



**Universidad  
Zaragoza**

Proyecto de fin de Carrera  
Ingeniería en Informática  
Curso 2015/2016

# **ExplorAragon: Juego social sobre Aragón. Desarrollo de una app para el aprendizaje sobre municipios de Aragón**

**Luis Cruz Pérez**

Director: Francisco Javier López Pellicer

Departamento de Informática e Ingeniería de Sistemas  
Centro Politécnico Superior  
Universidad de Zaragoza

Fecha 01 de Septiembre de 2016



# ExplorAragón: Juego social sobre Aragón.

## RESUMEN

En la actualidad el desarrollo de páginas web adaptables a los nuevos dispositivos o de aplicaciones para teléfonos móviles o tablets es uno de los principales motores de negocio de las consultoras TIC. Sin embargo, una importante serie de problemas de índole social, artístico, cultural, educativo,... Suelen quedar abandonadas, normalmente por la escasez de fondos y los precios del desarrollo de estas tecnologías.

ExplorAragón es una aplicación Android que ofrece experiencias de usuario únicas acercando los municipios de Aragón a quienes estén dispuestos a recorrerlos de una forma diferente. Para ello se disponen de dos juegos sociales en los que el usuario podrá competir por demostrar que es quien más sabe sobre esa comunidad.

ExplorAragon aprovecha los datos abiertos de distintas plataformas y un rastreador de información para crear una base de datos a partir de la que generar los juegos. El primero de ellos consiste en un juego de preguntas y respuestas en el que los jugadores podrán seleccionar la dificultad de las preguntas a las que deberán enfrentarse y competir por ser el primero en la clasificación. El segundo juego es un recorrido mediante la utilización de pistas que deberá resolver el usuario para completar un recorrido alrededor de distintos municipios de Aragón, de forma que se combina la diversión de participar en el juego, el conocimiento para resolver los acertijos y la posibilidad de aprender nueva información y visitar lugares que antes no se conocían.

Se ha conseguido realizar una aplicación con funciones atractivas y dinámicas y que puede dar muchas opciones de uso en muchos sectores (educativo, turístico, recreativo, etc.). A su vez, la sección del servidor, con carga de datos automatizada, puede servir como base a otros proyectos que aprovechen datos de municipios o puntos de interés de la Comunidad de Aragón, siendo este sistema utilizable en otros lugares.

La aplicación está disponible para ser descargada en Google Play Store en la dirección [play.google.com/store/apps/details?id=es.lcpdev.exploraragon](https://play.google.com/store/apps/details?id=es.lcpdev.exploraragon).



## Agradecimientos

Tras el duro trabajo de estos meses, resulta complicado escribir una sección de agradecimientos y espero no olvidar a nadie en ella.

En primer lugar, los mayores de mis agradecimientos y recuerdos están destinados a mis padres y a mi familia, que, sea la situación que sea, siempre se han mostrado dispuestos a apoyarme, acompañarme en esta odisea y ayudarme en lo posible. Por todo ello son los primeros en los que pienso al comenzar a escribir este breve apartado.

También merece una cita especial mi director de proyecto, Francisco Javier López, que ha tenido la paciencia de atender mis tutorías, dudas y ayudarme a darle forma a mis ideas, todo ello siempre proponiendo retos y alimentando la inquietud que me lanzó a realizar este proyecto.

No puedo olvidarme de mis amigos y de mis compañeros de carrera, muchos de los cuales han aportado ideas y observaciones que, de otra forma, habría pasado por alto y que han acompañado muchos de los momentos por los que hemos pasado en lo que fue el CPS y en lo que ahora es la EINA. De los amigos, ajenos a la ingeniería, que escuchaban con atención mis dilemas técnicos y que intentaban aconsejarme, a veces con disparates, a veces con ideas geniales, pero siempre con la mejor de sus intenciones.

Nombrar también a los compañeros y mentores de La Terminal, en especial a Sara Santolaria y a José Antonio Ruíz, quienes despertaron mis sueños emprendedores y me ayudaron a descubrir todo un mundo que hay más allá del desarrollo del software, como es la gestión de un producto, el marketing y el cliente.

Finalmente, pero no por ello menos importantes, mis compañeros de trabajo de *Hiberus Tecnología* en *Alliance Healthcare*, los que siguen allí y los que han ido dejando la compañía, quienes me han propuesto mejoras más que interesantes, me han ayudado a compaginar mi proyecto y el trabajo con un ambiente inmejorable y resultando una fuente de apoyo e ideas inigualable.

Seguro que hay muchas otras personas que me han apoyado y ayudado durante estos años y que estoy olvidando, por eso, continuar dando nombres sería complicado, así que espero que todos aquellos que haya podido olvidar sepan que siempre les agradeceré la ayuda que me han podido ofrecer.



# Índice general

<b>I</b>	<b>Memoria</b>	<b>1</b>
<b>1.</b>	<b>Introducción</b>	<b>3</b>
1.1.	Motivación y objetivos del proyecto . . . . .	3
1.2.	Contexto tecnológico . . . . .	5
1.3.	Organización de la memoria . . . . .	10
<b>2.</b>	<b>Trabajo Realizado</b>	<b>13</b>
2.1.	Análisis . . . . .	13
2.1.1.	Escenario . . . . .	13
2.1.2.	Determinación de requisitos . . . . .	15
2.1.3.	Definición de términos . . . . .	16
2.1.4.	Aplicaciones similares . . . . .	19
2.2.	Diseño . . . . .	21
2.2.1.	Arquitectura del sistema . . . . .	21
2.2.2.	Modelo de datos . . . . .	23
2.3.	Desarrollo de la aplicación . . . . .	24
2.3.1.	Metodología de desarrollo de la aplicación Android . . . . .	24
2.3.2.	Decisiones de diseño . . . . .	25
2.3.3.	Estructura de la aplicación . . . . .	28
2.3.4.	Dotación de interacción . . . . .	31
2.4.	Desarrollo del servidor . . . . .	32
2.4.1.	Base de datos . . . . .	33
2.4.2.	Servicios Web . . . . .	33
2.4.3.	Recopilación de datos . . . . .	36

2.5.	Servicios de Google APIs . . . . .	36
2.6.	Pruebas . . . . .	37
2.6.1.	Pruebas unitarias . . . . .	37
2.6.2.	Pruebas de componentes . . . . .	38
2.6.3.	Pruebas de sistema . . . . .	38
2.6.4.	Pruebas operacionales . . . . .	38
2.7.	Despliegue . . . . .	38
2.7.1.	Google Play Store . . . . .	39
2.7.2.	OpenShift . . . . .	39
<b>3.</b>	<b>Gestión del Proyecto</b>	<b>41</b>
3.1.	Planificación . . . . .	41
3.2.	Esfuerzos . . . . .	41
<b>4.</b>	<b>Conclusiones</b>	<b>45</b>
4.1.	Cumplimiento de objetivos propuestos . . . . .	45
4.2.	Trabajo futuro . . . . .	46
4.3.	Valoración personal . . . . .	47
<b>II</b>	<b>Anexos</b>	<b>51</b>
<b>A.</b>	<b>Tecnologías y herramientas utilizadas</b>	<b>53</b>
A.1.	Android . . . . .	53
A.2.	Datos abiertos . . . . .	53
A.2.1.	OpenData Aragón . . . . .	53
A.2.2.	Datos abiertos del Ayuntamiento de Zaragoza . . . . .	54
A.2.3.	Wikipedia . . . . .	54
A.3.	JBoss RestEasy - Web Service . . . . .	54
A.4.	L <sup>A</sup> T <sub>E</sub> X . . . . .	55
A.5.	Git . . . . .	55



<b>B. Historias de Usuario y Requisitos</b>	<b>57</b>
B.1. Historias de Usuario . . . . .	57
B.2. Requisitos . . . . .	63
B.2.1. Requisitos Funcionales . . . . .	63
B.2.2. Requisitos no Funcionales . . . . .	64
<b>C. Manual de usuario</b>	<b>65</b>
C.1. Inicio . . . . .	65
C.2. Juegos . . . . .	65
C.2.1. Preguntas y respuestas . . . . .	67
C.2.2. Gymkana . . . . .	67
C.3. Aprendizaje . . . . .	68
C.4. Mi cuenta . . . . .	69
C.4.1. Datos del usuario . . . . .	69
C.4.2. Amigos . . . . .	69
C.4.3. Invitaciones . . . . .	69
C.4.4. Gymkanas . . . . .	70
C.5. Menú superior . . . . .	71
<b>D. API del servidor de ExplorAragon</b>	<b>73</b>
D.1. Servicios disponibles . . . . .	73
D.2. Formato de la respuesta de error . . . . .	76



# Índice de figuras

1.1. Porcentaje de uso de lenguajes de programación [12]	6
2.1. Arquitectura del sistema	23
2.2. Modelo de clases de ExplorAragon	24
2.3. Diseño de navegación previa	26
2.4. Ejemplos de pantallas	26
2.5. Pantallas del menú de juegos. Ejemplo de uso de pestañas.	27
2.6. Pantallas de juego de Gymkana en modo inmóvil. Ejemplo de uso de los mapas.	29
2.7. Estructura y navegación de ExplorAragón	30
2.8. Gestos aplicables en ExplorAragon	32
2.9. Modelo de la base de datos (Parte 1)	34
2.10. Modelo de la base de datos (Parte 2)	35
3.1. Planificación de ExplorAragón	43
C.1. Página principal de ExplorAragón	66
C.2. Menú principal de ExplorAragón	66
C.3. Menús de juegos de ExplorAragón	67
C.4. Menú de creación o modificación de Gymkanas de ExplorAragón	70
C.5. Pantalla de información de localización de ExplorAragón	71
C.6. Pantalla de preferencias del usuario de ExplorAragón	72



# Índice de tablas

1.1. Uso de versiones de Android en smartphones . . . . .	7
3.1. Estimación de tareas . . . . .	42
B.1. Historias de usuario . . . . .	58



# Acrónimos

**API** Application Programming Interface

**CPS** Centro Politécnico Superior

**DAO** Data Access Object

**EINA** Escuela de Ingeniería y Arquitectura

**GPS** Global Positioning System

**HTTP** HyperText Transfer Protocol

**JDK** Java Development Kit

**JIT** Just In Time

**JSON** JavaScript Object Notation

**KB** KiloBytes

**PFC** Proyecto Fin de Carrera

**REST** Representational State Transfer

**SDK** Software Development Kit

**SQL** Structured Query Language

**TIC** Tecnologías de la Información y la Comunicación

**URI** Uniform Resource Identifier

**XML** eXtensible Markup Language





# Parte I

## Memoria



# Capítulo 1

## Introducción

En el presente documento se pretende resumir el proceso de realización del Proyecto Fin de Carrera titulado *ExplorAragón: Juego social sobre Aragón. Desarrollo de una app para el aprendizaje sobre municipios de Aragón*.

A lo largo del primer capítulo se tratarán las motivaciones y objetivos que han llevado a la realización del proyecto, el contexto tecnológico en el que se enmarca y sobre la estructura general en la que se organiza el presente documento.

### 1.1. Motivación y objetivos del proyecto

En la actualidad los dispositivos móviles, smartphones, y las tablets están tomando cada vez un mayor protagonismo, hasta el punto de estar cerca de alcanzar la venta de nuevos ordenadores. Todo ello hace que las empresas que centran su actividad en el sector de las tecnologías de la información y la comunicación apuesten cada vez más por la creación de equipos de desarrollo de aplicaciones para plataformas móviles, utilizando diferentes lenguajes, desde los nativos como Android o tecnologías como PhoneGap que permiten utilizar JavaScript para desarrollar aplicaciones mixtas.

Descargarse una aplicación ya es, entre los usuarios, algo cotidiano y se hace en casos como para controlar una cuenta corriente, para pedir comida a domicilio, para mantenerse en contacto con otras personas, para buscar trabajo o simplemente como entretenimiento. Existen una infinidad de aplicaciones con diferentes fines y concepciones. Y parece que el mercado de las aplicaciones continúa en auge por el, cada vez más, sencillo acceso, lo que supone una gran oportunidad.

Sin embargo, existen sectores en los que no se han desarrollado un gran número de aplicaciones. En concreto se trata de los sectores educativo y del turístico. La mayoría de las aplicaciones relacionadas con el turismo se limitan a un listado de puntos de interés de distintos tipos (restaurantes, atracciones turísticas, museos, etc.), en

ocasiones puntuadas por otros usuarios o con comentarios de estos. Sin embargo, no permite al usuario interactuar con esta información, resultando, mayormente, un método de consulta de información.

Algo similar ocurre con las aplicaciones relacionadas con el aprendizaje. Únicamente las aplicaciones en las que se incluyen mecanismos de preguntas y respuestas son las que permiten interactuar al jugador con la información. Sin embargo no se exploran métodos que permitan al usuario comprobar o recorrer sus averiguaciones o consultar la información.

Con la existencia de diversas fuentes de datos abiertos, cada vez es más sencillo acceder a una mayor cantidad de información. Desde OpenData Aragón se puede recopilar información general sobre la comunidad de Aragón, y el propio ayuntamiento de la ciudad de Zaragoza ofrece una sección de datos abiertos a los usuarios y a los desarrolladores para su consulta. Todo esto hace que el acceso a esa información sea cada vez más fácil de manipular y más sencillo de utilizar de formas nuevas y más atractivas para el usuario.

El objetivo de la aplicación ExplorAragón es ayudar a turistas y residentes en Aragón a conocer más sobre los municipios de esta comunidad autónoma y de incentivar la curiosidad por los diferentes lugares de importancia en los diferentes municipios. Todo ello rodeado por un entorno social que permita al usuario relacionarse con otras personas que utilicen la aplicación y generando una pequeña competición de los usuarios, convirtiéndolos en jugadores, para demostrar los conocimientos que adquieren o poseen sobre Aragón. Para ello se desarrollarán modos de juego atractivos e interactivos, que le permitan al usuario, además de dar respuesta a preguntas, participar en otro tipo de juegos que le lleven a diferentes lugares y que le permitan conocer cada vez más.

Para alcanzar estos objetivos se han fijado las siguientes tareas:

- La realización de un análisis previo de la aplicación. Estudiando la idea general que se desea alcanzar y los requisitos que se plantean cumplir para tener clara la funcionalidad de la aplicación.
- Estudio de las aplicaciones móviles con funcionalidades similares a las expuestas, estudiando la usabilidad de cada una de ellas para buscar una experiencia cada vez más sencilla y agradable para el usuario.
- Realización de un diseño inicial de la aplicación en el que se incluirían los casos de uso posibles del usuario y un esquema de navegación entre los diferentes apartados de la aplicación.
- Deliberación sobre el mecanismo de envío de datos entre el servidor y modelo MVC a utilizar, así como el modelado de datos para asegurar que la información requerida se envía correctamente entre el cliente y el servidor.

- Estudiar y conocer los métodos de recopilación de datos de los diferentes centros de información abierta utilizados. Así como de la forma de recopilar automáticamente esta información y almacenarla en la base de datos.
- Creación de los servicios web que responderán a las distintas peticiones.

## 1.2. Contexto tecnológico

Antes de entrar en detalle sobre el trabajo realizado y para comprender mejor su desarrollo, es conveniente describir brevemente las tecnologías utilizadas para el desarrollo y las herramientas utilizadas para llevar a cabo dicha tarea.

### Java

El lenguaje de programación Java [6] es uno de los más extendidos y utilizados entre los desarrolladores de aplicaciones (Figura 1.1). Pese a la disminución de su uso desde el año 2000, Java continúa siendo el lenguaje más utilizado a día de hoy al realizar nuevas aplicaciones. La posibilidad de desarrollar programas multiplataforma, que utilizan la máquina virtual de Java para ejecutarse, y la amplia documentación y librerías existentes hacen que este lenguaje continúe siendo muy atractivo para el desarrollo de software.

A principios de los 90, James Gosling, Mike Sheridan y Patrick Naughton dieron inicio a *Oak* (nombre que se le dio inicialmente al proyecto que más tarde se convertiría en Java) como lenguaje de programación. Su idea era desarrollarlo para diseñar una interfaz para la televisión, pero la industria de la televisión por cable no estaba lo suficientemente avanzada. Más tarde, el proyecto pasó a llamarse *Green* y, finalmente, *Java*. Sun Microsystems lanzó la primera versión del lenguaje de programación en el año 1995. En la actualidad, Java forma parte de la compañía Oracle, que ha anunciado el próximo lanzamiento del JDK 9 en el 2017.

Java es un lenguaje de programación de propósito general, por lo que permite su utilización para satisfacer diferentes usos, orientado a objetos y concurrente, lo que permite que se ejecuten varias tareas al mismo tiempo. Posee una sintaxis similar a la de los lenguajes C/C++ y se ejecuta sobre una máquina virtual de mismo nombre.

### Android

El sistema operativo Android [1] es más habitual en smartphones y tablets. La compañía Android apareció en el año 2003 y la primera intención de sus fundadores era crear un nuevo sistema operativo para cámaras fotográficas, sin embargo,

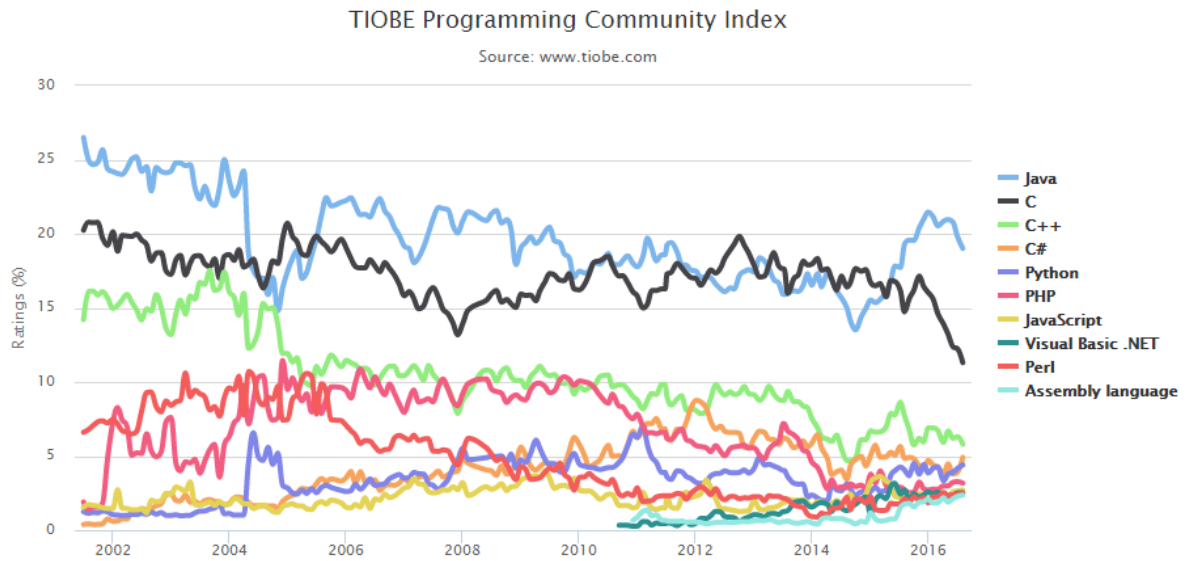


Figura 1.1: Porcentaje de uso de lenguajes de programación [12]

como el mercado no era lo suficientemente extenso, la compañía dirigió sus esfuerzos en el desarrollo de un sistema operativo para teléfonos móviles. En el año 2005, Google compra la compañía y en 2007 encabeza el consorcio tecnológico *Open Handset Alliance* cuya meta es conseguir estándares abiertos para dispositivos móviles. Android ve la luz, por primera vez en un dispositivo móvil en Octubre del año 2008.

El sistema operativo se basa en una distribución linux que, sobre ella, tiene una lógica de intercambio de información (middleware), librerías y APIs escritas en C y un marco de aplicaciones que incluye librerías compiladas en Java.

Hasta la versión 5.0, Android utiliza Dalvik, una máquina virtual de procesos basada en trazas de ejecución con compilación JIT.

Una de las principales ventajas de Android es la posibilidad de ejecutar aplicaciones desarrolladas para versiones anteriores. Mediante la utilización de librerías de compatibilidad, también es posible desarrollar aplicaciones utilizables por versiones anteriores a la actual. De este modo resulta relativamente sencillo el desarrollo de aplicaciones alcanzando el máximo de dispositivos posibles. La versión con la que se inició el desarrollo del proyecto fue la versión 5.0 (Lollipop), sin embargo, como se ha dicho, gracias a la utilización de ciertas librerías, sería posible ejecutar la aplicación a partir de la versión 4.1 (Jelly Bean). Esto significa, que la aplicación sería ejecutable en más del 95 % de los dispositivos en activo en la actualidad, como se puede ver en la siguiente tabla.

Tabla 1.1: Uso de versiones de Android en smartphones

Codename	Versión	Año de liberación	Distribución
Froyo	2.2 - 2.2.3	0.1	2010
Gingerbread	2.3.3 - 2.3.7	1.7	2010
Ice Cream Sandwich	4.0.3 - 4.0.4	1.6	2011
Jelly Bean	4.1.x	6.0	2012
	4.2.x	8.3	
	4.3	2.4	
KitKat	4.4 - 4.4.4	29.2	2013
Lollipop	5.0	14.1	2014
	5.1	21.4	
Marshmallow	6.0	15.2	2015
Nougat	7.0	N/D	2016

## SQLite

La base de datos relacionales SQLite [13] es la utilizada en los dispositivos Android. SQLite se trata de una librería de C que se caracteriza por ser relativamente pequeña, unos 275 KB.

A diferencia de otras bases de datos, basadas en gestión de tipo cliente-servidor, SQLite se integra en las aplicaciones, esto hace que la latencia en el acceso a la base de datos se reduzca considerablemente, ya que las llamadas terminan siendo más eficientes. El conjunto de las bases de datos se almacenan en una serie de ficheros en el dispositivo.

## Servicios web RESTful

REST [14] (Representational State Transfer) es, en origen, un estilo de arquitectura software para sistemas hipermedia distribuidos como en la World Wide Web. De REST se deriva el término RESTful, que se utiliza para describir cualquier interfaz que utilice HTTP para la obtención de datos en diversos formatos.

Se caracteriza por varios elementos clave. El primero de ellos es que se trata de un protocolo cliente/servidor sin estado, esto es, que cada mensaje contiene toda la información necesaria para completar la petición. Otra de las principales características es que se trata de un conjunto de operaciones bien definidas que se aplican a todos los recursos de información. Posee una sintaxis universal, esto es, cada recurso es direccionable únicamente a través de su URI (Uniform Resource Identifier). La

última de las características es la representación de estados HTML o XML, esto hace que sea posible navegar de un recurso REST a otro.

Los sistemas RESTful normalmente, aunque no siempre, se comunican utilizando el protocolo HTTP, utilizando los mismos verbos como instrucciones (GET, PUT, POST, DELETE, ...) que las páginas web para enviar y recibir datos desde un servidor remoto. Las interfaces de un servicio REST son identificadas por URIs.

Estos sistemas se caracterizan por:

- **Una dirección URL base**, a la cual se añadirán las URIs.
- **Un tipo de datos**. Esto es la forma en la que se van a representar los datos a través e internet. En nuestro caso se trata de JSON, que será descrito a continuación.
- **Los métodos estándar de HTTP**.

Los servicios web desarrollados se desplegaron en un servidor de tipo JBoss, ya que integra la implementación de estos servicios en el framework RestEasy [10]. La principal ventaja de este framework es que es fácilmente integrable con otros como Spring, Guice, etc., lo que hace que resulte sencilla la implementación de un modelo vista controlador.

## JSON

Uno de los principales estándares de transmisión de información se trata de JSON (JavaScript Object Notation) [3]. JSON es un formato de texto para serializar datos estructurados. La estructura del estándar se deriva de la notación de los literales en los objetos de JavaScript [4].

Este estándar será el utilizado para la transmisión de la información de los servicios web desarrollados. Una de las ventajas más importantes de este estándar es lo ligero que resulta el intercambio de datos y, debido a su popularidad, parece que esta resultando la alternativa al lenguaje XML cuando el ancho de banda o coste de procesamiento son limitados. A diferencia de XML, en JSON resulta más sencillo escribir un analizador sintáctico para crear los objetos enviados a través de este formato.

## Web scraping

A la extracción de datos de sitios web mediante programas software se le llama web scraping [17]. Estas técnicas son capaces de navegar a través de diversas páginas web, extrayendo la información que se precisa de cada una de ellas y almacenando estos datos en una base de datos. En la actualidad, la utilización de scrapers esta



sujeta a cierta controversia en algunos casos, ya que podría vulnerar la legalidad en función de la información que se extraiga de las diversas páginas. Por ello, existen diversos mecanismos que permiten a las páginas protegerse contra estos procesos o, incluso, filtrar aquellos que están autorizados a extraer los datos de las páginas. Sin embargo, el web scraping resulta una buena técnica para la recopilación de datos de temas muy diversos o de datos que se han descrito como datos abiertos.

## Google APIs

Desde hace unos años, Google ofrece una gran cantidad de APIs [16] que permiten acceso a sus diversos servicios. Esto facilita el acceso a gran cantidad de información para su uso tanto en las aplicaciones como en servidores o páginas web. El catálogo de APIs disponibles es muy extenso<sup>1</sup> y entre ellas destacan servicios de mapas, documentos de Google Drive, vídeos de Youtube o servicios relacionados con la publicidad o gestión de registro con cuentas de Google. Estas APIs también permiten el acceso a otras funcionalidades, como el análisis de uso de una aplicación o página web, lo que puede permitir analizar el comportamiento del usuario y ofrecerle mejoras más personalizadas.

Las librerías para el uso de las APIs de Google están disponibles en varios lenguajes, entre los que, por supuesto, se encuentra Java. Sin embargo, también es posible utilizarlo con Objective-C y JavaScript, lo que permite que sean utilizadas por más dispositivos móviles, y con otros lenguajes como PHP, Python o el framework .NET. Para su utilización únicamente es necesaria una cuenta de Google, sin embargo los servicios pueden ser gratuitos o tener una cuota de uso gratuita con planes de pago.

Google también facilita un sistema de scripts en la nube, que permite a los desarrolladores escribir código con el que manipular sus APIs y crear fácilmente extensiones para aplicaciones basadas en *chromium*.

## Git

Uno de los sistemas utilizados para el control de versiones en el desarrollo de software es Git [15]. Al tratarse de un control de versiones distribuido, esta orientado a ofrecer velocidad, integridad de los datos y soporte. Gracias a ello se puede almacenar el código en un servidor remoto, manteniendo el control de la versión sobre la que se está trabajando y dando la posibilidad de recuperar versiones anteriores de diferentes archivos, así como consultar el historial de los cambios realizados.

En el ámbito de las empresas tecnológicas, este tipo de control de versiones a través de Git<sup>2</sup> se está extendiendo cada vez más, ya que también permite la creación de diferentes ramas de trabajo en las que se pueden almacenar los estados

---

<sup>1</sup><https://developers.google.com/apis-explorer/>

<sup>2</sup><https://git-scm.com/>

del desarrollo y crear varios equipos de trabajo que se centren cada uno en un objetivo sin que las modificaciones de su código se entremezcle.

Git permite establecer un etiquetado de versiones, de forma que el usuario puede acceder a una versión concreta del software con sencillos comandos.

Resulta habitual añadir la metodología GitFlow [2], que permite, mediante una serie de extensiones, utilizar operaciones de alto nivel para ejercer el control sobre el repositorio y las ramas del mismo.

En el presente proyecto se utilizó GitHub<sup>3</sup>, una plataforma muy extendida entre desarrolladores de software de código abierto, para el control de versiones de la aplicación Android; y la plataforma OpenShift [9] para el servidor. Para la gestión de los comandos de Git se utilizó la herramienta SourceTree.

### 1.3. Organización de la memoria

Este documento se divide en diferentes partes. En primer lugar, la memoria, una descripción breve y de bajo nivel técnico acerca de la realización del presente PFC. A continuación se podrán consultar una serie de anexos y apéndices con datos mayoritariamente técnicos.

La memoria, como se ha dicho, da una visión global del proyecto sin entrar en detalles técnicos y de forma que la mayoría de las personas sean capaces de entender el alcance y los objetivos que se han perseguido y los que se han alcanzado y cómo esto se ha llevado a cabo.

El primero de los capítulos de esta parte pretende presentar e introducir la aplicación en la que se ha trabajado y su finalidad. En el segundo capítulo se expone el trabajo realizado, detallando cada uno de los principales procesos del desarrollo de software; análisis, diseño, desarrollo, evaluación de los resultados, pruebas y despliegue de la aplicación y del servidor. En el siguiente capítulo, se exponen las decisiones sobre la gestión del proyecto, la planificación y las dificultades surgidas relacionadas. Finalmente se exponen las conclusiones a las que ha conducido la realización del PFC y las posibles líneas de trabajo futuro que se podrían abarcar.

Después de presentar la memoria, se encontrarán una serie de anexos, en los que el lector podrá consultar datos de mayor profundidad técnica si considera necesario ampliar los detalles propuestos y ofrecidos a lo largo de la memoria.

El primero de los anexos se centra en la explicación más detallada de las tecnologías que se han utilizado en la elaboración del proyecto, así como de las herramientas que se han podido utilizar, como entornos de programación para el desarrollo, procesadores de texto para la elaboración de documentos, etc.

---

<sup>3</sup><https://github.com/>

En el siguiente anexo se listarán las historias de usuario que se analizaron y los requisitos funcionales y no funcionales que derivan de estas y se han tenido en cuenta para la elaboración del PFC.

En el tercer anexo se explicará mediante un manual el uso de la aplicación, explicando los diferentes menús y los modos de utilización.

Finalmente, en el último de los anexos, se describirán brevemente los servicios desarrollados y ofrecidos desde el servidor.



# Capítulo 2

## Trabajo Realizado

En este apartado se resume el trabajo realizado durante la elaboración del proyecto fin de carrera, tanto en el apartado de la aplicación Android como en el servidor.

### 2.1. Análisis

En esta primera sección del trabajo realizado se resumirá la labor de análisis de la aplicación, comenzando por una breve descripción del escenario que se ha tenido en cuenta para la elaboración de la aplicación Android. De este escenario, se deducen los requisitos que se desean alcanzar. Seguidamente se analizará el mercado en búsqueda de aplicaciones con funcionalidades similares a la realizada. Finalmente se definirán algunos términos importantes a tener en cuenta en el análisis de la aplicación.

#### 2.1.1. Escenario

En primer lugar, para la realización del proyecto se realizó un estudio sobre los diferentes escenarios en los que un usuario podría encontrarse al utilizar la aplicación. De esta forma se describen una serie de historias de usuario (ver apéndice B.1) y se pueden obtener los requisitos del sistema a desarrollar.

Gracias al análisis de los escenarios, se determinó la existencia de la siguiente posibilidad de actores y de la funcionalidad que esperaban de una plataforma como la que se ha pretendido desarrollar:

#### **Jugador social**

El primero de los escenarios tenía como objetivo satisfacer a un jugador social o competitivo. Esto es, un jugador que compite contra otros, ya sean sus amigos o personas desconocidas, y cuya meta es conseguir la máxima puntuación posible.

En la plataforma a desarrollar, este jugador pretenderá poder comparar su puntuación con la de los amigos que tenga o con la de los demás jugadores y para mejorarla jugará tanto al juego de preguntas y respuestas (ver 2.1.3) como participará en gymkanas (ver 2.1.3), hasta superar la puntuación de sus competidores. La forma más sencilla de facilitarle el acceso a este tipo de juegos es a través del teléfono móvil.

### **Jugador ocasional**

El jugador ocasional es aquel que juega por puro entretenimiento, sin que superar a otros jugadores sea su meta principal. Actualmente es más sencillo tener acceso a un teléfono móvil o a una tablet, por lo que una aplicación que le permita jugar durante unos minutos sería la decisión más habitual de un jugador de estas características. Los juegos más habituales para el actor de este escenario serían aquellos de rápida ejecución, que no requieran más de unos minutos y que pueda realizar desde dónde se encuentre. Los juegos de preguntas y respuestas son todo un clásico y siempre resultan atractivos para el usuario final.

### **Grupos de jugadores**

En el siguiente escenario, entran en juego varios actores, ya que se trataría de un grupo de amigos que quieren realizar alguna actividad en conjunto. Para ello, la plataforma a desarrollar debería poder facilitarles el acceso a alguna actividad que puedan realizar entre ellos, sin involucrar a otras personas. Normalmente, en estos casos, siempre entra en juego un pequeño factor competitivo.

### **Turista**

Otro escenario tenía como objetivo a un turista, cuya meta era recorrer un municipio a través de una serie de pistas que le facilitasen conocer nuevas localizaciones o consultando los datos de ese municipio.

### **Cultural**

Finalmente, el último de los actores a tener en cuenta, se trata de aquel que únicamente trata de explorar y aumentar su nivel de cultura. Para ello debería disponer del acceso a la información que desea adquirir, así como de un método para poder comprobar que sus conocimientos son correctos. Un juego de preguntas y respuestas o una gymkana pueden resultar dos métodos adecuados para ello.

## 2.1.2. Determinación de requisitos

El siguiente paso a realizar es la determinación de los requisitos que debe cumplir el sistema a realizar. Seguidamente se definirán los requisitos de la aplicación móvil sin entrar en detalle técnico, lo que facilitará tener una visión global de las acciones fundamentales que deben desarrollarse. Para un mayor detalle de estos requisitos, se recomienda al lector que se dirija la sección de anexos, al apéndice B.2.

### Requisitos funcionales

Los requisitos funcionales son aquellos que determinan el funcionamiento que debe tener la aplicación, indicando como se recibe, procesa y muestra la información.

- **Gestión de usuarios.** Deberá ser posible crear y editar usuarios a través de la aplicación móvil. Al crear un usuario se requerirán algunos datos para garantizar su acceso a la aplicación.
- **Juego de preguntas.** Se implementará una serie de métodos y funciones que permitirán al usuario acceder a un juego de preguntas sobre la comunidad de Aragón, como se describirá en la sección 2.1.3. También se dará la opción al usuario para consultar la clasificación de los mejores jugadores y de los mejores amigos del usuario.
- **Juego de gymkanas.** Se crearán funciones que permitirá al usuario participar en partidas de tipo gymkana. Estas se han descrito en la sección 2.1.3. El usuario podrá elegir entre tres tipos de modos de juego, así como la gymkana que desea jugar.
- **Gestión de gymkanas.** Un usuario será capaz de gestionar sus propias gymkanas. Se desarrollarán métodos que permitirán la creación, edición y el borrado de gymkanas creadas por diferentes usuarios. Será posible introducir localizaciones personalizadas, añadir pistas, seleccionar el número de localizaciones de la gymkana y el nombre. También será posible elegir si se desea hacer pública la gymkana para cualquier usuario o si solo será accesible por sus amigos.
- **Búsqueda de información de municipios.** Será posible consultar información sobre municipios introduciendo el nombre del mismo o mediante la geolocalización del usuario. Al usuario se le informará sobre el nombre, provincia, comarca, población y otros datos geopolíticos de interés sobre el municipio.
- **Búsqueda de información de usuarios.** Se implementarán métodos para consultar información de diferentes usuarios, así como su mejor puntuación en juegos de tipo trivia, el número de gymkanas completadas y el número de carreras ganadas.

- **Gestión de invitaciones.** Será posible realizar invitaciones a otros usuarios para establecer entre ellos una relación de amistad. Una vez establecida la amistad, también se podrán realizar invitaciones para participar en la carrera generada por el usuario. Las invitaciones podrán ser aceptadas o rechazadas.
- **Geoposicionamiento del usuario.** Se podrá geolocalizar al usuario, para la información sobre municipios o para localizar las posiciones en los juegos de tipo gymkana.
- **Carga de información de sistemas de terceros.** Se cargarán datos desde diferentes fuentes de datos abiertos o desde diferentes páginas con información sobre municipios, así como direcciones de imágenes y artículos de Wikipedia.

## Requisitos no funcionales

Los requisitos no funcionales son todos aquellos que, pese a no estar directamente relacionados con el tratamiento de la información del sistema, sí que determinan características como el rendimiento, la seguridad y otros aspectos que no están relacionados con el funcionamiento de la aplicación:

- Creación de una aplicación móvil para la plataforma Android, desarrollada en lenguaje nativo.
- Conexión con servicios web mediante protocolo RESTful y utilización de mensajes con formato JSON
- Uso de sistemas de datos abiertos como OpenData Aragón<sup>1</sup> y datos abiertos del ayuntamiento de Zaragoza<sup>2</sup>

### 2.1.3. Definición de términos

Durante la fase de análisis se procedió a la definición de una serie de términos que serán utilizados para describir algunas partes y el funcionamiento de la aplicación. A continuación se listan estos términos y se detallan para facilitar su entendimiento.

#### Trivia

Con trivia nos referimos a uno de los tipos de juego que se pueden realizar en la aplicación. En el caso de trivia se refiere a la realización de preguntas con la posibilidad de elegir la respuesta correcta entre otras que no lo son. Al responder

---

<sup>1</sup><http://opendata.aragon.es/>

<sup>2</sup><https://www.zaragoza.es/ciudad/risp/>



correctamente a una de las preguntas se acumulan puntos. Una vez respondidas diez preguntas, termina la partida de tipo trivia y se muestra la puntuación final.

Al iniciar una partida de tipo trivia, al usuario se le mostrará una pregunta, que puede ser de diversos tipos, y entre tres y cuatro posibles respuestas, de las cuales, únicamente una es correcta. En el momento en el que se muestra la pregunta, comienza una cuenta atrás de 15 segundos, tiempo en el que el jugador debe dar una respuesta.

Tras responder a la pregunta, si se da la respuesta correcta, la puntuación obtenida se calcula a partir del tiempo restante. Así, si al jugador le quedan 10 segundos o más, se le otorgan 1000 puntos (la puntuación máxima), a partir de ese momento, se multiplica el tiempo restante por 100 y así se obtiene la puntuación. Tanto si la respuesta es incorrecta, como si se acaba el tiempo, el jugador no obtendrá ningún punto.

Esta puntuación se va acumulando una pregunta tras otra, hasta completar una serie de diez preguntas, cuando termina la partida y se muestra la puntuación total.

Los tipos de preguntas son:

- **Provincia.** Se pregunta al usuario sobre la provincia a la que pertenece un municipio o el municipio que se encuentra en la provincia indicada.
- **Comarca.** Se pregunta al usuario sobre la comarca a la que pertenece un municipio o el municipio que se encuentra en la comarca indicada.
- **Población.** Se pregunta al usuario sobre la población de un municipio o el municipio que posee el número aproximado de habitantes indicados.
- **Imagen.** Se mostrará una imagen y se deberá elegir el municipio identificada con dicha imagen o se mostrarán cuatro imágenes y se pedirá que se indique la que identifica un municipio.

También se dará la posibilidad al usuario de elegir entre tres niveles de dificultad, fácil, medio y difícil. La dificultad fácil únicamente realizará preguntas sobre municipios con más de 500 habitantes y de tipo provincia y comarca. La dificultad media realizará cualquier tipo de pregunta sobre municipios con más de 100 habitantes. Por último, en el modo difícil, se podrán encontrar preguntas de cualquier tipo y acerca de cualquier municipio de Aragón.

Las puntuaciones se guardarán en función de la dificultad de la partida y será posible competir y consultarlas en una clasificación de usuarios o entre los amigos del usuario.

## **Gymkana**

Las gymkanas son el segundo modo de juego que permite la aplicación Android. Se define una gymkana como un recorrido siguiendo puntos de interés a través de la resolución de diferentes pistas. La comprobación de la solución de las pistas se realiza mediante la geolocalización del terminal móvil.

Al iniciar una partida de tipo Gymkana, se muestra al usuario una pista y un botón que le permitirá comprobar si su posición se corresponde, aproximadamente, con la posición que se desea encontrar. Si pasado cierto tiempo no se ha podido encontrar la localización correcta, se mostrará una segunda pista y, más tarde, una tercera y definitiva, que deberán llevar al usuario, a través de un reto de conocimiento, a diferentes puntos de interés o distintivos de un municipio. Como respuesta, el usuario debe desplazarse hasta las proximidades de la localización o punto que cree correcto e indicar a la aplicación que compruebe su localización. Esta estimará si la posición es la correcta.

Al contrario del modo trivia, en el modo gymkana no existe una puntuación de usuario, aunque si se contabilizan el número de gymkanas realizadas y completadas por cada usuario.

Las gymkanas podrán ser creadas tanto por los datos que posee el sistema, como por localizaciones y pistas introducidas por diferentes usuarios. Las gymkanas generadas por el sistema tienen la peculiaridad de dar localizaciones aleatorias y poder seguir diferentes órdenes de resolución de pistas.

Existen diferentes tipos de juegos de Gymkana. El modo normal es el descrito previamente. El modo inmóvil permite al jugador realizar una gymkana sin tener que desplazarse, indicando la posición que considere como respuesta en un mapa, sin embargo, este modo de juego no se contabilizará al tener en cuenta las gymkanas completadas. Finalmente tenemos el modo carrera, que será descrito a continuación.

## **Carrera**

La carrera es un modo de juego de tipo Gymkana. Este modo tiene de particular la posibilidad de participar en grupo y competir contra otros jugadores.

La metodología de juego del modo carrera es exactamente la misma que la Gymkana. Sin embargo, al poder participar al mismo tiempo varios equipos, estos pueden compartir entre sí. Un jugador deberá ser el anfitrión de la carrera y no podrá participar en ella, excepto si selecciona una de las gymkanas generadas automáticamente por el sistema. El anfitrión invitará a una serie de usuarios de su lista de amigos (se hablará de las invitaciones más adelante) y, cuando crea que hay suficientes participantes, dará inicio a la carrera.

Los participantes recibirán en ese momento las localizaciones y deberán resolverlas como se ha indicado anteriormente. En cuanto el primer usuario haya conseguido

encontrar todas las localizaciones conforme vayan apareciendo, se dará la partida por finalizada y este usuario será considerado como ganador. El número de carreras ganadas también se acumularán y se mostrarán en el perfil del usuario.

## **Invitación**

Una invitación es un modo de interactuar entre dos usuarios. Existen dos tipos de invitaciones: invitación de amistad e invitación de carrera.

A través de una invitación de amistad, un usuario puede pedirle a otro que se convierta en su amigo. De esta forma podrán verse mutuamente en la clasificación de amigos de juegos Gymkana, en el listado de amigos e invitarse a participar en una carrera.

Una invitación a una carrera únicamente podrá realizarse entre dos usuarios que sean amigos previamente. Al invitarse a una carrera, el usuario anfitrión invita a otro usuario a competir con otros de sus amigos para resolver una serie de pistas en una gymkana, como se ha descrito anteriormente.

### **2.1.4. Aplicaciones similares**

Durante el proceso de análisis de la aplicación se tuvo en cuenta las aplicaciones que ya habían sido desarrolladas y guardan ciertas similitudes con los requisitos fijados [11]. A continuación se expone un breve estudio de aplicaciones similares.

## **TouristEye**

TouristEye<sup>3</sup> es una de las aplicaciones turísticas más completas. Esta aplicación permite al usuario planificar rutas turísticas, organizar sus viajes e incluso leer sugerencias que realiza la aplicación. Es posible acceder tanto a través de una aplicación para smartphones o tablets como a través del portal web del que dispone. La principal ventaja que ofrece TouristEye es la cantidad de información disponible y la sencillez de su interfaz. Sin embargo, resulta complejo insertar sistemas por parte del usuario.

## **Foursquare**

Foursquare<sup>4</sup> es la aplicación completamente opuesta a la anterior. En esta aplicación son los usuarios los que añaden fotografías y comentarios sobre lugares de

---

<sup>3</sup><http://www.touristeye.es/>

<sup>4</sup><https://es.foursquare.com/about>

diferentes localidades. También como la anterior, dispone de un portal web y aplicaciones para smartphones y tablets con diferentes sistemas operativos. El principal punto fuerte de Foursquare es su aspecto social, ya que está orientada hacia la relación entre los usuarios y el intercambio de información. En contra, el sistema no ofrece rutas y se limita al intercambio anteriormente descrito.

## **TripAdvisor**

TripAdvisor<sup>5</sup> es una de las principales aplicaciones utilizadas para el ocio y en el sector turístico. Esta aplicación permite seleccionar entre una amplia variedad de hoteles, restaurantes, vuelos, destinos o puntos de interés con comentarios y puntuación de los demás usuarios. Dispone de una aplicación para smartphones o tablets y un portal web para realizar las consultas. Una de las principales ventajas es la gran cantidad de lugares de los que dispone y el número de usuarios activos que puntúan y comentan los diferentes lugares. En contra tiene que no da la posibilidad de añadir lugares a los usuarios ni sugiere rutas para seguir.

## **Ingress**

Ingress<sup>6</sup> es una aplicación de geoposicionamiento que consiste en dos equipos rivales que debe ocupar lo que llaman portales geoposicionados con la aplicación para móviles. La geolocalización del usuario es lo más característico de esta aplicación y lo más similar con ExplorAragon. Una cantidad importante de usuarios participa en este juego comunitario. La localización de los portales fue introducida en los primeros momentos de la aplicación por los usuarios, sin embargo, en la actualidad, ya no es posible crear nuevos portales y el usuario debe encontrar aleatoriamente los portales, en el sistema desarrollado se propone encontrar las localizaciones a través de pistas.

## **Atríviate**

Atríviate<sup>7</sup>, o Preguntados, como se llamaba anteriormente, es una aplicación para móviles o tablets que recrea el juego más famoso de preguntas y respuestas, tal y como es el Trivial Pursuit. Dispone de preguntas de diversas temáticas y permite jugar contra diferentes usuarios, estableciendo clasificaciones y niveles, de forma similar a lo que se quiere desarrollar en la aplicación ExplorAragon centrándose en el aspecto geopolítico de la comunidad de Aragón. En Atríviate el usuario puede añadir preguntas al sistema e, incluso, rebatir las que ya han sido introducidas. La principal desventaja es que no dispone de un sistema para consultar información

---

<sup>5</sup><https://www.tripadvisor.es/>

<sup>6</sup><https://www.ingress.com/>

<sup>7</sup><http://www.atriviate.com/>

relacionada con las preguntas, lo que obliga al usuario a cambiar de aplicación y consultar a través de un buscador web.

## **Otras aplicaciones**

Existen otras aplicaciones con similitudes con ExplorAragon, de estas, la más reseñable sería Google Maps, que permite analizar tiempo necesario para ir de un lugar a otro, ver reseñas sobre lugares, localizaciones cercanas, puntos de interés, así como otros servicios como hoteles, restaurantes, farmacias, etc. Además dispone de un buscador que permite al usuario localizar rápidamente un tipo de servicio o un lugar en concreto.

Otras aplicaciones como AroundMe, que ofrece información sobre los lugares cercanos al lugar dónde se encuentra el usuario; o Nuok, que también ofrece información similar, se encuentran en un mercado diferente, ya que la primera está disponible exclusivamente para sistemas iOS (iPhone o iPad) y la segunda sólo está disponible en idioma italiano.

## **2.2. Diseño**

En esta sección, centrada en el diseño tenido en cuenta para la realización del sistema, se comenzará dando una visión general de la arquitectura del mismo y de la comunicación entre cada una de las partes, lo que permitirá conocer el modelo de datos del sistema.

### **2.2.1. Arquitectura del sistema**

En el presente apartado se informará sobre la arquitectura del sistema y las tecnologías utilizadas en cada una de las partes que integran dicho sistema. La arquitectura se basa en los sistemas Cliente-Servidor.

El lado del cliente es la propia aplicación Android. Esto incluye tanto el código de la aplicación como los recursos utilizados (configuraciones, imágenes, sonidos, etc.). También formaría parte del cliente otras aplicaciones que podrían utilizarse y ya estuviesen instaladas en el dispositivo, tales como la galería de imágenes, (Gallery), los mapas de Google (Google Maps) o la cámara del dispositivo.

La aplicación se puede nutrir de datos internos o externos. Los internos son aquellos que pueden encontrarse en los diferentes almacenamientos de memoria del dispositivos, como la tarjeta SD o la memoria interna. También se considera una fuente de datos interna los datos procedentes de la base de datos SQLite. Se consideran datos externos a todos aquellos que provengan de fuentes de información ajenas al dispositivo, normalmente a través de la comunicación con los servicios web

del servidor desarrollado. La comunicación con las fuentes de datos externas se realizará a través de los mecanismos de acceso a internet de los dispositivos, la red de datos 4G o la conexión a una red Wi-Fi.

Tanto la aplicación como los datos utilizados en ella se pueden comunicar directamente o a través del *Application Framework*. Esto es, las herramientas del sistema operativo disponibles para el desarrollo de aplicaciones. En cierto modo, se trata de la capa que se encarga de comunicar la aplicación móvil con el exterior. Algunas de las herramientas disponibles en el *Application Framework* son las siguientes:

- **View System** (o Gestor de vistas). El sistema de vistas se utiliza para generar y describir los objetos que van a constituir la interfaz del usuario, esto es, los botones, textos, listas o elementos como fragmentos de vistas para Google Maps. El gestor de vistas permite implementar las vistas para dispositivos de diferentes tamaños o en orientación vertical u horizontal, añadiendo sufijos al archivo en el que se describe la vista.
- **Activity Manager** (o Gestor de actividades). Cada una de las vistas del usuario suelen ser, en su mayoría, una actividad diferente. El gestor de actividades se encarga de administrar la pila de actividades y su ciclo de vida. Dicho gestor posee un comportamiento por defecto que permite pasar de una actividad a otra, almacenando las actividades padre en la pila, de forma que sea posible retroceder a la actividad anterior. Este funcionamiento se puede modificar a través de diferentes comandos de programación que permitirán pasar a actividades que no sean la actividad padre.
- **Resources Manager** (o Gestor de recursos). El gestor de recursos administrará todos aquellos elementos utilizados ajenos al código de programación. Esto pueden ser, cadenas de texto, imágenes, iconos o sonidos. El gestor de contenidos de Android aporta una funcionalidad que permite al desarrollador añadir textos en diferentes idiomas, únicamente añadiendo un sufijo al archivo que permita diferenciar entre los diferentes idiomas.
- **Location Manager** (o Gestor de localización). El gestor de localización permite al dispositivo localizar al usuario mediante geoposicionamiento a través del GPS o analizando las redes disponibles. Esto permite, a su vez, trabajar con los mapas o, en el caso de ExplorAragon, permitirá comprobar si la posición del usuario es la correcta en una de las Gymkanas.

En el lado del servidor, el sistema se encargará de tres tareas; en primer lugar, el servidor recuperará la información de las diferentes fuentes de datos abiertos disponibles; otra de las funciones sería el almacenamiento de los datos recuperados en la base de datos orientada a dicho propósito; para terminar se podrá disponer de dicha información a través de los servicios web. Para desarrollar el servidor, se

ha utilizado la plataforma OpenShift<sup>8</sup>. Dicha plataforma ofrece una base de datos MySQL y un servidor JBoss, en el que se integrarán los servicios web mediante la utilización de librerías RestEasy. El gráfico de la figura 2.1 representa la arquitectura anteriormente descrita.

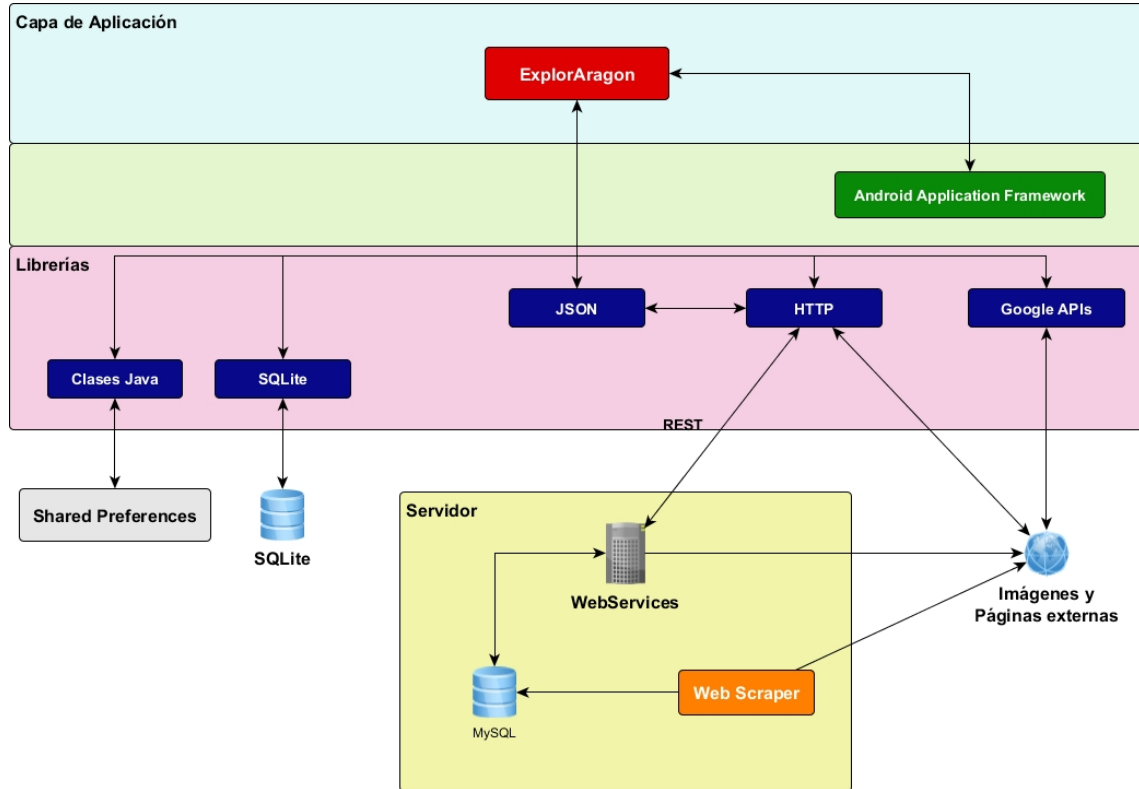


Figura 2.1: Arquitectura del sistema

### 2.2.2. Modelo de datos

Una vez expuestos los requisitos de la aplicación, definidas cada una de las partes que la conforman y descrita la arquitectura del sistema, el siguiente paso sería exponer el modelo de datos, es decir, la forma en la que se organiza la información que se utilizará en el sistema.

En primer lugar, obtenemos cada una de las entidades que conformarán el modelo de clases de la aplicación y definimos las relaciones que se establecerán entre cada una de esas entidades. El modelo resultante de la aplicación desarrollada será el presentado en la Figura 2.2.

La clase principal sobre la que se organizará el resto de la información será la clase usuario (*User*) en la que se detallan los datos de los usuarios. Cada una de

<sup>8</sup><https://www.openshift.com/>

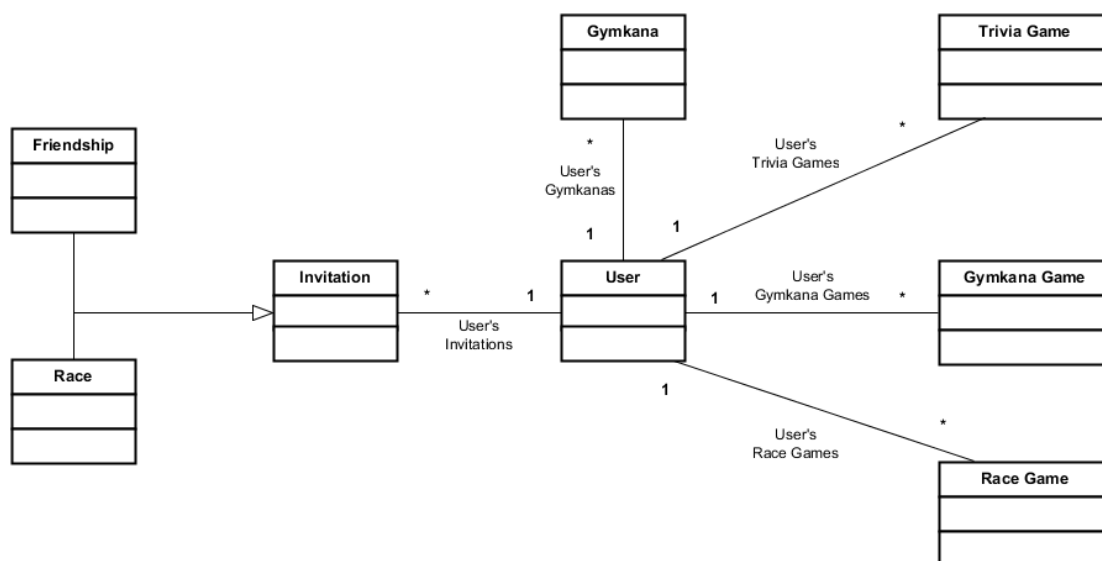


Figura 2.2: Modelo de clases de ExplorAragon

las demás clases es información adicional que se asignará a este usuario, excepto información de los municipios o la generación de preguntas.

La aplicación también pasará otros datos a través de las actividades y que no es necesario almacenar en los objetos, ya que se utilizarán a lo largo de toda la aplicación (como puede ser el identificador del usuario o la clave para identificarse en el servidor), o bien datos que se generan temporalmente para utilizarse en un momento puntual, como puede ser el identificador de otro usuario, el número actual de la pregunta o de la localización que se desea consultar o la puntuación, entre otros.

## 2.3. Desarrollo de la aplicación

La principal carga de trabajo en número de horas de este proyecto es el desarrollo de la aplicación móvil. En la presente sección se comentarán los puntos más importantes del desarrollo de la aplicación; la metodología, las decisiones de diseño y la estructura de la aplicación.

### 2.3.1. Metodología de desarrollo de la aplicación Android

Una de las principales características de Android es que permite realizar el diseño gráfico mientras se implementan las funciones. Pese a ello, para identificar las diferentes historias de usuario, se realizó un diseño previo con interacción para la navegación básica, algunos ejemplos pueden verse en la figura 2.3. También se han



desarrollado al mismo tiempo los servicios necesarios para la recopilación de información, lo que se detallará en la sección 2.4. Por todo ello, las diferentes partes del sistema han avanzado en paralelo.

Utilizando como base el diseño previo y la navegación, y utilizando los recursos gráficos ofrecidos por el SDK de Android, se comenzó con el desarrollo de la interfaz gráfica. A cada una de las ventanas de la aplicación le corresponde un código fuente independiente, lo que permite la modularidad del desarrollo y su sencillez. A su vez, es posible establecer un tema general para la aplicación que respetarán todas las ventanas, lo que permitirá que ciertos elementos tengan una representación común. También es posible modularizar diferentes partes de la interfaz como el menú, las listas de elementos, ... Para el desarrollo de la interfaz se han seguido los siguientes pasos:

- Creación del tema, editando una serie de archivos XML en los que se indican los colores principales y secundarios de la aplicación y la representación de los diferentes elementos de la aplicación. Para ello se utilizaron diferentes herramientas online<sup>9</sup> que facilitaron la generación de los archivos necesarios para representar el tema.
- Diseño de elementos comunes que se utilizarán en una o varias ventanas (menú, listas de elementos personalizadas y pestañas).
- Diseño gráfico de las ventanas editando los archivos XML, con la ayuda de un asistente gráfico que permite comprobar la representación de los diferentes elementos.
- Análisis de la información que debe mostrarse en cada una de las ventanas.
- Añadir interactividad a cada uno de los elementos de la ventana (botones, elementos de una lista, cuadros de texto, ...).
- Comprobación mediante “prueba y error” de cada uno de los componentes añadidos.

### 2.3.2. Decisiones de diseño

En este apartado se describirán algunas de las decisiones tomadas para realizar el diseño de las pantallas de la aplicación.<sup>2.4</sup>

---

<sup>9</sup><http://android-holo-colors.com/>



Figura 2.3: Diseño de navegación previa

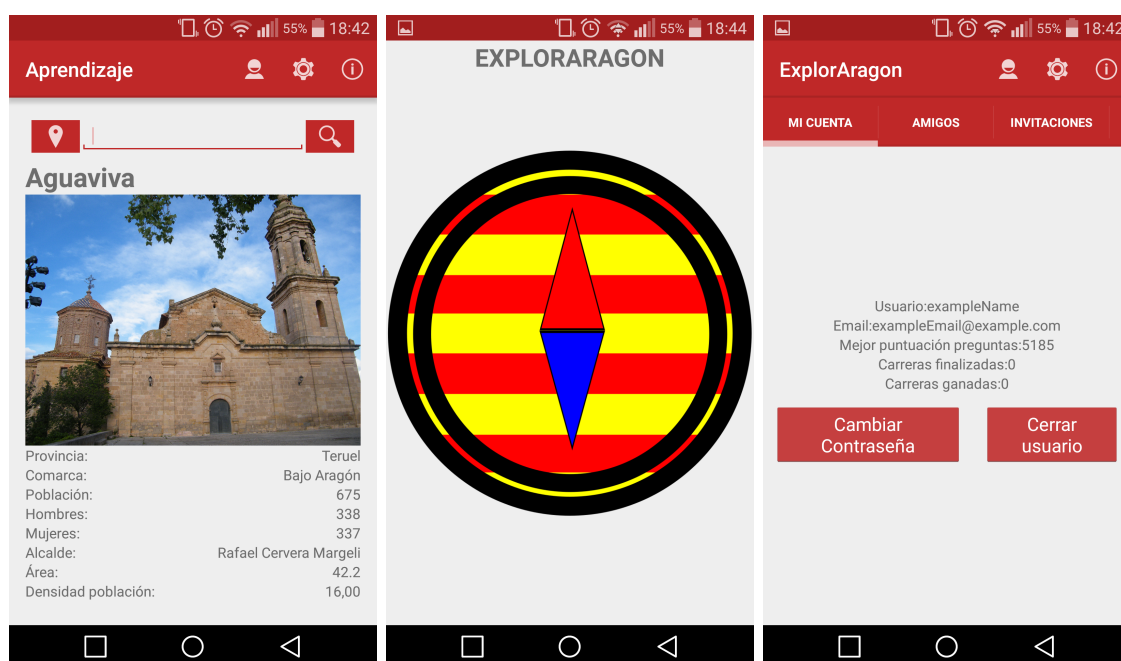


Figura 2.4: Ejemplos de pantallas

## Representación de la información

Al tratarse de una aplicación en la que se reúne tanto la actividad realizada en los juegos de preguntas y respuestas o gymkanas, como la posibilidad de consultar

información sobre los diferentes municipios, la representación de la información es uno de los puntos más importantes a tener en cuenta. Por ello se han elaborado diferentes tipos de listas, acceso a mapas, mensajes del sistema que informen al usuario del estado de la información o de los resultados de sus consultas.

**Listas:** Uno de los principales *widgets* en toda aplicación móvil son las listas. A través de una descripción de la información que debe mostrar cada uno de los elementos, se crean secuencias de elementos que posean esa información y que permiten un desplazamiento vertical. En ExplorAragon existen diversos tipos de listas: las clasificaciones, las invitaciones, las localizaciones de la gymkana de un usuario, los amigos,... Algunas de estas listas son representaciones de listas simples y otras poseen elementos más complejos.

**Pestañas:** Para la elaboración de ventanas en las que se quiera representar al mismo tiempo más de un tema relacionado pero diferente, El SDK de Android habilita la posibilidad de implementar pestañas. Al existir las pestañas en una ventana, se podrá ver el título de cada una de las pestañas disponibles bajo la barra de acciones. En la aplicación se podrán consultar las pestañas en el menú de juegos o en el perfil del usuario (Figura 2.5).

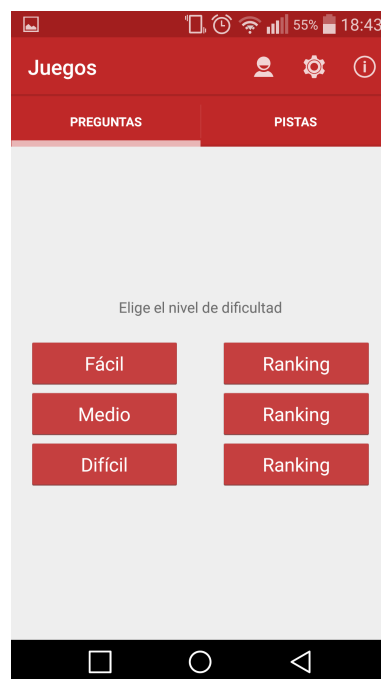


Figura 2.5: Pantallas del menú de juegos. Ejemplo de uso de pestañas.

**Toast:** Los toast o mensajes flotantes son mensajes que se muestran durante un breve periodo de tiempo y desaparecen una vez transcurrido ese tiempo, sin

que necesite interacción por parte del usuario. En los toast se notifican actualización de información del usuario, fallos de conexión u otras noticias puntuales de interés para el usuario.

**Diálogos:** Un diálogo es el equivalente a un pop-up de una página web, esto es, una pequeña ventana en la que aparece una información y se solicita al usuario que tome una decisión, ya sea introduciendo una información adicional o no. En la aplicación se utilizan en varios puntos. Se utiliza para recuperar la contraseña, introduciendo la dirección email del usuario; al informar sobre el resultado de una respuesta en el juego de preguntas y respuestas; o al resolver la posición en una Gymkana.

**Mapas:** Para la elaboración de los mapas se utiliza el servicio ofrecido por la API de Google Maps, ya que resulta muy completa y ofrece una amplia gama de funcionalidades. Se utilizarán dos de estas funcionalidades principalmente. El geoposicionamiento y la posibilidad de fijar un marcador sobre el mapa, que identifique un punto o localización. Un ejemplo del mapa puede verse en la figura 2.6.

**Menú:** En la barra de acciones se puede consultar un sencillo menú que dará acceso a algunas de las ventanas de la aplicación para la configuración o el perfil del usuario, así como información del desarrollador de la aplicación. También en la barra de acciones podremos consultar el nombre de la pantalla en la que se encuentre el usuario.

### 2.3.3. Estructura de la aplicación

En la presente sección se pretende describir la estructura de la aplicación móvil, la cual sigue el esquema representado en la figura 2.7. Aunque seguidamente se describirán las funciones principales de las distintas ventanas de la aplicación, se detallará su uso con más detalle en el Anexo C “Manual de usuario”.

Las principales funciones de las ventanas son las siguientes:

- **SplashActivity:** Se trata de una pantalla de introducción a la aplicación, en la que se realizará la carga de los datos básicos de la aplicación, si fuera necesario.
- **LoginActivity:** Será la primera ventana en la que el usuario podrá introducir datos. Desde esta ventana se podrá acceder a la creación de un nuevo usuario, en la ventana *CreateUserActivity*, o a introducir las credenciales para acceder al resto de la aplicación.
- **MainActivity:** Una vez autenticado el usuario, se accede a la ventana principal de la aplicación, desde donde se podrá acceder a cada una de las secciones de la aplicación, juegos, aprendizaje y perfil del usuario.



Figura 2.6: Pantallas de juego de Gymkana en modo inmóvil. Ejemplo de uso de los mapas.

- **LearnInfoActivity:** Permitirá al usuario la carga de información relativa a un municipio de Aragón.
- **GamesActivity:** Se divide en dos secciones principales, la relacionada con los juegos de tipo trivia y los de tipo gymkana.
- **TriviaActivity:** Se inicia una partida de tipo trivia, con preguntas y respuestas con la dificultad seleccionada por el usuario, como se ha descrito en la sección 2.1.3.
- **TriviaFinalActivity:** Una vez finalizadas las preguntas, se permitirá al usuario consultar su puntuación final y compartirla con otros usuarios. Desde esta aplicación se podrá comenzar una nueva partida con la misma dificultad, volver a GamesActivity o a MainActivity.
- **RankingActivity:** Mostrará un listado de las mejores puntuaciones obtenidas en las partidas de tipo trivia.
- **CluesActivity, RaceActivity, CluesLazyActivity:** Las tres comenzarán una partida de tipo gymkana, como se ha descrito en la sección 2.1.3, aunque con algunas diferencias. CluesActivity comenzará una partida normal de tipo gymkana. CluesLazyActivity permitirá al usuario jugar una partida de tipo gymkana, pero sin la necesidad de moverse hasta el lugar indicado por las



- **RaceWaitingActivity:** Es una pantalla en la que se indica al usuario que debe esperar al inicio de una partida gymkana tipo carrera, lo que decidirá el creador de la carrera.
- **InfoProfileActivity:** Se mostrará información relacionada a un usuario, permitiendo cancelar una amistad, enviar una invitación de amistad a otro usuario, ...
- **NewPasswordActivity:** Permitirá modificar la contraseña al usuario.

En todas las pantallas, excepto en las relativas a la participación en una partida de alguno de los juegos, se podrá acceder a través de un menú al perfil del paciente o a la configuración básica de la cuenta.

### 2.3.4. Dotación de interacción

Una de las partes más importantes de toda aplicación móvil es que sea interactiva con el usuario, para que le resulte más sencillo interactuar con ella. En esta sección se describirán algunas de las decisiones tomadas para adoptar a la aplicación de una mayor interacción.

#### Notificaciones

A través de un servicio de Google, se enviarán notificaciones al usuario, que le mantendrán informado sobre su actividad en la aplicación. Se enviarán diversos tipos de notificaciones:

- **Invitaciones:** Cuando un usuario sea invitado, ya sea de amistad o de participación en una carrera, recibirá una notificación en su teléfono, que le permitirá responder a la petición rápidamente.
- **Desarrollo de una carrera:** Cuando un jugador esté participando en una carrera, recibirá diferentes notificaciones cuando los otros equipos alcancen ciertos hitos, como el 50 % de las localizaciones recorridas o la pérdida del primer puesto en la carrera.
- **Modificación de la clasificación:** Cuando un jugador se encuentre entre los diez mejores en la clasificación general, recibirá una notificación si otro consigue una mayor puntuación en la misma dificultad. Lo mismo ocurrirá en el grupo de amigos del jugador.

## Gestos

A la hora de dotar de interacción a una aplicación, existen diversos tipos de gestos que pueden realizarse para interactuar con la interfaz de la aplicación<sup>2.8</sup>. No se tiene en cuenta el golpeo con el dedo en la pantalla como gesto, ya que es la funcionalidad básica de cualquier aplicación.

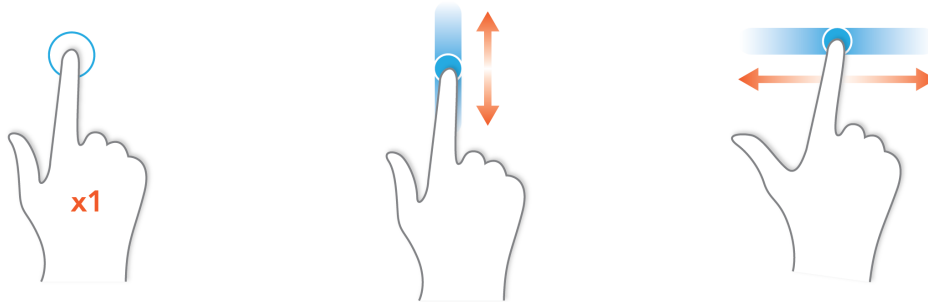


Figura 2.8: Gestos aplicables en ExplorAragon

En primer lugar, para desplazarse por pantallas en las que no cabe toda la información al mismo tiempo en la pantalla del smartphone, se habilita el movimiento de desplazamiento vertical de pantalla, lo que permite subir y bajar dentro de la misma ventana.

El segundo de los gestos habilitado está disponible en las ventanas en las que se dispone de varias pestañas, como son las vistas del perfil del usuario o el menú de juegos en la aplicación. Realizando un movimiento vertical con el dedo, es posible navegar entre las diferentes pestañas existentes.

Finalmente, los gestos habilitados en los mapas de Google, también estarán disponibles en aquellas ventanas en las que se tenga acceso a un mapa. Entre los diferentes gestos disponibles encontramos el desplazamiento alrededor del mapa; acercar o alejar el mapa, acercando o alejando la punta de los dedos, o dando dos golpes sobre el mapa; o rotar el mapa, dejando un dedo fijo mientras giramos otro alrededor del primero, entre otros gestos.

## 2.4. Desarrollo del servidor

La segunda parte del desarrollo del sistema es la creación del servidor y los servicios RESTful. En esta sección se describirán cada una de las partes en las que se divide el servidor y cómo se ha modularizado e implementado el código.



### 2.4.1. Base de datos

Una de las principales partes del servidor es la base de datos. Ésta se ha desarrollado teniendo en cuenta las tablas que pueden verse en las figuras 2.9 y 2.10.

A continuación se detallará la información almacenada en cada una de las tablas:

- **Users:** Es la tabla principal, en la que se almacenan los datos de acceso a la aplicación.
- **UserSettings:** Almacenará los datos de configuración de los usuarios, permitiendo actualizarlos cuando se acceda con una cuenta en diferentes dispositivos.
- **Friendships:** Establece una relación entre dos usuarios, inicialmente de tipo amistad, aunque es posible establecer otras relaciones.
- **UserTriviaGames:** Almacena las partidas de preguntas y respuestas que ha jugado el usuario.
- **Races:** Almacenará los datos necesarios para la realización de una partida gymkana de tipo carrera.
- **PlayedGymkanas:** Contendrá los datos relacionados con la realización de partidas de tipo gymkana.
- **Gymkanas:** En esta tabla se almacenará la información relacionada con las gymkanas generadas por los usuarios.
- **Locations:** La tabla almacenará información sobre posibles localizaciones de las gymkanas.
- **Places:** Contendrá datos de lugares o puntos de interés de diferentes municipios.
- **Cities:** Contendrá la información relacionada con los municipios.

### 2.4.2. Servicios Web

La función principal de un servidor es ofrecer servicios para la interacción con la información almacenada en el servidor. Para ello se han desarrollado una serie de servicios web, que permitirá a la aplicación acceder a la información almacenada en la base de datos previamente descrita. La documentación de los servicios ofrecidos puede consultarse, en inglés, online<sup>10</sup> o en el apéndice D.

---

<sup>10</sup><http://docs.exploraragon.apiary.io/>

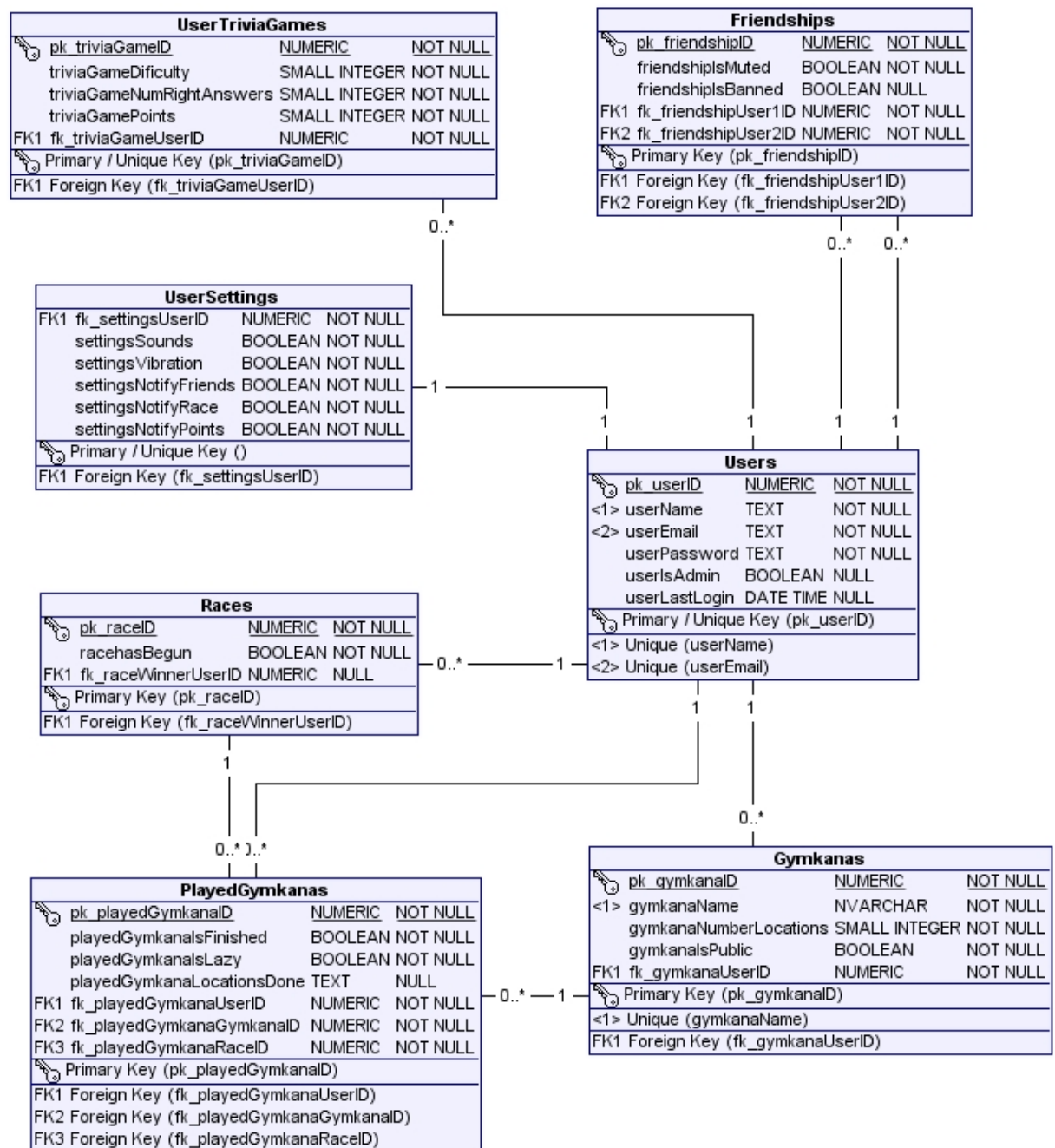


Figura 2.9: Modelo de la base de datos (Parte 1)

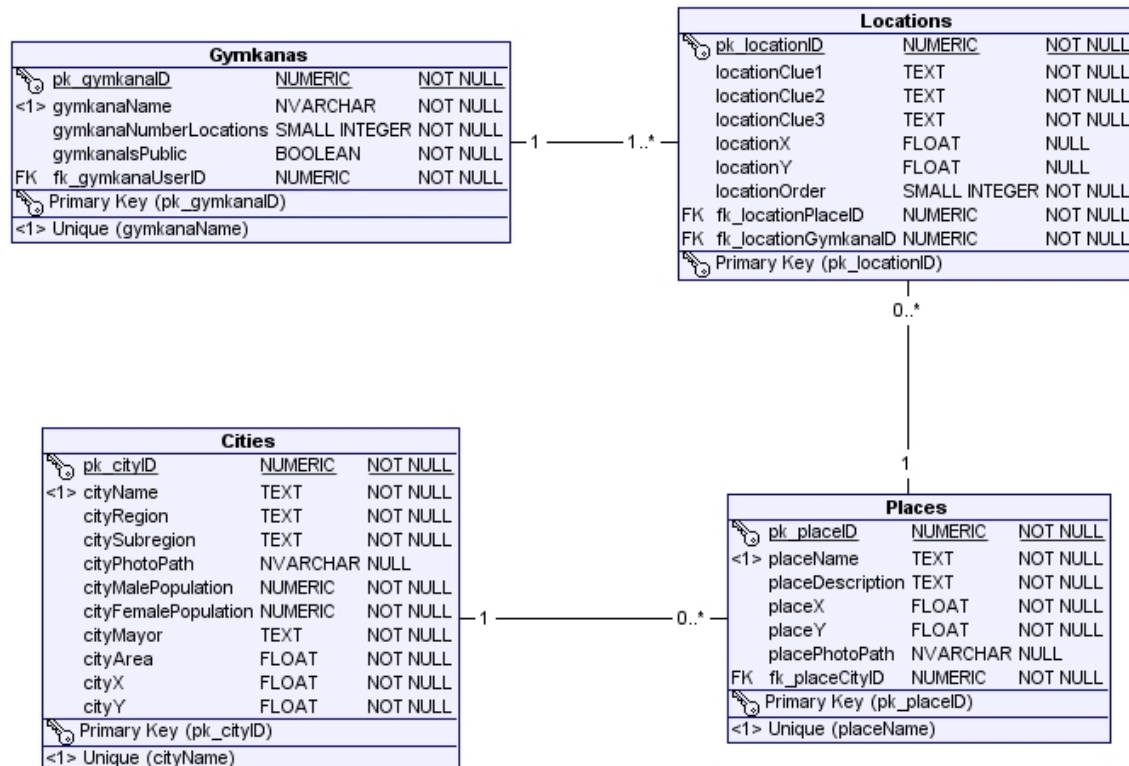


Figura 2.10: Modelo de la base de datos (Parte 2)

Para el desarrollo de los servicios web se han aprovechado las funciones ofrecidas por OpenShift<sup>11</sup>. Esto nos ha dado acceso a un servidor de tipo JBoss con una base de datos MySQL integrada. Al tratarse de un servidor de tipo JBoss, se ha decidido utilizar la librería RestEasy, la cual se incluye en los servidores del tipo indicado y permite la integración con sistemas de tipo modelo vista controlador (MVC).

El código del servidor se ha estructurado en tres partes diferenciadas:

- **Resources:** Se encargarán de la recepción de la petición, así como de la generación de la respuesta. Para ello solicitarán los datos a los *Managers*.
- **Managers:** En esta parte del código se realizarán las diferentes funciones de tratamiento de la información y se accederá a los *DAOs* para recuperarla.
- **Data Access Objects (DAOs):** Estas clases serán las encargadas de acceder a la base de datos y realizar las peticiones de forma segura, recuperando la información necesaria por los *Managers*.

<sup>11</sup><https://www.openshift.com/>

### 2.4.3. Recopilación de datos

La última parte del servidor es el proceso de recuperación de datos. El proceso se ejecutará diariamente, aunque la carga de información variará en función del número de días desde la última carga. Para ello el sistema accederá a diferentes fuentes de datos abiertos y a diferentes páginas web, desde las que extraer información. El proceso se divide en las siguientes partes:

- **Datos sobre municipios.** Recopilación de los datos básicos sobre los municipios de Aragón. Esta información se consultará a través de openData Aragón. Este proceso se ejecutará una vez al mes, ya que la información relacionada con los municipios está sujeta a pocas variaciones. La carga de imágenes se realizará diariamente, ya que la limitación de los servicios ofrecidos limita a 100 el número de búsqueda de imágenes realizadas, al no existir imágenes para los municipios en la base de datos de openData Aragón.
- **Datos sobre puntos de interés.** Recopilación de puntos de interés de los diferentes municipios. En el caso de Zaragoza, se accederá a los servicios de datos abiertos del Ayuntamiento de Zaragoza, dónde existe una sección relacionada con estos datos. En el caso de otros municipios, se buscarán puntos de interés relacionados estos a través de otras páginas y otros servicios de datos abiertos como la dbpedia<sup>12</sup>. Al ser un proceso pesado y tratarse de información que no es susceptible a muchos cambios, este proceso se realizará mensualmente.

## 2.5. Servicios de Google APIs

Para gran parte del tratamiento de la información se han utilizado algunos de los servicios de Google APIs. Para ello se habilitaron una serie de servicios, detallados a continuación, en la consola de Google<sup>13</sup>. Esto proporciona acceso a los servicios a través de una ApiKey, una cadena que identifica una aplicación o servidor.

### Google Maps Android API

En la aplicación Android se utilizó el servicio de mapas de Google, integrado en el SDK de Google y que permite tener acceso a todas las funcionalidades de la API.

### Custom Search API

Para la búsqueda automatizada de imágenes de municipios, se utilizó el servicio de búsquedas personalizadas. Este servicio permite, introduciendo una serie de

---

<sup>12</sup><http://es.dbpedia.org/>

<sup>13</sup><https://console.developers.google.com>

parámetros, realizar una búsqueda en Google Images, especificando los derechos de reutilización y páginas prioritarias de las que obtener la información. De este modo, con una simple consulta, se obtiene una respuesta JSON con las direcciones de las imágenes relacionadas con la búsqueda. Para introducir todas estas propiedades, se utilizó el panel de control de búsquedas personalizadas<sup>14</sup>. Este servicio está sujeto a una cuota de cien consultas diarias como máximo, en el programa utilizado durante el desarrollo, lo que ha supuesto una restricción a la hora de realizar la recopilación de datos.

## Google Cloud Messaging

Este servicio proporciona una lista de sencillos métodos con los que enviar notificaciones a dispositivos Android o páginas web. De esta forma se pueden enviar alertas a los jugadores de carreras de Gymkanas y a los de juegos Trivia, cuando su puntuación sea superada por un amigo o por otro jugador si su puntuación se encuentra entre los diez mejores.

## 2.6. Pruebas

Una de las partes más importantes del desarrollo del software es la realización de pruebas sobre el código.

Conforme se ha ido desarrollando el sistema, este se ha visto sujeto a una serie de pruebas de distintos tipos, lo que permitía comprobar que se ofreciese la funcionalidad deseada.

### 2.6.1. Pruebas unitarias

Las pruebas unitarias se centran en verificar que una parte del código cumple con los propósitos deseados.

Ya que tanto el servidor como la aplicación se han desarrollado utilizando Java y Android, respectivamente, las pruebas unitarias se pueden realizar utilizando la librería JUnit, en la que se pueden programar y verificar que cada método del código realiza la función para la que estaba diseñado.

En el caso del servidor, al haber sido desarrollado con la plataforma OpenShift, se utilizó la herramienta Jenkins<sup>15</sup>. Esta herramienta permite automatizar una serie de pruebas que comprueban que el código escrito cumple una serie de niveles de calidad tanto en el desarrollo como en la comprensión del mismo, ajustándose a los estándares del desarrollo Java.

---

<sup>14</sup><https://cse.google.com/>

<sup>15</sup><https://jenkins.io/>

### 2.6.2. Pruebas de componentes

Las pruebas de componentes comprueban el tratamiento de datos a través de de diversos módulos.

Para las pruebas de componentes en la aplicación, se replicaban los datos que se deberían obtener del servidor, lo que permitía acceder a cada una de las partes de la aplicación.

En el caso del servidor, se utilizan aplicaciones como *Postman*<sup>16</sup>, que permite realizar llamadas sobre un servidor y comprobar los datos recibidos de las mismas.

### 2.6.3. Pruebas de sistema

Las pruebas de sistema realizan pruebas sobre el sistema en general, comprobando que la integración final entre cada una de las partes se realiza de forma correcta.

Una vez superadas las pruebas de componentes, se llama al servidor desde la aplicación, que se encarga de tratar los datos reales recibidos y mostrarlos al usuario o de enviar los datos facilitados por el usuario al servidor. en las pruebas de sistema, se comprueba que esta comunicación se realiza de forma correcta.

### 2.6.4. Pruebas operacionales

Finalmente, se realizan una serie de pruebas operacionales. Estas son realizadas por una serie de personas ajenas al desarrollo y que se centran en comprobar la calidad del sistema.

Para ello se ha entregado la aplicación a una serie de personas no involucradas en el proyecto y con distinto grado de conocimiento tecnológico, que han probado la aplicación resaltando diferentes problemas o mejoras. Las personas involucradas van desde desarrolladores de software, diseñadores, jefes de proyectos y personas sin ningún conocimiento sobre el desarrollo de software.

## 2.7. Despliegue

Una vez finalizado el desarrollo del sistema y superadas todas las pruebas, el siguiente paso sería desplegar el código para el alcance del usuario. Esto se realiza por separado en el caso del servidor y de la aplicación.

---

<sup>16</sup><https://www.getpostman.com/>

### 2.7.1. Google Play Store

Para el despliegue de la aplicación para que sea accesible por los usuarios Android, esta debe ser suministrada a través de la aplicación Google Play Store. Aquí se centraliza la oferta de aplicaciones para que el usuario pueda acceder a un listado de forma sencilla y pueda encontrar la aplicación que busca rápidamente.

También es posible suministrar el archivo de instalación directamente al usuario final a través de una página web, pero para muchos usuarios esto resulta complejo y poco fiable, por lo que se ha decidido utilizar la otra opción.

Para publicar la aplicación en Google Play Store es necesario poseer una cuenta de desarrollador, cuyo coste es de 25 dólares estadounidenses. Al crear la cuenta será necesario suministrar datos acerca del desarrollador, tales como su nombre, dirección, etc.

El siguiente paso será rellenar los datos sobre la aplicación, entre los que se encuentra su nombre, una breve descripción, el nivel de edad para utilizarla, ... Además de una imágenes que reflejen el uso de la aplicación y ejemplifiquen sus propiedades.

Finalmente, se sube el archivo de instalación de Android (archivo de extensión *apk*) a la página de administración de Play Store. Google procederá a revisar la aplicación y decidirá si su uso está permitido bajo los acuerdos y licencias pertinentes. Este proceso suele llevar alrededor de 2 horas.

La aplicación está disponible para su descarga en el enlace <https://play.google.com/store/apps/details?id=es.lcpdev.exploraragon>.

### 2.7.2. OpenShift

El servidor se desarrolló aprovechando la plataforma OpenShift. Como se ha descrito previamente, esta plataforma permite el desarrollo gratuito de servidores de pequeño tamaño, con una cantidad limitada de espacio en la base de datos y con otras limitaciones. Sin embargo, al tratarse de datos de usuarios, en la actualidad estos servidores se ajustan perfectamente a lo necesario para la aplicación.

El despliegue en OpenShift se realiza a través de servicios Git. La plataforma provee un servidor Git al cual se puede subir el código del servidor. Una vez subido al servidor Git, este se compila y se despliega de forma automática en el servidor contratado para que los servicios sean finalmente visibles por los demás usuarios.

Además, la plataforma OpenShift permite la integración con Jenkins, lo que permite realizar una serie de pruebas sobre el código del servidor. Esto se ha descrito con más detalle en la sección 2.6.1.





# Capítulo 3

## Gestión del Proyecto

Este capítulo de la memoria se centra en la gestión del tiempo y la planificación al desarrollar el proyecto.

### 3.1. Planificación

Tras el análisis de los requisitos del sistema y del proyecto en general, se realizó una planificación temporal del coste de diseño e implementación del proyecto. Sin embargo esta planificación ha estado sujeta a varias rectificaciones con el paso del tiempo, ya que circunstancias personales han limitado el tiempo disponible para la realización del mismo. En concreto el hecho de trabajar desde el mes de junio de 2015 a jornada completa hizo que compaginar el desarrollo de un proyecto de esta envergadura en solitario resultase más complicado de lo esperado. Para tratar de ajustarse lo máximo posible al tiempo y tratando de realizar un desarrollo lo más eficiente posible, se analizaron diferentes herramientas de gestión de tiempo, como por ejemplo Sinnaps<sup>1</sup> o Jira<sup>2</sup>. En la figura 3.1 se puede ver un diagrama de Gantt en el que se representa la planificación final.

### 3.2. Esfuerzos

El esfuerzo dedicado cada semana variaba en función de la disponibilidad y de las exigencias en el entorno laboral, por lo que algunas tareas se prolongaron más o menos en el tiempo que otras, aunque se trató de respetar al máximo esta planificación. Finalmente el total de horas dedicadas ha podido encontrarse alrededor de las mil horas, contando desde el análisis, diseño, desarrollo y documentación. La división del esfuerzo realizado se divide de la forma expuesta a continuación.

---

<sup>1</sup><https://www.sinnaps.com/>

<sup>2</sup><https://es.atlassian.com/software/jira>

Tabla 3.1: Estimación de tareas

Fase	Tarea	Estimación en horas
Análisis	Historias de usuario	25
	Definición de requisitos	15
	<b>Subtotal</b>	40
Diseño	Arquitectura del sistema	25
	Prototipado de ventanas	35
	Modelo de datos	15
	<b>Subtotal</b>	75
Desarrollo aplicación	Navegación de ventanas	30
	Gestión de SQLite	20
	Implementación de listas	10
	Comunicación con el servidor	50
	Sección de usuario	40
	Sección de aprendizaje	35
	Sección trivia	40
	Sección social	75
	Sección gymkanas	100
	<b>Subtotal</b>	400
Desarrollo servidor	Arquitectura del servidor	25
	Creación de la base de datos	25
	Recopilación de datos abiertos	40
	Gestión de usuarios	40
	Gestión de trivia	40
	Gestión social	55
	Gestión de gymkanas	75
	<b>Subtotal</b>	300
Pruebas del sistema	Pruebas del sistema	30
Pruebas operacionales	Pruebas operacionales	30
Documentación	Documentación	50
Creación de la memoria	Creación de la memoria	75
<b>Trabajo total</b>	<b>Total</b>	1000

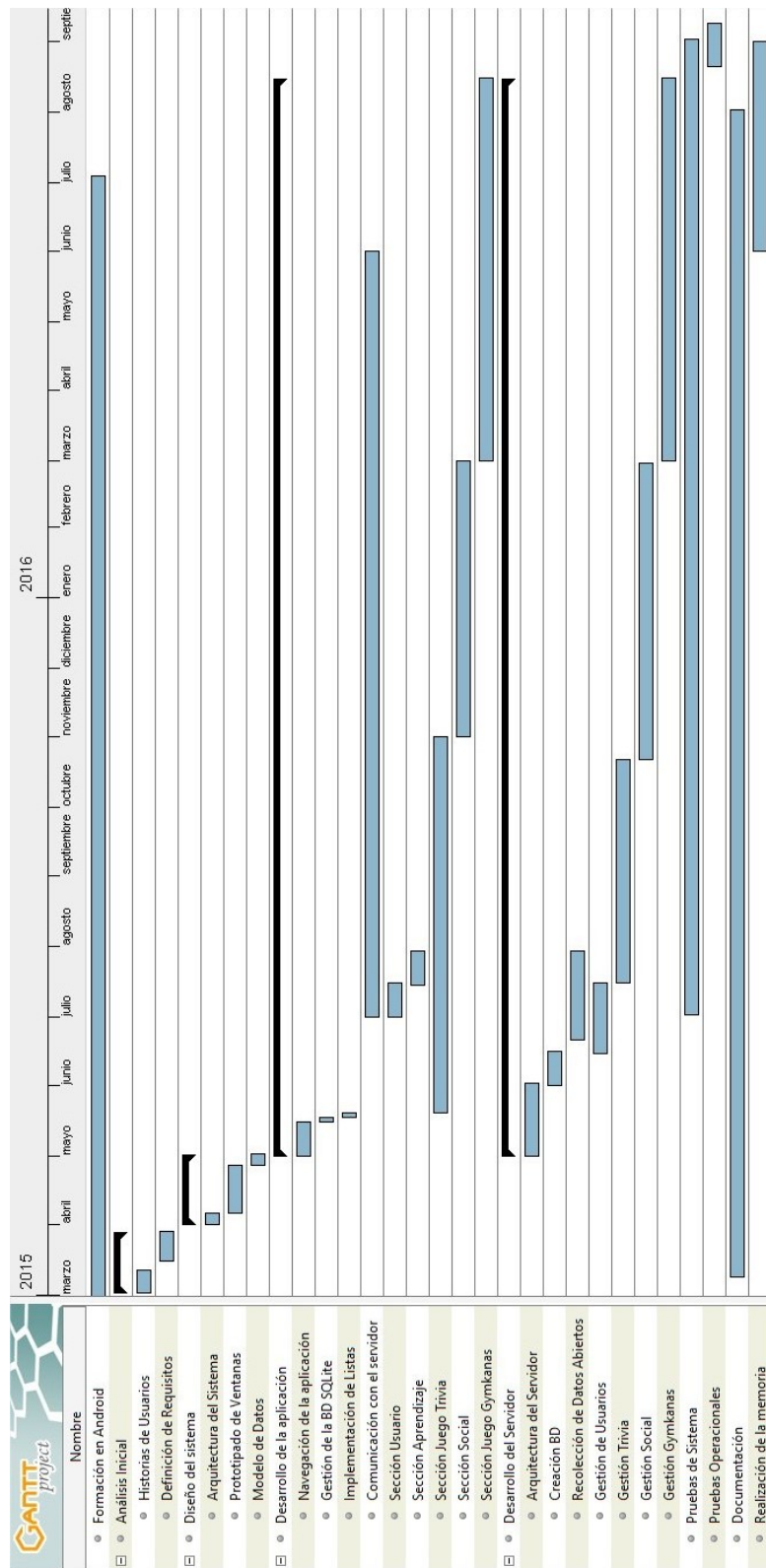


Figura 3.1: Planificación de ExplorAragón



# Capítulo 4

## Conclusiones

En este último capítulo de la memoria se comprobará el alcance del cumplimiento de los objetivos propuestos. A continuación, se analizarán y postularán una serie de ideas sobre el trabajo futuro realizable para mejorar la aplicación. Finalmente y como conclusión final de la memoria se dará una valoración personal de la realización, análisis, diseño y desarrollo de este PFC.

### 4.1. Cumplimiento de objetivos propuestos

El resultado más apreciable es el correcto funcionamiento del sistema software descrito en el presente documento.

Se puede afirmar que la aplicación desarrollada cumple todos los objetivos marcados al inicio del proyecto: utilización de datos abiertos, tanto de OpenData Aragón como del Ayuntamiento de Zaragoza; la extracción de datos de otras fuentes como Wikipedia; el desarrollo de una aplicación que favorezca el aprendizaje y el entretenimiento del usuario.

En resumidas cuentas se han realizado los siguientes objetivos:

- **Desarrollo de una aplicación Android.** Se ha desarrollado una aplicación Android que permite la geolocalización del usuario, la participación en un entorno competitivo en diversos juegos, la consulta de información sobre municipios de Aragón y la interacción de usuarios entre sí.
- **Desarrollo de servicios web RESTful.** Realización y publicación de una serie de servicios web que facilitan información en el lado de la aplicación móvil.
- **Desarrollo de un sistema de automatización de carga de datos.** Carga de datos desde una rutina de carga de información desde las distintas fuentes de datos abiertos indicadas.

- **Preparación y publicación del trabajo en el mercado de aplicaciones Android.** Publicación en el sistema Google Play Store<sup>1</sup> de la aplicación ExplorAragon.

## 4.2. Trabajo futuro

Existen una gran cantidad de líneas de trabajo en el futuro para la aplicación desarrollada, algunas de las cuales se enumerarán a continuación:

- **Introducción de gymkanas y trivias temáticos.** En el sistema de carga de gymkanas se podrían introducir gymkanas centradas en temas religiosos, artísticos, centradas en épocas como la romana o durante los sitios de la Guerra de la Independencia. Esto proporcionará unos datos más específicos y ayudará al usuario a ampliar sus conocimientos de la comunidad.
- **Ampliar la variedad de los tipos de preguntas.** Se podrían implementar nuevos tipos de preguntas, incluyendo sistemas de audio, puntos de interés, escudos y banderas de municipios y un largo etcétera de preguntas que han podido quedar sin reflejar y que podrían representar una mayor variedad, lo que supondría una mayor fuente de entretenimiento para el usuario.
- **Posibilitar la creación de preguntas por parte del usuario.** Al ser posible la creación de preguntas por parte del usuario, este podría sentirse más integrado en el sistema y afianzar su participación y mantenerlo como usuario activo de la aplicación.
- **Ampliar el ámbito de la aplicación a otras comunidades autónomas.** Una vez realizada la extracción de datos automatizada para Aragón, resultaría relativamente sencillo ampliar el sistema creando aplicaciones similares en otras comunidades autónomas, incluso alcanzando el ámbito nacional, únicamente cambiando algunos de los parámetros de la extracción de los datos. Esto supondría un mayor mercado y mayor número de usuarios posibles, generando toda una serie de aplicaciones como *ExplorCataluña*, *ExplorMadrid*, ...
- **Reutilización de los datos para otros propósitos.** Con los datos obtenidos podrían realizarse guías turísticas, pudiendo calcular la distancia entre los diferentes puntos de interés y el tiempo aproximado de visita de cada uno de ellos, datos que habría que calcular. Esto facilitaría al usuario el recorrido de los municipios de la comunidad aragonesa. Siguiendo esta línea, podría introducirse una sección de realidad aumentada, que permitiese al usuario obtener información detallada de un lugar, únicamente con su geoposicionamiento y con la ayuda de su cámara.

---

<sup>1</sup><https://play.google.com/store>

- **Inserción de anuncios y servicios *freemium*.** Para la monetización de la aplicación, se podrían realizar dos acciones diferentes. En primer lugar podrían introducirse anuncios utilizando los diferentes servicios de las APIs de Google o anuncios a través de empresas privadas que deseen publicitarse. Además de eso, ofrecer un servicio freemium que dará acceso a algunas de las opciones desarrolladas, como la creación de Gymkanas, o a otras opciones por implementar, como pistas adicionales en las Gymkanas o comodines en las partidas de preguntas y respuestas.

### 4.3. Valoración personal

Este proyecto fin de carrera ha supuesto un punto de inflexión en la mayoría de los aspectos de mi vida. Desde un principio quería realizar una aplicación móvil y desde el punto de vista académico me ayudó para aprender un nuevo lenguaje de programación y metodologías de trabajo. A su vez, tras haber aprendido el lenguaje Android y participar en diversas hackathones, esto me permitió continuar con la formación tanto académica como laboral y personal al conseguir una estancia en el coworking de La Terminal, como premio en una de estas hackathones. Esto me permitió conocer cómo otras personas desarrollaban aplicaciones, programas y sistemas software con diversas tecnologías y puntos de vista. También me permitió acceder a cursos de formación sobre la iniciación en el emprendimiento con metodología *Lean Start-up*. Y a nivel personal me ayudó comprender el modo en el que diversos autónomos luchaban por sacar adelante sus negocios, implementando mejoras y utilizando los lenguajes más novedosos.

La estancia en la terminal inició mi camino laboral, ya que una empresa tecnológica se interesó por mi perfil y terminé formando parte de esta compañía. Esto también me aportó una ayuda inesperada, ya que algunos de los expertos de dicha compañía, ya sean diseñadores, desarrolladores, jefes de proyectos o analistas, me apoyaron y dieron puntos de vista que no había tenido previamente en cuenta. Una vez más esto fue una importante contribución para mi crecimiento profesional y académico.

Finalmente, uno de los aspectos más importantes en el desarrollo de ExplorAragon, fue el hecho de aprender a ser mi propio cliente y el desarrollador al mismo tiempo. Sin lugar a duda, con la inestimable ayuda de mi director de proyecto, Francisco Javier López, que me ayudó a comprender las metodologías de análisis y diseño de una aplicación, así como el desarrollo de los casos de uso y el diseño de las ventanas de la aplicación. Con sus pautas y sus ideas me ha ayudado a entender y aprender la metodología a seguir para el desarrollo de software.

A grandes líneas, este proyecto me ha permitido abrir nuevas vías de conocimiento y ganas de continuar con el aprendizaje, una gran cantidad de contactos en el sector de los emprendedores y conocimientos sobre la metodología *Lean Start-up*,

el inicio de una carrera profesional de gran atractivo y rodeado por profesionales del sector.



# Bibliografía

- [1] *Android Developers*. <https://developer.android.com/index.html>.
- [2] *GitFlow*. <http://danielkummer.github.io/git-flow-cheatsheet/>.
- [3] *JSON*. <http://json.org/>.
- [4] *JSON - Definicion*. <https://tools.ietf.org/html/rfc7159>.
- [5] *LaTeX*. <http://www.latex-project.org/>.
- [6] *Lenguaje de programacion Java*. [https://www.java.com/es/about/whatis\\_java.js](https://www.java.com/es/about/whatis_java.js).
- [7] *LyX*. <http://www.lyx.org/>.
- [8] *MiKTeX*. <http://miktex.org/>.
- [9] *OpenShift*. <https://developers.openshift.com/>.
- [10] *RestEasy*. <http://resteasy.jboss.org/>.
- [11] Skyscanner. *Las mejoras 25 apps de viajes*. <https://www.skyscanner.es/noticias/las-25-mejores-apps-de-viajes>.
- [12] Tiobe. *Ranking de lenguajes de programacion*. <http://www.tiobe.com/tiobe-index/>.
- [13] Wikipedia. *Base de datos SQLite*. <https://en.wikipedia.org/wiki/SQLite>.
- [14] Wikipedia. *Estilo REST*. [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer).
- [15] Wikipedia. *Git*. [https://en.wikipedia.org/wiki/Git\\_\(software\)](https://en.wikipedia.org/wiki/Git_(software)).
- [16] Wikipedia. *Google APIs*. [https://en.wikipedia.org/wiki/Google\\_APIs](https://en.wikipedia.org/wiki/Google_APIs).
- [17] Wikipedia. *Web Scraping*. [https://en.wikipedia.org/wiki/Web\\_scraping](https://en.wikipedia.org/wiki/Web_scraping).

**Nota:** Las direcciones de internet que aparecen en la presente bibliografía están revisadas y son accesibles en la fecha de depósito del documento, en septiembre de 2016.



## Parte II

### Anexos



# Apéndice A

## Tecnologías y herramientas utilizadas

En este anexo se describirán las principales tecnologías utilizadas y las herramientas con las que han sido tratadas en cada caso.

### A.1. Android

La aplicación Android ha sido programada utilizando principalmente el lenguaje Java, en el que están basadas las aplicaciones nativas Android. En un principio se pensó en utilizar Eclipse como herramienta de trabajo, ya que se iba a utilizar para el servidor, sin embargo, finalmente, se optó por Android Studio. Esta herramienta, basada en IntelliJ IDEA, ha resultado muy práctica ya que ofrece una gran cantidad de funcionalidades que, hasta el momento, Eclipse no ha podido ofrecer. Algunas de estas serían la visualización y el renderizado de las pantallas o los nuevos emuladores en los que se puede comprobar el funcionamiento de la aplicación.

### A.2. Datos abiertos

Una de las partes más importantes de la aplicación es la accesibilidad a datos abiertos ofrecidos en diversos portales.

#### A.2.1. OpenData Aragón

La principal fuente de información sobre los municipios de Aragón sería, OpenDara Aragón<sup>1</sup>.

---

<sup>1</sup><http://opendata.aragon.es/>

En este portal se pueden encontrar una enorme cantidad de datos geopolíticos sobre la Comunidad Autónoma aragonesa. Dispone de diversos modos de visualización y recopilación de datos, un listado de aplicaciones disponibles (entre la que se encuentra *¿Conoces Aragón?*, precursora de este proyecto) y documentación para participar en el uso de los datos ofrecidos.

### A.2.2. Datos abiertos del Ayuntamiento de Zaragoza

Durante los últimos años, el Ayuntamiento de Zaragoza ha puesto a disposición de los ciudadanos cada vez más datos abiertos<sup>2</sup>. De este modo pretende ofrecer a aquellas personas preparadas, la posibilidad de gestionar esos datos para hacerlos accesibles y atractivos para el público general. De este modo se crean aplicaciones como *ViveZaragoza*<sup>3</sup>, que muestra datos sobre los autobuses urbanos, el tranvía o las estaciones Bizi; y se incentiva la creación de nuevas aplicaciones con otros propósitos, como *AdoptaPet*<sup>4</sup>, centrada en ofrecer al usuario la posibilidad de consultar las mascotas disponibles en el centro de protección animal.

Como puede comprobarse, la cantidad de datos es muy grande y crece cada día. Por ello se aprovecha este trabajo realizado para nutrir la aplicación ExplorAragón y poder ofrecer la mayor calidad posible de datos para el usuario final.

### A.2.3. Wikipedia

Wikipedia es, posiblemente, la mayor fuente de datos de internet en la actualidad. La posibilidad de colaborar por parte de los usuarios ha hecho que el número de artículos disponibles sea inmenso y, de esta forma, se trate de una fuente inagotable de información. Pese a todo ello, al poder ser manipulada por los usuarios, en muchas ocasiones, esta información debe ser contrastada. Por ello, para ExplorAragón, se utilizaran en su mayoría imágenes y vínculos que lleven al usuario a las páginas de la Wikipedia.

También es importante detallar la complejidad de la estructura del sistema de extracción de datos de los artículos de la Wikipedia, lo que hace más complejo el tratamiento de la información.

## A.3. JBoss RestEasy - Web Service

Diversos motivos laborales me han acercado, personalmente, al trato de servidores JBoss, que incorporan la librería RestEasy. Esta librería ayuda a conseguir

---

<sup>2</sup>[https://www.zaragoza.es/ciudad/risp/buscar\\_Risp](https://www.zaragoza.es/ciudad/risp/buscar_Risp)

<sup>3</sup><https://play.google.com/store/apps/details>

<sup>4</sup><https://play.google.com/store/apps/details>

de modo sencillo un estilo REST en la generación de servicios web. Por este motivo se tomó la decisión de utilizar este tipo de servidor, ya que las ventajas aportadas y la familiarización con este hacía que la posibilidad de avanzar fuera mayor. Otras alternativas habría sido la utilización del framework Spring en Java o del framework Symfony en PHP.

Aumentando aún más las ventajas de esta decisión, se encontró la posibilidad de contratar un servidor de forma gratuita con OpenShift. Esta plataforma esta desarrollada por el equipo de RedHat, que a su vez participa en el desarrollo de los servidores JBoss, por lo que la compatibilidad entre el servidor y la plataforma era clara.

Para programar los servicios web que se desplegarían en el servidor, se utilizó la herramienta Eclipse. Eclipse es una plataforma de desarrollo OpenSource y multi-plataforma basada en Java. Lo que hace que esta herramienta resulte especialmente atractiva es la posibilidad de completarla utilizando plug-ins o extensiones que complementan su funcionalidad. En este caso, el plug-in de OpenShift, se añadió a la configuración básica para poder gestionar el despliegue del código en el servidor a través de Git.

## A.4. $\text{\LaTeX}$

Para la elaboración del documento de la memoria, se ha utilizado  $\text{\LaTeX}$  [5], un procesador de texto basado en lenguaje marcado. Esto permite centrarse en el contenido del documento sin tener que preocuparse por el formato del mismo. Para poder utilizar  $\text{\LaTeX}$  en Windows, se precisó de la instalación de  $\text{\MiKTeX}$  [8], que permite mantener las librerías de  $\text{\LaTeX}$  actualizadas y gestionar fácilmente las que se desean instalar en el sistema. La principal herramienta utilizada para escribir el presente documento ha sido  $\text{\LyX}$  [7], ya que ofrece una gran cantidad de funciones que facilitan en gran cantidad la labor del redactor.

## A.5. Git

Git se ha convertido en una de las principales herramientas de control de versiones, como también lo es subversion<sup>5</sup> o mercurial<sup>6</sup>. Una de las principales ventajas de Git es la descentralización, lo que permite el acceso al repositorio de control de versiones desde prácticamente cualquier lugar con conexión a internet.

Las herramientas de control de versiones permiten al desarrollador de software mantener el control de los cambios aplicados y conservar una copia en remoto del

---

<sup>5</sup><https://subversion.apache.org/>

<sup>6</sup><https://www.mercurial-scm.org/>

archivo, de forma que en caso de pérdida de datos o problemas de corrupción de archivos, puede ser recuperada fácilmente.

Para el desarrollo de la aplicación se ha utilizado el portal GitHub, que esta tomando especial relevancia entre los desarrolladores de software de código abierto. Y para el servidor se utilizó el sistema de Git ofrecido por la plataforma OpenShift, por motivos ya descritos previamente en la memoria.

Para gestionar en ambos casos el código que se debía subir al servidor de versiones o que se conservaba en local a la espera de subir, se ha utilizado la herramienta SourceTree. Desarrollada por *Atlassian*, compañía que se ha encargado de otros repositorios como Bitbucket o sistemas de control de tareas y metodología ágil como Jira. SourceTree ofrece al usuario una interfaz gráfica en la que poder ejecutar los comandos de Git y que le permite tener un acceso más rápido y sencillo a los cambios realizados en los diversos documentos a lo largo del tiempo.



# Apéndice B

## Historias de Usuario y Requisitos

Una de las principales partes en el análisis del proyecto fue la preparación de diversas historias de usuario, como se ha descrito en la sección 2.1.1. En este apéndice se listarán las diferentes historias de usuario utilizadas para decidir los requisitos que debía satisfacer la aplicación.

### B.1. Historias de Usuario

A continuación se mostrará una tabla con las diferentes historias de usuario que se han tenido en cuenta.

Tabla B.1: Historias de usuario

ID	Enunciado de la historia		
	Rol	Característica - Funcionalidad	Razón - Resultado
10000	Jugador	Juego de preguntas y respuestas con 10 preguntas	ÉPICA
10001	Jugador	Comprobar si la respuesta es correcta	Saber si es la respuesta correcta
10002	Jugador	Comprobar si la respuesta no es correcta	Saber si el usuario ha cometido un error
10003	Jugador compulsivo, Jugador social	Cálculo de puntuación obtenida en una pregunta si la respuesta es correcta	Conocer la puntuación obtenida
10004	Jugador compulsivo, Jugador social	Cálculo de la puntuación acumulada a lo largo de las preguntas	Conocer la puntuación total de todas las preguntas
10005	Jugador social	Compartir la puntuación total obtenida a lo largo del juego	Competir con mis amigos o comparar con mis puntuaciones anteriores
10006	Jugador	Detener la partida en cualquier momento	Poder volver al menú o detener la partida en cualquier momento
10007	Jugador compulsivo, Jugador social	Volver a jugar una partida cuando haya terminado la anterior	Continuar jugando
10008	Jugador	Seleccionar el nivel de dificultad de las preguntas	Ajustar mi nivel de conocimiento al juego
10009	Jugador	Seleccionar la dificultad fácil	Para responder a preguntas de menor dificultad
10010	Jugador	Seleccionar la dificultad media	Para responder a preguntas de dificultad intermedia
10011	Jugador	Seleccionar la dificultad difícil	Para poder responder a cualquier tipo de preguntas
10012	Jugador	Poder responder a distintos tipos de preguntas	Ampliar la variedad de preguntas
10013	Jugador educativo, Padre jugador	Poder responder a preguntas sobre geografía política de Aragón	continuar con el juego

— Continúa en la página siguiente —

Tabla B.1: Historias de usuario

— Viene de la página anterior —

ID	Enunciado de la historia		
	Rol	Característica - Funcionalidad	Razón - Resultado
10014	Jugador educativo, Padre jugador	Responder a preguntas identificando una ciudad mediante una foto o seleccionar la foto que se corresponda a una ciudad	continuar con el juego
20000	Jugador	Juego de geolocalización	ÉPICA
20001	Jugador turístico	Facilitar información sobre el siguiente lugar que se debe visitar en el juego de geolocalización	Conocer el próximo lugar que debe visitarse
20002	Jugador turístico	Facilitar una lista con municipios en los que se puede jugar al juego de geolocalización	Jugar en el municipio que más se ajuste a mis deseos
20003	Jugador	Informar de la posición geolocalizada	Saber si es el lugar correcto en el juego de geolocalización
20004	Jugador	Comprobar si la posición es correcta	Saber si se debe continuar al siguiente lugar
20005	Jugador	Informar de la finalización del juego de geolocalización tras visitar todos los lugares correspondientes a ese municipio	Conocer el final del juego
20006	Jugador	Finalizar la partida de geolocalización	Poder salir de la partida en curso
20007	Jugador	Guardar la partida	Continuar en otro momento
20008	Jugador	Continuar partida guardada	Continuar con la última partida guardada
20009	Jugador	Jugar en modo inmóvil	Jugar sin moverme de mi localización
20010	Jugador	Seleccionar posición en un mapa	Para indicar la posición de la pista
30000	Estudiante	Sección de aprendizaje	ÉPICA
30001	Estudiante	Introducir un municipio	Buscar información sobre el municipio

— Continúa en la página siguiente —

Tabla B.1: Historias de usuario

— Viene de la página anterior —

ID	Enunciado de la historia		
	Rol	Característica - Funcionalidad	Razón - Resultado
30002	Estudiante sobre municipios de Aragón	Facilitar información sobre un municipio	Ampliar los conocimientos sobre el municipio
30003	Estudiante sobre municipios de Aragón	Botón para recurrir a un tercero	Ampliar la información sobre un municipio
30004	Estudiante sobre municipios de Aragón	Geolocalización del dispositivo	Para facilitar información directamente del lugar en el que se encuentra el usuario
40000	Usuario	Crear un juego propio de geolocalización	Retar a is amigos
40001	Usuario	Guardar mi localización actual	Crear un punto del juego de geolocalización
40002	Usuario	Guardar descripción del juego	Crear la pista para encontrar el lugar
40003	Usuario	Enviar partida de geolocalización	Retar a mis amigos
40004	Usuario	Jugar una gymkana en modo carrera	Competir con el resto de mis amigos
40005	Usuario	Invitar a participar a usuarios en una gymkana en modo carrera	Participan las personas a las que yo quiero invitar
40006	Usuario	Aceptar o rechazar una invitación a una gymkana modo carrera	Participar en una gymkana compitiendo con otros usuarios
40007	Usuario	Pantalla de gestión de Mis Gymkanas	Para poder decidir cómo gestionar mis gymkanas
40008	Usuario	Publicar una gymkana que haya creado	Para poder compartirla con mis amigos
40009	Usuario	Eliminar una gymkana	Para poder eliminar las gymkanas que ya no utilice
50000	Usuario	Notificación de petición de amistad	Para agregar amigos en mi aplicación
50001	Usuario	Notificación si se supera tu puntuación	Para poder mejorar la puntuación de mis amigos

— Continúa en la página siguiente —

Tabla B.1: Historias de usuario

— Viene de la página anterior —

ID	Enunciado de la historia		
	Rol	Característica - Funcionalidad	Razón - Resultado
50002	Usuario	Notificación de invitación a una carrera	Para competir contra mis amigos en una gymkana
50003	Usuario	Notificación de avances de la carrera	Para conocer el avance de mis amigos en una carrera de gymkana o si alguno ya ha terminado y ha ganado
50004	Usuario	Notificación de cancelación de carrera	Para saber si una carrera en la que participo ha sido cancelada
60000	Usuario	Crear una cuenta personal	Para poder participar en el juego
60001	Usuario	Añadir un nombre de usuario	Identificarme en el juego
60002	Usuario	Añadir una contraseña	Para que nadie juegue con mi cuenta
60003	Usuario	Añadir dirección de email	Para poder recuperar la contraseña si la olvido
70000	Usuario	Pantalla de gestión de cuenta	Para modificar los datos de mi cuenta
70001	Usuario	Cambiar contraseña	Para utilizar una contraseña nueva
70002	Usuario	Gestionar mis amigos	Para decidir mis amistades
70003	Usuario	Buscar otros amigos en la aplicación (por alias o por email)	Para poder localizar a mis amigos
70004	Usuario	Eliminar amigos	Para dejar de competir, recibir notificaciones o invitaciones de estas personas
70005	Usuario	Gestionar mis invitaciones	Para decidir que invitaciones acepto o rechazo
70006	Usuario	Aceptar/Rechazar invitaciones de amistad	Para aceptar/rechazar invitaciones de amistad
70007	Usuario	Aceptar/Rechazar invitaciones a carreras	Para aceptar/rechazar invitaciones a carreras
70008	Usuario	Información de usuario	Para comprobar la información de un usuario

— Continúa en la página siguiente —

Tabla B.1: Historias de usuario

— Viene de la página anterior —

ID	Enunciado de la historia		
	Rol	Característica - Funcionalidad	Razón - Resultado
80000	Usuario	Comprobar la clasificación de jugadores	Para conocer las mejores puntuaciones de los jugadores
80001	Usuario	Comprobar la clasificación de mis amigos	Para conocer las mejores puntuaciones de mis amigos
80002	Usuario	Gestionar mis preferencias	Para modificar las opciones de la aplicación
80003	Usuario	Iniciar/Detener reproducción de sonido	Para iniciar/dejar de escuchar los sonidos de la aplicación
80004	Usuario	Iniciar/Detener vibración	Para iniciar/detener la vibración de la aplicación
80005	Usuario	Mostrar/Ocultar email	Para mostrar/ocultar mi email en la información que ven los demás usuarios
80006	Usuario	Recibir/Bloquear notificaciones	Para seguir recibiendo o no las notificaciones de la aplicación

## B.2. Requisitos

De las anteriores historias de usuario se pueden deducir los siguientes requisitos que debe cumplir la aplicación.

### B.2.1. Requisitos Funcionales

**RF-1** Gestión de usuarios

**RF-2** Gestión de Gymkanas

**RF-3** Una Gymkana estará formada por un título y un número de localizaciones entre 5 y 10.

**RF-4** Una Localización estará formada por una posición GPS, un nombre y tres pistas que la identifiquen.

**RF-5** Posibilidad de participar en un juego de preguntas y respuestas.

**RF-6** Posibilidad de seleccionar el nivel de dificultad.

**RF-7** Establecer una puntuación por cada pregunta correctamente respondida.

**RF-8** Dar la puntuación final de la partida al usuario.

**RF-9** Establecer un sistema de clasificación de los usuarios con los puntos obtenidos en los juegos de preguntas y respuestas.

**RF-10** Posibilidad de consultar los diez primeros puestos de la clasificación de jugadores o de amigos.

**RF-11** Posibilidad de participar en una Gymkana.

**RF-12** Comprobación de la posición del usuario para determinar si se corresponde con la localización buscada en la Gymkana.

**RF-13** Información sobre la finalización de una Gymkana.

**RF-14** Guardar una gymkana para ser continuada más adelante

**RF-15** Dar la posibilidad de invitar a amigos del usuario a participar en una carrera.

**RF-16** Una carrera es similar a una gymkana, sin embargo participan varias personas al mismo tiempo y el primero en visitar todas las localizaciones es declarado el ganador de la carrera.

**RF-17** Determinar cuando una carrera ha sido finalizada.

- RF-18** Posibilitar la cancelación de una carrera.
- RF-19** Jugar a una partida de gymkanas sin necesidad de desplazarse.
- RF-20** Comparar la ubicación seleccionada en un mapa con la posición de una localización.
- RF-21** Consultar la información de un municipio introducido por el usuario.
- RF-22** Consultar la información del municipio en el que se encuentre el usuario.
- RF-23** Consultar la información de la cuenta del usuario.
- RF-24** Modificar la configuración de la cuenta del usuario.
- RF-25** Modificar la contraseña del usuario.
- RF-26** Ver el listado de amigos.
- RF-27** Buscar a un jugador.
- RF-28** Enviar a un jugador una petición de amistad.
- RF-29** Gestionar las invitaciones de amistad de otros usuarios.
- RF-30** Gestionar las invitaciones para participar en carreras.

### **B.2.2. Requisitos no Funcionales**

- RNF-1** Creación de una aplicación móvil para la plataforma Android, desarrollada en lenguaje nativo.
- RNF-2** Conexión con servicios web mediante protocolo RESTful y utilización de mensajes con formato JSON.
- RNF-3** Uso de sistemas de datos abiertos como OpenData Aragón<sup>1</sup> y datos abiertos del ayuntamiento de Zaragoza<sup>2</sup>.

---

<sup>1</sup><http://opendata.aragon.es/>

<sup>2</sup><https://www.zaragoza.es/ciudad/risp/>



# Apéndice C

## Manual de usuario

En este apéndice se indicarán las instrucciones para el uso de la aplicación ExplorAragon, desarrollada en este PFC.

### C.1. Inicio

En primer lugar, la instalación de la aplicación se realizará desde el mercado de aplicaciones Android, Google Play Store, disponible en todos los dispositivos Android. Para instalar la aplicación, se solicitará al usuario que acepte los permisos necesarios para la correcta ejecución de la aplicación. Una vez finalizada la instalación, tan solo habrá que pulsar el icono de ExplorAragon y se iniciará.

En primer lugar se cargará la pantalla de *splash*, en la que se cargarán o actualizarán los datos necesarios de la aplicación. Seguidamente se abrirá una pantalla de *login*. Si ya dispones de una cuenta de ExplorAragon, tan sólo habrá que introducir el nombre de usuario y la contraseña. En caso de no disponer de una cuenta, existe un botón que te permitirá acceder a la pantalla de creación de cuentas.

Para crear un nuevo usuario se solicitará un nombre de usuario, que no se encuentre en uso; una dirección de correo, con la que no se haya creado otra cuenta previamente; y una contraseña con la verificación de la misma, que permitirán al usuario un acceso seguro.

Una vez garantizado el acceso del usuario a la aplicación, se accederá al menú principal de la misma, desde dónde se podrá acceder a cada una de las tres secciones principales que la integran: Juegos, Aprendizaje y Mi cuenta.

### C.2. Juegos

En la sección Juegos, encontrarás dos pestañas, la primera hace referencia a los juegos de preguntas y respuestas y la segunda a las gymkanas.

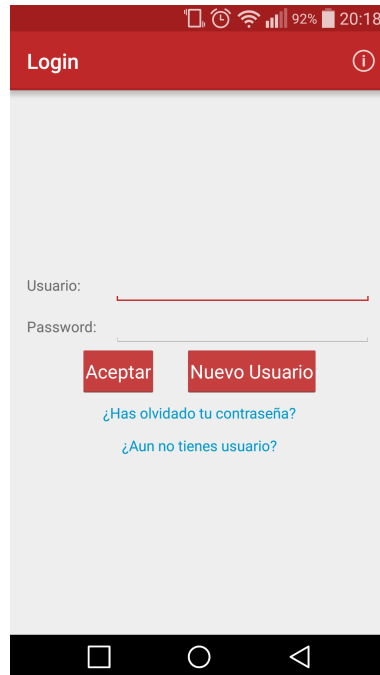


Figura C.1: Página principal de ExplorAragón

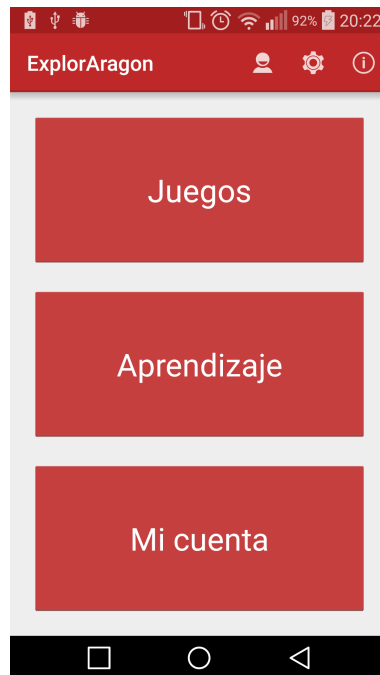


Figura C.2: Menú principal de ExplorAragón

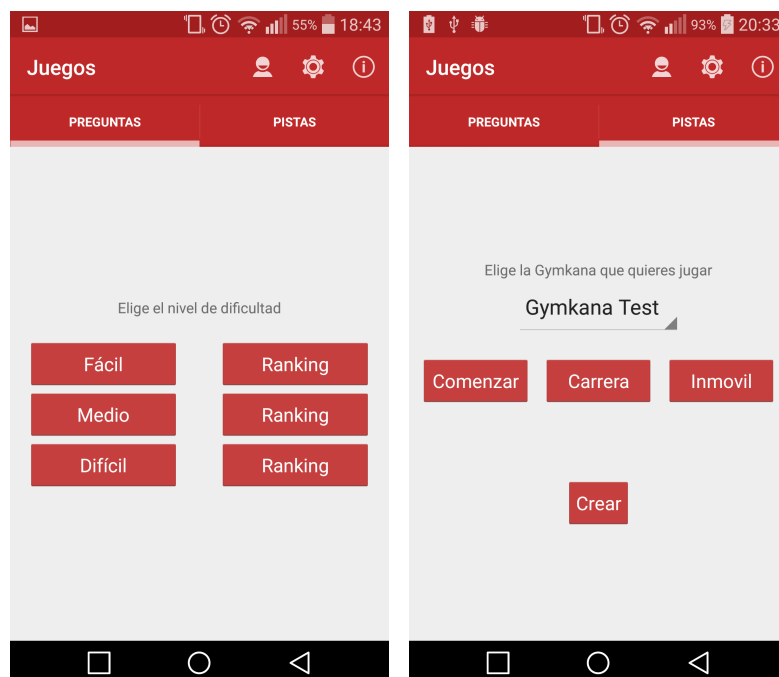


Figura C.3: Menús de juegos de ExplorAragón

### C.2.1. Preguntas y respuestas

El juego de preguntas y respuestas dispondrá de tres dificultades. Para participar en una partida de preguntas y respuestas, tan sólo hay que elegir el nivel de dificultad. La dificultad marca el tipo de preguntas que se realizarán y el número de habitantes de los municipios sobre los que se realizarán preguntas.

Una vez seleccionado un nivel de dificultad, se mostrará una pregunta y 3 ó 4 opciones, así como el número de la pregunta actual y la puntuación obtenida hasta ese momento. También dará comienzo una cuenta atrás que determinará el tiempo que tiene el jugador para responder la pregunta. En caso de no responder en el tiempo establecido, se considerará que la pregunta ha sido fallada y se concederán 0 puntos. Si el jugador escoge una respuesta incorrecta, también se le sumarán 0 puntos. Si el usuario elige la respuesta correcta, se calculará su puntuación en función del tiempo restante.

Una vez respondidas las diez preguntas del juego, se mostrará la puntuación final y se dará la opción de comenzar una nueva partida, volver al menú de juegos o volver al menú principal.

### C.2.2. Gymkana

Una gymkana es un tipo especial de juego. Este consiste en averiguar la localización de un lugar a través de una serie de pistas que se irán mostrando conforme pase

el tiempo. El usuario podrá escoger entre tres modalidades de juego de Gymkanas y un listado de gymkanas disponibles.

## **Gymkana**

La modalidad estándar del juego de gymkana mostrará, como se ha descrito previamente, una serie de pistas al usuario. Este deberá desplazarse físicamente hasta el lugar que crea se corresponde con la localización buscada. Una vez allí pulsará el botón de comprobar localización y la aplicación determinará si su posición geográfica es la correcta. De ser así, se procederá con la siguiente localización, hasta que complete todas las localizaciones de la gymkana, momento en el que terminará la partida.

En cualquier momento, el jugador puede guardar sus avances y continuar en otro momento con este tipo de partida. Sin embargo, no podrá comenzar otra sin terminar previamente la partida guardada.

## **Inmóvil**

La modalidad inmóvil es similar a la previamente descrita, sin embargo, en este caso, al jugador se le mostrará un mapa. El usuario ya no tendrá que desplazarse hasta el lugar sobre el que se formulan las pistas, sino que seleccionará la posición aproximada en el mapa. Este modo de juego no contabiliza como gymkana finalizada en las estadísticas del usuario.

## **Carrera**

La última de las modalidades es la carrera. Para ello se precisa más de un jugador y que uno de los usuarios sea el anfitrión. Este usuario invitará a una serie de jugadores de entre su lista de amigos, quienes decidirán si desean participar o no en la carrera. Cuando el anfitrión decida que hay suficientes jugadores podrá dar comienzo a la carrera. Esta se desarrollará de forma similar a un juego de Gymkana, sin embargo, el primer jugador en completar todas las localizaciones será considerado el ganador de la carrera, por lo que los jugadores deberán competir por ser el más rápido en completar todas las localizaciones.

## **C.3. Aprendizaje**

En la sección de aprendizaje podrás consultar datos sobre diferentes municipios de Aragón. En el cuadro de texto sólo tendrás que comenzar a introducir el nombre del municipio que quieras consultar, la aplicación te mostrará una serie de municipios

sugeridos y presionar el botón de la lupa. En poco tiempo, se mostrarán todos los datos del municipio que hayas buscado: Nombre, fotografía, provincia, comarca, población, nombre del alcalde, área en kilómetros cuadrados y densidad de población.

Si esperas unos segundos, verás que a la izquierda del cuadro de texto aparecerá un segundo botón. Este te localizará geográficamente y cargará la información del municipio en el que te encuentres.

## C.4. Mi cuenta

Podrás consultar datos relacionados con tu perfil en la sección *Mi cuenta*. Esta se dividirá en cuatro pestañas con diferente información.

### C.4.1. Datos del usuario

La primera de las pestañas mostrará tus datos, como tu nombre de usuario, tu dirección de correo, la mejor puntuación obtenida en juegos de tipos preguntas y respuestas, el número de gymkanas finalizadas y el número de carreras ganadas. Como observación, las gymkanas de tipo inmóvil no se contabilizarán como gymkanas finalizadas.

### C.4.2. Amigos

La segunda de las pestañas listará los amigos que actualmente tiene el usuario. Al seleccionar cada uno de tus amigos, podrás realizar diferentes acciones como silenciar al usuario, para no recibir notificaciones de este o cancelar la amistad.

En esta sección también podrás buscar a otras personas para añadirlas a tu lista de amigos.

### C.4.3. Invitaciones

Podrás consultar dos listas de invitaciones, la primera relacionada con las peticiones de amistad recibidas y la segunda con las carreras a las que te han invitado y aun puedes inscribirte. Para responder a una de las invitaciones, tan solo tienes que seleccionarla y accederás a una pantalla con un breve resumen de la información relacionada con la misma. Si se trata de una petición de amistad, verás la información del usuario que pide ser tu amigo. Si se trata de una invitación a una carrera, se verá el nombre de la gymkana y una breve descripción de la misma. En la pantalla dispondrás de dos botones, uno para aceptar la invitación y otro para rechazarla.

#### C.4.4. Gymkanas

En esta última pestaña, podrás gestionar las gymkanas que hayas creado o que estés en proceso de creación. Tendrás una lista con tus gymkanas y al seleccionar una podrás acceder al menú de creación o modificación de gymkanas.

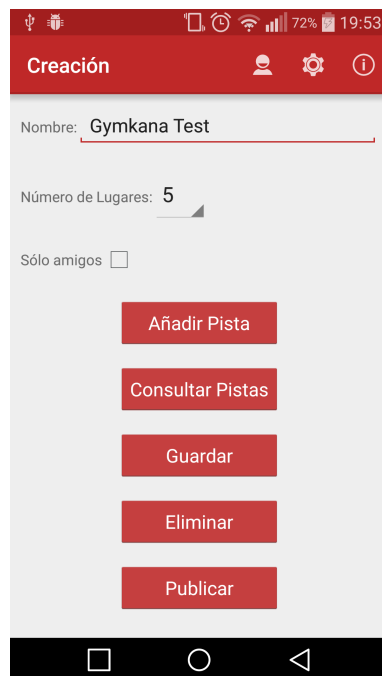


Figura C.4: Menú de creación o modificación de Gymkanas de ExplorAragón

Al crear una gymkana tendrás que rellenar una serie de datos. En primer lugar, deberás darle un nombre que la identifique sobre otras gymkanas. También deberás elegir el número de localizaciones que recorrer en tu gymkana. Estas pueden variar entre 5 como mínimo y 10 como máximo. En tercer lugar, podrás indicar que la gymkana sólo sea visible para aquellos usuarios que se encuentren en tu lista de amigos.

Seguidamente podrás añadir pistas o consultar las ya existentes. Para ello accederás a la pantalla de edición de localización.

Para añadir una nueva localización habrá que asignarle un nombre y tres pistas que ayuden a encontrarla. También deberás indicar en el mapa la posición aproximada, que permita al usuario geolocalizar la localización solución.

Una vez hayas añadido las localizaciones indicadas, de 5 a 10, como se ha dicho antes, podrás publicar la gymkana en el servidor. Hasta que no se cumplan los requisitos de número de localizaciones, sólo podrás guardar la gymkana temporalmente en tu dispositivo móvil y los demás usuarios no podrán disfrutar jugando tu gymkana. La última de las opciones disponible será el borrado de la gymkana del servidor, esto hará que la gymkana ya no sea accesible por los demás usuarios.

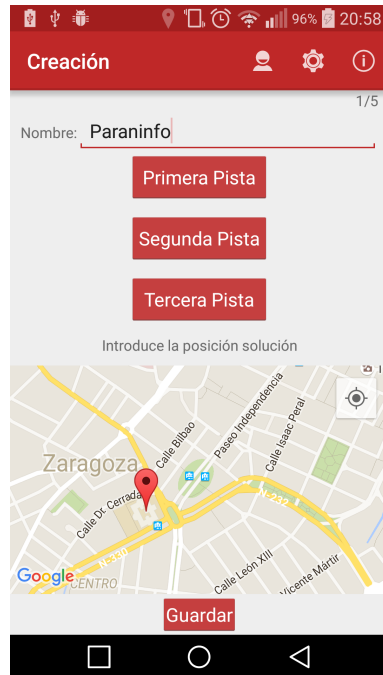


Figura C.5: Pantalla de información de localización de ExplorAragón

## C.5. Menú superior

Desde el menú superior se pueden acceder a diferentes opciones de la aplicación. El primero de los botones permitirá acceder al perfil del usuario ubicado en la sección *Mi cuenta*.

El segundo de los botones dará acceso a una serie de opciones del usuario, que le permitirán activar o desactivar diferentes opciones como el sonido, la vibración, mostrar el email del usuario o recibir notificaciones cuando se reciben invitaciones o para conocer el estado de una carrera.

El último de los botones mostrará información sobre el desarrollador de la aplicación.

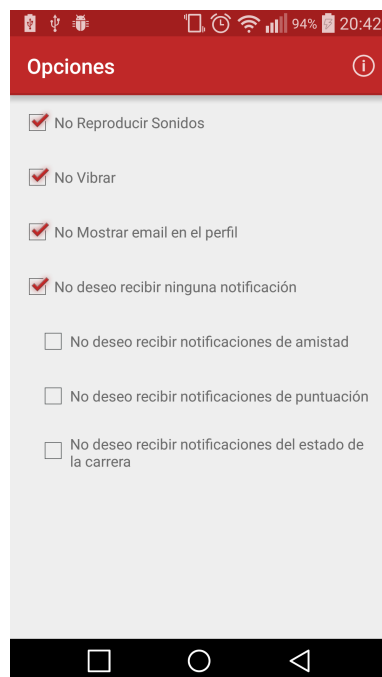


Figura C.6: Pantalla de preferencias del usuario de ExplorAragón



# Apéndice D

## API del servidor de ExplorAragon

Ya que la API desarrollada es bastante compleja, se presenta en este apéndice un breve resumen de los servicios web disponibles. Antes de comenzar el listado de los servicios, cabe indicar que todas las llamadas poseen una cabecera en la que se informará sobre el nombre de usuario y un código que se corresponderá con la encriptación mediante SHA256 de la contraseña del usuario. De este modo, evitamos que otras personas puedan hacerse con la contraseña de ese usuario.

### D.1. Servicios disponibles

A continuación se listarán los servicios disponibles con una breve descripción de la información facilitada en dicho servicio, para información más detallada se sugiere consultar la API publicada en el servicio de ApiAry <http://docs.exploraragon.apiary.io/>.

#### **/rest/users**

**GET** Recupera información sobre un usuario. Este será identificado mediante la cabecera de autenticación, por lo que este servicio nos servirá para realizar la entrada de un usuario en el sistema.

**POST** Añadiendo un objeto con la información de un usuario permitirá la creación de este. Este servicio no poseerá cabecera de autenticación.

**PUT** Actualiza información de un usuario.

#### **/rest/users/settings**

**GET** Recuperará la configuración de un usuario.

**PUT** Modificará la configuración de un usuario.

**/rest/users/all**

**GET** Recuperará un listado con todos los usuarios

**/rest/invitations**

**GET** Recuperará un listado de las invitaciones pendientes de un usuario, ya sean peticiones de amistad o participaciones en carreras.

**POST** Lanzará una invitación a un usuario. En el objeto enviado se determinará si se trata de una invitación de amistad o de participación en una carrera.

**PUT** Responde a una invitación.

**/rest/friends**

**GET** Recupera el listado de amigos del usuario.

**/rest/friends?<friendship\_id>**

**DELETE** Elimina la amistad identificada por el parámetro de la dirección *<friendship\_id>*.

**/rest/cities**

**GET** Recupera un listado con los nombres de todos los municipios.

**/rest/cities/<city\_name>**

**GET** Recupera la información detallada del municipio indicado en *<city\_name>*.

**/rest/trivia**

**PUT** Genera una nueva pregunta con los parámetros indicados en el objeto enviado.

**POST** Guarda el resultado de una partida de tipo trivia

**/rest/trivia/ranking?friend**

**GET** Recupera el listado de las diez mejores puntuaciones de los usuarios. Si la variable optativa de la dirección *friend* tiene valor *true*, se enviará únicamente la puntuación de los amigos del usuario.

**/rest/gymkanas**

**GET** Recuperará un listado con los nombres de todas las Gymkanas y su identificador.

**PUT** Creará o actualizará una gymkana creada por un usuario.

**/rest/gymkanas/<gymkana\_id>**

**GET** Recupera la información completa de la gymkana identificada mediante *<gymkana\_id>*.

**DELETE** Elimina la gymkana identificada por la variable *<gymkana\_id>*.

**/rest/gymkanas/id/<gymkana\_id>?<gymkana\_type>**

**PUT** Comienza una partida gymkana de la gymkana indicada por *<gymkana\_id>* y del tipo indicado por *<gymkana\_type>*.

**/rest/gymkanas/complete/<gymkana\_id>?<gymkana\_type>**

**PUT** Finaliza una partida gymkana identificada por la gymkana *<gymkana\_id>* y del tipo *<gymkana\_type>*, junto con la cabecera de identificación del usuario.

**/rest/gymkanas/user**

**GET** Recupera todas las gymkanas del usuario.

**/rest/gymkanas/locations/<location\_id>**

**GET** Recupera la información de la localización indicada por la variable *<location\_id>*.

**PUT** Actualiza la información de la localización identificada por la variable *<location\_id>*.

## D.2. Formato de la respuesta de error

El servidor devolverá diferentes formatos de respuesta, en función de la información que se espere. Sin embargo, en caso de darse un error, el servidor responderá siempre con el mismo formato compuesto por dos elementos.

El primero de ellos será *errorCode*. Este tendrá un formato numérico e indicará el tipo de error, correspondiente con los códigos de estado HTTP, que se ha dado al tratar de ejecutar la petición. Por ejemplo, 400 cuando se ha realizado una petición incorrecta. El listado de los códigos de estado HTTP y lo que significan están disponibles en la Wikipedia<sup>1</sup>.

El segundo de los elementos será *errorMessages*. Este será un listado de cadenas que dará información a la aplicación para que notifique al usuario del tipo de error que se ha producido.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)