entornos tecnológicos y comunicaciones, de almacenamiento, interfaz de usuarios y de comunicación con otros sistemas.

### 3.2.1 Nomenclatura

Los requisitos catalogados se han enumerado siguiendo la siguiente nomenclatura:



Siendo:

- **AAA:** Identificador del sistema a lo largo de todo su ciclo de vida, al que pertenece el desarrollo sobre el que se están definiendo los requisitos. En nuestro caso utilizaremos el anagrama CLR “*Capa Layar Rocódromos*”.
- **CCC:** Código del Proyecto, el cual identifica el desarrollo sobre el que se están definiendo los requisitos. En nuestro caso utilizaremos el código DSI “*Desarrollo del Sistema de Información*”.
- **RR:** Identificador de la categoría en que se encuadra el requisito, según se ha especificado junto al nombre de cada categoría en el apartado de Catalogación de Requisitos. A efectos prácticos los requisitos se englobarán dentro de las siguientes categorías que detallaremos en los apartados siguientes:
  - Requisitos de información (RI).
  - Requisitos funcionales:
    - Actores (AC).
    - Casos de uso (CU).
  - Requisitos no funcionales (RN).
- **XXX:** Identificador del requisito dentro de la categoría a la que pertenece (3 dígitos asignados de una forma lógica que permita la subdivisión/agrupación de varios requisitos).

## 3.2.2 Requisitos de información

Los requisitos de información hacen referencia a la información que el sistema debe almacenar para cumplir los objetivos.

### 3.2.2.1 <CLR.DSI.RI.001> Datos de los POIs

REQUISITO DE INFORMACIÓN	
<b>Identificación</b>	<b>CLR.DSI.RI.001</b>
<b>Nombre</b>	<b>Datos de los POI's</b>
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a los puntos de interés de los rocódromos.
<b>Datos</b>	<ul style="list-style-type: none"> <li>▪ Generales               <ul style="list-style-type: none"> <li>○ Identificación del POI</li> <li>○ Título</li> <li>○ Descripción</li> <li>○ Pie de página</li> <li>○ URL Imagen</li> </ul> </li> <li>▪ Geolocalización               <ul style="list-style-type: none"> <li>○ Lat</li> <li>○ Lon</li> <li>○ Alt</li> </ul> </li> <li>▪ Enlaces               <ul style="list-style-type: none"> <li>○ Texto</li> <li>○ URI</li> </ul> </li> </ul>
<b>Tipo</b>	<b>General</b>
<b>Restricciones</b>	Ninguna
<b>Prioridad</b>	<b>Alta</b>
<b>Estado</b>	Validado
<b>Comentarios</b>	Ninguno

## 3.2.3 Requisitos funcionales

En este apartado se exponen, en forma de casos de uso, los requisitos funcionales detectados durante la extracción de requisitos.

Para la especificación de los requisitos funcionales se utilizan los diagramas de casos de uso.

### 3.2.3.1 Actores del Sistema

Ante todo, lo primero que hay que especificar antes de empezar a describir los requisitos funcionales, son los actores del Sistema.

Los actores del Sistema son entes, exteriores al mismo, que desean interactuar con él. Éstos son tanto personas físicas como Sistemas ajenos que interactúan con el Sistema objeto de estudio.

Los actores que interactúan en el entorno del Sistema están relacionados entre ellos por las siguientes relaciones de especialización/generalización:

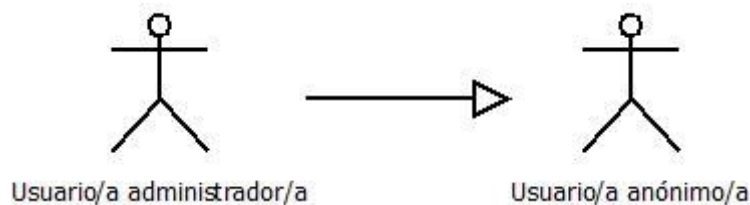


Fig. 2. Actores del sistema

Por otra parte, el actor que interactúa en el entorno web y que está relacionado con la aplicación de “Gestión de POIs” es el siguiente:



Fig. 3. Actor administrador

A continuación describimos los roles de cada actor del diagrama:

#### 3.2.3.1.1 <CLR.DSI.AC.001> Usuario/a administrador/a

ACTORES DEL SISTEMA	
<b>Identificación</b>	CLR.DSI.AC.001
<b>Nombre</b>	Usuario/a administrador/a
<b>Autor</b>	<ul style="list-style-type: none"> <li>▪ Nicolás Luque</li> </ul>
<b>Descripción</b>	Este actor representa a la persona que tiene perfil de “administrador” y contiene todos los permisos para acceder a la administración de los datos de los POIs para el subsistema de gestión de POIs.
<b>Prioridad</b>	Alta
<b>Estado</b>	Validado
<b>Comentarios</b>	Actor que solo interactuará con el subsistema de gestión de POIs.

### 3.2.3.1.2

### <CLR.DSI.AC.002> Usuario/a anónimo

ACTORES DEL SISTEMA	
<b>Identificación</b>	<b>CLR.DSI.AC.002</b>
<b>Nombre</b>	<b>Usuario anónimo</b>
<b>Autor</b>	<ul style="list-style-type: none"> <li>▪ Nicolás Luque</li> </ul>
<b>Descripción</b>	Este actor representa a la persona que accederá a la aplicación Layar para interactuar con la Capa de Rocódromos.
<b>Prioridad</b>	<b>Alta</b>
<b>Estado</b>	<b>Validado</b>
<b>Comentarios</b>	Ninguno

### 3.2.3.2 Diagramas de subsistemas

Para realizar una extracción detallada de requisitos funcionales del Sistema, se ha dividido lógicamente este mismo en subsistemas, según su funcionalidad. De esta forma se describe, de forma detallada, la propuesta del Sistema.

Los subsistemas que agrupan los requisitos funcionales del Sistema, son:

- Subsistema de gestión de datos de los POIs.
- Subsistema Capa Layar de Rocódromos.

A continuación se muestra el diagrama general del Sistema, en donde se incluyen los distintos subsistemas del mismo:

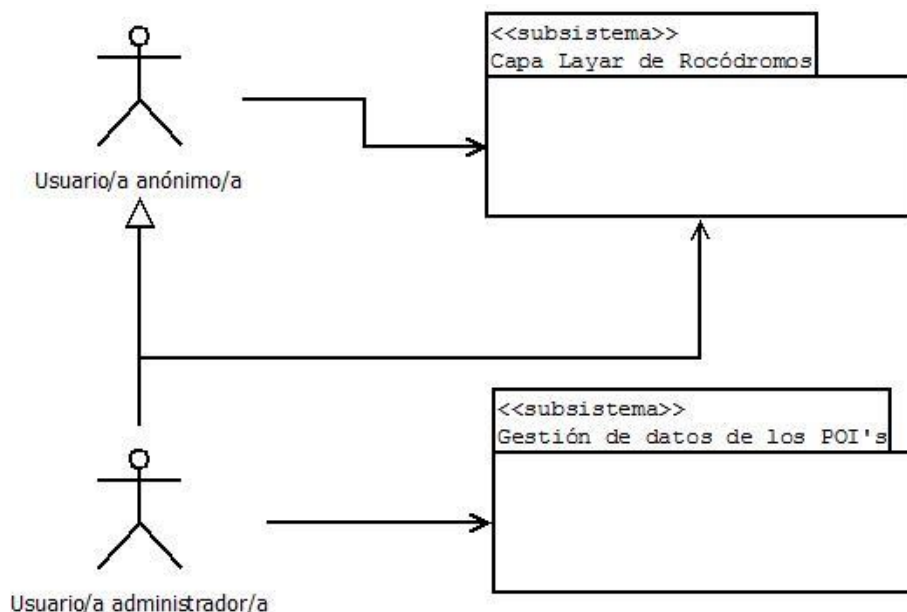


Fig. 4. Diagrama general del sistema

### 3.2.3.3 Subsistema de gestión de POIs

El subsistema de gestión de POIs cubre todos aquellos trámites de administración de los datos de los puntos de interés (POIs) de los rocódromos. Es decir, a través de una aplicación web y un fichero XML que contendrá los datos de los POIs, el subsistema de gestión de datos de los POIs deberá permitir mantener los puntos de interés del Sistema. Para ello, será directamente el/la Administrador/a quien gestione los datos del Sistema.

Sus funcionalidades y requisitos se enumeran a continuación:

- Funcionalidades de listado y visualización de datos de POIs.
- Funcionalidades de alta, baja y modificación de datos de POIs.
- Existirá siempre un/una usuario/a con el perfil para la administración total de los datos del Sistema. Se trata del actor "Usuario/a administrador/a". El equipo de mantenimiento dispondrá de los permisos necesarios para el desempeño de sus tareas.
- No existirá gestión de usuarios.
- Implementar paulatinamente funcionalidades para que exista gestión de POIs sobre Base de Datos.

El diagrama del subsistema de gestión de datos de los POIs queda como sigue:

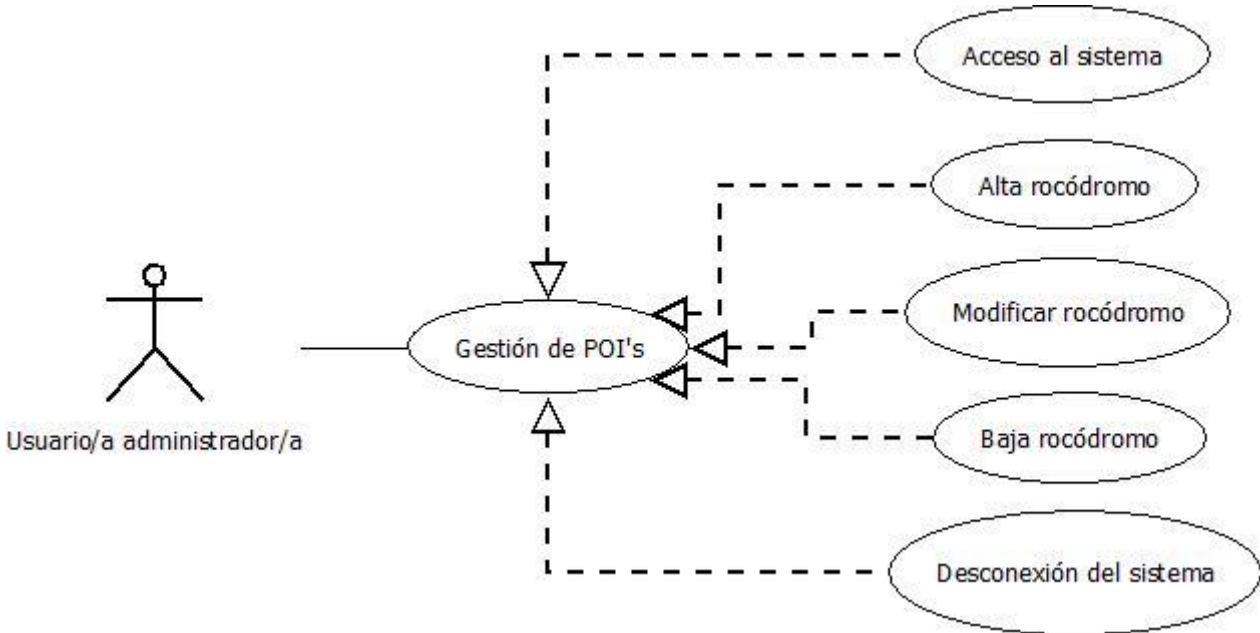


Fig. 5. Diagrama subsistema Gestión de POIs

En los siguientes apartados se detallan cada uno de los casos de uso del subsistema.

### 3.2.3.3.1 <CLR.DSI.CU.001> Acceso al sistema

CASO DE USO									
<b>Identificación</b>	<b>CLR.DSI.CU.001</b>								
<b>Nombre</b>	<b>Acceso al sistema</b>								
<b>Autor</b>	<ul style="list-style-type: none"> <li>Nicolás Luque</li> </ul>								
<b>Descripción</b>	Con esta función el actor puede: - Acceder al servidor web para gestionar los datos de los POI's.								
<b>Actor(es)</b>	Usuario/a administrador/a								
<b>Precondición</b>	El/La usuario/a administrador/a está creado en el sistema.								
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El actor Usuario administrador (CLR.DSI.AC.001) introduce el nombre de usuario y la contraseña en la pantalla principal de acceso al sistema de gestión de datos.</td> </tr> <tr> <td>2</td> <td>El actor Usuario administrador (CLR.DSI.AC.001) pulsa el botón de acceder.</td> </tr> <tr> <td>3</td> <td>El sistema autentifica al Usuario administrador en el Sistema.</td> </tr> </tbody> </table>	Paso	Acción	1	El actor Usuario administrador (CLR.DSI.AC.001) introduce el nombre de usuario y la contraseña en la pantalla principal de acceso al sistema de gestión de datos.	2	El actor Usuario administrador (CLR.DSI.AC.001) pulsa el botón de acceder.	3	El sistema autentifica al Usuario administrador en el Sistema.
	Paso	Acción							
	1	El actor Usuario administrador (CLR.DSI.AC.001) introduce el nombre de usuario y la contraseña en la pantalla principal de acceso al sistema de gestión de datos.							
	2	El actor Usuario administrador (CLR.DSI.AC.001) pulsa el botón de acceder.							
3	El sistema autentifica al Usuario administrador en el Sistema.								
<b>Postcondición</b>	El sistema muestra la pantalla principal donde aparecerá la siguiente información: <ul style="list-style-type: none"> <li>Mensaje de bienvenida</li> <li>Id del desarrollador de la capa Layar</li> <li>Nombre de la capa Layar</li> <li>Enlace para desconectar</li> </ul>								

	<ul style="list-style-type: none"> <li>▪ Enlace para ir a la pantalla principal</li> </ul>
<b>Excepciones</b>	<b>Paso</b>   <b>Acción</b>
	3   Si el usuario y/o contraseña es incorrecto el sistema muestra un mensaje de error.
<b>Prioridad</b>	Alta
<b>Estado</b>	Validado
<b>Comentarios</b>	Ninguno

### 3.2.3.3.2

### <CLR.DSI.CU.002> Alta rocódromo

CASO DE USO	
<b>Identificación</b>	<b>CLR.DSI.CU.002</b>
<b>Nombre</b>	<b>Alta rocódromo</b>
<b>Autor</b>	<ul style="list-style-type: none"> <li>▪ Nicolás Luque</li> </ul>
<b>Descripción</b>	Con esta función el actor puede: - Dar de alta nuevos POI's de rocódromos.
<b>Actor(es)</b>	Usuario/a administrador/a
<b>Precondición</b>	El/La usuario/a administrador/a está autenticado en el sistema y se encuentra en la pantalla de listado de POI's..
<b>Secuencia normal</b>	<b>Paso</b>   <b>Acción</b>
	1   El actor Usuario administrador (CLR.DSI.AC.001) pulsa sobre el botón "Crear POI".
	2   El sistema añade un nuevo POI a la lista con el nombre por defecto <Nuevo POI>.
	3   El actor Usuario administrador (CLR.DSI.AC.001) pulsa sobre el nuevo POI.
<b>Postcondición</b>	<p>El sistema muestra la pantalla de creación del POI con todos los datos que se listan a continuación vacíos a excepción del campo calculado ID.</p> <ul style="list-style-type: none"> <li>▪ ID</li> <li>▪ Título</li> <li>▪ Descripción</li> <li>▪ Pie de página</li> <li>▪ URL imagen</li> <li>▪ Geolocalización <ul style="list-style-type: none"> <li>○ Lat</li> <li>○ Lon</li> <li>○ Alt</li> </ul> </li> <li>▪ Enlaces <ul style="list-style-type: none"> <li>○ Texto</li> <li>○ URI</li> </ul> </li> </ul>
<b>Excepciones</b>	<b>Paso</b>   <b>Acción</b>
	-   -
<b>Prioridad</b>	Alta
<b>Estado</b>	Validado
<b>Comentarios</b>	Ninguno



**3.2.3.3.3**
**<CLR.DSI.CU.003> Modificar rocódromo**

CASO DE USO		
<b>Identificación</b>	<b>CLR.DSI.CU.003</b>	
<b>Nombre</b>	<b>Modificar rocódromo</b>	
<b>Autor</b>	▪ Nicolás Luque	
<b>Descripción</b>	Con esta función el actor puede: - Modificar datos de POI's existentes.	
<b>Actor(es)</b>	Usuario/a administrador/a	
<b>Precondición</b>	El/La usuario/a administrador/a está autenticado en el sistema y se encuentra en la pantalla de listado de POI's.	
<b>Secuencia normal</b>	<b>Paso</b> <b>Acción</b>	
	1	El actor Usuario administrador (CLR.DSI.AC.001) pulsa sobre el nombre del POI que desea modificar.
	2	El sistema muestra la pantalla de edición de datos del POI.
	3	El actor Usuario administrador (CLR.DSI.AC.001) modifica los datos que considere oportunos y pulsa sobre el botón "Guardar".
<b>Postcondición</b>	El sistema modifica los datos del POI y se mantiene en la pantalla de edición del POI.	
<b>Excepciones</b>	<b>Paso</b> <b>Acción</b>	
	-	-
<b>Prioridad</b>	Alta	
<b>Estado</b>	Validado	
<b>Comentarios</b>	Ninguno	

**3.2.3.3.4**
**<CLR.DSI.CU.004> Baja rocódromo**

CASO DE USO	
<b>Identificación</b>	<b>CLR.DSI.CU.004</b>
<b>Nombre</b>	<b>Baja rocódromo</b>
<b>Autor</b>	▪ Nicolás Luque
<b>Descripción</b>	Con esta función el actor puede: - Dar de baja POI's existentes.
<b>Actor(es)</b>	Usuario/a administrador/a
<b>Precondición</b>	El/La usuario/a administrador/a está autenticado en el sistema y se encuentra en la pantalla de listado de POI's.
<b>Secuencia normal</b>	<b>Paso</b> <b>Acción</b>
	1
<b>Postcondición</b>	El sistema se mantiene en la pantalla de listado de POI's actualizada sin el POI que se ha eliminado.
<b>Excepciones</b>	<b>Paso</b> <b>Acción</b>
	-
<b>Prioridad</b>	Alta
<b>Estado</b>	Validado
<b>Comentarios</b>	Ninguno

### 3.2.3.3.5

### <CLR.DSI.CU.005> Desconexión del sistema

CASO DE USO		
<b>Identificación</b>	<b>CLR.DSI.CU.005</b>	
<b>Nombre</b>	<b>Desconexión del sistema</b>	
<b>Autor</b>	<ul style="list-style-type: none"> <li>Nicolás Luque</li> </ul>	
<b>Descripción</b>	Con esta función el actor puede: - Desconectarse del servidor de datos de los POI's.	
<b>Actor(es)</b>	Usuario/a administrador/a	
<b>Precondición</b>	El/La usuario/a administrador/a está autenticado en el sistema y se encuentra en cualquier pantalla del sistema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Usuario administrador (CLR.DSI.AC.001) pulsa sobre el enlace "Salir" situado en la esquina superior izquierda.
<b>Postcondición</b>	El sistema desconecta al usuario del sistema y redirecciona a la pantalla principal de acceso.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	-	-
<b>Prioridad</b>	Alta	
<b>Estado</b>	Validado	
<b>Comentarios</b>	Ninguno	

### 3.2.3.4 Subsistema Capa Layar de Rocódromos

El subsistema Capa Layar de Rocódromos abarca la funcionalidad de visualización de POIs de rocódromos así como todas aquellas funcionalidades vinculadas a estos. Es decir, engloba funcionalidades tales como ver más información acerca del POI o enlazar con Google Maps para ver cómo llegar entre otras que se describirán a continuación. Cabe destacar que para acceder a este subsistema no será necesario tener ningún tipo de usuario ya que el acceso lo realizará un usuario anónimo.

Sus funcionalidades y requisitos se enumeran a continuación:

- Funcionalidad para buscar capa.
- Funcionalidad visualizar rocódromos.
- Funcionalidad filtros para cambiar las distancias en el rango de búsqueda.
- Funcionalidad para ampliar información del rocódromo.
- Funcionalidad para enlazar con Google Maps para ver cómo llegar.
- Se accederá siempre con un/una usuario/a anónimo/a. Se trata del actor "Usuario/a anónimo/a".
- Será necesario como mínimo tener conexión a internet para poder utilizar la funcionalidad. Asimismo, para utilizar el subsistema por completo será necesario tener también GPS.

El diagrama del subsistema Capa Layar de Rocódromos queda como sigue:

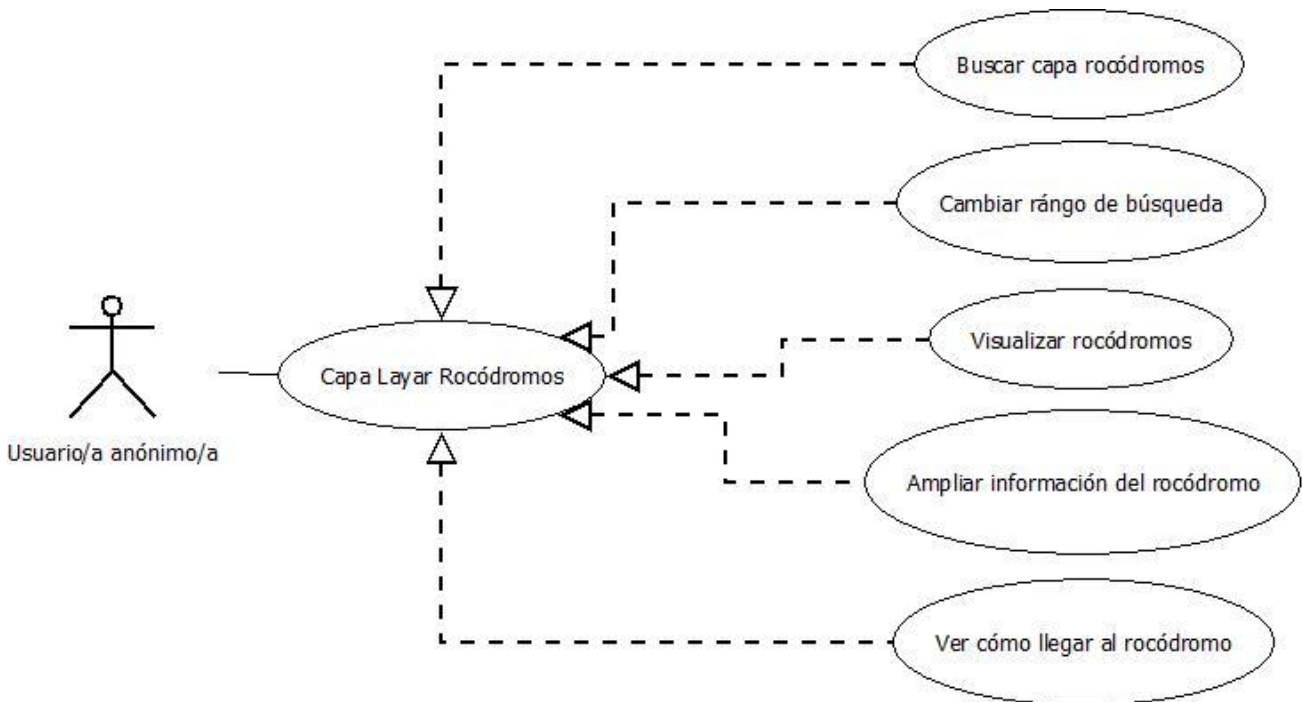


Fig. 6. Diagrama subsistema Capa Layar Rocódromos

En los siguientes apartados se detallan cada uno de los casos de uso del subsistema.

### 3.2.3.4.1 <CLR.DSI.CU.006> Buscar capa rocódromos

CASO DE USO									
<b>Identificación</b>	<b>CLR.DSI.CU.006</b>								
<b>Nombre</b>	<b>Buscar capa rocódromos</b>								
<b>Autor</b>	<ul style="list-style-type: none"> <li>Nicolás Luque</li> </ul>								
<b>Descripción</b>	Con esta función el actor puede: - Cargar la capa de rocódromos sobre el navegador Layar.								
<b>Actor(es)</b>	Usuario/a anonimo/a								
<b>Precondición</b>	El/La usuario/a anonimo/a ha accedido a la aplicación Layar y está en la pantalla principal.								
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El actor Usuario anonimo (CLR.DSI.AC.002) pulsa sobre el icono de Layar (ar) situado en la esquina superior izquierda para abrir el menú del navegador.</td> </tr> <tr> <td>2</td> <td>El actor Usuario anonimo (CLR.DSI.AC.002) pulsa sobre el enlace "Geo Layers" del menú y posteriormente sobre el enlace "Buscar capas".</td> </tr> <tr> <td>3</td> <td>El actor Usuario anonimo (CLR.DSI.AC.002) introduce el nombre de la capa (Rocódromos) en el buscador y pulsa sobre el icono de la lupa para iniciar la búsqueda. Asimismo, se puede realizar la búsqueda por categoría y/o por palabra clave (deporte, escalada, Sevilla, ...).</td> </tr> </tbody> </table>	Paso	Acción	1	El actor Usuario anonimo (CLR.DSI.AC.002) pulsa sobre el icono de Layar (ar) situado en la esquina superior izquierda para abrir el menú del navegador.	2	El actor Usuario anonimo (CLR.DSI.AC.002) pulsa sobre el enlace "Geo Layers" del menú y posteriormente sobre el enlace "Buscar capas".	3	El actor Usuario anonimo (CLR.DSI.AC.002) introduce el nombre de la capa (Rocódromos) en el buscador y pulsa sobre el icono de la lupa para iniciar la búsqueda. Asimismo, se puede realizar la búsqueda por categoría y/o por palabra clave (deporte, escalada, Sevilla, ...).
	Paso	Acción							
	1	El actor Usuario anonimo (CLR.DSI.AC.002) pulsa sobre el icono de Layar (ar) situado en la esquina superior izquierda para abrir el menú del navegador.							
	2	El actor Usuario anonimo (CLR.DSI.AC.002) pulsa sobre el enlace "Geo Layers" del menú y posteriormente sobre el enlace "Buscar capas".							
3	El actor Usuario anonimo (CLR.DSI.AC.002) introduce el nombre de la capa (Rocódromos) en el buscador y pulsa sobre el icono de la lupa para iniciar la búsqueda. Asimismo, se puede realizar la búsqueda por categoría y/o por palabra clave (deporte, escalada, Sevilla, ...).								

	4	El sistema realiza la búsqueda y muestra el resultado en formato lista.
	5	El actor Usuario anonimo (CLR.DSI.AC.002) selecciona la capa de la lista pulsando sobre ella.
	6	El sistema solicita que se seleccione un rango de búsqueda.
	7	El actor Usuario anonimo (CLR.DSI.AC.002) selecciona el rango de búsqueda sobre el que quiere que aparezcan los POIs.
<b>Postcondición</b>	El sistema carga la capa en el navegador y muestra los POIs encontrados en el rango seleccionado por el usuario al cargar la capa.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	El sistema no encuentra resultados acordes al valor introducido en el buscador.
<b>Prioridad</b>	Alta	
<b>Estado</b>	Validado	
<b>Comentarios</b>	La funcionalidad que realiza la búsqueda de capas está ya desarrollada por el navegador Layar.	

### 3.2.3.4.2

### <CLR.DSI.CU.007> Cambiar rango de búsqueda

CASO DE USO		
<b>Identificación</b>	<b>CLR.DSI.CU.007</b>	
<b>Nombre</b>	<b>Cambiar rango de búsqueda</b>	
<b>Autor</b>	<ul style="list-style-type: none"> <li>▪ Nicolás Luque</li> </ul>	
<b>Descripción</b>	Con esta función el actor puede: - Modificar el rango de búsqueda de POIs.	
<b>Actor(es)</b>	Usuario/a anonimo/a	
<b>Precondición</b>	El/La usuario/a anonimo/a ha accedido a la aplicación Layar y tiene la capa de rocódromos cargada.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Usuario anonimo (CLR.DSI.AC.002) pulsa sobre el icono de la capa de rocódromos situado en la esquina superior derecha para abrir el menú de la capa.
	2	El actor Usuario anonimo (CLR.DSI.AC.002) pulsa sobre el enlace "Filtros" del menú.
	3	El actor Usuario anonimo (CLR.DSI.AC.002) desliza el slide de rango de búsqueda, selecciona la que desee y pulsa el botón "Aceptar".
<b>Postcondición</b>	El sistema realiza de nuevo la búsqueda de rocódromos basándose en la nueva selección de rango de búsqueda y muestra en pantalla los resultados obtenidos (Ver Caso de Uso CLR.DSI.CU.008).	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	-	-
<b>Prioridad</b>	Alta	
<b>Estado</b>	Validado	
<b>Comentarios</b>	Ninguno	

**3.2.3.4.3**
**<CLR.DSI.CU.008> Visualizar rocódromos**

CASO DE USO							
<b>Identificación</b>	<b>CLR.DSI.CU.008</b>						
<b>Nombre</b>	<b>Visualizar rocódromos</b>						
<b>Autor</b>	<ul style="list-style-type: none"> <li>▪ Nicolás Luque</li> </ul>						
<b>Descripción</b>	Con esta función el actor puede: - Ver los POI de rocódromos que se encuentren a su alrededor y cumplan con el rango de búsqueda seleccionado en el Caso de Uso CLR.DSI.CU.007.						
<b>Actor(es)</b>	Usuario/a anonimo/a						
<b>Precondición</b>	El/La usuario/a anonimo/a ha accedido a la aplicación Layar y tiene la capa de rocódromos cargada.						
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El sistema muestra en la esquina superior derecha un radar con los POIs sobre rocódromos que cumplan con el rango de búsqueda (CLR.DSI.CU.007).</td> </tr> <tr> <td>2</td> <td>El actor Usuario anonimo (CLR.DSI.AC.002) mueve el dispositivo hacia los laterales en función de la posición que va indicando el radar para visualizar en pantalla el/los POIs que cumplan el rango de búsqueda (CLR.DSI.CU.007).</td> </tr> </tbody> </table>	Paso	Acción	1	El sistema muestra en la esquina superior derecha un radar con los POIs sobre rocódromos que cumplan con el rango de búsqueda (CLR.DSI.CU.007).	2	El actor Usuario anonimo (CLR.DSI.AC.002) mueve el dispositivo hacia los laterales en función de la posición que va indicando el radar para visualizar en pantalla el/los POIs que cumplan el rango de búsqueda (CLR.DSI.CU.007).
	Paso	Acción					
1	El sistema muestra en la esquina superior derecha un radar con los POIs sobre rocódromos que cumplan con el rango de búsqueda (CLR.DSI.CU.007).						
2	El actor Usuario anonimo (CLR.DSI.AC.002) mueve el dispositivo hacia los laterales en función de la posición que va indicando el radar para visualizar en pantalla el/los POIs que cumplan el rango de búsqueda (CLR.DSI.CU.007).						
<b>Postcondición</b>	El sistema va mostrando en pantalla sobre la imagen que transmite la cámara del dispositivo los POIs que cumplen con el Caso de Uso CLR.DSI.CU.007.						
<b>Excepciones</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El sistema no encuentra POIs dentro del rango de búsqueda seleccionado por el usuario.</td> </tr> </tbody> </table>	Paso	Acción	1	El sistema no encuentra POIs dentro del rango de búsqueda seleccionado por el usuario.		
	Paso	Acción					
1	El sistema no encuentra POIs dentro del rango de búsqueda seleccionado por el usuario.						
<b>Prioridad</b>	Alta						
<b>Estado</b>	Validado						
<b>Comentarios</b>	Cuando el usuario tenga el dispositivo orientado hacia la dirección donde se encuentra el POI aparecerá en la zona inferior de la pantalla información básica del rocódromo así como la distancia en metros a la que se encuentra el POI. Asimismo, se mostrará un enlace para ampliar información a través de una web externa y un enlace para conectar con Google Maps y que muestre en el mapa como llegar al POI.						

**3.2.3.4.4**
**<CLR.DSI.CU.009> Ampliar información del rocódromo**

CASO DE USO							
<b>Identificación</b>	<b>CLR.DSI.CU.009</b>						
<b>Nombre</b>	<b>Ampliar información del rocódromo</b>						
<b>Autor</b>	<ul style="list-style-type: none"> <li>▪ Nicolás Luque</li> </ul>						
<b>Descripción</b>	Con esta función el actor puede: - Ampliar la información del rocódromo que estará contenida en una web externa.						
<b>Actor(es)</b>	Usuario/a anonimo/a						
<b>Precondición</b>	El/La usuario/a anonimo/a está visualizando en pantalla un POI de rocódromo.						
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El sistema muestra en la zona inferior de la pantalla la información del rocódromo.</td> </tr> <tr> <td>2</td> <td>El actor Usuario anónimo (CLR.DSI.AC.002) pulsa sobre la información del rocódromo, posteriormente el sistema despliega el cuadro de información y</td> </tr> </tbody> </table>	Paso	Acción	1	El sistema muestra en la zona inferior de la pantalla la información del rocódromo.	2	El actor Usuario anónimo (CLR.DSI.AC.002) pulsa sobre la información del rocódromo, posteriormente el sistema despliega el cuadro de información y
	Paso	Acción					
1	El sistema muestra en la zona inferior de la pantalla la información del rocódromo.						
2	El actor Usuario anónimo (CLR.DSI.AC.002) pulsa sobre la información del rocódromo, posteriormente el sistema despliega el cuadro de información y						

		muestra los enlaces “Web” y “Cómo llegar”.
	3	El actor Usuario anónimo (CLR.DSI.AC.002) pulsa sobre el enlace “Ampliar info”.
<b>Postcondición</b>	El sistema redirecciona a la web externa de información acerca del rocódromo.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	El POI no tiene información externa por lo que no se muestra el enlace “Web”.
<b>Prioridad</b>	Alta	
<b>Estado</b>	Validado	
<b>Comentarios</b>	La funcionalidad que lanza el evento para enlazar con la web externa está ya desarrollada por el navegador Layar.	

### 3.2.3.4.5

### <CLR.DSI.CU.010> Ver cómo llegar al rocódromo

CASO DE USO		
<b>Identificación</b>	<b>CLR.DSI.CU.010</b>	
<b>Nombre</b>	<b>Ver cómo llegar al rocódromo</b>	
<b>Autor</b>	<ul style="list-style-type: none"> <li>▪ Nicolás Luque</li> </ul>	
<b>Descripción</b>	Con esta función el actor puede: - Enlazar con la aplicación Google Maps para ver sobre el mapa como llegar hasta el rocódromo.	
<b>Actor(es)</b>	Usuario/a anonimo/a	
<b>Precondición</b>	El/La usuario/a anonimo/a está visualizando en pantalla un POI de rocódromo.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema muestra en la zona inferior de la pantalla la información del rocódromo.
	2	El actor Usuario anónimo (CLR.DSI.AC.002) pulsa sobre la información del rocódromo, posteriormente el sistema despliega el cuadro de información y muestra los enlaces “Ampliar info” y “Cómo llegar”.
	3	El actor Usuario anónimo (CLR.DSI.AC.002) pulsa sobre el enlace “Cómo llegar”.
<b>Postcondición</b>	El sistema conecta con la aplicación Google Maps y muestra en el mapa la ruta a seguir para llegar al rocódromo.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	-	-
<b>Prioridad</b>	Alta	
<b>Estado</b>	Validado	
<b>Comentarios</b>	La funcionalidad que lanza el evento para enlazar con la aplicación Google Maps está ya desarrollada por el navegador Layar.	

### 3.2.4 Requisitos no funcionales

Las siguientes tablas muestran los requisitos no funcionales identificados en el proceso de extracción:

#### 3.2.4.1 <CLR.DSI.RN.001> Requisitos de entornos tecnológicos y comunicaciones

<b>Código</b>	CLR.DSI.RN.001
<b>Nombre</b>	Requisitos de entornos tecnológicos y comunicaciones
<b>Dependencias</b>	CLR.DSI.CU.006, CLR.DSI.CU.007, CLR.DSI.CU.008; CLR.DSI.CU.09, CLR.DSI.CU.010
<b>Descripción</b>	<p>La base principal de la aplicación será implementar una capa que corra sobre el navegador <b>Layar</b> de Realidad Aumentada que se instalará en el dispositivo, por lo que éste debe contar con conexión a internet (3G o 4G).</p> <p>Asimismo, el sistema debe contar con los sensores necesarios para poder funcionar con normalidad como son:</p> <ul style="list-style-type: none"> <li>• <b>GPS:</b> utilizado para conocer la localización del observador y situarse en el entorno. Permite conocer la posición a través de las coordenadas GPS, pero no la orientación.</li> <li>• <b>Brújula digital:</b> con este sensor se puede conocer hacia qué dirección estamos observando. Nos proporciona la dirección hacia donde estamos apuntando con la cámara, para así sólo mostrar aquellos puntos en forma de etiquetas que estén aproximadamente sobre la misma dirección.</li> <li>• <b>Acelerómetro:</b> una vez conocidas la posición y dirección, es recomendable conocer la orientación de la cámara (punto de vista del observador). Esto se realiza en base al funcionamiento de este sensor, que nos proporciona unos valores en X, Y, Z, que convenientemente tratados, nos permitirán conocer cuál es la orientación de la cámara con respecto a un plano vertical (el suelo se considera horizontal).</li> </ul> <p>No menos importante para el funcionamiento de la aplicación es que el dispositivo debe tener como hardware necesario lo siguiente: <b>pantalla</b>, donde se verá reflejada la suma de lo real y lo virtual que conforma la realidad aumentada y <b>cámara</b> que es la encargada de capturar la realidad y transmitirla al software.</p>
<b>Prioridad</b>	Alta
<b>Estado</b>	Validado
<b>Comentarios</b>	La aplicación debe permitir la portabilidad a otros dispositivos que tengan como sistema operativo IOS y/o Symbian OS.

### 3.2.4.2 <CLR.DSI.RN.002> Requisitos de almacenamiento

<b>Código</b>	CLR.DSI.RN.002
<b>Nombre</b>	Requisitos de almacenamiento
<b>Dependencias</b>	CLR.DSI.CU.001, CLR.DSI.CU.002, CLR.DSI.CU.003; CLR.DSI.CU.004, CLR.DSI.CU.005
<b>Descripción</b>	El sistema necesitará un servidor externo para gestionar y servir los datos de los POIs.
<b>Prioridad</b>	Medio
<b>Estado</b>	Validado
<b>Comentarios</b>	El servidor deberá ser PHP por disponer de más posibilidades de servidores gratuitos.

### 3.2.4.3 <CLR.DSI.RN.003> Requisitos de interfaz de usuarios

<b>Código</b>	CLR.DSI.RN.003
<b>Nombre</b>	Requisitos de interfaz de usuarios
<b>Dependencias</b>	CLR.DSI.CU.001, CLR.DSI.CU.002, CLR.DSI.CU.003; CLR.DSI.CU.004, CLR.DSI.CU.005, CLR.DSI.CU.006, CLR.DSI.CU.007, CLR.DSI.CU.008; CLR.DSI.CU.09, CLR.DSI.CU.010
<b>Descripción</b>	<p>La forma de interactuar de los usuarios con el sistema será a base de toques de pantalla que le permitirá acceder y navegar a través de las funcionalidades expuestas en el apartado de requisitos funcionales.</p> <p>El actor Usuario administrador (CLR.DSI.AC.001), interactuará con el servidor Web para gestionar (alta, baja y modificación) los puntos de interés a través de un navegador web y accediendo mediante usuario y password.</p>
<b>Prioridad</b>	Alta
<b>Estado</b>	Validado
<b>Comentarios</b>	Más adelante, en el apartado <b>5. Prototipo de interfaz de usuario</b> , se detallaran los prototipos de las diferentes pantallas que compondrán el sistema.

### 3.2.4.4 <CLR.DSI.RN.004> Requisitos de comunicación con otros sistemas

<b>Código</b>	CLR.DSI.RN.004
<b>Nombre</b>	Requisitos de comunicación con otros sistemas



<b>Dependencias</b>	CLR.DSI.CU.001, CLR.DSI.CU.002, CLR.DSI.CU.003; CLR.DSI.CU.004, CLR.DSI.CU.005, CLR.DSI.CU.006, CLR.DSI.CU.007, CLR.DSI.CU.008; CLR.DSI.CU.09, CLR.DSI.CU.010
<b>Descripción</b>	<p>La principal comunicación necesaria para el funcionamiento de la aplicación será basada en el protocolo HTTP y se utilizará para realizar peticiones a un servidor PHP que devolverá, en formato JSON, los datos que necesita la capa Layar para mostrar los diferentes POI en pantalla.</p> <p>Asimismo, realizará conexiones siguiendo el anterior protocolo (HTTP) para realizar las redirecciones a las páginas Web externas con información sobre los rocódromos.</p> <p>Finalmente, el navegador de Layar realizará conexiones sobre el sistema Google Maps, a través de la API de éste, para geo posicionar los rocódromos y mostrar la mejor ruta a seguir hasta llegar al POI en cuestión.</p>
<b>Prioridad</b>	Alta
<b>Estado</b>	Validado
<b>Comentarios</b>	Todas estas conexiones y arquitectura se explican más detalladamente en el apartado <b>4. Diseño de Arquitectura.</b>

## 4. Diseño arquitectura

### 4.1 Arquitectura de Layar

Para el diseño de una capa de Realidad Aumentada sobre Layar, es muy importante entender la arquitectura de dicha plataforma. El usuario final no se dará cuenta al ejecutarla, pero detrás de ella existe toda una compleja estructura.

Se puede dividir el esquema de funcionamiento en tres grandes bloques: el navegador instalado en el dispositivo móvil del usuario, el servidor de Layar y el servidor que contiene la capa que estamos utilizando (“Hosting”). En la siguiente imagen se ven reflejados los tres bloques:

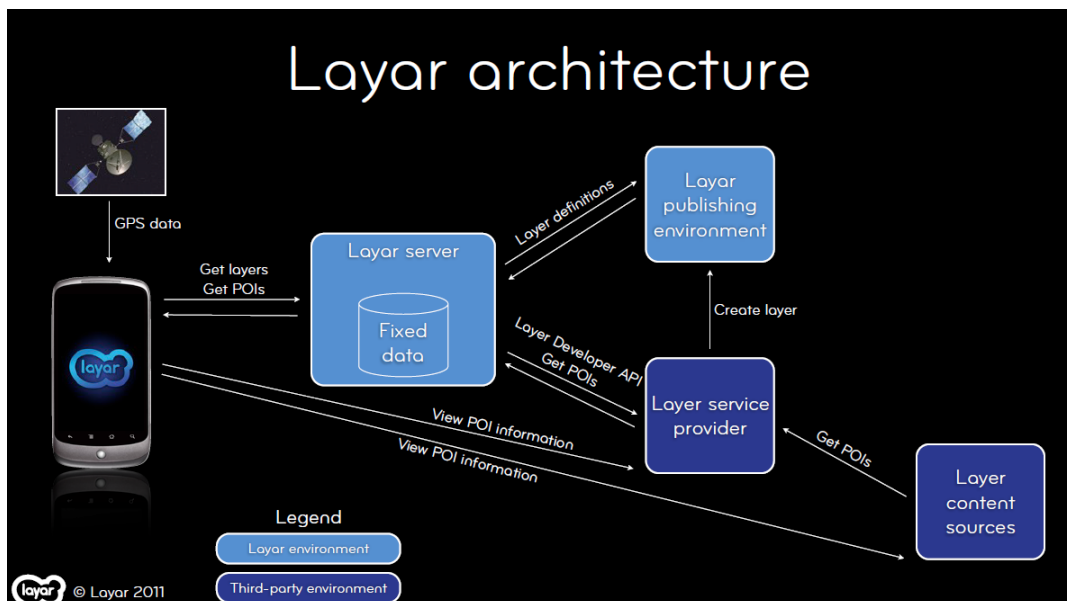


Fig. 7. Arquitectura Layar

Como ya se ha comentado, Layar dispone de un navegador de RA que el usuario instala en su móvil. Además, el móvil debe disponer de la cámara, el GPS y demás elementos que se comentaron como indispensables al explicar este tipo de RA.

Como segundo gran bloque tenemos los servidores de Layar. El cliente entrará en el navegador de RA de Layar y seleccionará la capa. Es en este momento cuando Layar recibe lo que el usuario está pidiendo y detecta en su Base de Datos si dicha capa existe. En caso de que exista, Layar re direccionará la petición al tercer gran bloque, comunicándole a éste cuál es la posición GPS y otros datos que puedan ser necesarios.

El tercer bloque está formado por un servidor Web y un servidor de Base de Datos, en el caso particular del TFC será un servidor PHP y prescindiremos de la BD que sustituiremos por un fichero

estático XML para mantener los datos de los POIs. El primer servidor recibirá la petición que le está enviando Layar y realizará una consulta en la Base de Datos (consulta sobre fichero XML). Finalmente devolverá la respuesta en formato JSON, que será recibida en el aplicativo Layar instalado en el móvil del usuario. Esta respuesta son los llamados Puntos de Interés (POIs), que serán mostrados en pantalla como objetos.

Cabe destacar en este punto que la visualización de los POIs es una de la ventaja más importante de haber utilizado una aplicación existente como Layar, ya que se reutiliza elementos disponibles en Layar pero que pueden ser configurados por el desarrollador.



Fig. 8. Logo Layar.

## 4.2 Servidor Web PHP

Para la gestión y servir los datos de los POIs utilizaremos un servidor que contendrá toda la lógica de negocio necesaria para el funcionamiento de la capa de Layar cuando necesite los datos.

Para la implementación del servidor de POIs se ha reutilizado el código de licencia libre BSD PorPOISe (*Portable Point-of-Interest Server for Layar*). La decisión de reutilizar código se ha tomado basándonos en que PorPOISe es un servidor PHP lo que facilita su implantación en hosting gratuito o de bajo coste. Asimismo, al disponer de todo el código fuente podemos adaptarlo a nuestras necesidades al ser de licencia libre y para acabar de convencernos, en 2011 pasó a formar parte del desarrollo de Layar. Como veremos en el apartado **6 Implementación**, El código original PorPOISe se ha traducido, simplificado y publicado para la comunidad en la Forja de GitHub.

PHP, fue uno de los primeros lenguajes de programación del lado del servidor que se podían 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. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

En 2007, más de 20 millones de dominios de Internet tenían servicios web con PHP instalado y hoy en día se utiliza en el 75% de todos los sitios web como lenguaje de programación del lado del servidor.



Fig. 9. Logo PHP.

### 4.3 Google Maps API

Como ya se ha comentado, el navegador de Realidad Aumentada Layar utiliza la API de Google Maps para insertar rutas en los mapas de Google.

Hay muchas formas de insertar rutas de Google Maps en las aplicaciones. Allá por junio de 2005 Google lanzó Google Maps API para integrar Google Maps en sus sitios web como un servicio gratuito. Gracias al API de matriz de distancia, los usuarios pueden encontrar las mejores rutas en coche y conocer el tiempo necesario para llegar a su destino.

Una de las funcionalidades de este proyecto tiene como objetivo mostrar la información geo localizada de una manera amigable por lo que el API de Google Maps parece una buena solución , ya que hace posible integrar Mapas en un sitio web externo y proporciona un gran número de códigos de muestra y documentación sobre los casos de uso.

En junio de 2010, más de 350.000 sitios web utilizan la API de Google Maps y hoy en día la página de documentación de desarrolladores de Google tiene más de 390.000 usuarios diarios.



Fig. 10. Logo Google Maps.

## 4.4 Funcionamiento y conexiones entre servidores (Layar, Servidor de POI, Google Maps)

En alguna ocasión se ha comentado que **Layar** ha sido creado como un navegador especial para Realidad Aumentada que, a partir de nuestro posicionamiento GPS, la brújula, la cámara y una conexión a internet, nos permite mostrar por pantalla las capturas del mundo real, añadiendo una capa virtual con información y posicionamiento de puntos de interés (POI).

En este navegador se pueden ir agregando layers (capas) que funcionan de una manera similar a los complementos de un navegador web normal. Cada capa agrega información y complejidad a la realidad aumentada. Es por ello que, cuando nos conectamos a Layar, se debe elegir una capa a la que conectarse.

Una vez seleccionada la capa, la aplicación Layar obtiene la posición GPS del usuario y hace una petición de POIs enviando al servidor de Layar diferentes campos, como el nombre de capa, posición GPS, etc. Cuando Layar recibe dicha información, comprueba si existe esa capa en su base de datos y de ser así, la base de datos (BD) de Layar ya tendrá asociada una URL del servidor Web externo que hace de proxy entre el servidor Layar y la BD del autor. Se reenvía la petición a dicha URL y ésta realiza la consulta en la BD (en nuestro caso el fichero XML) que dispone y devuelve (mediante JSON) los puntos que están en el rango. Una vez recibidos los puntos, la aplicación se encarga de posicionarlos en la pantalla según la orientación del usuario.

Finalmente, las conexiones que se realizarán con Google Maps serán a través de la API que éste provee y que desde Layar se invocará con los parámetros necesarios para mostrar la mejor ruta hasta el POI.

A continuación se muestra, paso a paso, como viajarán las peticiones durante el funcionamiento del aplicativo que se pretende implementar para el TFC:

1. El usuario se conecta a Layar en su móvil, busca la capa y selecciona la capa "Rocódromos".
2. El usuario seleccionará el rango máximo donde quiere que aparezcan los POI en el mapa.
3. Layar busca en su Base de Datos y detecta que ésta capa existe.
4. Layar redirige la petición a la URL del servidor de POIs (servidor web PHP) en el que se encuentra la BD (fichero XML) con los datos de los POI y el servidor web hace la consulta sobre el XML creado con los datos de los rocódromos. Cabe destacar que, en esta redirección, Layar enviará los parámetros necesarios para que el servidor de POI devuelva los POI necesarios. Los parámetros son:
  - **lan**: corresponde al lenguaje
  - **countryCode**: código del país
  - **userId**: id del usuario
  - **lon**: longitud de la posición del usuario
  - **lat**: latitud de la posición del usuario

- **radius**: radio de detección de POI
  - **layerName**: nombre de la capa
5. El servidor de POIs envía la respuesta a Layar en formato JSON con los datos que cumplan con los parámetros recibidos en la petición Layar.
  6. Layar mostrará al usuario aquellos POIs que estén dentro de la distancia seleccionada anteriormente por el usuario.
  7. Si el usuario pulsa sobre el enlace de cómo llegar, el servidor Layar conectará a través de la API de Google Map con la aplicación para mostrar en el mapa el camino hasta llegar al POI.
  8. Si el usuario pulsa encima del enlace a más información, el servidor Layar conectará a través de HTTP con la página web que contenga la información.

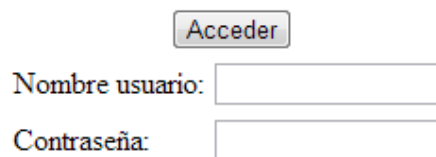
## 5. Prototipo de interfaz de usuario

En este apartado veremos los prototipos de las pantallas más importantes que compondrán el servidor Web y la capa Layer de rocódromos.

### 5.1 Pantallas del servidor de POIs

#### 5.1.1 Pantalla de acceso

Una vez que accedamos a la URL del gestor de POIs nos aparecerá una pantalla de login tal como se muestra a continuación.



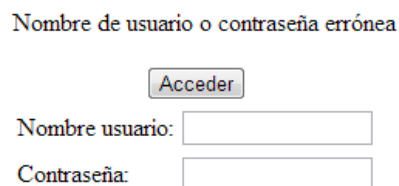
Acceder

Nombre usuario:

Contraseña:

Fig. 11. Pantalla login de usuario.

Si el usuario/contraseña es incorrecto se mostrará un mensaje de error en la misma pantalla.



Nombre de usuario o contraseña errónea

Acceder

Nombre usuario:

Contraseña:

Fig. 12. Login erróneo de usuario.

## 5.1.2 Pantalla principal de datos de la capa

A continuación se muestra el prototipo de pantalla a la que se llegará una vez logado en el sistema. En ella apreciaremos el nombre de la capa que queremos editar además del ID del desarrollador y los botones de Salir y Principal.

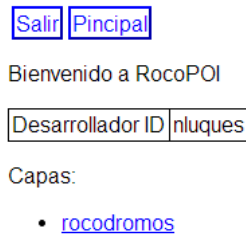


Fig. 13. Pantalla principal.

## 5.1.3 Pantalla detalle de capa (listado de POIs)

El prototipo que se muestra a continuación es para la pantalla de detalle de la capa en la que se mostrará un listado de todos los POIs y desde la cual se podrá ver detalle/modificar/eliminar uno de ellos, así como crear uno nuevo. Asimismo, tendremos la posibilidad de salir del sistema o ir a la pantalla principal.

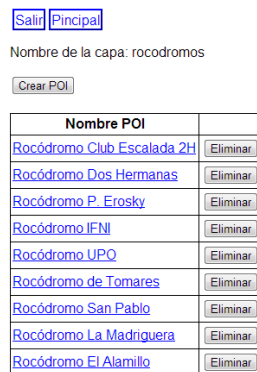


Fig. 14. Pantalla listado POIs.



### 5.1.4 Pantalla de alta/edición/detalle de POI

Finalmente, el siguiente prototipo es para la pantalla de alta/edición/detalle de un POI. Asimismo, al igual que en las anteriores pantallas, el usuario podrá Salir del sistema o ir a la pantalla principal.

[Salir](#) [Principal](#)  
[Volver a rocodromos](#)

General			
ID	6		
Título	Rocódromo IFNI		
Descripción	Rocódromo situado cerca de Avda. de		
Pie de página	En necesario estar federa		
URL Imagen	<a href="http://rocopoi.net23.net/ro">http://rocopoi.net23.net/ro</a>		
Geolocalización			
Lat/lon/alt	37.354141	-5.982673	0
Enlaces			
Enlace	Texto	Web	
<a href="#">Eliminar</a>	URI	<a href="http://www.clubelbuz.es/i">http://www.clubelbuz.es/i</a>	
<a href="#">Nuevo Enlace</a>			
<a href="#">Guardar</a>			

Fig. 15. Pantalla gestión de POI.

## 5.2 Pantallas de la capa Layar

### 5.2.1 Pantalla búsqueda de capas

Desde el menú principal del navegador de realidad aumentada Layar tendremos, entre otras opciones, un buscador de capas desde el que accederemos a la pantalla de búsqueda de capas para realizar la búsqueda de nuestra capa de rocodromos.



Fig. 16. Menú principal.

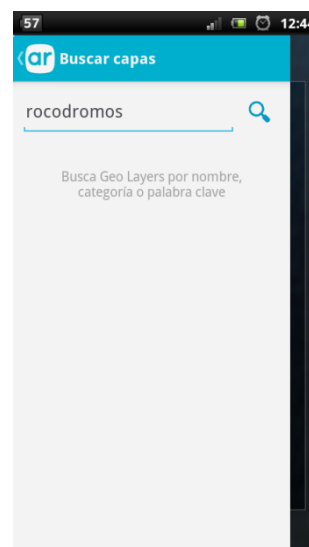


Fig. 17. Buscador de capas

## 5.2.2 Pantalla ajustes de usuario

La siguiente pantalla es la que contiene todos los accesos a las posibles configuraciones de usuario que permite Layar.

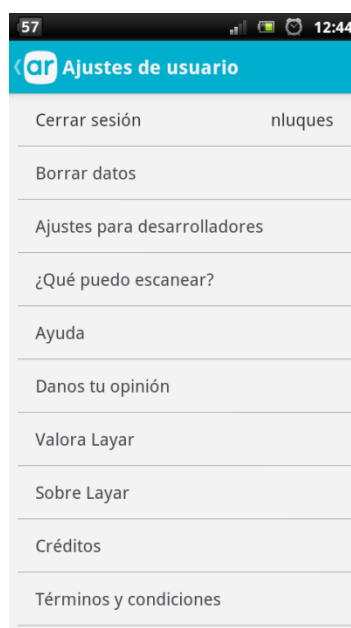


Fig. 18. Menú de ajustes de usuario.

## 5.2.3 Pantalla de capas recientes

A continuación se muestra la pantalla donde se encontrarán las distintas capas recientemente utilizadas.

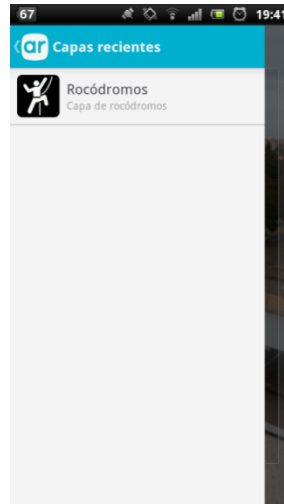


Fig. 19. Acceso a capas recientes.

### 5.2.4 Pantalla ajustes de capa (Rango de búsqueda)

Las siguientes pantallas serán el menú de propiedades de la capa y la barra de rango de búsqueda desde donde se controlará el radio máximo donde se mostrarán POIs (rocódromos).

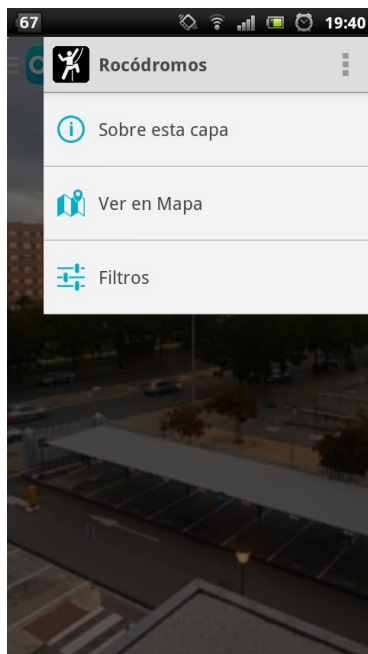


Fig. 20. Menú de la capa.

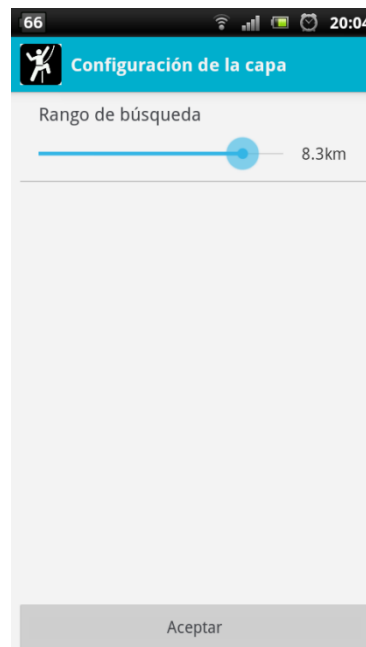


Fig. 21 Rango de búsqueda.

## 5.2.5 Pantalla principal de capa

Las siguientes pantallas muestran algunos de los estados en los que se puede encontrar la pantalla principal de la capa de rocódromos. Los diferentes estados son:

- Detectando ubicación: mostrará la imagen que obtenga la cámara y el texto “Detectando ubicación...”.
- Trayendo resultados: mostrará la imagen que obtenga la cámara y un texto “Trayendo resultados...”.
- Puntos encontrados: mostrará la imagen que obtenga de la cámara y un texto “Se encontraron XXX puntos de interés”.
- No se encontraron resultados: mostrará la imagen que obtenga de la cámara y un texto “No se encontraron resultados”.

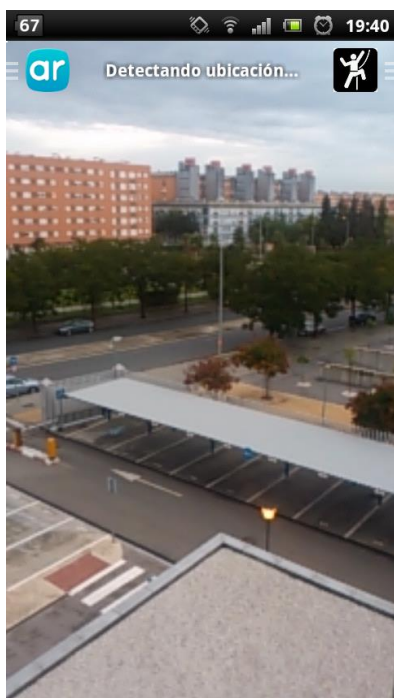


Fig. 22. Detección de ubicación.

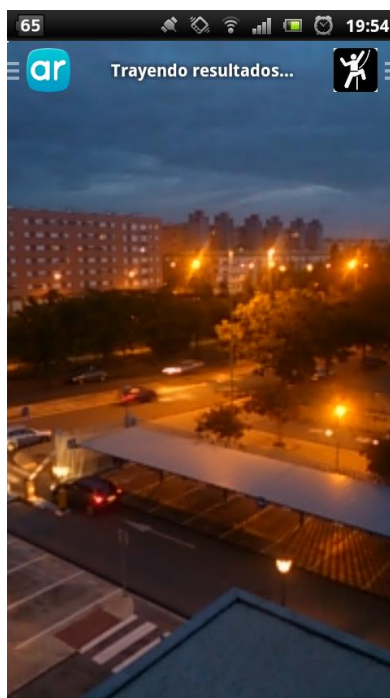


Fig. 23. Importando resultados desde servidor de POIs.



Fig. 24. Mostrando resultados en pantalla.

## 5.2.6 Pantalla de ampliar información y cómo llegar

A continuación se muestran las pantallas de ampliar información (Web) y la de Google Maps para la opción de Cómo llegar.



Fig. 25. Web externa que amplía información.



Fig. 26. Google Maps para la opción de Cómo llegar.

## 6. Implementación

### 6.1 Servidor de POIs

Como ya se comentó en el apartado de arquitectura del sistema, para la implementación del servidor de POIs se ha reutilizado el código de licencia libre (New BSD License) PorPOISe (*Portable Point-of-Interest Server for Layar*).

El código PorPOISe original ha sido adaptado a nuestras necesidades mejorando algunos problemas que tenía a la hora de gestionar el fichero de datos XML. Asimismo, se ha traducido al castellano y simplificado para que solo aparezcan los campos necesarios a la hora de gestionar los POIs. Para dar un valor añadido, se ha publicado el código como un clon del original en la forja de GitHub y se puede descargar de la siguiente dirección: <https://github.com/nluque/PorPOISe-clone>.

#### 6.1.1 Estructura de carpetas

A continuación se lista la estructura principal de carpetas del servidor

- **/rocopoi**: en la raíz se encuentran las clases necesarias para conectar con Layar y para realizar las transformaciones y conexión para crear el XML de los POIs así como la clase que indica la ubicación de los ficheros de configuración y XML.
- **/rocopoi/config5684**: en esta carpeta encontraremos los ficheros de configuración y de mantenimiento de los datos.
- **/rocopoi/web/dashboard**: en esta carpeta se encuentran todas las clases necesarias para la gestión de los POIs incluidas las interfaces. Asimismo, se encuentran las clases necesarias para el login de usuario.
- **/rocopoi/web/img**: en esta carpeta se encuentran las imágenes de cada uno de los POIs que posteriormente se mostrarán en la capa Layar.

#### 6.1.2 Clases modificadas

En este apartado realizaremos una descripción de las clases modificadas y del funcionamiento de las mismas. Para facilitar el entendimiento de las mismas utilizaremos tablas que mantendrán ordenados los cambios.

##### 6.1.2.1 Configuración

En este apartado se describen todas las clases que se utilizan para la configuración del servidor.

<b>Nombre</b>	<b>config.php</b>
<b>Ubicación</b>	rocopoi/
<b>Descripción</b>	Clase donde se configura el directorio donde se encuentra el fichero de configuración de los datos en XML y, si fuera el caso, de la Base de Datos.
<b>Modificaciones</b>	<ul style="list-style-type: none"> <li>Modificado el nombre de la carpeta de configuración.</li> </ul> <pre>if (!defined('PORPOISE_CONFIG_PATH')) {     define("PORPOISE_CONFIG_PATH", "config5684"); }</pre>
<b>Estado</b>	Estable
<b>Comentarios</b>	N/A

<b>Nombre</b>	<b>config.xml</b>
<b>Ubicación</b>	rocopoi/config5684
<b>Descripción</b>	Fichero donde se configuran los datos del fichero XML que mantendrá los datos además del usuario, contraseña de la cuenta Layar donde se encuentra la capa a la que enviará el código JSON con los datos solicitados por ésta.
<b>Modificaciones</b>	<ul style="list-style-type: none"> <li>Configuración de la cuenta Layar (usuario, contraseña y nombre de la capa).</li> <li>Configuración del tipo de conector XML y nombre del XML que contendrá los datos.</li> </ul> <pre>&lt;porpoise-configuration&gt;   &lt;developer-id&gt;androidTFC&lt;/developer-id&gt;   &lt;developer-key&gt;xxxxxxx&lt;/developer-key&gt;   &lt;connectors&gt; &lt;/connectors&gt;   &lt;layers&gt;     Rocodromos ciudad de Sevilla   &lt;layer&gt;     &lt;name&gt;rocodromos&lt;/name&gt;     &lt;connector&gt;XMLPOIConnector&lt;/connector&gt;     &lt;source&gt;rocodromos_sevilla.xml&lt;/source&gt;   &lt;/layer&gt; &lt;/porpoise-configuration&gt;</pre>
<b>Estado</b>	Estable
<b>Comentarios</b>	N/A

<b>Nombre</b>	<b>rocodromos_sevilla.xml</b>
<b>Ubicación</b>	rocopoi/config5684
<b>Descripción</b>	Es el fichero que contendrá los datos de los POI sobre rocódromos.
<b>Modificaciones</b>	La modificación de este fichero será dinámica y se realizará a través de la página de edición de POIs ( <i>dashboard</i> ).
<b>Estado</b>	Estable
<b>Comentarios</b>	N/A

<b>Nombre</b>	<b>users.inc.php</b>
<b>Descripción</b>	Clase que mantiene los datos del usuario administrador encriptados en md5
<b>Modificaciones</b>	<ul style="list-style-type: none"> <li>Incluido el siguiente código que es necesario para logarnos con un usuario administrador. La contraseña está codificada en MD5.  <code>\$_access["administrador"] = '\$1\$B7dLV7eL\$W0qZbGIRAH51qVI5/azps.';</code> </li> </ul>
<b>Estado</b>	Estable
<b>Comentarios</b>	N/A

### 6.1.2.2 Login

A continuación se muestra la clase que se utiliza para la autenticación de usuarios.

<b>Nombre</b>	<b>authorize.inc.php</b>
<b>Descripción</b>	Es la clase que ejecuta el login de usuario.
<b>Modificaciones</b>	<ul style="list-style-type: none"> <li>Traducción de textos al castellano.</li> </ul>
<b>Estado</b>	Estable
<b>Comentarios</b>	Esta clase utiliza el fichero <b>users.inc.php</b> que se ha explicado anteriormente.

### 6.1.2.3 Clases gestión de POI

En este apartado se describen las clases que se utilizan para toda la gestión de los datos de los rocódromos.



<b>Nombre</b>	<b>gui.class.php</b>
<b>Ubicación</b>	rocopoi/web/dashboard
<b>Descripción</b>	Es la clase principal del <i>dashboard</i> , en ella se encuentra toda la lógica para la interfaz gráfica y se utiliza para listar los POI, editarlos, grabarlos y eliminarlos.
<b>Modificaciones</b>	<ul style="list-style-type: none"> <li>• Traducción de textos al castellano.</li> <li>• Eliminados campos innecesarios para la capa.</li> </ul>
<b>Estado</b>	Estable
<b>Comentarios</b>	N/A

<b>Nombre</b>	<b>script.js</b>
<b>Ubicación</b>	rocopoi/web/dashboard
<b>Descripción</b>	Fichero JavaScript que se utiliza para acciones dinámicas necesarias en la pantalla de administración de POI, por ejemplo para incluir dinámicamente campos de enlace a páginas externas a la capa.
<b>Modificaciones</b>	<ul style="list-style-type: none"> <li>• Traducción de textos al castellano.</li> </ul>
<b>Estado</b>	Estable
<b>Comentarios</b>	N/A

#### 6.1.2.4 Transformador XML

El punto más trascendente de la reutilización del código fuente de PorPOISe ha sido para solventar un problema que trae el código a la hora de crear los ficheros XML que contendrán los datos de los POI.

En la labor de investigación y entendimiento del código fuente original, llegamos a comprobar que a la hora de grabar el fichero XML en disco, PorPOISe incluye una etiqueta (<transform/>) que crea conflicto con el JSON que espera Layar con la consecuencia de que no muestra los iconos en forma de disco de los POI en pantalla.

La solución del error ha pasado por modificar la clase **xmlpoiconnector.class.php** modificando la parte del código donde se incluía dicha etiqueta en el montaje del XML y que posteriormente se grababa en el fichero *rocódromos\_sevilla.xml*. Esta etiqueta es indiferente para nuestra implementación por lo que se ha comentado el código siguiente para que se muestren correctamente los iconos.

A continuación se muestra el código que se ha modificado.

```

else if ($key == "transform") {
    /*$transformElement = $poiElement->addChild("transform");
    $rotateElement = $transformElement->addChild('rotate');
    $rotateElement->addChild('rel', $poi->transform->rotate['rel!']);
    $axisElement = $rotateElement->addChild('axis');
    foreach(array('x','y','z') as $elementName) {
        $axisElement->addChild($elementName, $poi->transform-
    >rotate['axis'][$elementName]);
    }
    $rotateElement->addChild('angle', $poi->transform->rotate['angle!']);
    $translateElement = $transformElement->addChild('translate');
    foreach(array('x','y','z') as $elementName) {
        $translateElement->addChild($elementName, $poi->transform-
    >translate[$elementName]);
    }
    $transformElement->addChild('scale', $poi->transform->scale);*/
}
}

```

## 6.2 Capa Layar – Rocódromos

A continuación se muestra mediante capturas de pantalla el trabajo de configuración sobre la plataforma Layar para la implementación de la capa de Rocódromos que se ha desarrollado. Cabe destacar que la explicación de los campos de cada una de las pantallas se basará solo en los que se han utilizado para la creación de la capa rocódromos.

### 6.2.1 Nueva capa (Create a Layer)

El primer paso que hay que realizar antes de crear la capa es registrarse en la Web Layar.com. Una vez registrado, iremos al menú “My layers” y allí pulsaremos el botón “Create a new layer!” y nos aparecerá la siguiente ventana.

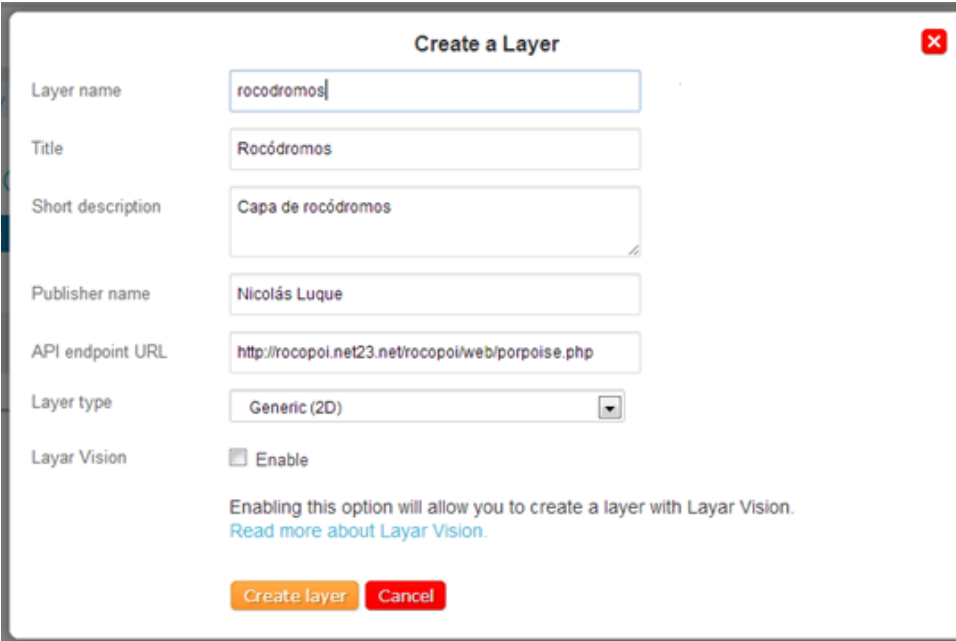


Fig. 27. Pantalla creación de capa.

En esta ventana tendremos que incluir los datos siguientes con las consideraciones que se indican:

- **Layer name:** nombre de la capa, tendrá que ser un nombre único que no exista en la plataforma Layar. Será el ID de la capa.
- **Title:** Título de la capa. Es por este nombre por el que se buscará la capa.
- **Short description:** Pequeña descripción de la capa a crear.
- **Publisher name:** Nombre del desarrollador.
- **API endpoint URL:** Dirección del servidor que contendrá los datos de los diferentes POIs y que devolverá el JSON.

## 6.2.2 Punto de acceso (API endpoint)

Una vez creada la capa, tendremos que configurar cada uno de los parámetros necesarios para que esta funcione. Para ello comenzaremos por la zona de API endpoint cuya pantalla se muestra a continuación y en la que, en principio, no se debería modificar nada puesto que ya en la pantalla de creación hemos dicho la URL del punto de acceso al servidor de POI.

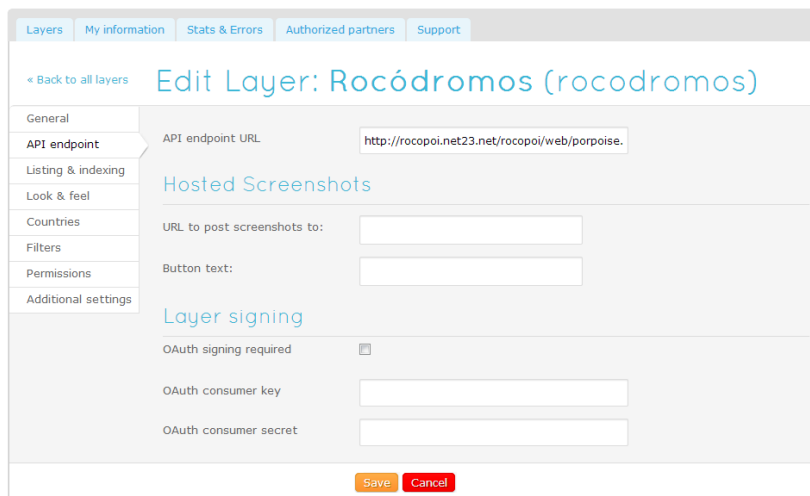


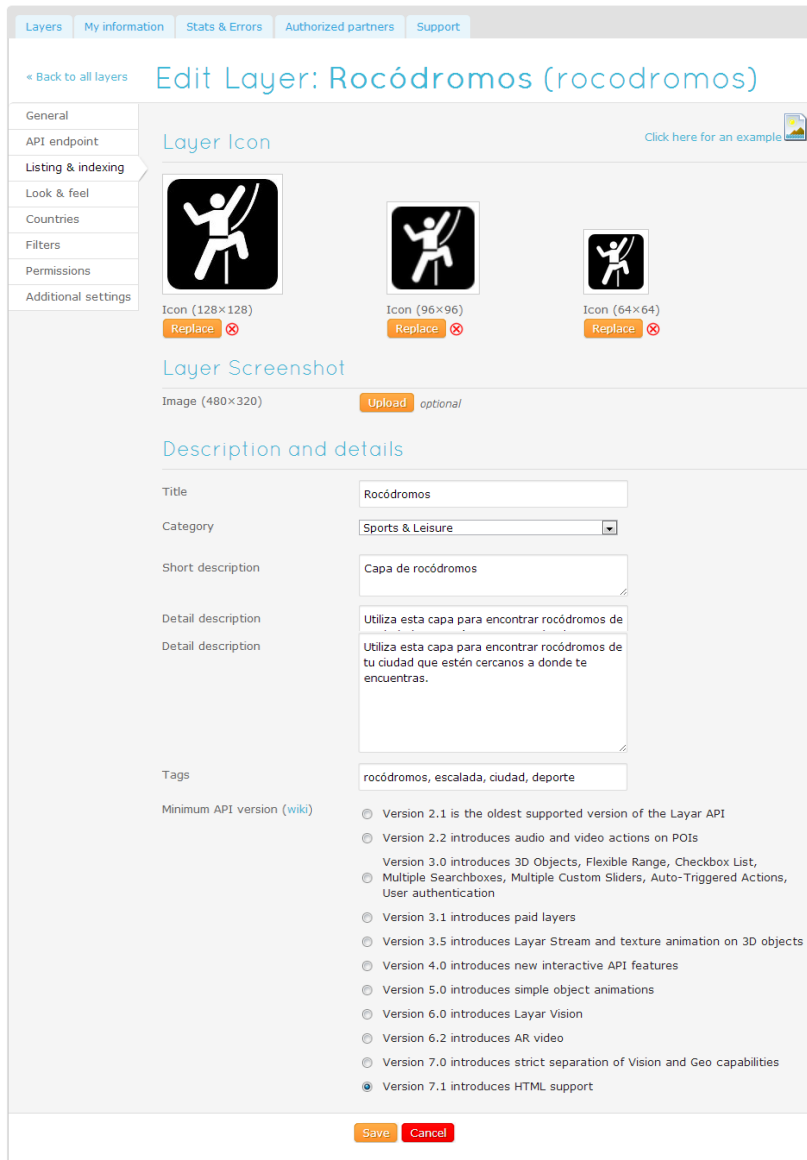
Fig. 28. Pantalla configuración punto de acceso

## 6.2.3 Iconos y detalle (Listing & indexing)

En la siguiente pantalla se configura principalmente los logotipos de la capa, la descripción y detalles de la misma, una serie de etiquetas que nos sirvan para encontrar la capa con las búsquedas a través del navegador Layar y por último, el mínimo de versión de API que soportará la capa. Esto último se utiliza para poder incluir nuevas funcionalidades que se permiten en las últimas versiones de Layar.

Los campos que se han configurado son:

- **Layer Icon:** donde configuraremos el logo vinculado a la capa a alta resolución siguiendo las proporciones (128x128) y que el sistema utilizará para crear el de media (96x96) y baja resolución (64x64).
- **Description and details**
  - **Title:** Título de la capa (vendrá precargado de la pantalla de creación de capa).
  - **Category:** Es la categoría dentro de la que se ubicará la capa, es indispensable para que aparezca en las búsquedas.
  - **Short description:** Descripción corta de la capa (vendrá precargado de la pantalla de creación de capa).
  - **Detail description:** Descripción detallada de la utilidad de la capa.
  - **Tags:** Etiquetas que sirven para facilitar la aparición de la capa en las búsquedas.
  - **Minimum API version:** Versión de Laya que se soporta en la capa.



The screenshot shows the 'Edit Layer' interface for 'Rocódromos (rocodromos)'. It features a navigation menu on the left with options like 'General', 'API endpoint', 'Listing & indexing', 'Look & feel', 'Countries', 'Filters', 'Permissions', and 'Additional settings'. The main content area is divided into sections: 'Layer Icon' with three icon options (128x128, 96x96, and 64x64), 'Layer Screenshot' with an 'Upload' button, and 'Description and details' with form fields for Title, Category, Short description, Detail description, Tags, and Minimum API version. The 'Minimum API version' section includes a list of versions from 2.1 to 7.1, with 7.1 selected. At the bottom, there are 'Save' and 'Cancel' buttons.

Fig. 29. Pantalla de iconos y detalles.

## 6.2.4 Diseño y estilos (Look & feel)

La siguiente pantalla de configuración se utilizará para incluir el Banner de la aplicación, para configurar los colores del texto y fondo del marco de información y para indicar los colores que tendrán los POI sobre la pantalla en sus diferentes estados.

Los campos que se han configurado son:

- **Upload a Banner Icon:** donde configuraremos el banner a alta resolución (High-res) siguiendo las proporciones (120x52) y que el sistema utilizará para crear el de media (Medium-res) y baja resolución (Low-res).
- **Set Colors**
  - **Banner text color:** Color del texto para el título del POI.
  - **Banner Background color:** Color de fondo para el título del POI.
  - **POI Spot Color:** Color del POI en el radar al estar visualizado.
  - **POI Inner Color:** Color del POI en pantalla cuando no está apuntado directamente.
  - **BIW Background Color:** Color de fondo para el detalle del POI.
  - **BIW Title Color:** Color del título del POI.
  - **BIW Text Color:** Color del texto del detalle del POI.

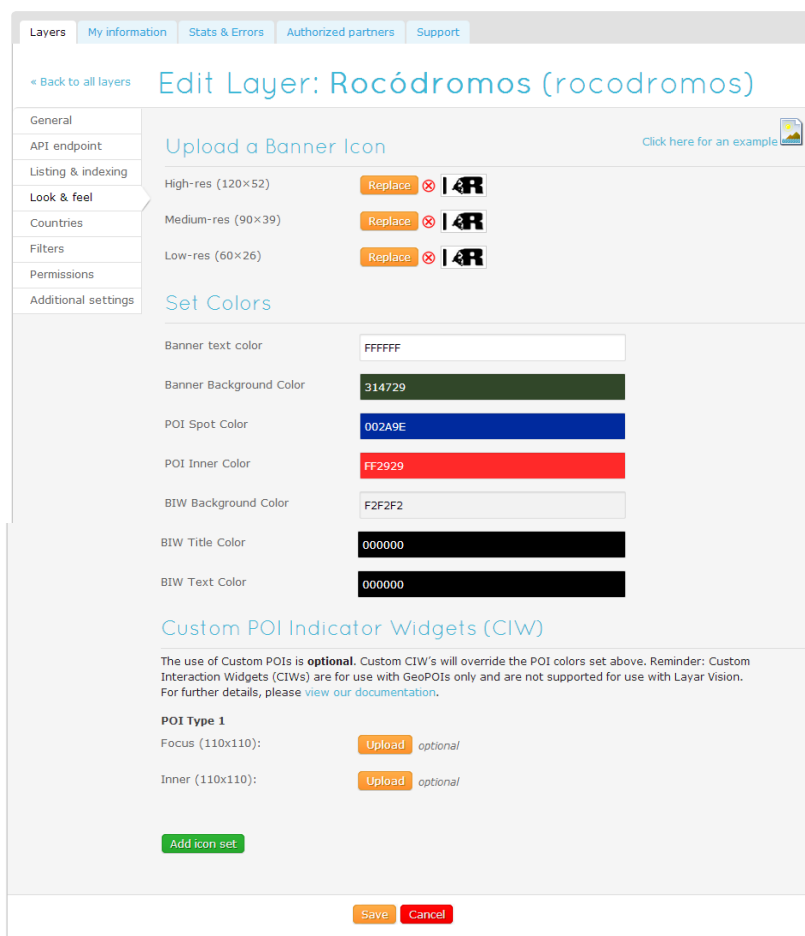


Fig. 30. Pantalla diseño y estilos.

## 6.2.5 País (Country)

La pantalla de configuración de país nos servirá para indicar en que país funcionará la capa que se está creando.

Los campos a seleccionar son básicamente el traspaso de la tabla **Available options** a la tabla **Selected** que serían los países seleccionados para hacer uso de la capa.

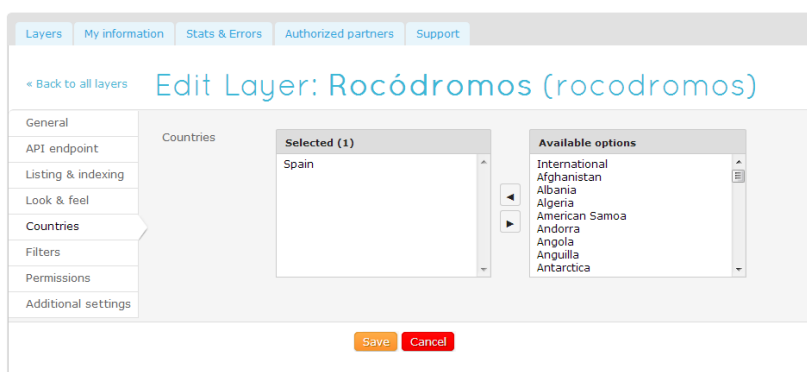


Fig. 31. Pantalla configuración país.

## 6.2.6 Filtros (Filters)

Finalmente, tenemos la pantalla de configuración de filtros. En esta pantalla podremos crear varios tipos de filtros que luego aparecerán en la pantalla de opciones de la capa.

En el caso de la capa que nos ocupa tan solo se ha creado el filtro “Rango de búsqueda” que nos servirá para mostrar resultados de POI que se encuentren dentro del rango seleccionado.

Los campos que disponemos son:

- **Label:** Título del filtro.
- **Minimum value:** Valor mínimo que tendrá la barra de rango.
- **Default value:** Valor por defecto si no se selecciona rango.
- **Maximum value:** Valor máximo que tendrá la barra de rango.

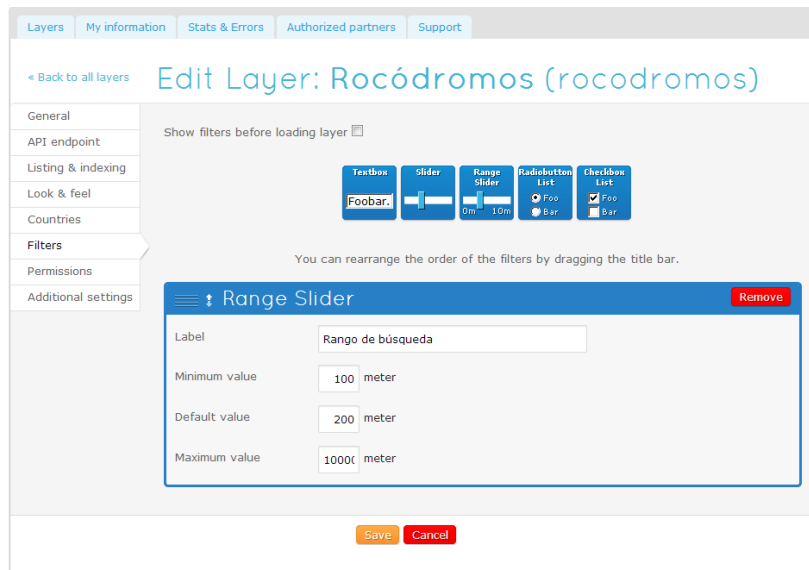


Fig. 32. Pantalla creación de filtros.

### 6.2.7 Prueba y publicación de la capa

Una vez que todo ha quedado completamente configurado se pulsará el botón “Save” y quedará la capa guardada y configurada para poder probarla.

Para probar la capa podemos usar el botón Test y nos abrirá una ventana en la que aparece Google Maps y los filtros que se han configurado en la capa, asimismo, podemos realizar la prueba de nuestra capa a través del dispositivo móvil haciendo login con nuestro usuario y accediendo a las capas de prueba (*Ver punto 7.3. Prueba de capa Layer – Rocódromos*).

Finalmente, una vez que hemos probado la capa y hemos realizado todas las pruebas pertinentes sobre ésta, procederemos a la publicación de la capa. Para ello, haremos una petición a Layar a través del botón “Publish”, a partir de ahí, Layar tardará como mínimo 5 días en publicar nuestra capa.

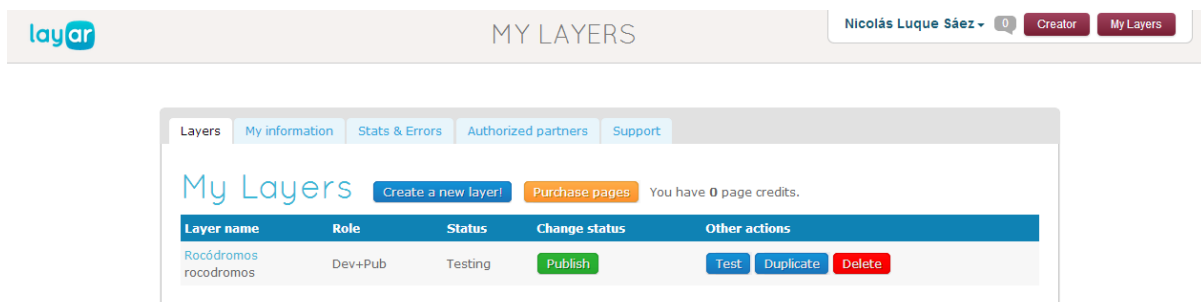


Fig. 33. Pantalla principal de mis capas.

## 7. Implantación

Para usar la capa Layar es necesario instalar un servidor donde se mantendrán los datos de los rocódromos. Asimismo hay que tener instalado en el dispositivo el navegador de realidad aumentada Layar.

A continuación se muestran cada uno de los pasos a seguir para instalar los sistemas necesarios.

### 7.1 Instalación del servidor de POIs

Aunque el servidor de POIs de rocódromos ya se encuentra completamente instalado y configurado para servir POIs a la capa Layar de Rocódromos, vamos a mostrar los pasos necesarios para instalar dicho servidor por si se quisiera migrar de hosting.

Datos para acceder a la zona de administración de POIs del servidor actualmente instalado: <http://rocofois.webcindario.com/rocopoi/web/dashboard/index.php>.

- Usuario: administrador
- Contraseña: bscw5684

#### 7.1.1 Pasos para la instalación

- a. Descomprimir el fichero **rocopoi.zip** ubicado en la carpeta
- b. Subir la carpeta obtenida (rocopoi) al servidor PHP en una carpeta que tenga visibilidad desde internet.
- c. Finalmente, para comprobar que todo ha ido bien, acceder a la siguiente dirección: <http://dominio.com/rocopoi/web/porpoise.php> donde [dominio.com](#) será el nombre del dominio del servidor. Si todo ha ido bien debe mostrar el siguiente código JSON:

```
{"layer":"unspecified","errorCode":20,"errorString":"Missing parameter:
userId","hotspots":[],"nextPageKey":null,"morePages":false}
```

NOTA: si devolviera alguno de estos errores

- **Error 404** → Verifica la ruta
- **Error 403** → Verifica los permisos



### 7.1.1.1 Configuración

El servidor se ha dejado totalmente configurado a falta de las URL de las imágenes de los rocódromos puesto que estas son direcciones absolutas a la ubicación de la imagen en el servidor.

Es por ello que, una vez instalado, debemos acceder a la gestión de POIs para configurar la ruta de las imágenes. Para ello accederemos a través de la URL: <http://dominio.com/rocopoi/web/dashboard/index.php> donde [dominio.com](http://dominio.com) será el nombre del dominio del servidor.

- Usuario: administrador
- Contraseña: bscw5684

Para realizar dicha configuración tendremos que acceder a cada uno de los POIs y cambiar en la ruta de la imagen el nombre del dominio. Es decir, dentro de cada uno de los POIs tendremos que cambiar el campo **URL Imagen** donde cambiaremos el nombre del dominio.

Por ejemplo, el formato para la imagen del rocódromo IFNI tendría que ser el siguiente: <http://dominio.com/rocopoi/web/img/ifni.jpg> donde [dominio.com](http://dominio.com) será el nombre del dominio del servidor.

## 7.2 Instalación navegador Layar

La instalación del navegador Layar no presenta ninguna dificultad añadida a la que tenga cualquier otra aplicación existente en Play Store de Android.

### 7.2.1 Pasos para la instalación

- a. Abrir la aplicación Play Store de Android.
- b. Buscar la aplicación “Layar” a través del buscador.
- c. Instalar la aplicación.

Una vez instalada la aplicación ya podemos abrir el programa para realizar las pruebas de la capa.


## 7.3 Prueba de capa Layar - Rocódromos


Actualmente la capa aún se encuentra en fase de pruebas puesto que no ha habido tiempo material para realizar la solicitud de publicación y que desde Layar puedan publicarla por lo que para probarla lo haremos desde el entorno de pruebas que tiene la aplicación Layar.

### 7.3.1 Pasos para probar la aplicación

Como se indica en la documentación de análisis y diseño, en esta primera versión la capa de Rocódromos se ha implementado con datos de la ciudad de Sevilla por lo que para poder probar la capa en cualquier otro punto habrá que indicarle al navegador que nos encontramos en un punto fijo de la ciudad de Sevilla para que suplante la ubicación actual.

Por todo ello, para probar la aplicación con los datos de los rocódromos de la ciudad de Sevilla tendremos que realizar los siguientes pasos:

- a. Tener conexión a internet y el GPS habilitado.
- b. Abrir la aplicación Layar.
- c. Dentro del menú que se despliega pulsando sobre el icono de Layar () en la esquina superior derecha, Ir a → **Ajustes de usuario** → **Usuario** y logarse con los siguientes datos:
  - Usuario: androidTFC
  - Contraseña: bscw5684
- d. Ir a → **Ajustes de usuario** → **Ajustes para desarrolladores**, una vez en esta pantalla marcar el check de **Usar ubicación fija** e indicar los siguientes valores:
  - Latitud: 37.355558
  - Longitud: -5.982381
- e. Volveremos al menú principal y para cargar la capa iremos a → **Geo Layers** → **Capas de prueba** → **Rocódromos**.
- f. Finalmente, una vez cargada la capa nos solicitará el rango de búsqueda dentro del cual queremos visualizar rocódromos (entre 200m y 10000m)

Para la prueba que acabamos de configurar nos hemos ubicado a menos de 200 m de uno de los rocódromos (Rocódromo IFNI) por lo que aparecerá en pantalla como uno de los puntos más cercanos. Si aumentamos el rango de búsqueda a más de 200m (distancia por defecto) aparecerán más rocódromos, para ampliar el rango podremos ir al menú de la capa pulsando sobre el icono de la capa () ubicado en la esquina superior izquierda → **Filtros** y en esa ventana aumentar el rango.

## 8. Conclusiones

El proyecto, y las metas marcadas en los objetivos, se han desarrollado según los pasos marcados en la planificación y se han cumplido los hitos establecidos.

Con respecto a los riesgos del proyecto hemos conseguido minimizarlos en tiempo utilizando un servidor Web ya existente de código abierto que se ha modificado y adaptado a las necesidades de la capa Layar que se ha diseñado. Como ya se comentó en el punto **6. Implementación**, este código mejorado y traducido al castellano ha sido publicado como un clon del original y se puede descargar de la siguiente dirección: <https://github.com/nluque/PorPOISe-clone>.

En ésta línea, hemos superado las dificultades que supone el corto espacio de tiempo, con lo que se ha de mostrado el aprovechamiento del tiempo disponible, haciendo más productivas las horas dedicadas al proyecto.

Finalmente, me gustaría añadir como opinión personal que durante la realización del presente proyecto he puesto en práctica aptitudes adquiridas a lo largo de mis años de estudio y que por las características iniciales del proyecto intuía que sería lo más importante. Lo que quiero decir es que, al enfrentarme a un proyecto en el cual no he creado un CORE Android desde su SDK, sí que he tenido que utilizar mucho la capacidad analítica y la síntesis de información a la hora de realizar los estudios que me han llevado al diseño y creación de la capa Layar. Por otra parte y muy importante también, he aprendido conocimientos nuevos de Realidad Aumentada y del lenguaje de programación PHP, lenguaje que desconocía.

## 9. Nuevas líneas de ampliación y mejoras

El proyecto que se ha realizado para este Trabajo de Fin de Carrera se puede ampliar con las siguientes funcionalidades y/o mejoras:

- Crear una **Base de Datos** para el mantenimiento de datos de los POI de rocódromos. Esta base de datos estaría creada bajo MySQL y sería atacada desde el servidor web PHP que en la fase actual trabaja con los datos sobre XML.
- Aplicaciones Android auxiliares:
  - Crear una **aplicación Android** que sirva para mejorar la funcionalidad de ampliar información sobre rocódromos. Actualmente, la capa desarrollada está pensada para mostrar más información enlazando con una web que ya exista y contenga dicha información. Con esta nueva aplicación se tendría una solución completa que permitiría, además de ir a una web externa, mostrar información de los rocódromos que no tengan web de información.
  - Crear una **aplicación de Realidad Aumentada** para mostrar las vías de escalada dentro del rocódromo. Un rocódromo está formado por muchas piezas, donde vas colocando manos y pies, que forman las vías. La aplicación que se propone se

enlazaría desde la capa Layar y permitiría apuntar con la cámara hacia la pared del rocódromo y superpondría con realidad aumentada las vías escalables que tiene esa zona del rocódromo.

## 10. Bibliografía

La documentación e información consultada para realizar el sistema de información se ha obtenido de las siguientes fuentes:

- Documentación e información sobre Layar: [www.layar.com](http://www.layar.com)
- Documentación e información sobre Wikitude: <http://www.wikitude.com/>
- Documentación e información sobre API Google Maps: <https://developers.google.com/maps/?hl=es>
- Información, ejemplos y documentación sobre PorPOISe, servidor de POIs: <https://code.google.com/p/porpoise/>
- Información acerca de PHP: <http://php.net/>
- Blog sobre montañismo: <http://destreando.blogspot.com.es/>
- 19 Aplicaciones de realidad aumentada: <http://blogs.elperiodico.com/masdigital/afondo/19-aplicaciones-de-realidad-aumentada>