



TRABALLO DE FIN DE GRADO
GRADO EN ENXEÑERÍA INFORMÁTICA
Mención en Sistemas de Información

Aplicación móvil para actividades a campo abierto

Alumno: Alberto Ramil Fernández
Directores: Javier Parapar López
Óscar Pedreira Fernández

A Coruña, 2 de enero de 2018

Datos del trabajo

Título del Trabajo:	Aplicación móvil para actividades a campo abierto
Clase a la que pertenece:	Trabajo Clásico de Ingeniería
Alumno:	Alberto Ramil Fernández
Directores:	Javier Parapar López Óscar Pedreira Fernández
Miembros del tribunal:	
Fecha de lectura y defensa:	A Coruña, 2 de enero de 2018
Calificación:	

JAVIER PARAPAR LÓPEZ
Profesor de Universidad
Departamento de Computación
Universidade da Coruña

y

ÓSCAR PEDREIRA FERNÁNDEZ
Profesor Predoctoral FPU
Departamento de Computación
Universidade da Coruña

CERTIFICAN: Que la memoria titulada *Aplicación móvil para actividades a campo abierto* ha sido realizada por ALBERTO RAMIL FERNÁNDEZ bajo su dirección y constituye su Trabajo de Fin de Grado en el Grado de Enxeñería Informática.

En A Coruña, a 2 de enero de 2018

JAVIER PARAPAR LÓPEZ
Director

ÓSCAR PEDREIRA FERNÁNDEZ
Director

Dedicatoria

Agradecimientos

I would like to thank...

Author name

A Coruña, 2 de enero de 2018

Resumen

This is the summary of the project. No more than one page, please.

Palabras clave

FIRST KEYWORD, SECOND KEYWORD, ETC.

Índice general

1. Introducción	5
1.1. Motivación	5
1.2. Objetivos	5
1.3. Estructura de la memoria	5
1.4. Plan de trabajo	6
2. Conceptos	7
3. Tecnología	9
4. Proceso de ingeniería	11
5. Desarrollo	13
5.1. Análisis de Requisitos	13
5.1.1. Actores	14
5.1.2. Requisitos funcionales	14
5.1.3. Requisitos no funcionales	16
5.2. Arquitectura propuesta	16
5.2.1. Esquema general de la arquitectura	17
5.2.2. Diagrama de despliegue	17
5.3. Componente A	17
5.3.1. Análisis	17
5.3.2. Diseño e implementación	18

6. Conclusiones y trabajo futuro	19
6.1. Investigación realizada	19
6.2. Características del proyecto	19
6.3. Trabajo futuro	20
Índice de Tablas	21
Índice de Figuras	23
Apéndices	25
A. Glosario	25
B. Bibliografía	29

Capítulo 1

Introducción

Intro

1.1. Motivación

1.2. Objetivos

- Y
- X

1.3. Estructura de la memoria

La memoria del presente proyecto está estructurada del siguiente modo:

- **Introducción** Explica el contexto en el que se enmarca el proyecto, introduce la problemática a tratar y detalla el alcance y objetivos del mismo desde un punto de vista global. También muestra la estructura de la memoria y el plan de trabajo seguido.
- **Tecnología** Describe y justifica las principales tecnologías empleadas para desarrollar el objeto del proyecto atendiendo a los requisitos del mismo.
- **Conceptos**

- **Proceso de ingeniería** Detalla el proceso de ingeniería: la metodología, la planificación y la gestión del proyecto.
- **Desarrollo** Realiza una descripción detallada del análisis, diseño, implementación, pruebas y despliegue del sistema.
- **Conclusiones y trabajo futuro** Proporciona una evaluación global del producto obtenido así como futuras líneas de trabajo que se podrían explotar alrededor del proyecto.
- **Apéndices** Está compuesto por las siguientes secciones complementarias:
 - **Glosario** Define los términos y acrónimos técnicos empleados en la memoria del proyecto.
 - **Bibliografía** Recoge la documentación bibliográfica sobre la que se apoya el proyecto.

1.4. Plan de trabajo

Capítulo 2

Conceptos

Capítulo 3

Tecnología

La selección de la tecnología adecuada es una condición necesaria, aunque no suficiente, para llevar a cabo un proyecto con éxito. En este capítulo se detallarán las elecciones tecnológicas y se justificarán debidamente.

Este capítulo contiene una sección para explicar las tecnologías de cada uno de los componentes. Adicionalmente, se incluye un apartado donde se describen las herramientas utilizadas para dar soporte a la gestión y desarrollo del proyecto.

Con estas tecnologías se ha llevado a cabo la construcción de los distintos componentes del sistema siguiendo los pasos que comentaremos en el próximo capítulo.

Capítulo 4

Proceso de ingeniería

En este capítulo se justifica y se describe la metodología de desarrollo sobre la que se apoya el proceso de ingeniería. En el caso del presente proyecto se ha optado por emplear una aproximación ágil al proceso unificado de desarrollo. Se estudiarán también el resto de aspectos relacionados con el proceso de ingeniería como son el ciclo de vida, la descomposición en tareas, la organización, la estimación de costes y duración, la gestión de proyecto, la planificación y la gestión de riesgos.

Capítulo 5

Desarrollo

En este capítulo se detalla el proceso de desarrollo del proyecto. En primer lugar, se analizarán los requisitos del proyecto para definir la arquitectura general del sistema. Luego se describirán el análisis, el diseño y la complementación de los componentes. Por último, se mostrarán datos ofrecidos por las herramientas de soporte al desarrollo.

5.1. Análisis de Requisitos

En esta aplicación móvil para actividades a campo abierto objetivo de este proyecto, se establecieron una serie de requisitos generales que debería cumplir la aplicación.

El registro del usuario para poder comenzar a usarla.

La creación de puntos de interés marcados en un mapa con nombre, descripción y un punto en el mapa con o sin señal GPS clasificados en tipo, caza o pesca.

Se podrán guardar las rutas seguidas por un usuario en sus caminatas por cualquier tipo de terreno.

El usuario también podrá crear grupos con los usuarios que quiera y los integrantes del mismo poder añadir a otros, el resultado de esta funcionalidad es la que permitirá posteriormente crear rutas conjuntas. Para crear una ruta conjunta primero se

elige el grupo del que se hará el seguimiento y se enviarán las invitaciones para participar en él a cada integrante del grupo. Estas invitaciones en el caso de ser aceptadas llevarán al usuario a un mapa y periódicamente se irán realizando actualizaciones de las posiciones del resto de integrantes del grupo anteriormente indicado. Finalmente se podrán ver las rutas conjuntas igual que las individuales.

5.1.1. Actores

Los únicos actores que se presentan en la aplicación son los siguientes:

- **Usuario no autenticado.** Usuario que no está autenticado en la aplicación y que se le permite registrarse en el sistema o iniciar sesión si la ya se registro en otro momento.
- **Usuario autenticado.** Usuario autenticado que puede acceder a todas as funcionalidades del sistema.

5.1.2. Requisitos funcionales

A continuación, en esta sección, se exponen los requisitos funcionales que surgen de los requisitos generales planteados en el punto anterior.

• Usuario no autenticado

- **R1 Registrarse en la aplicación.** El usuario podrá darse de alta en el sistema introduciendo sus datos en el formulario que se le indican. Una vez registrado se iniciará sesión automáticamente con el nuevo perfil.
- **R2 Iniciar sesión en la aplicación.** El usuario ya registrado podrá, con sus credenciales, autenticarse en el sistema. Se pedirá o nombre del usuario la aplicación y su contraseña. Se guardará el estado en el terminal hasta que el usuario decida desconectarse.

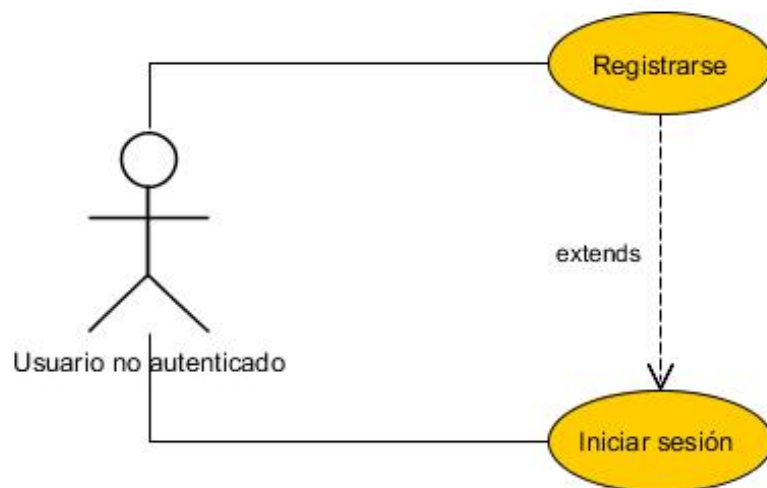


Figura 5.1: Casos de uso del actor Usuario No Autenticado

- **Usuario autenticado**

Para hacer un poco más comprensible dividiré los casos de uso del usuario autenticado por grupos funcionales.

- Gestión de puntos de interés
 - R-PDI-1 Guardar Punto De Interés caza
 - R-PDI-2 Guardar Punto De Interés pesca
 - R-PDI-3 Eliminar PDI
 - R-PDI-4 Buscar los PDI
- Gestión de grupos
 - R-G-1 Crear grupo
 - R-G-1.1 Añadir integrantes
 - R-G-1.2 Eliminar integrantes
 - R-G-1.3 Ver grupos
 - R-G-1.4 Ver integrantes grupo
- Gestión de rutas

- R-R-1 Crear ruta privada
 - R-R-1.1 Iniciar ruta
 - R-R-1.2 Parar ruta
 - R-R-1.3 Guardar ruta
- R-R-2 Crear ruta compartida
 - R-R-2.1 Iniciar ruta
 - R-R-2.2 Parar ruta
 - R-R-2.3 Finalizar ruta
- R-R-3 Listar rutas
- R-R-4 Ver ruta en mapa

5.1.3. Requisitos no funcionales

Por otro lado, no debemos olvidarnos de los requisitos no funcionales. Estos son aquellos que especifican criterios a cumplir por el sistema (en lugar de funciones específicas como indican los requisitos funcionales). Estos dan lugar a decisiones de diseño en la arquitectura y sus componentes así como en las elecciones tecnológicas detalladas en la Sección 3. A continuación se recogen los requisitos no funcionales educidos tras el proceso de ingeniería de requisitos.

-
-
-

5.2. Arquitectura propuesta

En base a los requisitos enunciados en la sección anterior se propone una arquitectura basada en

Componente A

Componente B

Componente C

5.2.1. Esquema general de la arquitectura

El resultado del proceso de ingeniería seguido ha dado lugar al esquema general de la arquitectura del sistema que se ilustra a continuación en la Figura 5.2. En este se pueden apreciar las características arquitectónicas de cada uno de los componentes de la plataforma.

Figura 5.2: Esquema general de la arquitectura del sistema

5.2.2. Diagrama de despliegue

Una vez creada la arquitectura, hay que determinar las elecciones tecnológicas que dan soporte a la misma. Dicho estudio se encuentra recogido en el Capítulo 3. La configuración del despliegue de la aplicación se recoge en la Figura 5.3.

Figura 5.3: Diagrama de despliegue del sistema

5.3. Componente A

En esta sección se estudiará el componente A...

5.3.1. Análisis

Este componente del sistema da respuesta al requisito «R? xxx»

Casos de uso

En base a los requisitos educidos, se exponen los casos de uso de este subsistema.

En la Figura 5.4 se ilustra el diagrama de casos de uso correspondiente al componente A.

Figura 5.4: Diagrama de casos de uso relativos al componente A

Modelo de datos

5.3.2. Diseño e implementación

En esta sección se presenta el diseño seguido para implementar el motor de recomendación. Se empleará el lenguaje de modelado UML para ilustrar la estructura de este subsistema.

Capítulo 6

Conclusiones y trabajo futuro

Bla, bla, bla...

6.1. Investigación realizada

-
-

6.2. Características del proyecto

El uso de los paradigmas adecuados junto con herramientas consolidadas nos ha permitido construir una plataforma en la que servir una aplicación web donde los usuarios puedan buscar, valorar y obtener recomendaciones de contenidos audiovisuales.

A continuación se muestran las principales características del producto construido:

-
-

6.3. Trabajo futuro

Una vez finalizado este proyecto, se abren nuevas vías de trabajo, tanto de explotación comercial como de investigación científica:

-
-
-

Índice de Tablas

Índice de Figuras

5.1. Casos de uso del actor Usuario No Autenticado	15
5.2. Esquema general de la arquitectura del sistema	17
5.3. Diagrama de despliegue del sistema	17
5.4. Diagrama de casos de uso relativos al componente A	18

Apéndice A

Glosario

ACID Atomicity, Consistency, Isolation and Durability: es un conjunto de propiedades que garantizan la fiabilidad de las transacciones de una base de datos. La atomicidad implica que cada transacción se ejecuta de forma completa o no se ejecuta. La consistencia garantiza que cada transacción llevará la base de datos de un estado consistente a otro consistente. El aislamiento permite que parezca que las transacciones se ejecutan de forma serializada cuando en realidad existe concurrencia. Por último, la durabilidad garantiza que una vez finalizada con éxito una transacción sus cambios se guardarán de forma persistente.

AM Agile Modeling: una metodología complementaria para modelar y documentar sistemas software mediante una serie de buenas prácticas basadas en el desarrollo ágil.

Big Data Es el término que se utiliza para llamar a los sistemas que procesan grandes colecciones de datos que son inmanejables bajo las herramientas tradicionales.

CF Collaborative Filtering: los algoritmos de filtrado colaborativo son una familia de algoritmos de recomendación cuyo funcionamiento se basa en generar sugerencias basándose en las preferencias de otros usuarios.

HDFS Hadoop Distributed File System: es un sistema de fichero distribuido, escalable y portable escrito en java y diseñado para servir de forma de almacenamiento a Hadoop.

IR Information Retrieval: la recuperación de información es la ciencia que estudia la obtención de información relevante a partir de una colección de documentos.

-
- ML *Machine Learning*: el aprendizaje automático es una rama de la inteligencia artificial que busca construir y estudiar sistemas que sean capaces de aprender de los datos.
- MVC *Model-view-controller*: es un patrón arquitectónico para implementar interfaces de usuario. Divide el software en tres componentes. El modelo es el encargado de gestionar los datos y la lógica de negocio. La vista se encarga de la representación gráfica de la información. Por último, el controlador es el encargado de poner en comunicación los otros componentes.
- NoSQL *Not Only SQL*: clase de sistemas de gestión de bases de datos que difieren del modelo relacional y cuyo principal objetivo es proporcionar escalabilidad horizontal.
- ORM *Object-Relational Mapping*: El mapeo objeto-relacional es una herramienta de programación que establece una correspondencia entre clases y objetos de un lenguaje de programación orientado a objetos con tablas y filas de una base de datos relacional siguiendo el patrón arquitectónico *active record*.
- RDBMS *Relational DataBase Management System*: un sistema de gestión de base de datos relacional es un software diseñado para gestionar la creación, modificación y consulta de bases de datos bajo el esquema relacional.
- RecSys *Recommender Systems*: los sistemas de recomendación tienen como objetivo predecir la preferencia de un usuario hacia un producto.
- RM *Relevance Models*: forma abreviada empleada en esta memoria referirse a los *Relevance-based Language Models*, una técnica de IR que introduce el concepto de relevancia en los modelos de lenguaje estadísticos.
- RPC *Remote Procedure Call*: es un tipo de comunicación entre procesos que permite la ejecución de una rutina en un equipo remoto.
- RUP *Rational Unified Process*: se trata de la propuesta de IBM para implementar una metodología de desarrollo basada en el marco que establece el Proceso Unificado.
- SQALE *Software Quality Assessment based on Lifecycle Expectations*: es un método genérico para evaluar la calidad de un código mediante unos índices y unos indicadores.

CAPÍTULO A.

Glosario

WSGI *Web Server Gateway Interface*: es una interfaz estándar de Python para la comunicación entre servidores y aplicaciones web.

Apéndice B

Bibliografía