

# Integración de Foursquare y Geolocalización en una Aplicación Móvil para la Creación de Rutas Turísticas

---

Manuel Eduardo Sánchez Gomis

Supervisores:

Vicente Pelechano Ferragud

Pablo Muñoz Julve



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

*DSIC*  
DEPARTAMENTO DE SISTEMAS  
INFORMÁTICOS Y COMPUTACIÓN



# Agradecimientos

---

A mis padres, que siempre me han apoyado.

A Pablo, que me ha guiado y ayudado en este trabajo.

A Vicente Pelechano, por darme la oportunidad de realizar esta tesis.

A mis profesores, por darme la formación que me ha permitido llegar hasta aquí.

A Víctor y Alberto, por ayudarme con el logo y otros problemas gráficos.



# Resumen

---

El proyecto realizado en este trabajo tiene como objetivo desarrollar una aplicación que permita crear rutas turísticas a los usuarios. Esta aplicación proporciona principalmente tres funcionalidades: la **creación de una ruta turística**, su **finalización y publicación en un servidor Web** y la **descarga y el seguimiento punto a punto de una ruta**. Con este objetivo, la aplicación móvil integra varias tecnologías: **geolocalización**, **redes sociales** y **servicios Web**. Todas ellas se han desplegado en un **dispositivo móvil Android**.

La **geolocalización** es una tecnología que permite calcular la posición de un elemento. Para realizar este cálculo, se utilizan los siguientes elementos: Un dispositivo **GPS**, que calcula la localización mediante la red de satélites **GPS**, y una conexión a **internet**, que calcula la localización mediante el uso de los nodos **3G** que dan soporte a los dispositivos móviles.

Las **redes sociales** son un nuevo tipo de tecnología en la que un gran número de usuarios están conectados entre sí. Éstas se pueden agrupar por temáticas. Entre estas temáticas se encuentra la geolocalización. Una red social de geolocalización permite obtener los lugares y establecimientos cercanos a la localización de un usuario. Este trabajo está basado en la red social Foursquare, especializada en el ámbito de la geolocalización.

Los **servicios Web** proporcionan la capacidad de almacenar información para que otras aplicaciones puedan acceder a ella. Estos servicios proporcionan una serie de funcionalidades que permiten tanto almacenar la información como extraerla para su uso. En este trabajo se utiliza un servicio Web para almacenar las rutas turísticas en un servidor Web de forma que todos los usuarios puedan acceder a ellas.

El desarrollo de este trabajo pretende mostrar cómo se han integrado todas estas tecnologías para dar soporte al proceso de creación de rutas turísticas. En la aplicación, un usuario obtiene su localización mediante la geolocalización. Esta localización es transmitida a la red social Foursquare para obtener todos los lugares y establecimientos cercanos. Con esta información, el usuario puede añadir estos puntos a su ruta turística. Posteriormente, cuando la ruta se finaliza, se publica a través de un servicio Web almacenándose en un servidor Web para que otros usuarios puedan acceder a ella.

# Índice de Contenido

<b>CAPÍTULO 1: INTRODUCCIÓN .....</b>	<b>1</b>
1.1.    MOTIVACIÓN .....	1
1.2.    OBJETIVOS DE LA APLICACIÓN .....	3
1.3.    SOLUCIÓN PROPUESTA .....	4
1.4.    CONTEXTO DE LA TESIS .....	5
1.5.    ESTRUCTURA DE LA TESIS .....	6
<b>CAPÍTULO 2: CONTEXTO TECNOLÓGICO.....</b>	<b>8</b>
2.1.    REDES SOCIALES.....	8
2.2.    REDES SOCIALES BASADAS EN LA GEOLOCALIZACIÓN .....	11
Google Latitude .....	12
Facebook Places .....	13
Foursquare .....	14
Gowalla .....	15
2.3.    DISCUSIÓN.....	17
2.4.    DISPOSITIVOS MÓVILES .....	18
2.4.1.    Android .....	19
2.4.2.    Características .....	19
2.4.3.    Arquitectura.....	20
2.4.4.    Fundamentos de las Aplicaciones Android .....	21
2.4.5.    Componentes de una Aplicación.....	22
<b>CAPÍTULO 3: ESTUDIO DE APLICACIONES MÓVILES SIMILARES .....</b>	<b>25</b>
3.1.    TRIPJOURNAL .....	25
3.2.    TRIPIT .....	26
3.3.    FOURSQUARE APP.....	27
3.4.    GOWALLA.....	29
3.5.    GOOGLE LATITUDE APP .....	30
3.6.    MINUBE APP.....	31
3.7.    TRIPADVISOR APP .....	32
3.8.    CONCLUSIONES.....	33
<b>CAPÍTULO 4: INTEGRACIÓN DE FOURSQUARE, GEOLOCALIZACIÓN Y SERVICIOS WEB .....</b>	<b>35</b>
4.1.    GEOLOCALIZACIÓN .....	36
4.1.1.    Integración de la geolocalización .....	38
4.2.    FOURSQUARE .....	39
4.2.1.    OAUTH(13).....	39
4.2.2.    Integración de Foursquare.....	40
Obtención de las credenciales de Foursquare .....	40
Peticiones a Foursquare .....	42
4.3.    SERVICIOS WEB .....	43
4.3.1.    Integración del Servicio Web .....	43
4.4.    CONCLUSIONES.....	44
<b>CAPÍTULO 5: ESTRUCTURA Y FUNCIONALIDAD DEL SISTEMA .....</b>	<b>47</b>
5.1.    ESTRUCTURA DEL SISTEMA .....	48
5.1.1.    Capa de presentación .....	49

Interfaz Principal .....	51
Creación de rutas turísticas .....	54
Finalizar rutas .....	57
Descargar rutas .....	57
Interfaces adicionales.....	58
<b>5.1.2. Capa de lógica .....</b>	<b>59</b>
Creación de rutas turísticas .....	59
Finalización y publicación de las rutas en un servicio Web .....	60
Descarga de rutas turísticas para su seguimiento .....	60
Diagrama UML.....	60
<b>5.1.3. Capa de persistencia.....</b>	<b>62</b>
Diagrama Entidad-Relación .....	63
Acceso a la base de datos local: Content Provider .....	65
DBSQLiteHelper .....	66
DBProvider .....	68
Acceso al Servicio Web del sistema.....	70
Foursquare .....	70
<b>5.2. CONCLUSIONES.....</b>	<b>73</b>
<b>CAPÍTULO 6: MANUAL DE USUARIO.....</b>	<b>75</b>
<b>6.1.1. Creación de Rutas Turísticas.....</b>	<b>76</b>
Acceso a la aplicación.....	76
Añadir Puntos a la Ruta .....	78
Editar y eliminar un punto.....	81
Eliminar rutas .....	83
<b>6.1.2. Finalizar ruta.....</b>	<b>83</b>
<b>6.1.3. Descargar rutas .....</b>	<b>84</b>
Descargar rutas turísticas .....	84
Seguimiento de una ruta turística .....	84
<b>CAPÍTULO 7: CONCLUSIONES .....</b>	<b>87</b>
<b>7.1. TRABAJO FUTURO .....</b>	<b>88</b>

# Listado de Figuras

Figura 1. Representación de las redes sociales.....	8
Figura 2. Estadísticas mundiales de usuarios de internet .....	9
Figura 3. Listado de redes sociales con más usuarios registrados .....	10
Figura 4. Google Latitude .....	13
Figura 5. Contactos Facebook Places .....	14
Figura 6. Medallas Foursquare .....	15
Figura 7. Trips Gowalla .....	16
Figura 8. Arquitectura Android.....	20
Figura 9. Capturas de la aplicación TripJournal.....	26
Figura 10. Capturas de la aplicación Triplt .....	27
Figura 11. Capturas de Foursquare App .....	28
Figura 12. Capturas de la aplicación Gowalla.....	29
Figura 13. Capturas de la aplicación Google Latitude .....	30
Figura 14. Capturas de la aplicación Minube .....	32
Figura 15. Capturas de la aplicación TripAdvisor .....	33
Figura 16. Funciones de las tecnologías .....	36
Figura 17. Triangulación de los satélites GPS .....	37
Figura 18. Trilateración.....	38
Figura 19. Triangulación .....	38
Figura 20. Workflow Oauth .....	41
Figura 21. Resumen peticiones Foursquare .....	42
Figura 22. Resumen interfaces gráficas.....	49
Figura 23. Login Foursquare .....	50
Figura 24. Web Foursquare .....	50
Figura 25. Pestaña rutas .....	51
Figura 26. Pestaña mapa .....	52
Figura 27. Pestaña puntos .....	53
Figura 28. Detalles punto .....	54
Figura 29. Nueva ruta .....	54
Figura 30. Realizar “check-in” .....	55
Figura 31. Añadir localización.....	56
Figura 32. Eliminar puntos.....	57
Figura 33. Descargar rutas.....	58
Figura 34. Eliminar rutas.....	58
Figura 35. UML de la aplicación.....	61
Figura 36. Diagrama de la base de datos.....	63
Figura 37. Pantalla de bienvenida .....	77
Figura 38. Pantalla de acceso a Foursquare .....	77
Figura 39. Nueva ruta .....	78
Figura 40. Formulario nueva ruta.....	78
Figura 41. Menú contextual de un punto.....	79

Figura 42. Formulario añadir punto .....	79
Figura 43. Añadir localización.....	80
Figura 44. Formulario nueva localización .....	80
Figura 45. Filtrar Puntos .....	80
Figura 46. Menú contextual filtrar puntos .....	80
Figura 47. Puntos cercanos.....	81
Figura 48. Puntos obtenidos.....	81
Figura 49. Eliminar Puntos.....	82
Figura 50. Listado eliminar puntos .....	82
Figura 51. Editar punto .....	82
Figura 52. Formulario editar punto .....	82
Figura 53. Eliminar rutas.....	83
Figura 54. Listado eliminar rutas .....	83
Figura 55. Descargar rutas.....	84
Figura 56. Formulario descargar rutas .....	84
Figura 57. Seguimiento de una ruta .....	85

# Listado de Tablas

Tabla 1. Tabla comparativa entre redes sociales .....	17
Tabla 2. Clase usuario .....	61
Tabla 3. Clase rutas.....	62
Tabla 4. Clase checkins .....	62
Tabla 5. Tabla usuario.....	64
Tabla 6. Tabla rutas .....	64
Tabla 7. Tabla checkins .....	65

# Capítulo 1

## Introducción

---

A todo el mundo le gusta viajar, ver mundo, conocer nuevos lugares, ver ciudades desconocidas y monumentos famosos, convivir con otras culturas y observar su forma de vida. Pero para salir de viaje y poder hacer turismo, se requiere una gran cantidad de información que permita conocer en todo momento aquellas zonas y lugares que son visitados.

El trabajo realizado en esta tesis tiene como objetivo solucionar algunos de estos problemas mediante una aplicación que permite crear **rutas turísticas**. Primero, un usuario debe **crear una nueva ruta turística** añadiéndole todos los puntos que la conforman. Cuando la ruta se **finalice**, debe ser publicada mediante un **servicio Web** para que otros usuarios puedan acceder a ella. Finalmente, cualquier usuario puede **descargar esta ruta y realizar su seguimiento**.

Para realizar esto, se deben integrar las siguientes tecnologías: la **geolocalización**, que permite obtener la localización de un usuario, una **red social** especializada en geolocalización, que en función de la posición de un usuario obtiene todos los lugares y establecimientos que le rodean y los **servicios Web**, que permiten almacenar información en un servidor Web para que varias aplicaciones accedan a ella. Todos estos elementos tecnológicos deben ser integrados en un **dispositivo móvil Android**, que proporciona al usuario la posibilidad de interactuar con ellos.

Este capítulo está organizado de la siguiente manera: la primera sección introduce el conjunto de tecnologías relacionadas con el desarrollo de este trabajo. A continuación, en la siguiente sección se plantean los objetivos de este trabajo. El tercer punto, expone la solución propuesta que cubre los objetivos de la aplicación. El cuarto punto, realiza una breve descripción de aquellos conocimientos que han hecho falta para poder realizar este proyecto, así como los nuevos conocimientos obtenidos durante la realización de este. Por último, se estructura el resto del contenido de este documento.

### 1.1. Motivación

El objetivo de este trabajo es poder crear una aplicación que gestione la creación y la distribución de rutas turísticas. Para ello se deben integrar un conjunto de tecnologías. Ente estas tecnologías se encuentran los **dispositivos móviles**. Los dispositivos móviles son pequeños aparatos que hoy en día ofrecen un gran número de funcionalidades al usuario. Un ejemplo concreto es el que puede verse cada día en multitud de ciudades:

una persona se sienta a leer su correo electrónico, otra escucha música o un grupo de jóvenes se ríen mientras observan un video, funciones que solo podían realizarse con un ordenador de sobremesa y que han sido trasladadas a estos dispositivos.

Otro de los aspectos fuertes de estos dispositivos es que hoy en día tienen casi las mismas prestaciones hardware que un ordenador personal, dando una mayor capacidad de almacenamiento o de ejecución de procesos, lo cual permite ejecutar aplicaciones más pesadas y que realizan tareas más costosas.

Con todo esto podemos observar que el desarrollo de aplicaciones móviles es un campo interesante para los desarrolladores, ya que los dispositivos móviles actuales permiten acceder a un gran número de sistemas de información en cualquier momento y lugar, mediante aplicaciones que antes solo podían disponer los ordenadores de sobremesa.

Otro elemento a integrar son las **redes sociales**, un nuevo concepto tecnológico en el cual un conjunto de personas comparten información con un objetivo común, como por ejemplo ocio o trabajo y que permite recaudar en poco tiempo un gran conjunto de datos de interés para sus consumidores. Todo esto ofrece un mercado de información muy útil para las empresas, las cuales pueden desarrollar aplicaciones o campañas publicitarias mediante unos datos totalmente gratuitos.

En las redes sociales, son los mismos usuarios los que introducen la información ayudando a capturar las nuevas tendencias y modas de las personas así como poder observar la aceptación que podría tener un producto en el mercado actual antes de ser lanzado o una vez ya ha sido puesto a la venta.

Las redes sociales tienen también un papel fundamental en el desarrollo del software, proporcionando una capa de persistencia muy amplia, permitiendo integrar nuevos productos en una gran aplicación global que se visita diariamente. Entre los beneficios que se pueden obtener de estas podemos destacar:

- Proporcionan una gran base de datos constantemente actualizada por los mismos usuarios que usan la red social.
- Dan la posibilidad de integración de productos software en una macro aplicación con un gran número de usuarios.
- Permiten acceder a un gran número de servicios web de forma gratuita.
- Permiten compartir una gran cantidad de información entre los distintos usuarios.

La **geolocalización** permite mostrarnos en todo momento la posición exacta de un individuo mediante dispositivos, ya sean móviles u otros. Esto se puede realizar mediante el uso de un dispositivo GPS o de internet.

El dispositivo GPS calcula la localización de un elemento mediante la red de satélites GPS. Para ello, el dispositivo envía una señal y mediante la triangulación de tres satélites, se calcula la localización de este dispositivo.

Para calcular la localización de un individuo mediante internet, se utilizan los nodos 3G existentes. Estos nodos reciben una señal de un dispositivo, y la triangulan para obtener su localización.

Por último, los **servicios Web**, permiten a las aplicaciones compartir información entre ellas. Estos servicios proporcionan un conjunto de métodos que permiten almacenar los datos. Además de estos métodos, proporcionan otros que permiten acceder a la información almacenada.

## 1.2. Objetivos de la Aplicación

La idea principal de este proyecto es crear una aplicación que permita al usuario (1) **crear rutas turísticas**, (2) **publicarlas en un servidor Web**, y (3) **descargar rutas de otros usuarios desde este servidor Web para que éstas sean recorridas punto a punto**.

La realización de esta aplicación plantea una serie de objetivos, los cuales se exponen a continuación de manera breve:

- Necesidad de obtener un conjunto de datos para poder realizar las rutas turísticas. Estos deben estar en constante actualización para que el proyecto no quede obsoleto.
- La información obtenida anteriormente debe ser mostrada al usuario de manera clara, para que este pueda interactuar con ella sin que le resulte tedioso y complicado.
- El usuario debe poder complementar la anterior información con nueva información permitiendo ampliar los datos mostrados.
- La información introducida por el usuario debe ser almacenada para futuras consultas.
- La información almacenada por el usuario debe ser fácilmente accesible por otros usuarios.

A continuación se presenta la solución propuesta de aplicación móvil que cubre dichas necesidades.

### 1.3. Solución Propuesta

La propuesta de aplicación móvil que se presenta como solución a las necesidades anteriores utiliza e integra diferentes tecnologías. En primer lugar, **la aplicación se integra con la red social Foursquare**, una red social especializada en lugares, sitios de ocio o de visita. La integración con esta red social permite que la aplicación muestre, utilice e incluya este conjunto de datos para la creación de rutas turísticas. En segundo lugar, **la aplicación está basada en la localización del usuario**. Esto significa que los puntos o lugares de ocio se obtienen en función de la posición de este. Finalmente, **la aplicación se integra con un servidor web** en el que los usuarios pueden publicar las rutas creadas. De esta manera, un usuario puede buscar, descargar y recorrer rutas creadas por otros usuarios.

Todas estas tecnologías son necesarias en el proceso de creación de rutas turísticas. El uso de **Foursquare**, una red social especializada en geolocalización, permite obtener una gran cantidad de información referente a lugares y establecimientos cercanos a la localización en la que se encuentra un usuario. Aunque existen otras redes sociales especializadas en este campo, se ha optado por esta en concreto ya que permite añadir a cualquier usuario un nuevo lugar, ampliando con ello la información que se puede obtener al realizar una búsqueda. Otro factor importante para su elección es su gran cantidad de usuarios, los cuales se amplían notablemente al estar integrada con Facebook. La **geolocalización**, una tecnología basada en calcular la posición actual de un usuario, ofrece un gran número de beneficios para éste. El beneficio principal que se obtiene de la geolocalización consiste en obtener en cualquier momento información sobre el mundo que rodea al usuario. Por ejemplo, en caso de tener un pinchazo o una avería en el coche, gracias a la geolocalización se puede buscar un taller cercano. Pero no es el único servicio que se ofrece a los usuarios que utilizan la geolocalización. Si una persona participa en una red social, gracias a la geolocalización se puede obtener la posición en la que se encuentran sus contactos. Por último, **el servidor Web** encargado de almacenar las rutas turísticas, proporciona una capa de persistencia que puede ser utilizada por todos los usuarios. Esta capa permite almacenar de forma constante la información creada y ser compartida a nivel global. Esto permite que un usuario que necesite consultar una ruta turística de una ciudad en concreto, por ejemplo París, pueda descargársela y seguirla.

Integrando todas estas tecnologías se ha desarrollado una aplicación que permite crear rutas turísticas. Este proceso se inicia cuando el usuario crea una ruta de viaje nueva. Una vez creada, obtiene su posición mediante el uso de la geolocalización. Esta información es enviada a Foursquare, que filtra los puntos almacenados en su red social en base a ella y los devuelve al dispositivo. Utilizando estos puntos y añadiendo información adicional, como comentarios o valoraciones de los lugares, el usuario va construyendo su ruta punto a punto. Una vez finalizada, es enviada al servicio Web para ser almacenada, permitiendo que otros usuarios puedan descargarla y seguirla, pudiendo disfrutar las experiencias vividas por este.

## 1.4. Contexto de la Tesis

Esta tesis ha sido desarrollada para incrementar los conocimientos del alumno y poner a prueba todos aquellos que se han aprendido mediante la realización del *Máster en Ingeniería de Software, Métodos Formales y Sistemas de Información*.

Para ello, se han utilizado los siguientes conceptos de las siguientes asignaturas impartidas en el máster:

- Desarrollo de aplicaciones en Java. El contenido de esta asignatura fue de suma importancia para poder desarrollar la aplicación ya que el proyecto se realiza para la plataforma de dispositivos móviles Android, que emplea el lenguaje de programación Java.
- Modelado, diseño e implementación de servicios Web. Asignatura muy útil para entender los conceptos que permiten interactuar con la red social que proporciona la información y crear un servicio capaz de almacenar las rutas creadas por el usuario.
- Técnicas en ingeniería de requisitos. Asignatura que explica distintas técnicas de captar los requisitos de una aplicación. Enseña técnicas necesarias para entender el problema que se intenta abarcar y proporcionar la capacidad de crear la solución de forma fácil.

Por otro lado, esta tesis ha ofrecido al alumno la posibilidad de incrementar o adquirir un gran número de conceptos nuevos incrementando los ya obtenidos mediante la realización del máster. Entre estos conceptos se hallan:

- Desarrollo de aplicaciones para dispositivos móviles. Con una experiencia muy ínfima de la programación de estos dispositivos, mediante este trabajo se podría decir que se ha alcanzado un gran nivel en el desarrollo de aplicaciones para la plataforma Android.
- Interacción con redes sociales. A través de la recaudación de información de la red social empleada para mostrar los datos necesarios para que el usuario pueda interactuar con la aplicación, se ha adquirido una experiencia de su funcionamiento.
- Manejo de la API de Google Maps. Por último, se ha desarrollado un conocimiento por el api proporcionada por Google Maps, que permite mostrar un mapa en el dispositivo y poder interactuar con el plasmando toda la información anteriormente recaudada.

## 1.5. Estructura de la Tesis

El documento está organizado en seis capítulos:

- El capítulo 2 introduce las tecnologías que se han utilizado para desarrollar este trabajo: **Redes sociales, Geolocalización y Dispositivos móviles.**
- El capítulo 3 realiza un breve estudio sobre un conjunto de aplicaciones móviles similares a la desarrollada en este trabajo, finalizando con un pequeño resumen sobre aquellas características carentes en ellas y que han sido implementadas en la aplicación.
- El capítulo 4 muestra el proceso de integración de la **geolocalización**, las **redes sociales** y los **Servicios Web** en el desarrollo de este trabajo.
- El capítulo 5 expone como se ha estructurado la aplicación de este trabajo, en una arquitectura de tres capas compuesta por una capa de **presentación**, que permite al usuario interactuar con el sistema, una capa de **lógica**, que ofrece toda la funcionalidad del sistema y una capa de **persistencia**, que permite al sistema almacenar la información que utiliza.
- El capítulo 6 expone un manual de usuario que explica el funcionamiento de la aplicación desarrollada en este trabajo.
- El capítulo 7 muestra las conclusiones y posibles futuros trabajos a realizar en este trabajo.



## Capítulo 2

# Contexto Tecnológico

---

En este capítulo se describen aquellas tecnologías y conceptos que se utilizan en el desarrollo de esta tesis de máster. De esta forma, este capítulo introduce por una parte el concepto de red social, y por otra, el de geolocalización. Posteriormente se muestran y comparan varias redes sociales que introducen la geolocalización entre sus funcionalidades. Finalmente, se proporciona una visión general de la plataforma software Android, sobre la cual se ha realizado la implementación de este trabajo.

A continuación se estructura el contenido del resto del capítulo. En la sección 2.1 se introduce el concepto de red social. En la sección 2.2 se describe el concepto de geolocalización y se muestran varias redes sociales especializadas en este ámbito. En la sección 2.3 se realiza una comparativa de estas redes sociales. Por último, la sección 2.4, introduce al lector al área de los dispositivos móviles y en particular, a la plataforma software utilizada en el desarrollo de esta tesina de máster: Android.

### 2.1. Redes Sociales

Una red social es una comunidad virtual donde los usuarios ingresan para interactuar con personas de todo el mundo con gustos en común. Se pueden encontrar un gran número de temáticas en las redes sociales, entre las cuales destacan las profesionales o las de ocio, pudiendo encontrar tanto redes encargadas de búsqueda de trabajo como otras encargadas en ver el día a día de los amigos que tengamos agregados.

Una red social se suele representar como un grafo, donde cada elemento de la red es un nodo de este grafo que se relaciona mediante aristas con otros nodos de la red:



Figura 1. Representación de las redes sociales

Las redes sociales han supuesto un gran impacto en la vida de la gente, creando comunidades con una gran cantidad de usuarios. Un ejemplo concreto es Facebook(1), que con sus más de 750 millones de usuarios<sup>1</sup>, es la red social actual que cuenta con una mayor participación diaria. Teniendo en cuenta este dato y la tabla mostrada en la figura 2, extraída de la web Internet World Stats, donde se puede observar que más de dos mil millones de personas usan internet, se puede asegurar que cerca de la mitad de la población que usa internet utiliza Facebook. Además, si se suman todos los usuarios registrados en las redes sociales de la figura 3, se obtienen más de 2.000 millones de usuarios registrados a una red social, que es casi la misma cantidad de usuarios que utilizan internet. Este dato es erróneo, ya que muchos usuarios de una red social, están registrados en otras redes sociales, pero muestra que existen una gran cantidad de usuarios registrados en las redes sociales.

<b>WORLD INTERNET USAGE AND POPULATION STATISTICS</b>						
<b>March 31, 2011</b>						
<b>World Regions</b>	<b>Population ( 2011 Est.)</b>	<b>Internet Users Dec. 31, 2000</b>	<b>Internet Users Latest Data</b>	<b>Penetration (% Population)</b>	<b>Growth 2000-2011</b>	<b>Users % of Table</b>
<a href="#">Africa</a>	1,037,524,058	4,514,400	118,609,620	11.4 %	2,527.4 %	5.7 %
<a href="#">Asia</a>	3,879,740,877	114,304,000	922,329,554	23.8 %	706.9 %	44.0 %
<a href="#">Europe</a>	816,426,346	105,096,093	476,213,935	58.3 %	353.1 %	22.7 %
<a href="#">Middle East</a>	216,258,843	3,284,800	68,553,666	31.7 %	1,987.0 %	3.3 %
<a href="#">North America</a>	347,394,870	108,096,800	272,066,000	78.3 %	151.7 %	13.0 %
<a href="#">Latin America / Carib.</a>	597,283,165	18,068,919	215,939,400	36.2 %	1,037.4 %	10.3 %
<a href="#">Oceania / Australia</a>	35,426,995	7,620,480	21,293,830	60.1 %	179.4 %	1.0 %
<b>WORLD TOTAL</b>	<b>6,930,055,154</b>	<b>360,985,492</b>	<b>2,095,006,005</b>	<b>30.2 %</b>	<b>480.4 %</b>	<b>100.0 %</b>

**Figura 2. Estadísticas mundiales de usuarios de internet**

<sup>1</sup> Estadísticas oficiales de la red social Facebook: <http://www.facebook.com/press/info.php?statistics>

Name	Description/Focus	Date launched	Registered users
Facebook	General.	February 2004	640,000,000+ <sup>[77]</sup>
Qzone	General. In Simplified Chinese; caters for mainland China users		480,000,000 <sup>[77][241]</sup>
Habbo	General for teens. Over 31 communities worldwide. <a href="#">Chat Room</a> and <a href="#">user profiles</a> .		200,000,000 <sup>[125][126][127]</sup>
Twitter	General. Micro-blogging, RSS, updates	July 15, 2006	175,000,000 <sup>[291][292]</sup>
Renren	Significant site in China. Was known as 校内 (Xiaonei) until August 2009.		160,000,000 <sup>[245]</sup>
Windows Live Spaces	Blogging (formerly MSN Spaces)		120,000,000 <sup>[313]</sup>
LinkedIn	Business and professional networking	May 2003	120,000,000 <sup>[166]</sup>
Bebo	General	July 2005	117,000,000 <sup>[16]</sup>
Vkontakte	Social Network for Russian-speaking world including former Soviet republics.	September 2006	110,578,500 <sup>[294]</sup>
Tagged	General.		100,000,000 <sup>[273]</sup>
Orkut	General. Owned by Google Inc. Popular in <a href="#">India</a> and <a href="#">Brazil</a> . <sup>[222]</sup>	January 22, 2004	100,000,000 <sup>[223]</sup>
Myspace	General	August 2003	100,000,000+ <sup>[199][200]</sup>

**Figura 3. Listado de redes sociales con más usuarios registrados**

Las redes sociales ofrecen diversos beneficios a sus usuarios. Entre estos beneficios se puede encontrar la socialización. Una red social permite conectar un gran número de usuarios distintos, que se conocen entre sí o no, pero que suelen tener gustos y aficiones en común. Esto ayuda a conocer a otras personas, con estos mismos gustos y aficiones.

Existen beneficios de ámbito profesional. Las redes sociales permiten a las empresas crear campañas publicitarias y estudios de mercado. Con este fin, una empresa crea su perfil en una red social. Los usuarios interesados en esta empresa pueden realizar un seguimiento de ella en la red social. Cuando la empresa anuncia un nuevo producto, todos los usuarios que están siguiendo su perfil son informados inmediatamente de ello. Esto proporciona a las empresas un medio de comunicación directo gratuito con sus clientes o usuarios. Además, si un usuario expone su opinión en el perfil de la empresa sobre el anuncio anterior, esta puede hacer un estudio de mercado sobre un la aceptación de su producto.

Por último, las redes sociales proporcionan una gran cantidad de información. Por ejemplo, la red social Facebook, ofrece una gran cantidad de información personal ya que sus usuarios opinan y realizan comentarios a diario en su perfil. Otro ejemplo diferente al anterior es Foursquare, una red social de geolocalización que proporciona a sus usuarios todos los lugares y establecimientos cercanos a su posición. LinkedIn es otro ejemplo de red social. Esta red social está especializada en la búsqueda de empleo, proporcionando a sus usuarios información de ofertas de trabajo.

Aparte de los beneficios anteriores, las redes sociales proporcionan otros beneficios adicionales a los desarrolladores de aplicaciones software. Entre estos beneficios se encuentran las APIs, que permiten crear nuevas aplicaciones que interactúen con la red

social u obtengan información de ella. Por ejemplo, utilizando la red social LinkedIn se puede crear una aplicación móvil que informe a los usuarios cuando se publique una oferta de trabajo acorde con sus características profesionales. La ventaja de crear una aplicación nueva para una red social reside en poder disponer de un conjunto de funcionalidades prediseñadas bastante útiles. Citando el ejemplo anterior de Foursquare, gracias a esta red social un desarrollador solo debe indicar la posición en la que se encuentra un individuo para obtener todos los restaurantes cercanos a él y mostrarlo en su aplicación.

## 2.2. Redes Sociales Basadas en la Geolocalización

La geolocalización es un concepto que define la capacidad de conocer la localización geográfica exacta de un elemento en cualquier momento.

Existen dos mecanismos para obtener la localización de un usuario: mediante el uso de un dispositivo **GPS** o mediante **internet**.

El **GPS** es un dispositivo que calcula la posición mediante la red de satélites que rodea el planeta. Este dispositivo viene integrado en la mayoría de los teléfonos de última generación, dispositivos acuñados con el término inglés “*Smartphone*”. La ventaja del GPS es que es más preciso que la localización a través de internet, pero como inconveniente es más lento al principio, ya que debe encontrar la posición de un satélite y realizar la conexión.

Para realizar el cálculo mediante **internet**, se utilizan las antenas que proporcionan cobertura 3G a los dispositivos móviles. Para ello, se calcula la intensidad de la señal a de estas antenas para posteriormente triangular la posición estimada en un mapa. La desventaja de calcular la posición a través de internet es que es menos preciso que mediante el cálculo con GPS, pudiendo dar una pequeña desviación, pero a cambio cuenta con una mayor velocidad inicial en comparación con el GPS.

Los dispositivos móviles tienen integrados dispositivos GPS y una conexión a internet. Esto les proporciona la capacidad de poder calcular su localización. Esta localización puede ser enviada a una red social especializada en geolocalización para obtener información adicional.

Existen un número distinto de redes sociales especializadas en geolocalización, tales como Gowalla(2), Facebook Lugares(1), Foursquare(3), Brightkite(4), Loopt(5), Yelp(6), etc. Estas redes sociales se basan en la posición de un individuo para obtener toda la información perteneciente a ella que se encuentre cercana. Por ejemplo, si un usuario desea saber que restaurantes se encuentran cerca de su posición, mediante una red social de geolocalización puede obtener esta información. Estas redes sociales también se enriquecen de la información que introducen los usuarios. Entra esta

información se encuentran los comentarios u opiniones de un sitio en concreto o la creación de un nuevo lugar o establecimiento.

En esta sección se describen las redes sociales basadas en localización: Facebook Places(1), Google Latitude(7), Foursquare(3) y Gowalla(2). En particular, se han seleccionado estas redes sociales porque cuentan con la mayor cantidad de usuarios registrados.

### **Google Latitude**

Google Latitude(7) es una aplicación para dispositivos móviles creada por Google que permite mostrar (1) la ubicación actual de un individuo, (2) aquellos lugares registrados en la red de Google, y (3) la posición del conjunto de contactos del usuario.

El funcionamiento de Google Latitude es muy sencillo ya que una vez abierta la aplicación, se calcula la posición del usuario y la de todos sus contactos. Debido a la privacidad ofrecida por Google Latitude, si un usuario quiere registrar un nuevo contacto en su lista de contactos, primero debe enviarle una petición de amistad. Cuando esta petición de amistad es aceptada, el contacto pasa a formar parte de la lista. Cuando un contacto pertenece a la lista de contactos, se puede ver su posición en el mapa tal como vemos en la Figura 4.

A medida que los usuarios van examinando el mapa, pueden observar aquellos puntos registrados en Google. Estos puntos disponen de información de interés, como el número de teléfono o la dirección, entre otros. Pero no solo se limitan a proporcionar información, también ofrecen algunas opciones funcionales como la posibilidad de indicar la ubicación del usuario en un determinado lugar, operación que dentro de las redes sociales basadas en geolocalización se conoce con el término “*check-in*”, la cual publica dicha información dentro de la propia red social a los contactos del usuario que realiza dicha operación.



**Figura 4. Google Latitude**

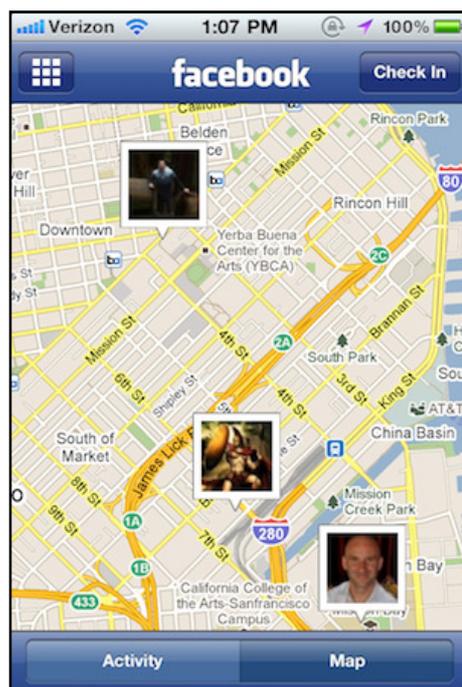
### **Facebook Places**

Facebook Places(1) es la funcionalidad basada en geolocalización desarrollada e integrada en la aplicación móvil de la red social Facebook. De esta forma, Facebook Places se beneficia de la gran cantidad de usuarios que hacen uso de esta red: más de 750 millones de usuarios.

El funcionamiento de este servicio es parecido al del resto de redes sociales basadas en geolocalización. Para ello, el usuario debe identificarse con su cuenta de Facebook. Una vez identificado, puede realizar “*check-in*” o buscar amigos y lugares que estén cerca de su posición a partir de la capacidad de geolocalización de su dispositivo móvil. Así, cuando se realiza un “*check-in*”, éste es publicado en su cuenta pudiendo ser visto por los contactos agregados de la misma forma en que se ve una actualización relacionada con la publicación de un comentario o con la subida de una fotografía.

Como se puede observar en la figura 5, Facebook Places proporciona la opción de ver a otros usuarios de esta plataforma que estén cerca de la posición actual del usuario. Esto puede suponer un problema de privacidad ya que un usuario no siempre puede querer mostrar su ubicación. Para ello, Facebook Places permite desactivar esta opción.

Junto a esta utilidad, Facebook Places ha lanzado un API que permite a los desarrolladores realizar aplicaciones usando su plataforma. Hay que tener en cuenta que a día de hoy, solo está disponible a nivel global el API de lectura. Esta API proporciona la capacidad de obtener toda la información accesible de la red social. En cambio, el API de escritura se encuentra en fase beta, y solo es accesible para empresas, evitando con ello que los desarrolladores puedan crear aplicaciones que permitan añadir nuevos lugares o establecimientos a la red social.



**Figura 5. Contactos Facebook Places**

## **Foursquare**

Foursquare(3) es una red social de geolocalización en la cual el usuario, a través de un dispositivo móvil, es capaz de obtener su posición y mostrar los lugares o establecimientos cercanos a ésta, como por ejemplo un restaurante. Posteriormente, Foursquare permite realizar *check-in* en estos establecimientos indicando que el usuario se encuentra en dicho lugar y comentar sus impresiones y valoraciones acerca de estos lugares.

Además, también muestra publicidad o promociones de aquellos establecimientos cercanos a la posición del usuario. Esto permite a las empresas promocionarse mediante el uso de las redes sociales a un gran número de clientes potenciales.

Asimismo, Foursquare cuenta también con aplicaciones móvil para casi cualquier dispositivo móvil como iPhone, Blackberry y Android, siendo éstas gratuitas. Esto permite a Foursquare abarcar un gran número de usuarios.

Una de las ventajas de Foursquare es la posibilidad de introducir un nuevo lugar o establecimiento a su red. Por ejemplo, si un usuario tiene la costumbre de visitar un bar para ver el partido de futbol de los domingos, y este bar no se encuentra en Foursquare, rellenando un formulario es añadido a la red social. Con esto, se crea una sensación de participación que da a entender al usuario que éste no es meramente un cliente más, si no un miembro de la comunidad que hace crecer el contenido de un entorno real y compartido ubicado en la red social.

Otro de los atractivos de Foursquare, el cual le ha dado más fama es que ha convertido su red en un juego. El usuario puede acceder a distintos premios simbólicos, como por ejemplo las medallas, mostradas en la figura 6, que se otorgan a los usuarios cuando se han cumplido una serie de objetivos. Otro premio simbólico consiste en convertirse en el *mayor* de un lugar. El *mayor* es aquella persona que ha realizado más *check-ins* en ese lugar.

Por último destacar que Foursquare cuenta con integración para las redes sociales Twitter y Facebook, permitiendo publicar en otras redes sociales la actividad realizada en ella.



Figura 6. Medallas Foursquare

## Gowalla

Gowalla(2)es una red social de geolocalización que permite mediante un dispositivo móvil, obtener los lugares y establecimientos cercanos a un usuario, permitiendo que éste realice *check-in* en ellos.

Al igual que Foursquare, Gowalla cuenta con aplicaciones para iPhone, WebOS y Blackberry y Android. Además, cuenta con una web mediante la cual un usuario puede hacer *check-in* con cualquier dispositivo móvil que no tenga una aplicación nativa de este servicio.

Otro de los alicientes de Gowalla es poder compartir los *check-in* en Facebook y Twitter, mostrando a todos los contactos agregados los puntos visitado, dando la opción de ser comentados al igual que una fotografía o comentario. Además, este servicio

también permite añadir amigos pertenecientes a estas redes sociales al igual que en los contactos de Gmail o de Windows Live.

Gowalla cuenta con una opción accesible para cualquier usuario que permite añadir lugares y establecimientos a su red social. Esto permite incrementar la información mostrada por Gowalla no solo a las empresas, como es el caso de Facebook Places.

Otro de los puntos interesantes de Gowalla son los Trips, mostrados en la figura 7. Trips es una funcionalidad que permite visualizar rutas interconectando puntos de paso. Estas rutas vienen incluidas en la aplicación nativa de Gowalla para móviles, con todo un conjunto de comentarios disponibles por sus creadores. El principal inconveniente de este servicio reside en que actualmente no permite crear rutas a los usuarios, teniendo que depender de aquellas rutas que crean los responsables de esta red social.

Por último, al igual que en Foursquare, Gowalla cuenta con un mini juego de visitas, La diferencia radica que así como en Foursquare existe el concepto de “Mayor”, que es la persona que más veces ha realizado *check-in* en un lugar, en Gowalla hay un ranking de personas que han pasado por ese lugar. Las recompensas a esto son medallas para el usuario que simplemente tienen un valor simbólico, permitiendo que se vean más valorados que otros usuarios.

Gowalla también incluye la funcionalidad necesaria para introducir nuevos puntos a su red. Este servicio está muy limitado ya que los únicos datos a introducir, son: el tipo de sitio que es, su localización y una descripción.

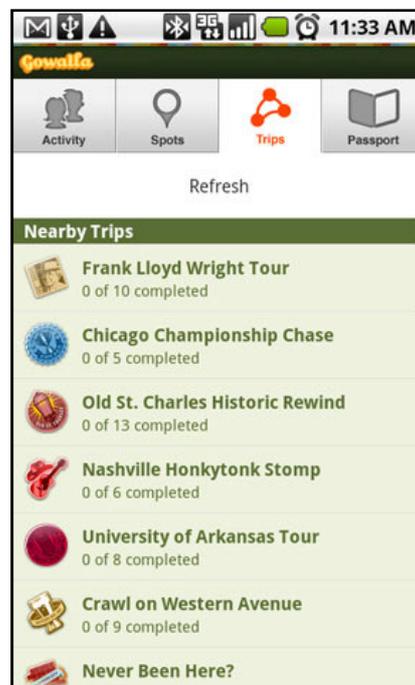


Figura 7. Trips Gowalla

### 2.3. Discusión

En la anterior sección se han introducido cuatro redes sociales basadas en geolocalización. Todas ellas proporcionan al usuario la posibilidad de conocer su posición exacta y disponer de información cercana en base a dicha posición. A continuación se realiza una comparativa entre estas redes sociales que refleja de forma resumida sus atributos:

	<b>Google Latitude</b>	<b>Facebook Places</b>	<b>Foursquare</b>	<b>Gowalla</b>
<b>Número de Usuarios</b>	Más de 10 millones	Más de 500 millones	Más de 8 millones	-
<b>Integración con otras redes sociales</b>	No	Con Foursquare y Gowalla	Con Facebook	Con Facebook
<b>Permite publicidad</b>	Si	Si	Si	Si
<b>Plasma la información sobre un mapa</b>	Si	Si	Si	Si
<b>Integración de amigos</b>	Si	Si	Si	Si
<b>Publicación de los Check-ins</b>	En Google	En Facebook	En Foursquare y Facebook	En Gowalla y Facebook
<b>Permite al usuario añadir localizaciones</b>	No	No	Si	Si
<b>Trips o rutas turísticas</b>	No	No	No	Si
<b>API de Escritura abierta</b>	No	No	Si	Si

**Tabla 1. Tabla comparativa entre redes sociales**

Analizando la tabla se puede observar como Foursquare proporciona aquellas prestaciones necesarias a la hora de crear una aplicación que gestione rutas turísticas. El API de lectura proporciona todos los lugares cercanos a la localización de un usuario. El API de escritura proporciona la posibilidad de añadir un nuevo punto a la red social, incrementando con ello la información que se puede mostrar.

Asimismo, Foursquare cuenta con un mayor número de usuarios que Gowalla, y esto indica una actualización de la información más frecuente, la cual puede aportar al usuario un valor añadido. Además, esta red social se integra con Facebook, permitiendo en un futuro incrementar el trabajo realizado en esta tesina, para que las rutas turísticas sean publicadas y comentadas en Facebook.

En definitiva, esta comparativa muestra que Google Latitude y Facebook Places no permiten utilizar el API de escritura a cualquier usuario, evitando con esto que se puedan añadir puntos nuevos. Por otra parte, pese a que Gowalla cuenta con esta

funcionalidad, tiene un menor número de usuarios, ofreciendo por tanto un flujo de información menor que Foursquare.

## 2.4. Dispositivos Móviles

Los Smartphone o teléfonos inteligentes son dispositivos móviles capaces de proporcionar al usuario las mismas prestaciones que un ordenador personal. Entre estas prestaciones se encuentran: leer el correo electrónico, visitar una Web, visualizar y editar documentos, planificar reuniones, etc.

Aunque parecen un sustituto de los ordenadores de sobremesa, estos dispositivos cuentan con un problema que evita que esto sea posible: son dispositivos más pequeños. Esto influye en la cantidad de información que son capaces de mostrar y en su interacción con ellos. Por ejemplo, si un usuario quiere leer un archivo de texto, cuenta con un menor número de líneas de lectura. Por otro lado, si desea modificarlo, el dispositivo proporciona un teclado más pequeño e incómodo, dificultando esta tarea.

Otro problema a tener en cuenta es la capacidad de sus baterías. Al ser dispositivos más pequeños y finos, cuentan con una batería más reducida. Si a esto le sumamos un gran número de aplicaciones multimedia que consumen muchos recursos, obtenemos un dispositivo que en menos de un día debe ser recargado.

Además, estos dispositivos suelen tener integrado un dispositivo GPS, que junto a la conexión a internet, proporcionan la posibilidad de saber en cualquier momento la posición actual y mostrarla en un mapa.

Actualmente, el sistema operativo de un Smartphone, es uno de los componentes más importantes, siendo las tres grandes casas competidoras en Estados Unidos (Microsoft, Apple, Google) las que están en la disputa por llevarse el mercado en este aspecto.

- Windows Phone<sup>2</sup>, anteriormente llamado Windows Mobile es el sistema operativo propuesto por Microsoft. Se trata de la versión de Windows para móviles.
- iOS<sup>3</sup>(anteriormente denominado iPhone OS) es el sistema operativo planteado por Apple, y que se usa tanto en su dispositivo móvil iPhone, como en otros dispositivos como iPad o iPod Touch.
- Android<sup>4</sup> es el sistema operativo de Google. Está basado en Linux y fue desarrollado originalmente para dispositivos móviles, tales como teléfonos

---

<sup>2</sup> <http://www.microsoft.com/windowsphone/es-mx/default.aspx>

<sup>3</sup> <http://www.apple.com/es/iphone/ios4/>

inteligentes, tablets, pero que actualmente se encuentra en desarrollo para usarse en netbooks y PCs.

En particular, la aplicación que se presenta en este trabajo se ha realizado para la plataforma Android. Por este motivo, a continuación se profundiza en sus componentes y funcionamiento.

#### **2.4.1. Android**

Android es el sistema operativo de Google para dispositivos móviles. Basa su funcionamiento en un conjunto de actividades que son procesadas mediante una pila, donde la última en ser ejecutada es la que se muestra en primer plano, dejando en reposo al resto. Google provee de herramientas y APIs necesarios para comenzar a desarrollar aplicaciones en la plataforma Android, utilizando el lenguaje de programación Java.

#### **2.4.2. Características**

- Framework de aplicación que permite la reutilización y sustitución de los componentes.
- Máquina virtual Dalvik optimizada para dispositivos móviles.
- Navegador integrado basado en el motor Webkit de código abierto.
- Gráficos optimizados impulsado por una biblioteca de gráficos 2D, gráficos3D basado en la especificación OpenGL ES 1.0(aceleración de hardware opcional)
- SQLite para almacenamiento estructurado de datos.
- Soporte a los medios de comunicación de audio, videos y formatos de imagen comunes(MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- GSM Telephony.
- Bluetooth, EDGE, 3G, y WiFi.
- Cámara, GPS, brújula, y acelerómetro.
- Entorno de desarrollo completo que incluye un emulador de dispositivos, herramientas para la depuración, la memoria y de perfiles de rendimiento, y un plugin para el IDE de Eclipse.

---

<sup>4</sup> <http://www.android.com/>

### 2.4.3. Arquitectura

El siguiente diagrama muestra los componentes principales del sistema operativo Android. Cada sección se describe con más detalle a continuación.

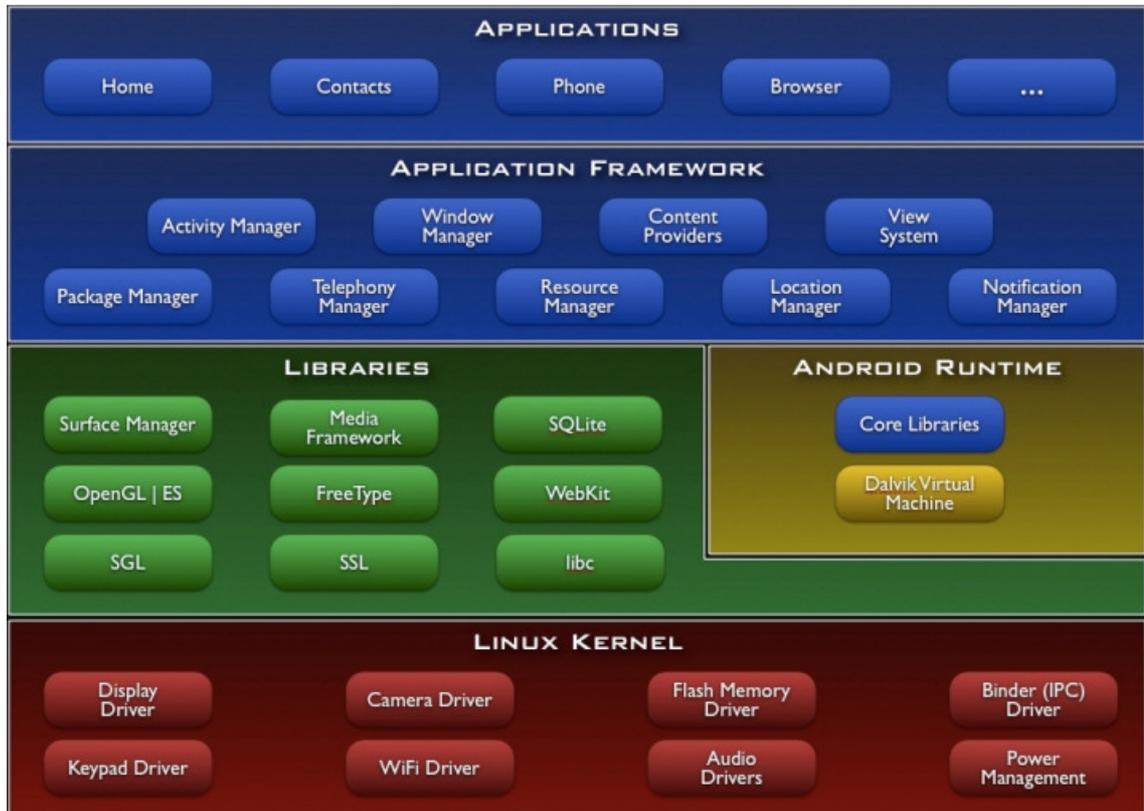


Figura 8. Arquitectura Android

**Aplicaciones:** Las aplicaciones base incluyen un cliente de email, programa de SMS, calendario, mapas, navegador, contactos, y otros. Todas las aplicaciones de Android están escritas en el lenguaje de programación Java.

**Framework de aplicaciones:** Los desarrolladores tienen acceso completo a las mismas APIs del framework usadas por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Éste mismo mecanismo permite que los componentes sean reemplazados por el usuario.

**Librerías:** Android incluye un set de librerías C/C++ usadas por varios componentes del sistema. Estas capacidades se exponen a los desarrolladores a través del framework de aplicaciones de Android. Algunas son: System C library (implementación de una librería estándar del lenguaje C), librerías multimedia, librerías de gráficos, 3D, SQLite, entre otras.

**Runtime de Android:** Android incluye un set de librerías que provee la mayor parte de las funcionalidades del lenguaje de programación Java. Cuando se ejecuta una aplicación, se crea una nueva instancia de la máquina virtual Dalvik, encargada de ejecutar los procesos de esta aplicación. Dalvik proporciona la capacidad de ejecutar de forma eficiente, distintas instancias de la máquina virtual.

**Núcleo - Linux:** Android se basa en una versión de Linux 2.6 para los servicios base del sistema, como seguridad, gestión de memoria, gestión de procesos, pila de red, y modelo de drivers. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

#### 2.4.4. Fundamentos de las Aplicaciones Android

Las aplicaciones de Android están escritas en el lenguaje de programación Java. Las herramientas de SDK de Android compilan el código, junto con los datos y archivos de recursos en un paquete de Android, un archivo con un sufijo “.apk”. Este paquete es un instalador de la aplicación.

Una vez instalado en un dispositivo, cada aplicación Android tiene su propio entorno seguro:

- El sistema operativo Android es un sistema multiusuario de Linux en el que cada aplicación tiene un identificador de usuario diferente.
- Por defecto, el sistema asigna a cada aplicación un único ID de usuario. El sistema establece permisos para todos los archivos en una aplicación y sólo el ID de usuario asignado a esa aplicación puede acceder a ellos.
- Cada proceso tiene su propia máquina virtual (VM). Esto implica que el código de una aplicación se ejecute de forma aislada a otras aplicaciones.
- Por defecto, cada aplicación se ejecuta en su proceso Linux. Android inicia el proceso cuando alguno de los componentes de la aplicación debe ser ejecutados. Posteriormente se cierra el proceso cuando ya no se necesita o cuando el sistema debe recuperar la memoria para otras aplicaciones.

De esta manera, el sistema Android aplica el principio de mínimo privilegio. Este principio se basa en que cada aplicación, por defecto, sólo tiene acceso a los componentes que necesita para hacer su trabajo. Esto crea un ambiente muy seguro en el que una aplicación no puede acceder a partes del sistema si no tiene permisos.

Sin embargo, existen formas de compartir datos entre aplicaciones y de que una aplicación pueda acceder a los servicios del sistema:

- Si dos aplicaciones comparte el mismo ID, estas pueden acceder a los mismos archivos y ejecutar el mismo proceso en la máquina virtual.
- Una aplicación puede solicitar permiso para acceder a los datos del dispositivo, tales como los contactos del usuario, los mensajes SMS, el almacenamiento (tarjeta SD), cámara, Bluetooth, etc. Todos los permisos de la aplicación deben ser concedidos por el usuario durante la instalación.

#### **2.4.5. Componentes de una Aplicación**

Los componentes son los elementos esenciales de una aplicación Android. Cada componente es un punto de entrada diferente. No todos los componentes son verdaderos puntos de entrada para el usuario y algunos dependen de otros, pero cada uno existe como una entidad propia y desempeña un papel específico. Cada uno es una pieza única que ayuda a definir el comportamiento general de la aplicación.

Hay cuatro tipos diferentes de componentes. Cada tipo tiene un propósito y un ciclo de vida distinto, que define cómo se crea y destruye. Estos son los cuatro tipos de componentes de una aplicación:

##### **Componentes de tipo Activity**

Una actividad representa una pantalla única con una interfaz de usuario. Por ejemplo, una aplicación de correo electrónico puede tener una actividad que muestra una lista de correos electrónicos nuevos, otra actividad para componer un correo electrónico, así como otra actividad para leer correos electrónicos. Aunque las actividades de trabajo forman una experiencia de usuario coherente en la aplicación de correo electrónico, cada una es independiente de las otras. Por lo tanto, una aplicación diferente puede iniciar cualquiera de estas actividades (si la aplicación de correo electrónico lo permite). Por ejemplo, una aplicación que utilice la cámara fotográfica del dispositivo, puede iniciar una actividad en la aplicación de correo electrónico que permita componer nuevos mensajes de correo, para que el usuario pueda enviar una fotografía.

##### **Componentes de tipo Service**

Un servicio es un componente que se ejecuta en segundo plano para realizar operaciones de larga duración o para realizar trabajos para los procesos remotos. Por ejemplo, un servicio puede reproducir música en segundo plano mientras el usuario está en otra aplicación. Otros componentes, tales como una actividad, pueden iniciar el servicio y dejarlo funcionando o hacer que se una a ella con el fin de interactuar con él.

Un servicio no proporciona una interfaz de usuario.

##### **Componentes de tipo Content Provider**

Un proveedor de contenido maneja un conjunto compartido de datos de aplicación. Puede almacenar los datos en el sistema de archivos, en una base de datos SQLite, en la

web, o cualquier otro lugar de almacenamiento permanente al que una aplicación pueda tener acceso. A través del proveedor de contenidos, otras aplicaciones pueden consultar o modificar los datos (si el proveedor de contenido lo permite). Por ejemplo, el sistema Android proporciona un proveedor de contenidos que gestiona la información de contacto del usuario. Por lo tanto, cualquier aplicación con los permisos adecuados puede hacer una consulta por parte del proveedor de contenido y leer o escribir información sobre una persona en particular.

Los proveedores de contenido también son útiles para la lectura y escritura de datos que son privados de una aplicación y no se comparten. Por ejemplo, la aplicación del Bloc de Notas utiliza un proveedor de contenido para guardar notas.

Un proveedor de contenido se implementa como una subclase de Content Provider y debe implementar un conjunto estándar de API que permita a otras aplicaciones realizar transacciones.

### **Componentes de tipo Broadcast Receiver**

Un receptor de radiodifusión es un componente que responde a los anuncios de difusión en todo el sistema. Muchas emisiones se originan en el sistema, por ejemplo, una emisión que anuncia que la pantalla se ha apagado, la batería está baja, o una imagen capturada. Las solicitudes también pueden iniciar emisiones, por ejemplo, para permitir que otras aplicaciones sepan que algunos datos han sido descargados en el dispositivo y está disponible para que los utilicen. A pesar de que los receptores de radiodifusión no muestran una interfaz de usuario, se puede crear una notificación en la barra de estado para alertar al usuario cuando un evento de difusión se produce. Sin embargo, un receptor es sólo una "*puerta*" a otros componentes y tiene la intención de hacer una cantidad de trabajo mínimo.



## Capítulo 3

# Estudio de Aplicaciones Móviles Similares

---

Este capítulo introduce el conjunto de las distintas aplicaciones relacionadas con la aplicación desarrollada en este trabajo. En primer lugar, se introducen aquellas proporcionadas por las anteriores redes sociales basadas en geolocalización: Foursquare(3), Gowalla(2) y Google Latitude(7). Posteriormente, se describen otras aplicaciones relacionadas.

Cada sección de este capítulo describe brevemente una aplicación: en la sección 3.1 se describe TripJournal(8), una aplicación que permite crear un diario de viajes. En la sección 3.2 se describe TripIt(9), que permite planificar los viajes. En la sección 3.3 se describe la aplicación nativa de Foursquare(3). En la sección 3.4 se describe la aplicación nativa de Gowalla(2) y en la 3.5 la aplicación nativa de Google Latitude(7). En la sección 3.6 se realiza una breve descripción de Minube(10), también perteneciente a una red social. En la sección 3.7 se plantea un nuevo tipo de aplicación, que aunque parecida a las descritas sobre las redes sociales, no pertenece a una red social en sí, TripAdvisor(11). Para finalizar, el punto 3.8 muestra las conclusiones obtenidas al realizar el estudio de las aplicaciones mencionadas anteriormente.

### 3.1. TripJournal

TripJournal(8) es una aplicación que permite crear rutas turísticas a las que se añaden de forma manual notas y fotografías de un lugar. Por ejemplo, si un usuario que está realizando un viaje y que acude a un monumento de su interés, puede añadir una foto (o varias) del lugar y añadir notas del autor. En resumen, TripJournal realiza la función de un diario de viajes que permite plasmar y anotar todas las experiencias del viajero.

Para ello, primero se debe indicar el punto de partida, punto llamado “waypoint” por la aplicación, de una ruta de viaje utilizando la capacidad de geolocalización del dispositivo móvil. Este proceso puede ser repetido en más de una ocasión para conseguir una señal más precisa de la ubicación del dispositivo. Una vez realizada la localización del usuario, la aplicación rueda en segundo plano capturando la ruta que el usuario realiza utilizando una cantidad menor de batería que si ejecutara en primer plano. Posteriormente, cuando el usuario desee añadir un punto a la ruta en fase de creación, éste debe rellenar los datos necesarios incluyendo fotografías o anotaciones

adicionales del lugar. Finalmente, para finalizar con la ruta, el usuario debe marcar dicha ruta como concluida quedando ésta almacenada en el dispositivo.

Como características adicionales de la aplicación, TripJournal permite calcular diversos atributos relacionados. Entre estos datos se encuentran; el tiempo de viaje, la distancia recorrida, la velocidad media a la hora de realizar este viaje, o la altitud máxima y mínima.

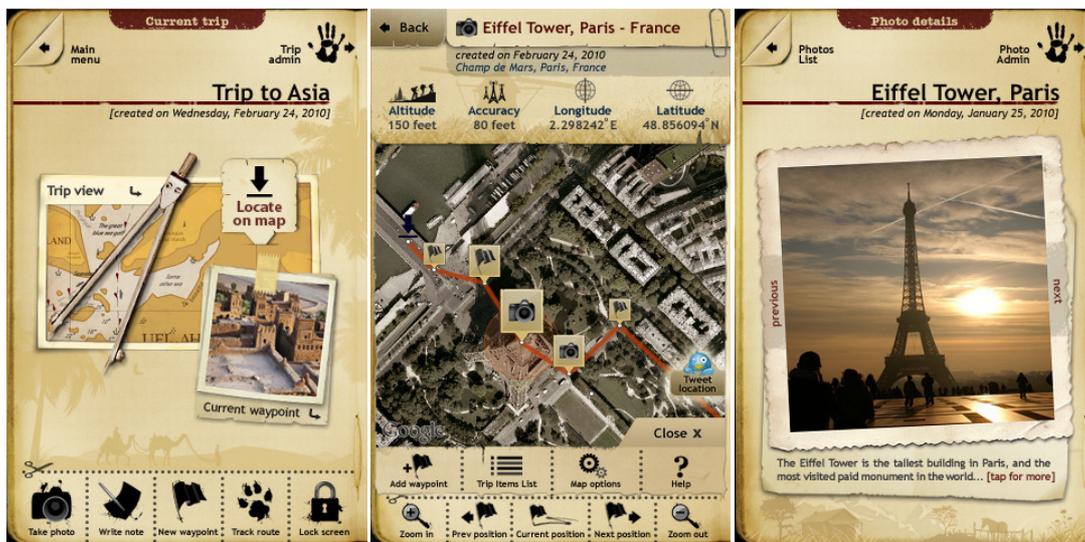


Figura 9. Capturas de la aplicación TripJournal

Finalmente, con el objetivo de hacer visible la ruta creada a otros usuarios, TripJournal se integra con las redes sociales Foursquare y Twitter, lo que permite que el usuario publique la ruta de viaje con otros usuarios.

### 3.2. TripIt

TripIt(9) es una aplicación para dispositivos móviles que permite planificar un viaje. Para ello, la aplicación permite reenviar el correo electrónico que un usuario recibe de confirmación al realizar una reserva de un vuelo o de una habitación de hotel mediante el cual, TripIt obtiene los datos indicados en este para confeccionar horarios, mapas, pasos necesarios, etc.

Su uso es sencillo: Cuando se recibe una confirmación de una reserva de hotel, de avión, etc. se debe enviar dicha confirmación a la dirección de correo electrónico registrada en TripIt. En un momento, el itinerario es enviado al dispositivo móvil para ser visualizado en la aplicación. En ella se muestran horarios, *check-ins* en hoteles, tiempo previsto, mapas asociados y otros datos de interés a la hora de realizar el viaje.

Una de las ventajas de esta aplicación es que si el usuario tiene una cuenta de correo de Google, TripIt permite sincronizarla de tal manera que no será necesario

reenviar los correos a su dirección, si no que automáticamente utilizara esta para obtener los datos necesarios que permiten elaborar el trayecto de la ruta.

Otro punto a destacar es la posibilidad de compartir rutas turísticas y otros datos entre los distintos amigos que se tengan en TripIt, como por ejemplo, precios de hoteles y de vuelos realizados, tiempo de viaje, lugares que el usuario tiene pensado visitar y día de ida y de vuelta.

TripIt permite añadir amigos a su aplicación, permitiendo saber en cualquier momento si cerca de la ubicación actual se encuentra alguien que también este realizando un viaje, permitiendo contactar con él.

Por último, TripIt permite crear un calendario de eventos personalizado para cada viaje permitiendo al usuario organizar su tiempo durante la realización de estos.

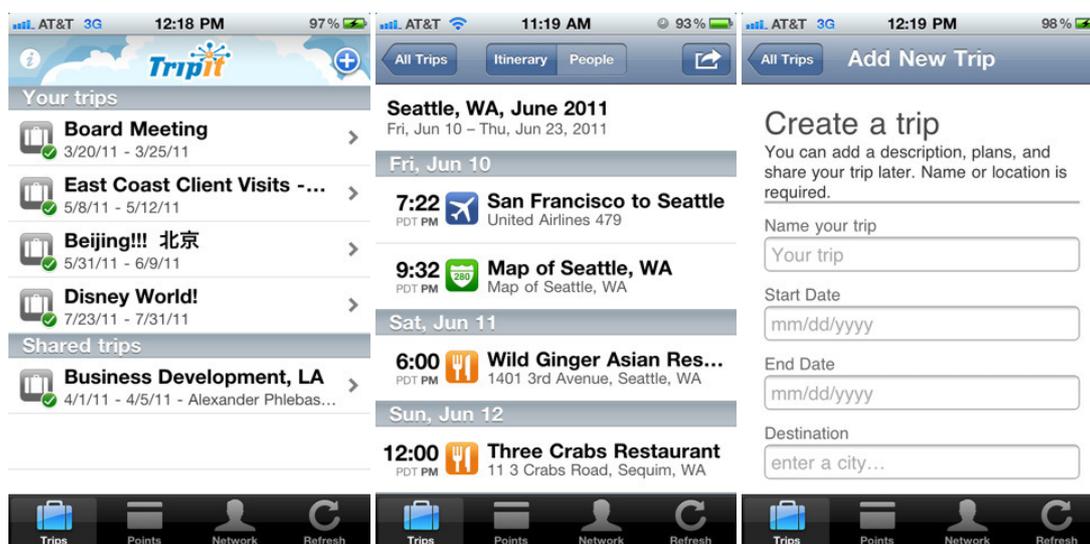


Figura 10. Capturas de la aplicación TripIt

### 3.3. Foursquare App

En este documento se ha introducido la red social (3), en qué se basa y cuáles son sus virtudes y defectos. A continuación se realiza una breve descripción sobre la aplicación que proporciona para dispositivos móviles, tanto iPhone, Android o Blackberry.

La aplicación de Foursquare se basa en la capacidad de mostrar un conjunto de lugares en función de la posición en la cual se encuentra un usuario para poder realizar *check-in* en ellos. Para ello, el usuario se identifica en la aplicación con su cuenta. En caso de no tener una cuenta, se le proporciona un enlace al formulario de creación de cuentas.

Una vez identificado, el usuario puede listar un conjunto de puntos cercanos a su ubicación y, en caso de existir, muestra las promociones realizadas por estos lugares. En

caso de no mostrarse un lugar en concreto, la aplicación cuenta con un buscador para intentar encontrarlo en la red. En caso de no existir este lugar, permite añadirlo mediante un formulario.

Cabe la posibilidad de que el usuario no quiera visualizar todos aquellos puntos cercanos, sino un conjunto de ellos según categorías. En este caso, Foursquare permite mediante un conjunto de filtros, obtener el listado de aquellas localizaciones que cumplen con las características indicadas, las cuales pueden mostrarse sobre un mapa.

Una vez realizado un *check-in*, el usuario puede compartirlo en Facebook o en Twitter, permitiendo a todos sus contactos visualicen la información presentada por este y comentarla.

Foursquare también cuenta con un gestor de amigos que permite añadir otros usuarios a la lista de contactos mediante peticiones de amistad. Para ello, se deben buscar aquellas personas que tengan la aplicación instalada y que pueden ser localizadas mediante la libreta de direcciones de teléfonos, Facebook o Twitter. Una vez un usuario acepte una petición de amistad, Foursquare permite visualizar aquellos *check-ins* realizados por él.

Foursquare también incluye una funcionalidad dedicada a mostrar cosas que el usuario tenga por hacer mostrando aquellas actividades añadidas recientemente o de lugares cercanos, permitiendo llevar un recordatorio de todo aquello que puede o debe realizar en cada momento.

Para finalizar, Foursquare también incluye un juego dentro de su aplicación en el cual se pueden ir consiguiendo medallas al hacer *check-in* en los distintos lugares que se visiten. Esto permite convertirse en el alcalde de un lugar, siendo este aquella persona que ha realizado más veces *check-in* en este lugar.

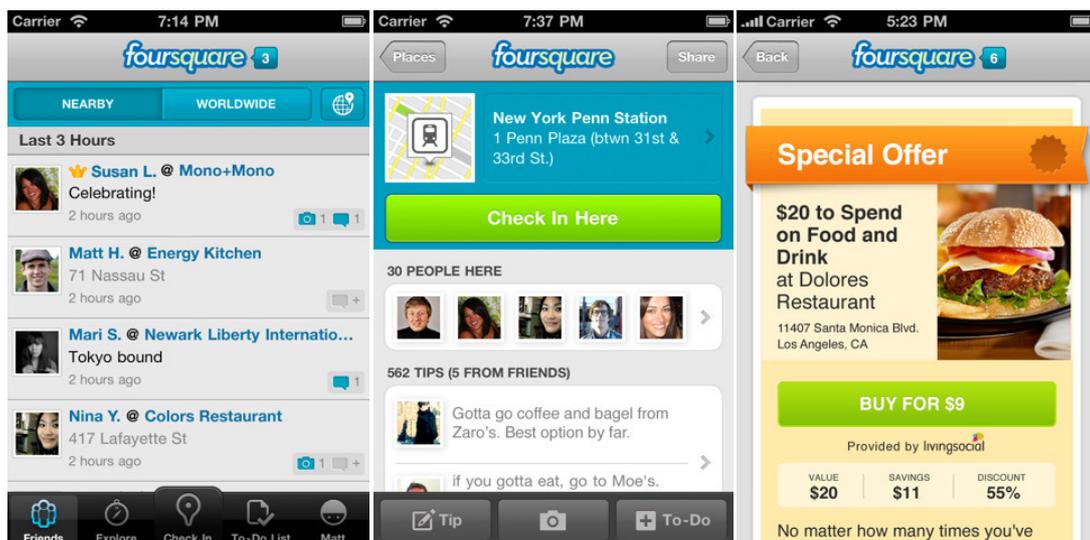


Figura 11. Capturas de Foursquare App

### 3.4. Gowalla

Gowalla(2) proporciona al usuario una interfaz gráfica amigable, muy intuitiva y con la funcionalidad claramente definida. Esto representa un gran aliciente frente a otras aplicaciones de geolocalización.

En particular, Gowalla permite obtener aquellos puntos cercanos a la ubicación del usuario y listarlos en una página para posteriormente realizar *check-in* en ellos. La diferencia con otras aplicaciones reside cuando se realiza un *check-in*. Gowalla permite añadir una foto realizada por el usuario e incluirla en el *check-in*.

En caso de no existir una localización, Gowalla cuenta con un formulario para añadir este a su red. Este formulario permite introducir un conjunto de datos menor que otras aplicaciones, dando una información en el futuro mucho más reducida que otras redes sociales.

Gowalla también cuenta con una sección de amigos, pudiendo añadir estos mediante la libreta de teléfonos, Facebook o Twitter. Una vez aceptada la solicitud de amistad, el usuario podrá observar aquellos *check-ins* realizados por sus amigos.

La opción más interesante de Gowalla reside en poder observar un conjunto de rutas turísticas existentes en Gowalla llamadas Trips. Estas rutas muestran un conjunto de puntos representados en un mapa de Google Maps y permiten realizar el trazado de la ruta realizada por el usuario, pudiendo ver en todo momento las fotos y comentarios subidas por este a la ruta.

Esta herramienta es muy útil y puede significar un duro competidor para el proyecto realizado en esta tesis en un futuro si se permite a los usuarios crear rutas turísticas, funcionalidad que en estos momentos solo puede ser realizada por los desarrolladores.



Figura 12. Capturas de la aplicación Gowalla

### 3.5. Google Latitude App

Es la aplicación proporcionada por Google basada en la geolocalización y una alternativa interesante a las presentadas por las redes sociales especializadas en este campo. La mayor diferencia reside en que en todo momento, el usuario puede ver su posición y la de sus contactos en un mapa de Google.

Para acceder a Google Latitude(7) el usuario se debe identificar con su cuenta de Google. Una vez identificado, la aplicación muestra un mapa con la posición actual del usuario. Esta puede ser calculada con mayor precisión mediante la utilización de dispositivos más precisos como el GPS.

El usuario tiene acceso a un listado de aquellas localizaciones cercanas registrada en la red de Google permitiendo mostrar sus datos, realizar un *check-in* en ellos o incluso, si tienen el teléfono registrado, realizar una llamada telefónica mediante el uso de su dispositivo móvil.

Google Latitude también permite compartir los *check-ins* en su red, al igual que aquellos comentarios o valoraciones realizadas por los usuarios. Otra de las posibilidades de Google Latitude es añadir un conjunto de amigos obteniendo estos de aquellos contactos que el usuario tenga en su cuenta de Google mediante una petición de amistad. Una vez aceptada esta, el usuario puede consultar su ubicación en el mapa.

En caso de no querer mostrar su ubicación, el usuario tiene la opción, mediante las opciones de privacidad, de cancelar la posibilidad de que sus contactos sepan donde se encuentra.



Figura 13. Capturas de la aplicación Google Latitude

### **3.6. Minube App**

Minube App(10) es la aplicación desarrollada por la red social Minube. Esta red social no ha sido comentada anteriormente en este documento debido a su poca cantidad de usuarios, si se compara con otras redes sociales.

Al igual que ocurre con las otras aplicaciones de redes sociales, para acceder a Minube App primero el usuario se debe identificar con su cuenta, y en caso de no tener una, se permite crear una mediante un formulario.

Una vez identificado, los usuarios pueden acceder a un conjunto de fotos y localizaciones añadidas por otros usuarios. Estas localizaciones pueden ser añadidas a favoritos, permitiendo, incluso sin conexión a internet, que el usuario las visualice en cualquier momento.

Al igual que ocurre con las otras redes sociales, Minube permite al usuario añadir su propio contenido y publicarlo en la red social, para que otros usuarios accedan a él. Estos contenidos también pueden ser publicados en Facebook o Twitter, permitiendo que esta información sea visualizada y comentada por más gente.

Entre las opciones destacadas, cabe la posibilidad de crear viajes, que son un conjunto de lugares visitados por los usuarios que han añadido estos lugares a la red social, formando así una ruta que muestra aquellos lugares de interés para el viajero.

Por último, Minube permite buscar los perfiles de otros viajeros y ser seguidos, permitiendo ver el contenido creado y publicado por estos, al igual que sus viajes.

Minube proporciona una nueva aplicación dentro del mundo de las redes sociales de la geolocalización en la cual los usuarios tiene un gran protagonismo añadiendo todos los lugares y comentarios referidos a estos, al igual que ofrece la posibilidad de crear viajes, dando la posibilidad a otros usuarios con menos experiencia, de realizar un viaje guiado.



Figura 14. Capturas de la aplicación Minube

### 3.7. TripAdvisor App

TripAdvisor App (11) es la aplicación de la Web TripAdvisor, encargada de reunir un conjunto de opiniones y consejos de viajeros para poder ser consultadas por cualquier persona que acceda a su web.

TripAdvisor App permite acceder a estas opiniones y consejos en cualquier lugar, mediante un dispositivo móvil, obsequiando al viajero con una guía de referencia de lugares portable.

TripAdvisor permite obtener los lugares cercanos a la posición actual del usuario y plasmarlos en un mapa, pudiendo saber la distancia y los comentarios de la gente sobre ellos.

En caso de no mostrarse un lugar por no estar cerca de él, TripAdvisor permite buscarlo y saber a qué distancia se encuentra de la ubicación del usuario si este se encuentra registrado en su red social.

Por último, mediante su cuenta, el usuario podrá añadir nuevo contenido a la Web para que otros usuarios puedan visualizarlo en un futuro.

Como se puede observar, esta aplicación permite crear una gran red de información creada por los usuarios para futuras consultas, dando una gran muestra de consejos y opiniones útiles a la hora de realizar un viaje.



Figura 15. Capturas de la aplicación TripAdvisor

### 3.8. Conclusiones

Como se ha podido observar, aquellas aplicaciones que proporcionan una visión más amplia de las localizaciones son las que vienen integradas con las redes sociales, como Foursquare. De todas estas redes sociales, Gowalla es la única que cuenta con rutas turísticas, también llamadas Trips. El desarrollo de estos Trips no es accesible a los desarrolladores, impidiendo crear aplicaciones que permitan crear rutas turísticas.

La aplicación desarrollada en esta tesis cubre este problema mediante la muestra de los puntos que tiene almacenados Foursquare en su red social y permite crear rutas de viaje, añadiendo comentarios y valoraciones a modo de notas como haría TripJournal, para posteriormente ser compartidas y almacenadas en un servidor Web para su futura descarga y visualización.

Otro de los puntos positivos que tiene este proyecto es el hecho de que las aplicaciones que se basan en las redes sociales no suelen mostrar por defecto los puntos mostrados en un mapa, problema contemplado y resuelto también en esta aplicación.

Otra diferencia a tener en cuenta reside en que el contenido realizado por los usuarios no se almacena en la red social, si no en un servidor Web al cual la aplicación conecta mediante un servicio Web propio. Este detalle permite tener una capa de persistencia que viene dada por la red social utilizada, en este caso Foursquare, pero en caso de ser necesario cambiar esto, el contenido creado por los usuarios no se pierde.

Por último, ampliando este último punto, al no estar ligada a ningún sistema de datos definido, como es el caso de una red social, se pueden incluir más campos a la hora de realizar un *check-in* o descartar en el futuro aquellos que se consideren menos útiles.



## Capítulo 4

# Integración de Foursquare, Geolocalización y Servicios Web

---

Los dispositivos móviles proporcionan un gran número de servicios y funcionalidades a sus usuarios solo accesibles, hasta hace poco tiempo, a los ordenadores de sobremesa. Los más utilizados están relacionados con el ocio y la comunicación: Leer la prensa, leer el correo electrónico, visitar una página Web, jugar a un juego, realizar un comentario en una red social, etc.

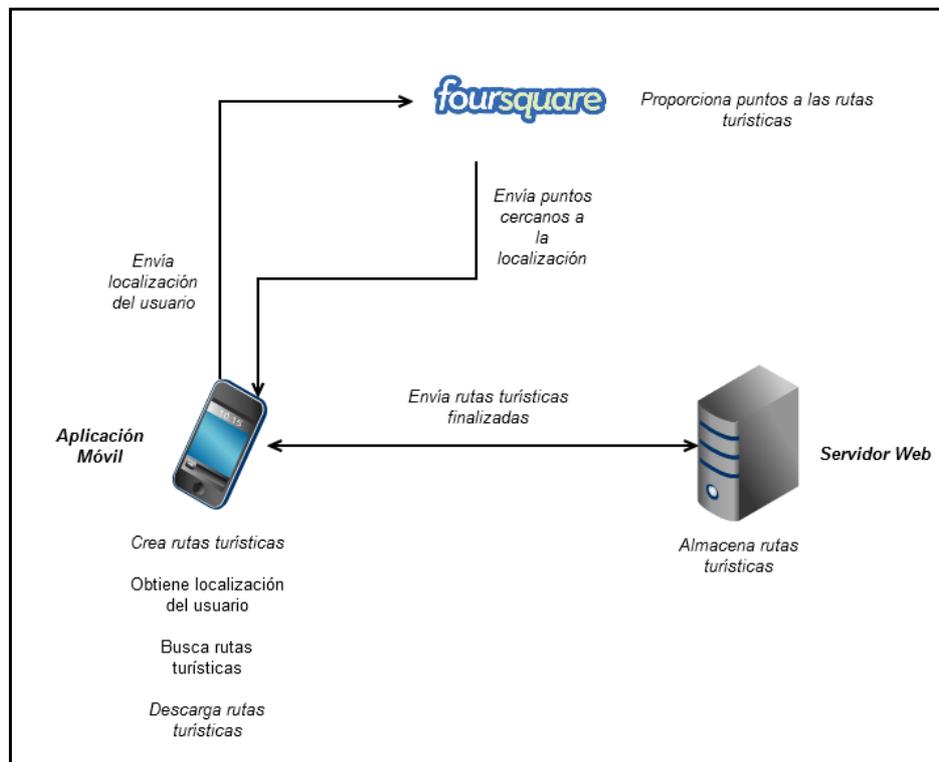
En este trabajo se pretende añadir una nueva funcionalidad a los dispositivos móviles: la creación de rutas turísticas. Esta funcionalidad contempla tres procesos: **creación de las rutas turísticas, finalización y publicación de las rutas en un servidor Web y descarga y seguimiento de las rutas**. Con el fin de poder realizar estos tres procesos, se han integrado en un dispositivo móvil las siguientes tecnologías: **La geolocalización, redes sociales y servicios Web**.

La geolocalización proporciona la localización de un usuario en un momento concreto. Esta localización suele ser plasmada en un mapa, ayudando al usuario a poder observar con total claridad su posición actual. Mediante la geolocalización, la aplicación desarrollada en este trabajo, captura las coordenadas de la localización de un usuario. Una vez obtenidas, son enviadas a una red social especializada en geolocalización para poder obtener información de los lugares y establecimientos cercanos al usuario.

Las redes sociales especializadas en la geolocalización, permiten obtener todos los lugares y establecimientos cercanos a un usuario. Para ello, es necesario proporcionarles las coordenadas de la localización actual del usuario. Entre todas las redes sociales existentes, se ha optado en este trabajo por la red social Foursquare. Los dos motivos principales para esta elección son: (1) la gran cantidad de usuarios registrados en Foursquare, y (2) la posibilidad de que cualquier usuario pueda añadir un nuevo lugar o establecimiento a Foursquare.

Los servicios Web ofrecen la posibilidad de almacenar datos en un servidor, permitiendo que estos datos sean accesibles a un gran número de aplicaciones. Estos servicios cumplen con unos protocolos y estándares para realizar esta función. En este trabajo se ha optado por almacenar las rutas turísticas en un servicio Web por dos razones: (1) permitir que todos los usuarios puedan acceder a las rutas turísticas, (2) poder almacenar los datos que se creen necesarios en el proceso de creación de una ruta turística.

La integración de las tecnologías y sus funciones dentro del sistema se pueden observar en la figura 16.



**Figura 16. Funciones de las tecnologías**

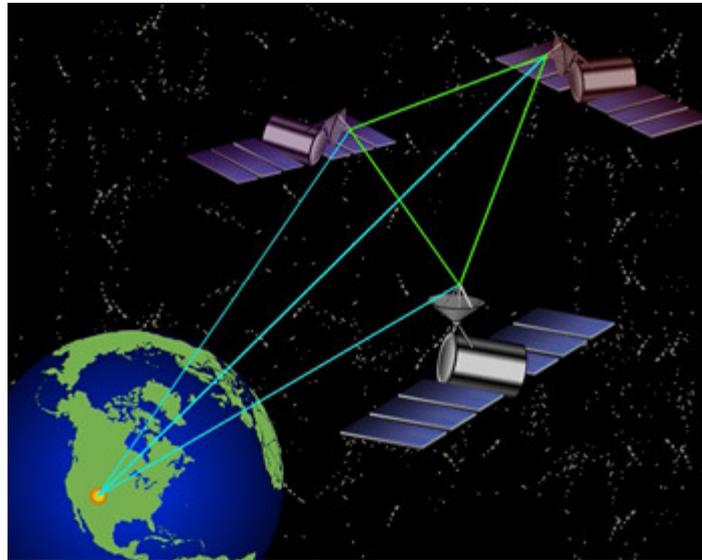
En este capítulo se van a describir las tecnologías utilizadas y el proceso necesario para integrar cada una de ellas. En la sección 4.1 se habla de la geolocalización, una herramienta que permite capturar la posición de un usuario. En la sección 4.2 se expone la red social Foursquare, y el proceso de integración en la aplicación desarrollada en este trabajo. En la sección 4.3 se describen los servicios Web, así como la integración y el papel desarrollado por estos en este trabajo. En la sección 4.4 se muestran un conjunto de conclusiones.

## 4.1. Geolocalización

La geolocalización es una herramienta que permite obtener las coordenadas geográficas de un elemento en cualquier momento. Este proceso se puede realizar mediante el uso de un dispositivo GPS o mediante una conexión a internet.

Los dispositivos GPS son elementos que utilizan la red de satélites especializados en este campo, permitiendo obtener la posición de un elemento en cualquier lugar del mundo. Esta red está compuesta por 27 satélites (24 en funcionamiento y tres extra por si falla alguno de los anteriores), situados todos ellos a una distancia de 12.645 millas (unos 19.300 Km.).(12)

El proceso que siguen los dispositivos GPS para calcular la localización es(12): El dispositivo emite una señal indicando la hora y la información de la localización donde se encuentra. Esta señal es recibida por los satélites GPS, y comparada por tres de ellos, para calcular la posición del elemento. Esta triangulación de satélites se puede observar en la figura 17.



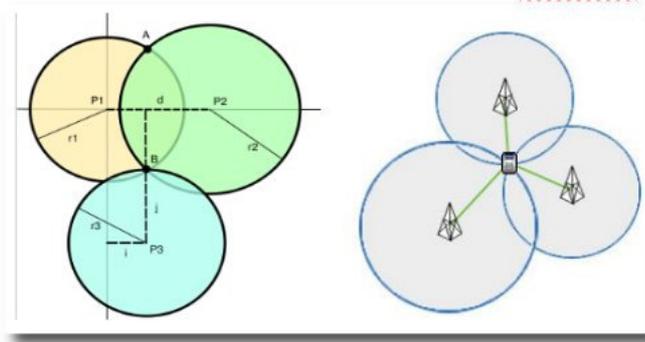
**Figura 17. Triangulación de los satélites GPS**

Los satélites GPS cuentan con sistemas muy precisos que permiten calcular sin error la hora exacta en la que se encuentra. Además, estos dispositivos cuentan con sistemas de monitorización de los satélites para poder obtener con total precisión su posición.

A causa de tener que establecer una primera conexión a la red de satélites, estos sistemas son un poco lentos, pudiendo tardar entre dos o tres minutos en realizar esta primera conexión. Una vez realizada esta conexión, la señal se actualiza con mayor facilidad y permite calcular con una gran precisión (margen de error de centímetros) la localización de cualquier elemento.

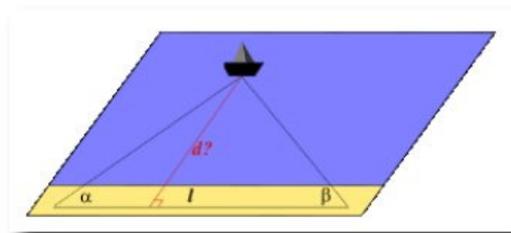
El otro método para obtener la localización de un individuo es mediante internet. Mediante la medición de las señales de radio emitidas por un dispositivo móvil, se puede calcular su localización en función de la intensidad de estas señales. Para ello existen dos técnicas: La **trilateración** y la **triangulación** (13).

La **trilateración** consiste en medir la distancia de las señales para calcular la localización de un dispositivo móvil en un plano bidimensional. Para realizar esto, se utilizan al menos tres antenas 3G, que trazan circunferencias sobre las que se encuentra el punto de medición. Dos circunferencias que interseccionan, lo hacen en dos puntos. La tercera determina el punto donde se encuentra el terminal móvil. La figura 18 representa este cálculo.



**Figura 18. Trilateración**

La **triangulación** utiliza un conjunto de nodos 3G fijos cuya posición es conocida. Cada nodo debe ser capaz de determinar la dirección donde se encuentra el punto a calcular. Para realizar este proceso, se debe captar mediante una antena direccional la señal emitida por el dispositivo móvil. Este proceso permite obtener la dirección de la señal respecto a uno de estos nodos 3G. Para calcular la distancia, se utiliza una segunda antena que se encuentra a una distancia conocida del primer nodo. Esta antena determina la dirección en la que se encuentra la señal del dispositivo. Conociendo las dos direcciones y la dirección y distancia entre las dos antenas, se crea un triángulo. Como también son conocidos los ángulos que continúan a al lado del triángulo que une los dos nodos, mediante operaciones trigonométricas se puede calcular la distancia de los nodos a la señal, que son también dos lados del triángulo. Si el dispositivo móvil estuviera en la recta que forman los dos nodos, sería necesario el uso de un tercer nodo para poder calcular la localización. La figura 19 muestra este proceso.



**Figura 19. Triangulación**

#### 4.1.1. Integración de la geolocalización

La gran mayoría de los dispositivos móviles actuales cuentan con la posibilidad de realizar una conexión a internet. Además de esto, también suelen disponer de un dispositivo GPS integrado. Para poder acceder a ellos en un dispositivo Android, es necesario añadir una serie de permisos a la aplicación. Estos permisos se definen en el archivo *AndroidManifest.xml* de la aplicación. A continuación se muestra cómo se pueden dar permisos para acceder a internet y/o al GPS(10).

Una vez asignados estos permisos, se debe crear una instancia de la clase *LocationManager*(11) que permite calcular la posición del individuo utilizando cualquiera de estas dos variantes. Para indicar mediante que método se desea calcular la posición, se debe indicar cuándo se realice la petición para calcular la localización actual.

## 4.2. Foursquare

Foursquare(3) es una red social especializada en la geolocalización. Mediante la recepción de las coordenadas de longitud y latitud de un usuario, es capaz de mostrar todos los elementos cercanos a este que tenga almacenados en su base de datos. Una vez mostrados estos elementos, los usuarios pueden realizar *check-in* en ellos indicando con ello que han pasado por este lugar.

Foursquare no es la única red de geolocalización existente. La elección de Foursquare para realizar este trabajo viene dada por los siguientes motivos:

**Permite a cualquier usuario añadir un nuevo lugar a su red.** No todas las redes sociales tienen un API de escritura que permita realizar esta funcionalidad. La ventaja que ofrece un API de escritura en el desarrollo de este trabajo reside en que permite al usuario añadir un nuevo punto a su ruta turística que no exista en la red social. Además, permite incrementar la cantidad de información disponible para todos los usuarios.

**Tiene un gran número de usuarios.** Foursquare cuenta con cerca de 8 millones de usuarios (12). Esto asegura que la cantidad de información almacenada en la red social sea mayor gracias a la contribución de todos los usuarios, ya que como se ha indicado en el punto anterior, Foursquare tiene el API de escritura abierta a los desarrolladores.

Foursquare es indispensable en el proceso de creación de rutas turísticas. Para ello, recibe las coordenadas en las que se encuentra un usuario y calcula todos aquellos lugares almacenados en su red. Una vez calculados, estos son devueltos a la aplicación en formato JSON, para ser mostrados al usuario.

A continuación se mostrara el proceso de integración de Foursquare en la aplicación desarrollada. Primero se presenta al lector el sistema de seguridad OAuth, utilizado en Foursquare. Posteriormente, se muestran los pasos necesarios para poder integrar Foursquare en la aplicación desarrollada en este trabajo.

### 4.2.1. Oauth

Antes de poder mostrar el proceso realizado para integrar Foursquare a la aplicación desarrollada en este trabajo, se debe explicar cómo funciona el sistema de seguridad

OAuth(13). Este sistema de seguridad es utilizado en Foursquare y otras redes sociales para gestionar el acceso a su plataforma.

OAuth (Open Authorization) es un estándar abierto que permite incluir seguridad en la autenticación para aplicaciones web y de escritorio. Para crear esta seguridad, OAuth restringe el acceso a los recursos y permite que el usuario acceda a ellos sin tener que utilizar el nombre de usuario ni la contraseña. En lugar de esto, OAuth utiliza unas credenciales que permitirán el acceso a uno o varios recursos por un periodo de tiempo determinado. En el caso de Foursquare, la credencial es un identificador único que tienen cada usuario y mediante el cual se pueden realizar las distintas operaciones solicitadas.

Para mostrar las ventajas de OAuth, veamos el siguiente ejemplo obtenido de la Web <http://www.prise.es/>: *Una aplicación nueva incluye un servicio de avisos de fechas importantes que deberá importar de otra aplicación, por ejemplo, la agenda del gestor de correo. Para poder activar el servicio es necesario que el sujeto en cuestión le dé su nombre de usuario y contraseña para así acceder al gestor de correo. El resultado obtenido es que ambas aplicaciones funcionan correctamente, pero existe un inconveniente primordial: la aplicación que da el servicio de avisos ya tiene su contraseña, lo cual puede llevar a un problema de seguridad grave.*

*OAuth intenta solucionar esta problemática **restringiendo** el **acceso** a los **recursos**. En el ejemplo anterior, mediante OAuth se habría conseguido que la aplicación servidora sólo pudiera acceder a la información del calendario, evitando así tener que darle el nombre de usuario y contraseña y que hubiera un acceso ilimitado a la información contenida en el gestor de correo.*

#### **4.2.2. Integración de Foursquare**

Foursquare basa su sistema de seguridad en OAuth. Esto hace necesario obtener la credencial de un usuario para que pueda acceder a los recursos de la red social. Una vez obtenida esta credencial, mediante un conjunto de peticiones HTTP, se realizan todos los accesos a la plataforma de Foursquare.

En esta sección se explica el proceso de integración de Foursquare. En primer lugar se muestra la obtención de la credencial necesaria para poder acceder al sistema de seguridad de Foursquare.

##### **Obtención de las credenciales de Foursquare**

En la figura 20 se expone un Workflow obtenida de la Web <http://www.returngis.net/en> en el cual se muestra el proceso para las credenciales necesarias para acceder Foursquare.

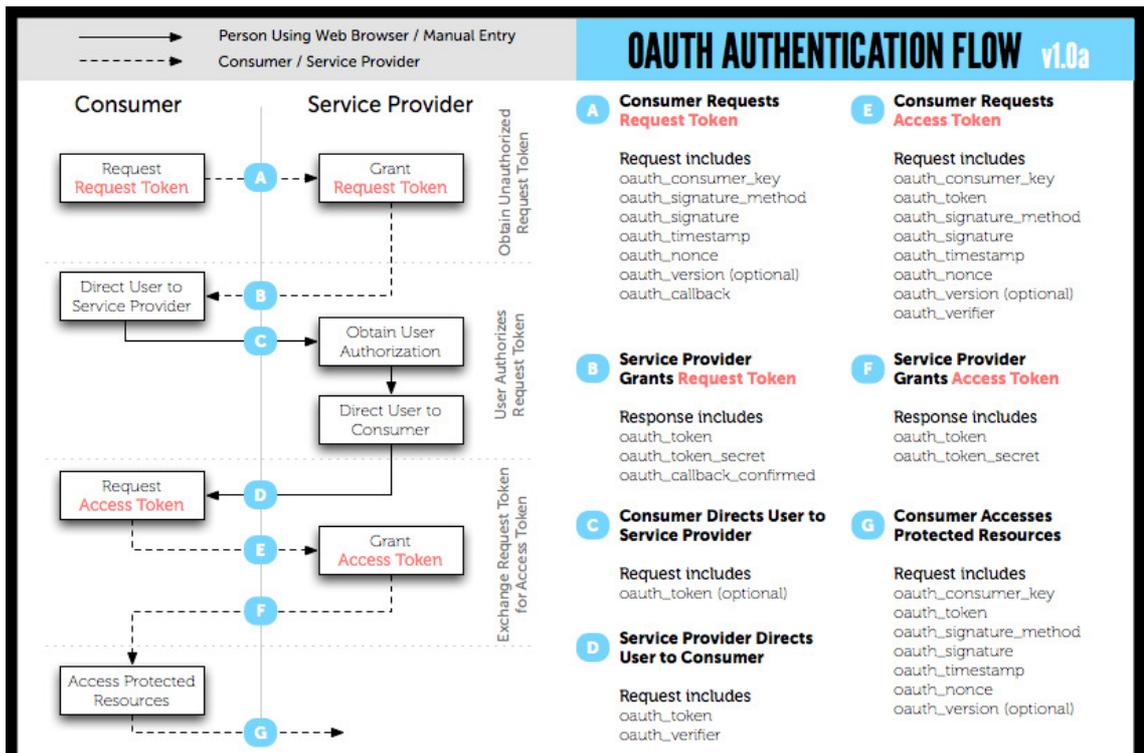


Figura 20. Workflow Oauth

En este Workflow se pueden observar un conjunto de pasos para obtener las credenciales. Previos a estos pasos, el desarrollador debe **registrar** la aplicación en Foursquare. Para registrar la aplicación, un desarrollador debe insertar los siguientes datos: el **nombre de la aplicación**, la **Web de la aplicación** y una dirección web llamada **Callback URL**. La **Callback URL** es utilizada en el proceso de autenticación. Cuando se realiza la petición del token temporal, Foursquare re direcciona al usuario a esta URL añadiendo el token solicitado. Por ejemplo, si la dirección registrada fuese [www.direccion.es](http://www.direccion.es), la respuesta de Foursquare seria [www.direccion.es/token](http://www.direccion.es/token). Una vez registrada esta aplicación, Foursquare(14) proporcionará dos credenciales necesarias para realizar el proceso de autenticación: **Consumer Key** y **Consumer Secret**. Estas credenciales son únicas para cada aplicación.

El primer paso para realizar la autenticación consiste en **solicitar un token temporal**. El token temporal es una credencial que se crea en el intercambio de información entre Foursquare y el usuario. La obtención de este token inicia el proceso de autenticación. Para obtener este token, se debe realizar una petición HTTP con los siguientes parámetros: **Consumer Key** y **Callback URL**. Al realizar la petición del token temporal, el usuario debe estar identificado con su usuario y contraseña. En caso de no estarlo, Foursquare solicitará estos datos.

Una vez obtenido el token temporal, se debe realizar un **intercambio del token temporal por un Access Token**. El Acces Token es la credencial correspondiente a

cada usuario. Esta credencial es única y cada usuario dispone de una. Una vez el usuario ha obtenido el token temporal, debe realizar una petición a Foursquare para obtener el Acces Token. En esta petición debe incluir los siguientes parámetros: **Consumer Key**, **Consumer Secret**, **Callback URL** y el token temporal obtenido anteriormente.

### Peticiones a Foursquare

Una vez obtenido el AccesToken, Foursquare permite realizar una serie de peticiones HTTP mediante las cuales se acceden a los servicios proporcionados por esta red social. La figura 21, obtenida de la API de Foursquare (15) , muestra el conjunto de peticiones que se pueden realizar a Foursquare.

	General	Aspects	Actions
users	leaderboard search requests	badges checkins friends tips todos venuehistory	request unfriend approve deny setpings
venues	add categories explore search trending	herenow tips photos links	marktodo flag edit proposeedit
checkins	add recent		addcomment deletecomment
tips	add search		marktodo markdone unmark
photos	add		
settings	all		set
specials	search		

Figura 21. Resumen peticiones Foursquare

En ella se puede observar como las peticiones están agrupadas según el tipo de información que se quiera obtener.

### **4.3. Servicios Web**

La última tecnología utilizada en este trabajo son los servicios Web. Los servicios Web permiten almacenar información y proporcionan funcionalidades adicionales a las aplicaciones. Por ejemplo, un servicio Web podría ofrecer el estado meteorológico de la ciudad de Valencia. Los servicios Web se encuentran alojados en servidores accesibles a través de internet.

La principal ventaja que ofrecen los servicios Web reside en que cualquier aplicación pueda acceder a ellos, ya que como se ha indicado anteriormente, se puede acceder a ellos a través de internet. Esto permite crear una capa de almacenamiento y de funcionalidad accesible a los programadores que tengan acceso a ellos, permitiendo incrementar los servicios ofrecidos por cualquier aplicación desarrollada por ellos.

La aplicación desarrollada en este trabajo utiliza un servicio Web para realizar dos de sus procesos:

- Creación de las rutas turísticas. El servicio Web almacena todos los comentarios y valoraciones realizados por los usuarios acerca de los lugares visitados. Esta información es accesible a la hora de crear una ruta de viaje, permitiendo a un usuario tener una mayor información sobre un lugar.
- Finalización y publicación de las rutas en un servicio Web. Una vez el usuario ha completado su ruta de viaje, puede finalizarla y enviarla al servicio Web desarrollado en este trabajo. Estas son almacenadas para su futura descarga tanto por él como por el resto de usuarios.
- Descarga y seguimiento de las rutas. Cualquier usuario puede acceder al servicio Web para descargar rutas de viaje. Mediante una petición a este, el usuario descargara estas rutas y podrá visualizarlas en su dispositivo móvil.

A continuación se muestra el proceso de integración del servicio Web en la aplicación desarrollada en este trabajo.

#### **4.3.1. Integración del Servicio Web**

Como se ha indicado anteriormente, el Servicio Web permite tanto almacenar las rutas turísticas realizadas por los usuarios, como descargar éstas para ser vistas y seguidas en el dispositivo móvil.

El servicio cuenta con una base de datos MySQL. Esta base de datos es un espejo de la base de datos local que tiene el dispositivo móvil, encargada de almacenar las rutas turísticas, los puntos que las componen y los datos del usuario que las ha creado.

Al igual que ocurre en Foursquare, para acceder a los datos ofrecidos por el Servicio Web, se realizan peticiones HTTP. Estas peticiones están compuestas por un conjunto de parámetros que permiten indicar que datos deben ser transmitidos entre el Servicio Web y el dispositivo móvil. Las tres funcionalidades que componen el Servicio Web son:

- **Almacenar rutas.** Permite al usuario almacenar sus rutas y todos los puntos asociados a esa ruta. Además de esto, también se envía información referente al usuario que crea las rutas, para almacenarla y poder ser consultada en un futuro. Los datos son enviados en formato JSON y posteriormente, interpretados en el servidor para su almacenamiento en la base de datos. Una vez finalizado este proceso, devuelve un mensaje de confirmación para indicar que la operación se ha realizado con éxito.
- **Buscar rutas.** Con un conjunto de parámetros introducidos en el dispositivo el usuario es capaz de buscar rutas en el servidor que serán automáticamente descargadas. Entre estos parámetros, se encuentra la opción de buscar rutas de otros usuarios o del mismo usuario que realiza la búsqueda, permitiendo con esto recuperar aquellas rutas turísticas que se realizaron en el pasado.
- **Buscar más información sobre un punto.** Una vez descargados los puntos en el mapa, el usuario podrá ver la información que otras personas han introducido y subida al servidor a través de sus rutas turísticas. Esto permite obtener la opinión y la valoración de los usuarios sobre un lugar en concreto para tener una mayor valoración sobre este lugar y decidir si quieren visitarlo o no. Los datos devueltos no incluyen las valoraciones y comentarios realizados por el usuario que ha realizado la petición.

#### 4.4. Conclusiones

En este capítulo se han expuesto las tecnologías utilizadas en el desarrollo de este trabajo y el proceso de integración de estas en la aplicación desarrollada. Como se ha podido observar, cada tecnología se integraba de manera distinta:

Para poder integrar la geolocalización, es necesario conceder permisos de acceso a la aplicación de los recursos de internet y/o del GPS del dispositivo móvil. Estos permisos son concedidos en el archivo *AndroidManifest.xml* de la aplicación. Una vez realizado esto, se debe crear una instancia de la clase *LocationManager* que permite calcular la posición del usuario con cualquiera de estos dos elementos.

Para integrar Foursquare, antes se deben obtener las credenciales necesarias para poder acceder. Estas credenciales vienen indicadas por el sistema de seguridad OAuth y

son únicas para cada usuario. Una vez se han obtenido, mediante un conjunto de peticiones HTTP, el usuario puede acceder a los recursos ofrecidos por Foursquare.

Por último, para integrar el sistema Web, es necesario conocer los parámetros de interacción con este. Estos parámetros ayudan al intercambio de información entre el servicio Web y el dispositivo móvil. Una vez se conocen estos parámetros, mediante tres peticiones HTTP distintas, se puede realizar este intercambio de información.

La complejidad en la integración de estas tecnologías no es muy elevada, siendo el proceso más difícil, aquel que permite obtener las credenciales de Foursquare.



## Capítulo 5

# Estructura y Funcionalidad del Sistema

---

A lo largo de este documento ya se ha indicado cual es el objetivo principal que debe realizar la aplicación desarrollada en este trabajo: crear rutas turísticas. Con este fin, se han integrado varias tecnologías, la geolocalización, las redes sociales y los Servicios Web, en un dispositivo móvil. Pero, ¿Cómo están estructuradas estas tecnologías? ¿Qué funcionalidades realiza el sistema para crear rutas turísticas?

Los tres procesos principales del sistema son: creación de rutas turísticas, finalización y publicación de las rutas turísticas en el servicio Web, descarga y seguimiento de las rutas turísticas.

El proceso de creación de rutas turísticas se inicia cuando el usuario rellena el formulario que crea una nueva ruta. Una vez rellenado, accede a un mapa que realiza tres funciones: (1) calcular la localización del usuario, (2) enviar estas coordenadas a Foursquare para calcular los puntos cercanos, (3) mostrar los lugares devueltos por Foursquare. Una vez mostrados estos lugares, el usuario puede añadirlos a su ruta turística junto a los comentarios realizados por él. Cuando el usuario finalice la ruta, ésta es publicada en un servicio Web. En este proceso el sistema cierra la ruta, evitando con ello que pueda ser ampliada en un futuro. Una vez finalizada, envía los datos de la ruta, junto a los puntos que la componen y los datos del usuario, al servicio Web. El servicio recibe estos datos, los procesa y almacena en su base de datos para futuras consultas. El proceso de descarga y seguimiento de las rutas turísticas, permite al usuario descargar las rutas del servicio Web y poder visualizarlas en su dispositivo móvil. El proceso se inicia cuando el usuario rellena el formulario de búsqueda de rutas con una serie de parámetros. Una vez rellenado, se realiza una petición al servicio Web, en la que se adjuntan estos parámetros para filtrar las rutas. El servicio Web procesa esta información y busca aquellas rutas que cumplan con los requisitos. Una vez encontradas, son devueltas al dispositivo móvil, el cual las almacena en su base de datos.

Para realizar estos tres procesos, se ha contado con las siguientes tecnologías: La geolocalización, la red social Foursquare y un servicio Web. Estas tres tecnologías han sido integradas en un dispositivo móvil Android. Este conjunto de elementos, están estructurados en un sistema clásico de tres capas. La capa de presentación, proporciona al usuario una interfaz que permite interactuar con el sistema. Esta capa está completamente desarrollada en el dispositivo móvil Android. La capa de lógica,

gestiona toda la funcionalidad del sistema. En esta capa se encuentra la geolocalización, integrada en el dispositivo móvil. Esta capa está completamente gestionada por el dispositivo móvil Android. La capa de persistencia, gestiona la información y los datos del sistema. Está dividida entre: El dispositivo móvil Android, que proporciona una base de datos local para almacenar las rutas de viaje creadas por el usuario y las descargadas para su seguimiento. La red social Foursquare, que proporciona aquellos lugares y establecimientos que pueden ser añadidos a una ruta de viaje. El servicio Web, que almacena las rutas turísticas de todos los usuarios.

En este capítulo se describe tanto la funcionalidad del sistema, como su estructura. En la sección 5.1 se muestra la estructura del sistema en una arquitectura de tres capas destacando la capa de lógica, que muestra la funcionalidad del sistema. En la sección 5.2 se exponen las conclusiones del capítulo.

## 5.1. Estructura del sistema

La estructura del sistema desarrollado en este trabajo, se basa en una arquitectura de tres capas [8]. Estas capas gestionan tanto la interacción del usuario con el sistema, como el funcionamiento de este sistema y los datos utilizados en el. Para ello, el sistema está dividido en estas tres capas:

- **Capa de presentación.** Contempla todas las actividades que permiten al usuario interactuar con el sistema. Una actividad Android es un elemento que presenta una funcionalidad al usuario y permite interactuar con ella a través de su interfaz gráfica. Entre estas interfaces se puede destacar el mapa de la aplicación, que permite mostrar al usuario su localización y la de aquellos lugares y establecimientos cercanos a ella.
- **Capa de lógica.** Gestiona la funcionalidad del sistema. Esta funcionalidad está compuesta por actividades Android. Entre estas funcionalidades se puede destacar la obtención de la localización del usuario. Esta localización es calculada mediante la conexión a internet del dispositivo móvil.
- **Capa de persistencia.** La capa de persistencia se encarga de almacenar los datos en la base de datos del sistema y gestionar sus accesos. Esta capa está compuesta por tres elementos: (1) base de datos del dispositivo móvil Android, que permite almacenar las rutas de viaje creadas por el usuario y las descargas de otros usuarios; (2) la red social Foursquare, que mediante la localización obtenida en la capa de lógica, permite obtener todos aquellos lugares y establecimientos cercanos; y (3) la base de datos residente en el servicio Web, que se encarga de almacenar las rutas turísticas para su futura descarga.

En esta sección se describen estas capas. Primero se expone la capa de presentación mediante unos diagramas realizados con la herramienta Moskiitt(16). En segundo lugar se habla de la capa de lógica, mediante la explicación de los casos de uso que componen

la funcionalidad del sistema y con un diagrama de clases UML. Por último, se muestra la capa de persistencia, mostrando los datos y los accesos a estos, así como un diagrama de entidad-relación.

### 5.1.1. Capa de presentación

La capa de presentación es la encargada de proporcionar un conjunto de interfaces gráficas que permitan la interacción al usuario con el sistema. En un dispositivo móvil Android, estas interfaces se definen mediante archivos *xml*, que posteriormente son incluidas en las actividades que las utilizan. Estos archivos presentan un parecido con las hojas de estilo en cascada, *css*, ya que el conjunto de comandos que permiten definir la posición de los elementos son parecidos a los utilizados en estas hojas de estilo.

En la figura 22 se muestra un resumen de todas las interfaces gráficas del sistema:

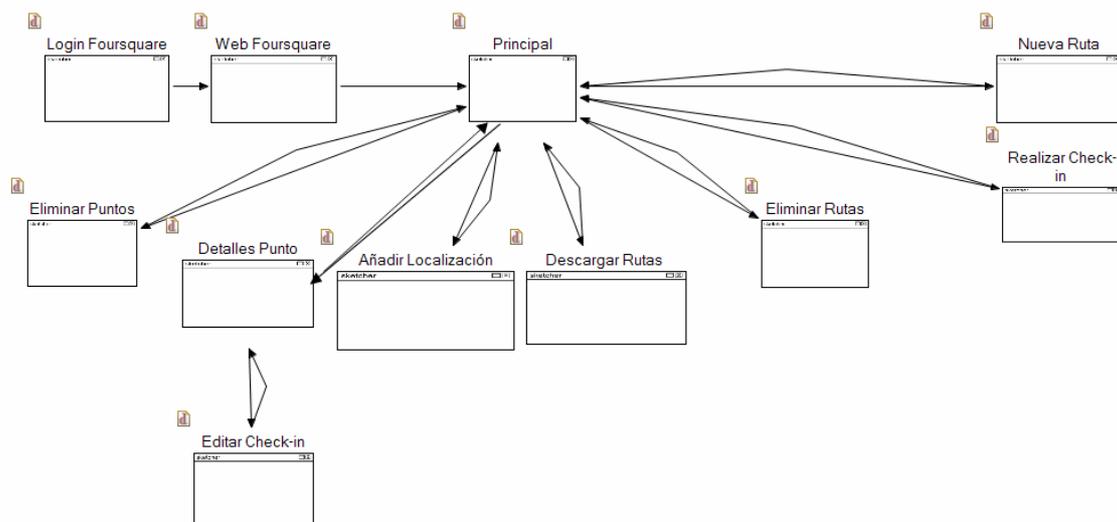
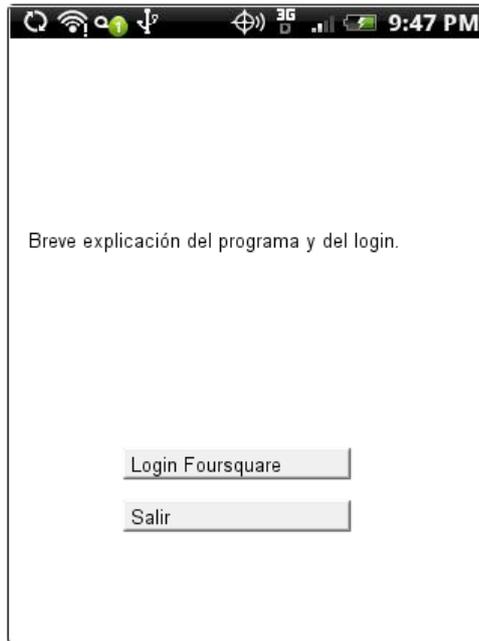


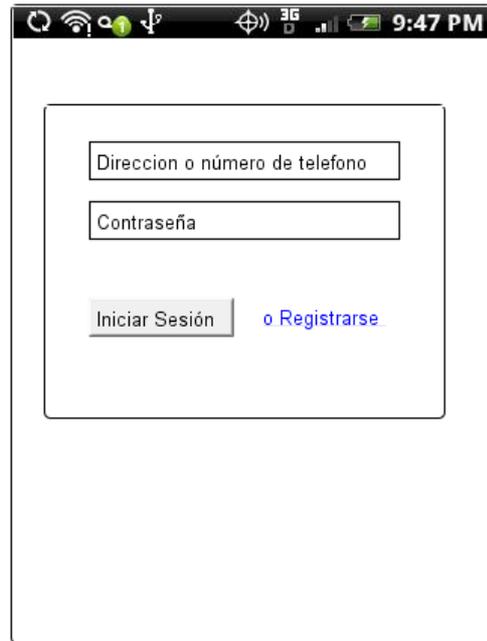
Figura 22. Resumen interfaces gráficas

En ella se puede observar como la interfaz con el nombre *principal* se encarga de proporcionar acceso a todas las demás. Para acceder a esta interfaz, el usuario previamente debe identificarse en la aplicación a través de sus credenciales de Foursquare.

Las interfaces gráficas que gestionan este proceso se pueden ver en las figuras 23 y 24.



**Figura 23. Login Foursquare**



**Figura 24. Web Foursquare**

La figura 18 muestra la interfaz gráfica inicial de la aplicación. Esta interfaz permite realizar dos acciones, identificarse en Foursquare o salir. Si el usuario decide identificarse en Foursquare, el dispositivo móvil abre una nueva ventana de navegador que lleva al usuario a la página Web que le permite identificarse en Foursquare. Esta ventana del navegador se puede observar en la figura 24. El usuario debe rellenar los campos con sus datos e iniciar sesión. En caso de que no disponga de una cuenta Foursquare, se le permite crear una nueva cuenta. Una vez identificado, la aplicación almacena su Acces Token, y muestra la interfaz *principal* del sistema. En caso de que el usuario decida pulsar el botón salir de la figura 23, la aplicación se cerrará.

Con el fin de estructurar un poco mejor esta sección, se van a comentar las interfaces gráficas que componen los tres procesos principales del sistema: **Creación de rutas turísticas, finalización y publicación de las rutas en el servicio Web y descarga y seguimiento de las rutas.** Todos estos procesos partirán de la interfaz gráfica *principal*, que se comenta también a continuación.

## Interfaz Principal

La interfaz gráfica *principal*, es la encargada de gestionar todas las funcionalidades del sistema. En caso de que el usuario quiera realizar una funcionalidad no incluida en la interfaz principal, esta realizara una llamada a la interfaz que permita realizarla.

La interfaz gráfica *principal* está compuesta por tres pestañas. Estas pestañas se pueden observar en las figuras 25, 26 y 27.

La pestaña rutas, permite al usuario crear y descargar las rutas turísticas. Está compuesta por dos listados, uno referente a las rutas creadas por el usuario y otro que muestra las rutas descargadas del servicio Web pertenecientes al resto de usuarios. Para acceder a cada uno de estos listados, el usuario dispone de dos botones, *Mis rutas* y *Otras rutas*.

Cada uno de estos listados cuenta con un menú. El menú es común a la pestaña *Mis rutas*, pero según el listado en el que se encuentre el usuario, mostrara unos elementos u otros. Para el listado que muestra las rutas turísticas del usuario, el menú muestra dos opciones: nueva ruta y eliminar rutas. En el caso del listado de las rutas descargadas pertenecientes a otros usuarios, el menú muestra las siguientes opciones: descargar rutas y eliminar rutas. Como ya se ha indicado, la figura 25 muestra el contenido de esta pestaña.

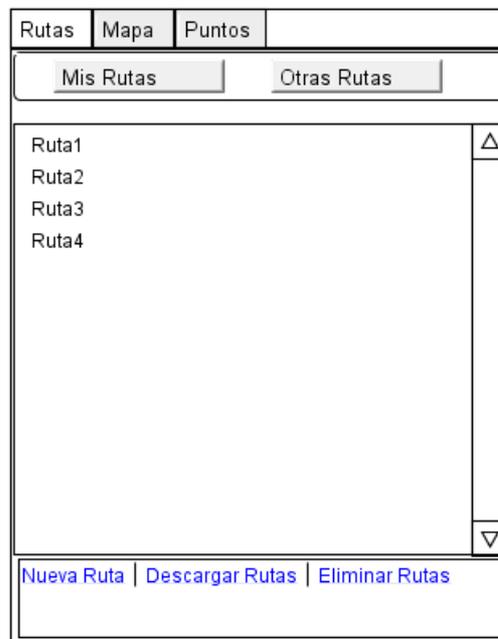


Figura 25. Pestaña rutas

La siguiente pestaña de la interfaz *principal* muestra al usuario el mapa donde puede observar su localización y aquellos elementos obtenidos de la red social Foursquare. Este mapa lo proporciona el API de Google, mediante su plataforma Google Maps (21). En la figura 26 se puede observar el contenido de esta pestaña:



**Figura 26. Pestaña mapa**

Como se puede observar en la figura, la pestaña está compuesta por varios elementos. En primer lugar, existen dos botones, los cuales permiten al usuario continuar una ruta de viaje, en caso de no estar finalizada o visualizar una ruta de viaje, que permite plasmar en el mapa aquellos puntos que componen la ruta y seguirlos en orden.

Otro elemento que compone esta pestaña es el menú. En función de si se está continuando la ruta o visualizándose, el menú muestra un contenido u otro. En la figura 26 solo se muestra el contenido del menú perteneciente a la funcionalidad de continuar rutas. En él se pueden observar 4 opciones: Puntos cercanos, que refresca la localización del usuario y los puntos cercanos a él. Filtrar puntos, que muestra un menú contextual que permite filtrar los puntos mostrados según la elección del usuario. Por ejemplo, una opción del menú contextual es *cafeterías*. Si el usuario pulsa esta opción, el mapa solo muestra las cafeterías cercanas al usuario. Añadir localización, que permite añadir una nueva localización a la red de Foursquare. Finalizar ruta, que finaliza y publica la ruta turística en el servicio Web.

El menú perteneciente a la opción de visualizar las rutas, está compuesto por tres elementos: Primer punto, siguiente punto, último punto. La opción primer punto, posiciona al usuario en la localización donde se encuentra el primer punto añadido a la ruta turística. La opción siguiente punto, posiciona al usuario en el siguiente punto

añadido a la ruta de viaje al menos que este sea el último. La opción último punto, posiciona al usuario en la localización del último punto añadido a la ruta turística. Con estas tres opciones, la aplicación permite al usuario realizar un seguimiento de la ruta turística creada por cualquier usuario.

El último elemento de esta pestaña, es un menú contextual que proporciona al usuario dos opciones: añadir un nuevo punto a la ruta, mediante la opción check-in y obtener información adicional creada por otros usuarios, mediante la opción información. Este menú es desplegado cuando un usuario pulsa sobre un punto obtenido de Foursquare.

La última pestaña que compone la interfaz *principal*, puntos, se expone en la figura 27. Esta pestaña muestra un listado de todos los puntos añadidos a una ruta turística. Si el usuario pulsa en cualquier elemento de este listado, el sistema mostrara una nueva interfaz que permite ver los detalles del punto. Estos detalles están compuestos por los datos introducidos por el creador de la ruta turística, así como por un mapa que muestra la localización de este punto. Esta interfaz cuenta con un menú que permite realizar dos opciones si la ruta no está finalizada: Eliminar el punto de la ruta turística o editarlo para corregir los datos introducidos por el usuario. Esta interfaz se puede observar en la figura 28. Además de proporcionar el listado de los puntos, la pestaña cuenta con un menú que permite eliminar puntos de la ruta turística. Este menú solo será mostrado al usuario si la ruta no está finalizada.

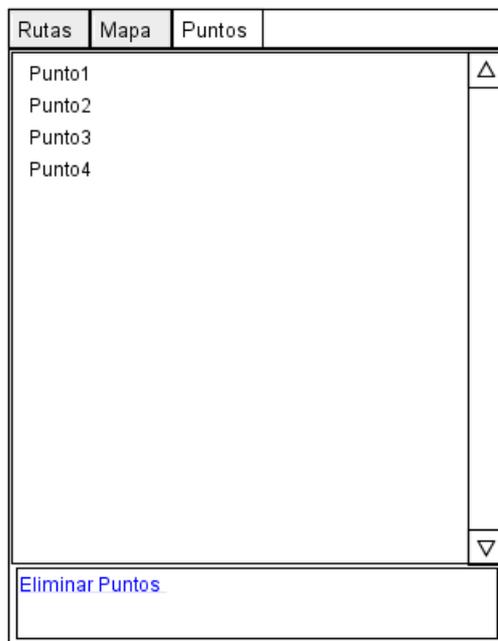


Figura 27. Pestaña puntos

Datos del Lugar:

Nombre del Lugar

Tipo de Lugar

Valoración del Lugar:

Precio por Persona

Valoración

Comentario

Localización:

Editar Punto | Eliminar Punto

**Figura 28. Detalles punto**

### Creación de rutas turísticas

El proceso de creación de rutas turísticas empieza cuando el usuario pulsa la opción del menú mostrado en la figura 29, Nueva ruta. Mediante esta opción, el usuario accede a un formulario que permite crear una nueva ruta, mostrado en la figura 29.

Datos de la Ruta:

Nombre de la Ruta

Ciudad de Origen

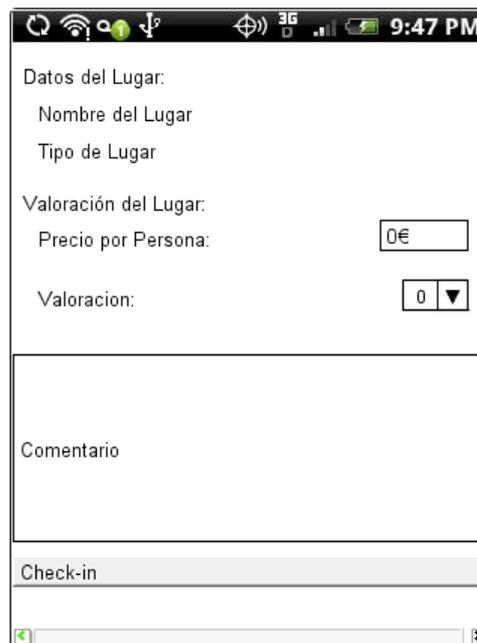
Tipo de Ruta:

Cultural ▼

Nueva Ruta

**Figura 29. Nueva ruta**

Una vez el usuario ha rellenado este formulario, el sistema crea una nueva ruta y la almacena en su base de datos. Para añadir un nuevo punto a la ruta, el usuario debe acceder a la pestaña Mapa, mostrada en la figura 30. Una vez se han calculado todos los puntos cercanos al usuario a través de su localización y de la red social Foursquare, el usuario podrá pulsar en cualquiera de ellos y seleccionar la opción Check-in, que permite añadir un nuevo punto. Para ello, el usuario debe rellenar una serie de campos de un formulario en el que se indica, entre otros elementos, un comentario acerca de este lugar realizado por el usuario. Este formulario se puede observar en la figura 30.



The image shows a mobile application interface for performing a 'check-in'. At the top, there is a status bar with icons for Wi-Fi, signal strength, and battery, along with the time 9:47 PM. The main form is divided into several sections:

- Datos del Lugar:** This section contains two text input fields: 'Nombre del Lugar' and 'Tipo de Lugar'.
- Valoración del Lugar:** This section contains two input elements: 'Precio por Persona:' with a text box containing '0€', and 'Valoración:' with a dropdown menu showing '0' and a downward arrow.
- Comentario:** A large text area for entering a comment.
- Check-in:** A button labeled 'Check-in' at the bottom of the form.

At the very bottom of the screen, there is a standard Android navigation bar with a back arrow on the left and a right-pointing arrow on the right.

**Figura 30. Realizar “check-in”**

Si el usuario desea añadir un punto a su ruta turística, el cual no pertenece a la red social Foursquare, antes debe crearlo. Para ello, el usuario debe acceder a la opción de menú, Añadir localización, y rellenar el formulario mostrado en la figura 31.

Datos de la Localización:

Nombre de la localización

Ciudad

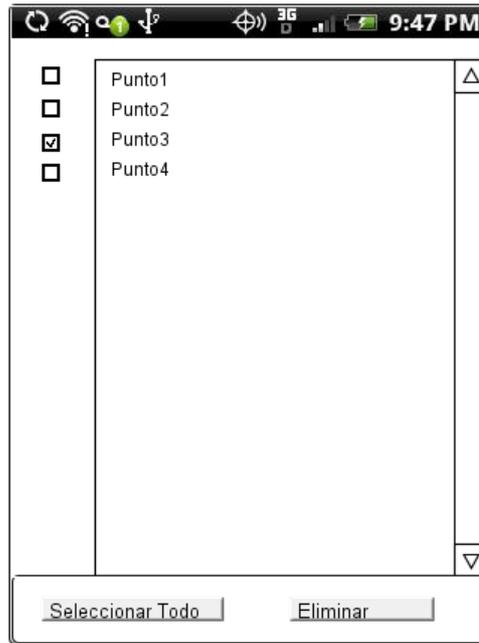
Tipo de Localización:

Bar

Nueva Localización

**Figura 31. Añadir localización**

Si el usuario desea borrar un punto de la ruta, debe acceder a la pestaña Puntos, mostrada en la figura 32. Una vez en ella, el usuario debe pulsar la opción del menú Eliminar puntos. Esta opción muestra un listado compuesto por elementos de tipo *checkbox* que permiten seleccionar los puntos a eliminar y dos botones: Seleccionar todo, que selecciona todos los puntos que componen la ruta y Eliminar, que elimina los puntos seleccionados. Esta interfaz se puede observar en la figura 32.



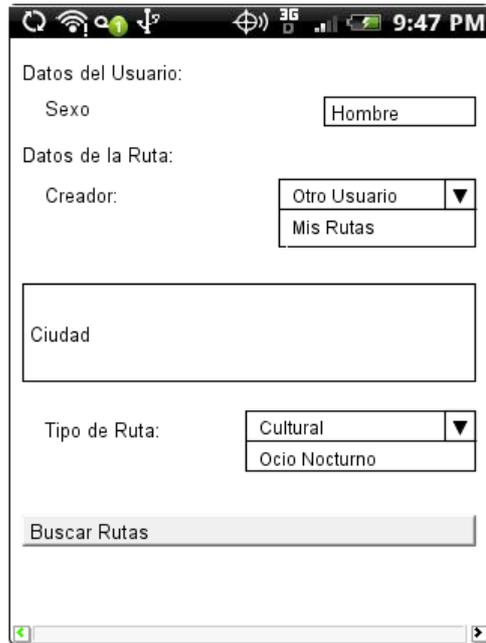
**Figura 32. Eliminar puntos**

### **Finalizar rutas**

El proceso de finalizar una ruta de viaje no contempla un conjunto de interfaces. Para poder finalizar una ruta, el usuario simplemente debe acceder al menú de la pestaña Mapa, mostrado en la figura 26. Una vez pulsada la opción, el sistema finalizada la ruta y la publica en el servicio Web.

### **Descargar rutas**

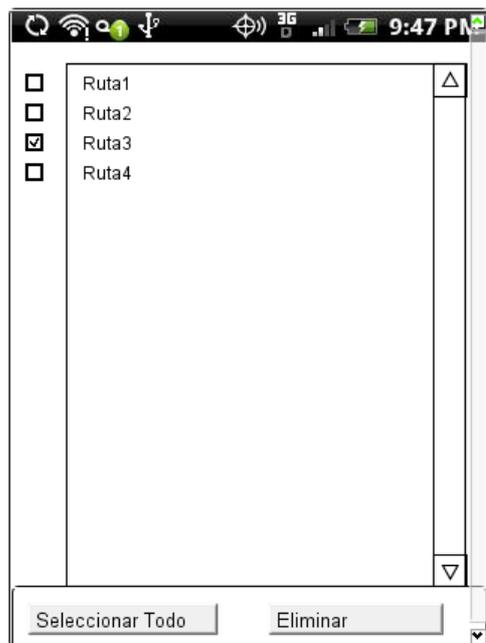
La figura 33 muestra la interfaz gráfica que permite descargar rutas. Para acceder a esta interfaz, el usuario debe pulsar en la opción descargar rutas de la pestaña Rutas, mostrada en la figura 25. Una vez el sistema muestre esta interfaz, el usuario debe rellenar un formulario con los campos necesarios para que el servicio Web pueda buscar las rutas.



**Figura 33. Descargar rutas**

### Interfaces adicionales

El sistema cuenta con una interfaz adicional, que permite eliminar las rutas turísticas al usuario. Esta interfaz es accesible mediante el menú de la pestaña Rutas. Una vez pulsada esta opción, el usuario accede a la interfaz mostrada en la figura 34.



**Figura 34. Eliminar rutas**

En ella se puede observar un listado compuesto por *checkboxes*, que permiten seleccionar aquellas rutas almacenadas en el sistema. Además del listado, la interfaz también cuenta con dos botones: **seleccionar todo**, que permite seleccionar todas las rutas turísticas y **eliminar**, que elimina del sistema las rutas seleccionadas.

### 5.1.2. Capa de lógica

La capa de lógica del sistema, define la funcionalidad de este. En una aplicación Android, esta funcionalidad viene definida por aquellas actividades que la componen. Una actividad es un elemento que muestra al usuario una interfaz gráfica que permite interactuar con él. Estas actividades se comunican entre ellas a través de llamadas. Para gestionar que actividad se ejecuta en cada momento, Android dispone de una pila de actividades que ejecuta en primer plano la última actividad lanzada por el sistema. Todas las otras actividades están en reposo en la pila, esperando a que el sistema termine con la actividad que se está ejecutando.

En esta sección del capítulo, se muestra la funcionalidad del sistema. Esta funcionalidad viene dada por tres procesos: Creación de rutas turísticas, finalización y publicación de las rutas en un servicio Web y descarga y seguimiento de las rutas turísticas. Como información adicional, se expone un diagrama UML que muestra las clases del sistema.

#### Creación de rutas turísticas

Este proceso contempla una serie de pasos que permiten al usuario crear rutas turísticas.

En el primer paso, el usuario realiza una petición al sistema para crear una nueva ruta turística. Para ello, el usuario debe rellenar los campos que componen el formulario que permite crear la ruta. En el segundo paso, el usuario debe acceder al mapa de la aplicación para que el sistema calcule la localización en la que se encuentra. En el tercer paso, el sistema realiza una petición a la red social Foursquare enviándole las coordenadas obtenidas anteriormente. En el cuarto paso, Foursquare recibe estas coordenadas, y realiza una búsqueda en su base de datos de todos aquellos puntos cercanos. En el quinto paso, Foursquare devuelve al dispositivo móvil aquellos puntos obtenidos. En el sexto paso, el sistema procesa estos puntos y los muestra en el mapa, junto a la posición actual del usuario. En el séptimo paso, el usuario ya puede añadir puntos a su ruta turística rellenando un formulario que permite incluir la valoración y comentarios de un lugar que tiene el usuario. Este conjunto de pasos se pueden repetir a partir del segundo paso si el usuario cambia su localización. En caso de que un usuario quiera añadir más de un punto cercano a su localización, el usuario debe repetir el último paso.

## **Finalización y publicación de las rutas en un servicio Web**

Este proceso contempla todos los pasos necesarios que permiten al usuario dar por finalizada la ruta de turística, enviando los datos de esta al servicio Web, para ser almacenados.

El primer paso consiste en realizar la petición al sistema para finalizar la ruta. En ese momento, el sistema obtiene la fecha actual para poder incluirla en la ruta turística. Una vez ha obtenido esta fecha, en el segundo paso, el sistema crea una cadena JSON con los datos necesarios que deben ser almacenados: Los **datos de la ruta turística**, los **puntos que componen esta ruta** y los **datos del usuario** que crea la ruta. Esta cadena es enviada al servicio Web en el tercer paso para que los datos sean almacenados. En el cuarto paso, el servicio Web recibe estos datos, procesa la cadena y almacena los datos en su base de datos: (1) primero comprueba si el usuario existe en la base de datos del servicio Web, en caso de no existir almacena sus datos, (2) después almacena los datos correspondientes a la ruta turística, (3) finalmente, almacena todos los puntos añadidos a esta ruta. Una vez los datos han sido almacenados, en el quinto paso, el sistema envía un mensaje de confirmación al dispositivo móvil. Cuando el dispositivo móvil recibe esta confirmación, finaliza la ruta evitando con ello que se puedan añadir puntos adicionales.

## **Descarga de rutas turísticas para su seguimiento**

Este proceso permite al usuario descargar rutas del servidor. Contempla los pasos desde que el usuario realiza la petición para descargar las rutas turísticas, hasta que el sistema las almacena en su base de datos.

En el primer paso el usuario debe realizar una petición al sistema para poder descargar las rutas turísticas. Una vez realizada la petición, el sistema muestra al usuario un formulario que debe rellenar. En el segundo paso, el sistema realiza una petición al servicio Web, enviando los datos rellenados en el formulario como parámetros. En el tercer paso, el servicio Web procesa esta petición y obtiene todas las rutas que cumplan con la petición. Una vez se han obtenido, en el cuarto paso, son enviadas al dispositivo móvil, que procesa esta información y almacena las rutas en su base de datos.

## **Diagrama UML**

Un diagrama UML es un esquema que permite mostrar las clases que componen el sistema de una aplicación. En este trabajo, el diagrama UML es muy reducido, ya que cuenta con solo tres clases que gestionan todo el sistema: Usuarios, Rutas y Checkins. Este diagrama se puede observar en la figura 35.

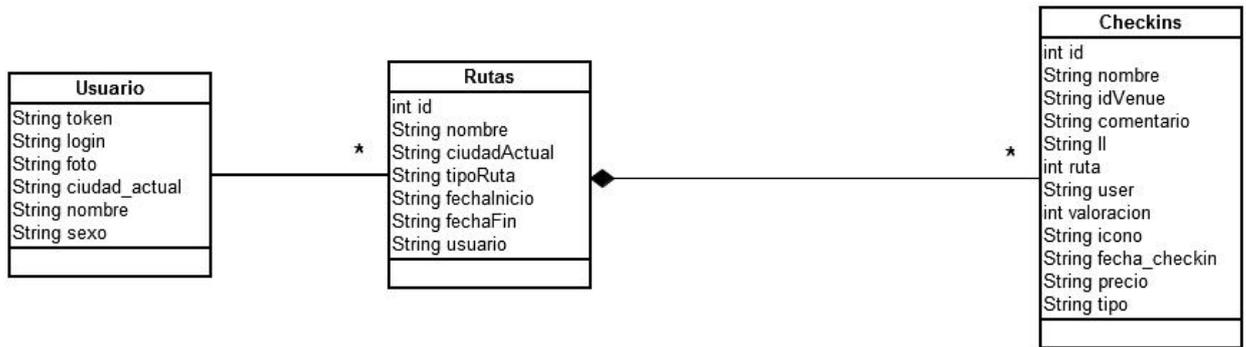


Figura 35. UML de la aplicación

El diagrama está compuesto por tres clases: La clase **usuario**, que gestiona todos los datos del usuario, (1) la clase **ruta**, que gestiona los datos de la ruta turística. Un usuario puede tener  $n$  rutas, pero una ruta solo puede pertenecer a un usuario, (3) la clase **checkins**, que gestiona todos los datos de los puntos añadidos a la ruta turística. Una ruta puede tener  $n$  checkins, pero un checkin solo puede pertenecer a una ruta. En las tablas 2, 3 y 4, se realiza una descripción de aquellos parámetros que pertenecen a la clase **usuario**, **ruta**, y **checkins** respectivamente.

Usuario		
Atributo	Tipo de parámetro	Descripción
Token	String	AccessToken obtenido de Foursquare para poder interactuar con sus datos.
Login	String	Login utilizado por Foursquare. Suele ser una dirección de correo electrónico.
Foto	String	Dirección en la que Foursquare almacena la foto del usuario. Se accede a ella a través de una llamada Http.
Ciudad_Actual	String	Ciudad a la que pertenece el usuario y que ha introducido en Foursquare como tal.
Nombre	String	Nombre del usuario y sus apellidos.
Sexo	String	Genero del usuario.

Tabla 2. Clase usuario

<b>Rutas</b>		
Atributo	Tipo de parámetro	Descripción
Id	Int	Campo identificador de la ruta.
Nombre	String	Nombre que utiliza el usuario para almacenar la ruta.
CiudadActual	String	Ciudad introducida por el usuario al crear la ruta.
TipoRuta	String	Tipo de ruta elegido entre los dos existentes, ocio nocturno y cultural.
FechaInicio	String	Fecha de inicio de la ruta, almacenado en String ya que Sqlite no permite Date y de este modo es más fácil la interacción.
FechaFin	String	Fecha en la cual se finaliza la ruta
Usuario	String	Usuario que crea la ruta.
Seleccionada	Int	Este campo solo esta a nivel local y sirve para indicar si se ha seleccionado una ruta previamente. Es útil para cuando se cierra la aplicación y al volverla a iniciar, se puede continuar con la ruta seleccionada.

**Tabla 3. Clase rutas**

<b>Checkins</b>		
Atributo	Tipo de parámetro	Descripción
Id	Int	Campo identificador de la ruta.
Nombre	String	Nombre del punto proporcionado por Foursquare.
idVenue	String	Identificador del punto según Foursquare.
Comentario	String	Comentario introducido por el usuario al realizar el check-in del punto.
Ll	String	Longitud y latitud del punto.
Ruta	Int	Id de la ruta a la que pertenece.
User	String	Login del usuario y a su vez, identificador de este.
Valoración	Int	Valoración del sitio introducida por el usuario al realizar el check-in. Valores ente 0 y 10.
Icono	String	Ruta de la imagen utilizada por Foursquare para mostrar el punto.
Fecha_checkin	String	Fecha en la cual el usuario ha realizado el check-in en este punto. Se almacena en un String ya que Sqlite no contempla el campo Datetime y así se simplifica la interacción.
Precio	String	Precio por persona introducido por el usuario que ha realizado el check-in. Esta en formato String para poder almacenar el símbolo del € o cualquier otro símbolo correspondiente a una moneda.
Tipo	String	Tipo de punto obtenido de la red Foursquare.

**Tabla 4. Clase checkins**

### 5.1.3. Capa de persistencia

La capa de persistencia permite al sistema almacenar los datos de forma permanente. Esta capa gestiona no solo las bases de datos del sistema, sino también sus

accesos. En este trabajo, la capa de persistencia contempla tres fuentes distintas: La base de datos del dispositivo móvil, Foursquare y el Servicio Web.

La base de datos del dispositivo móvil permite almacenar las rutas turísticas creadas por un usuario junto a todos los puntos que la forman y los datos personales proporcionados por Foursquare del usuario. Además, también permite almacenar aquellas rutas turísticas que son descargadas del servicio Web para su seguimiento.

Foursquare proporciona todos los puntos necesarios para crear una ruta turística. Mediante las coordenadas de longitud y latitud de la localización del usuario, Foursquare puede filtrar estos puntos y devolverlos al usuario para poder incluirlos en su ruta turística.

El Servicio Web proporciona una capa de persistencia a nivel global para toda la aplicación. Aquí es donde son almacenadas todas las rutas turísticas creadas por todos los usuarios de la aplicación, para su futura descarga. Además, junto a estas rutas, el servicio Web almacena los datos del usuario como información adicional.

En esta sección se muestra el diagrama entidad-relación que compone la base de datos del dispositivo móvil y el servicio Web, así como los accesos que permiten acceder a los datos de los tres elementos que componen esta capa de persistencia.

### Diagrama Entidad-Relación

Un diagrama entidad-relación permite mostrar el conjunto de tablas que componen una base de datos así como la relación que existe entre ellas. En la figura 36 se puede observar el diagrama entidad-relación desarrollado en este trabajo. Este diagrama contempla tanto la base de datos del dispositivo móvil como la del Servicio Web.

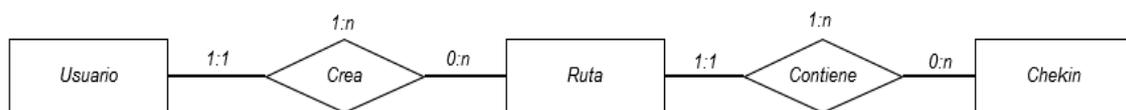


Figura 36. Diagrama de la base de datos

Como se puede observar, el sistema consta de tres tablas: **Usuario**, que se encarga de almacenar los datos del usuario, **Ruta**, que almacena las rutas turísticas y **Checkin**, que almacena todos los puntos añadidos a una ruta. Un usuario puede crear de 1 a  $n$  rutas. Una ruta puede tener de 1 a  $n$  puntos. Las tablas 5, 6 y 7 muestran todos los datos que componen las tablas de la base de datos.

<b>Usuario</b>		
Atributo	Tipo de parámetro	Descripción
Token	String	Accesstoken obtenido de Foursquare para poder interactuar con sus datos. Este token no se almacena en el servicio Web.
Login	String	Login utilizado por Foursquare. Suele ser una dirección de correo electrónico.
Foto	String	Dirección en la que Foursquare almacena la foto del usuario. Se accede a ella a través de una llamada Http.
Ciudad_Actual	String	Ciudad a la que pertenece el usuario y que ha introducido en Foursquare como tal.
Nombre	String	Nombre del usuario y sus apellidos.
Sexo	String	Genero del usuario.

**Tabla 5. Tabla usuario**

<b>Rutas</b>		
Atributo	Tipo de parámetro	Descripción
Id	Int	Campo identificador de la ruta. Se autoincrementa en la base de datos a medida que vamos almacenando nuevas rutas.
Nombre	String	Nombre que utiliza el usuario para almacenar la ruta. Se crea al crear la ruta.
CiudadActual	String	Ciudad introducida por el usuario al crear la ruta.
TipoRuta	String	Tipo de ruta elegido entre los dos existentes, ocio nocturno y cultural.
FechaInicio	String	Fecha de inicio de la ruta, almacenado en String ya que sqlite no permite Date.
FechaFin	String	Fecha en la cual se finaliza la ruta.
Usuario	String	Usuario que crea la ruta.
Seleccionada	Int	Este campo solo esta a nivel local y sirve para indicar si se ha seleccionado una ruta previamente. Es útil para cuando se cierra la aplicación y al volverla a iniciar, se puede continuar con la ruta seleccionada.

**Tabla 6. Tabla rutas**

<b>Checkins</b>		
Atributo	Tipo de parámetro	Descripción
Id	Int	Campo identificador del “check-in”. Se autoincrementa en la base de datos a medida que vamos almacenando nuevos “check-ins”.
Nombre	String	Nombre del punto proporcionado por Foursquare. Es interesante almacenarlo para no tener que realizar peticiones futuras para obtenerlo.
idVenue	String	Identificador del punto según Foursquare. Útil para buscar más información sobre este en la red social.
Comentario	String	Comentario introducido por el usuario al realizar el check-in del punto.
Li	String	Longitud y latitud del punto. Muy útil para posicionarlo en el mapa sin tener que realizar una nueva llamada a Foursquare para buscarlo.
Ruta	Int	Id de la ruta a la que pertenece.
User	String	Login del usuario y a su vez, identificador de este.
Valoración	Int	Valoración del sitio introducida por el usuario al realizar el check-in. Valores ente 0 y 10.
Icono	String	Ruta de la imagen utilizada por Foursquare para mostrar el punto. Es útil almacenarla para no tener que obtenerla realizando peticiones a Foursquare.
Fecha_checkin	String	Fecha en la cual el usuario ha realizado el check-in en este punto. Se almacena en un String ya que sqlite no contempla el campo datetime.
Precio	String	Precio por persona introducido por el usuario que ha realizado el check-in. Esta en formato String para poder almacenar el símbolo del € o cualquier otro símbolo correspondiente a una moneda.
Tipo	String	Tipo de punto obtenido de la red Foursquare. Es útil almacenarlo para no tener que realizar peticiones futuras a Foursquare para obtenerlo.

**Tabla 7. Tabla checkins**

### **Acceso a la base de datos local: Content Provider**

Un Content Provider es un componente de un sistema Android que permite gestionar los accesos a una base de datos. La ventaja que proporciona un Content Provider es que más de una aplicación puede acceder a él, permitiendo que varias aplicaciones puedan acceder al mismo conjunto de datos. Por ejemplo, la agenda de contactos de un dispositivo móvil Android está gestionada por un Content Provider. A este Content Provider pueden acceder la aplicación que permite realizar llamadas telefónicas y la que permite enviar mensajes de texto.

Un Content Provider está compuesto de dos elementos, que son a su vez clases java: Un SQLiteOpenHelper, encargado de realizar la conexión con la base de datos y de modificar el esquema de esta y el Content Provider en sí, encargado de realizar todas

las operaciones sobre la base de datos mediante la definición de métodos más accesibles para el desarrollador.

En este apartado se expone cómo se ha construido el Content Provider de esta aplicación describiendo las clases que realizan esta función y su contenido.

## DBSQLiteHelper

Es la clase SQLiteOpenHelper de este proyecto y la encargada de realizar la conexión a la base de datos y, en caso de que no existan, crear las tablas necesarias. Para ello, se definen primero los campos de las tablas:

```
publicstaticfinal String DATABASE_NAME ="tesis.db";
publicstaticfinalintDATABASE_VERSION = 1;
publicstaticfinal String TABLE_USER ="user";
publicstaticfinal String TABLE_RUTA ="ruta";
publicstaticfinal String TABLE_CHEKIN ="chekin";
//Usuario
publicstaticfinal String FIELD_ID ="id" ;
publicstaticfinal String FIELD_NOMBRE ="nombre" ; //tbpararuta y chekin
publicstaticfinal String FIELD_LOGIN ="login" ;
publicstaticfinal String FIELD_FOTO ="foto" ;
publicstaticfinal String FIELD_SEXO ="sexo" ;
publicstaticfinal String FIELD_CIUADADA ="ciudad_actual" ; //tbpararuta
publicstaticfinal String OAUTH_TOKEN ="oauthToken" ;
//ruta
publicstaticfinal String FIELD_USER ="user";
publicstaticfinal String FIELD_TIPO ="tipo_ruta" ;
publicstaticfinal String FIELD_FINICIO ="fecha_inicio" ;
publicstaticfinal String FIELD_FFIN ="fecha_fin" ;
publicstaticfinal String FIELD_SELECCIONADA = "seleccionada";
//Chek-in
publicstaticfinal String FIELD_VENUE ="idVenue" ;
//public static final String FIELD_CATEGORIA ="categoria" ;
publicstaticfinal String FIELD_COMENTARIO ="comentario" ;
publicstaticfinal String FIELD_LL ="ll" ;
publicstaticfinal String FIELD_RUTA ="ruta";
publicstaticfinal String FIELD_VALORACION ="valoracion";
publicstaticfinal String FIELD_PRECIO ="precio";
publicstaticfinal String FIELD_TIPO_PUNTO = "tipo";
publicstaticfinal String FIELD_ICONO ="icono";
publicstaticfinal String FIELD_FECHA_CHEKIN ="fecha_chekin";
publicstaticfinal String PROJECTION_ALL_FIELDS_USER[] = newString []{
    FIELD_ID, OAUTH_TOKEN, FIELD_NOMBRE, FIELD_LOGIN, FIELD_FOTO, FIELD_SEXO, FIEL
D_CIUADADA
};
publicstaticfinal String PROJECTION_ALL_FIELDS_RUTA[] = new String []{
    FIELD_ID, FIELD_NOMBRE, FIELD_USER, FIELD_CIUADADA, FIELD_TIPO,
FIELD_FINICIO, FIELD_FFIN, FIELD_SELECCIONADA
};
publicstaticfinal String PROJECTION_ALL_FIELDS_CHEKIN[] = new String []{
    FIELD_ID, FIELD_NOMBRE, FIELD_VENUE, FIELD_COMENTARIO, FIELD_LL,
FIELD_RUTA, FIELD_VALORACION, FIELD_PRECIO, FIELD_TIPO_PUNTO,
FIELD_USER, FIELD_ICONO, FIELD_FECHA_CHEKIN
```

Como se puede observar, los campos están definidos en variables estáticas, ya que sus valores no van a cambiar en toda la aplicación y se van a usar de manera continua en ella.

Posteriormente se define el constructor, el cual realiza una llamada al constructor de la clase padre para indicarle la versión de la base de datos y el archivo que se creara:

```
public DBSQLiteHelper (Context context)
{
    //Constructor que hay que llamar obligatoriamente
    super (context, DATABASE_NAME, null, DATABASE_VERSION);
}
```

Por último, se realiza una llamada al método onCreate() para crear una instancia de esta clase y crear el esquema de la base de datos.

```
@Override
public void onCreate (SQLiteDatabase db)
{
    /* Esto se ejecutara solo si se va a crear la base de datos. Si solo se va a acceder a ella no se ejecutara */
    db.execSQL("CREATE TABLE IF NOT EXISTS " + TABLE_USER +
        "(" + FIELD_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " + OAUTH_TOKEN
        + " text, " + FIELD_NOMBRE + " text, " + FIELD_LOGIN + " text, " + FIELD_FOTO +
        text, " + FIELD_SEXO + " text, " + FIELD_CIUADADA + " text)");

    String ruta = "CREATE TABLE IF NOT EXISTS " + TABLE_RUTA +
        "(" + FIELD_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        FIELD_NOMBRE + " text, " + FIELD_USER + " text, " + FIELD_CIUADADA + " text, " +
        FIELD_TIPO +
        " text, " + FIELD_FINICIO + " text, " + FIELD_FFIN +
        text, " + FIELD_SELECCIONADA + " INTEGER)";

    db.execSQL(ruta);

    db.execSQL("CREATE TABLE IF NOT EXISTS " + TABLE_CHEKIN +
        "(" + FIELD_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " + FIELD_NOMBRE +
        text, " + FIELD_VENUE + " text, " + FIELD_COMENTARIO + " text, " + FIELD_LL +
        text, " + FIELD_RUTA + " INTEGER, " + FIELD_USER + " text, " + FIELD_VALORACION +
        INTEGER, " + FIELD_PRECIO + " text, " + FIELD_TIPO_PUNTO +
        text, " + FIELD_ICONO + " text, " + FIELD_FECHA_CHEKIN + " text)");
}
```

## DBProvider

La clase DBProvider es la que realizara la función en el proyecto de **Content Provider**. Antes de pasar a explicar la funcionalidad de esta clase, se exponen un par de conceptos:

- Todos los Content Provider se localizan en base a URIs.
- Aunque los Content Provider pueden ser utilizados para acceder a cualquier tipo de datos su diseño parece haber sido pensado para acceder a bases de datos relacionales.

Teniendo como referencia estas indicaciones, en primer lugar se define la URI del Content Provider. La URI sirve para identificarlo el Content Provider. Como la aplicación cuenta con tres tablas, se ha optado por incluir una URI para cada una de estas tablas:

```
publicstaticfinal Uri CONTENT_URI_USER = Uri.parse(CONTENT_PREFIX
+AUTHORITY_PART +"/Usuario");
publicstaticfinal Uri CONTENT_URI_RUTA = Uri.parse(CONTENT_PREFIX
+AUTHORITY_PART +"/Ruta");
publicstaticfinal Uri CONTENT_URI_CHEKIN =
Uri.parse(CONTENT_PREFIX +AUTHORITY_PART +"/Chekin");
```

Como se puede observar, el URI está formado por:

- El prefijo del protocolo **content://**.
- La autoridad **dsic.upv.es**: Este fragmento deberá utilizarse en la declaración del archivo *AndroidManifest.xml*.
- El recurso **/Usuario**, **/Ruta** o **/Chekin**, según la tabla.

Posteriormente se debe crear un UriMatcher, un objeto que analizara los URIS de los recursos que se piden y actuar en consecuencia:

```
static {
    URI_MATCHER=newUriMatcher(UriMatcher.NO_MATCH);
    URI_MATCHER.addURI(AUTHORITY_PART, "Usuario", CODE_ALL_ITEMS_USER);
    URI_MATCHER.addURI(AUTHORITY_PART, "Usuario/#", CODE_SINGLE_ITEM_USER);
    URI_MATCHER.addURI(AUTHORITY_PART, "Ruta", CODE_ALL_ITEMS_RUTA);
    URI_MATCHER.addURI(AUTHORITY_PART, "Ruta/#", CODE_SINGLE_ITEM_RUTA);
    URI_MATCHER.addURI(AUTHORITY_PART, "Chekin", CODE_ALL_ITEMS_CHEKIN);
    URI_MATCHER.addURI(AUTHORITY_PART, "Chekin/#", CODE_SINGLE_ITEM_CHEKIN);
}
```

Como se puede observar, la primera instancia del URIMatcher es el URI NO\_MATCH, que proporciona una respuesta en caso de que no se encuentre el URI buscado. Posteriormente, se añaden las URIs para seleccionar un elemento en concreto de las tablas, o todos los de una tabla. Por ejemplo, el urimatcher para seleccionar todos los elementos de una ruta:

```
URI_MATCHER.addURI(AUTHORITY_PART,"Ruta",CODE_ALL_ITEMS_RUTA);
```

El siguiente paso es definir el método getType() que establece los MIME que se usaran en la aplicación. En general existe una convención sobre cómo establecer los mime types distinguiendo entre un recurso único y una colección de recursos:

- **Colecciones:** vnd.parteautoridad.dir/vnd.parteautoridad
- **Recursos:** vnd.parteautoridad.item/vnd.parteautoridad

```
public String getType(Uri uri) {
    switch (URI_MATCHER.match(uri)) {
        case CODE_ALL_ITEMS_USER:
            return MIME_TYPE_ALL_ITEMS;
        case CODE_ALL_ITEMS_RUTA:
            return MIME_TYPE_ALL_ITEMS;
        case CODE_ALL_ITEMS_CHEKIN:
            return MIME_TYPE_ALL_ITEMS;

        case CODE_SINGLE_ITEM_USER:
            return MIME_TYPE_SINGLE_ITEM;
        case CODE_SINGLE_ITEM_RUTA:
            return MIME_TYPE_SINGLE_ITEM;
        case CODE_SINGLE_ITEM_CHEKIN:
            return MIME_TYPE_SINGLE_ITEM;

        default:
            throw new IllegalArgumentException(INVALID_URI_MESSAGE +
uri);
    }
}
```

Por último, se debe crear el método onCreate() el cual define la especialización de la clase SQLHelper y proporciona permisos de lectura y escritura:

```
public boolean onCreate() {
    dbHelper = new DBSQLiteHelper(this.context);
    database = dbHelper.getWritableDatabase();
    return database != null && database.isOpen();
}
```

Una vez definidos todos estos elementos, se deben crear los métodos de acceso para realizar las operaciones pertinentes: Insert, Delete, Update y Query.

### **Acceso al Servicio Web del sistema**

El servicio Web del sistema permite almacenar las rutas turísticas creadas por los usuarios. Esto proporciona dos ventajas: (1) permite que cualquier usuario descargue una ruta de viaje, (2) proporciona al usuario la posibilidad de almacenar sus rutas de viaje por si son borradas del dispositivo móvil.

El acceso a estos datos viene explicado en el capítulo que comenta la integración de las distintas tecnologías. En él se muestran como mediante tres peticiones HTTP, el usuario puede acceder a todos los datos almacenados en el servicio Web. Para obtener más información de esto, consultar el capítulo 4.

### **Foursquare**

Foursquare proporciona los puntos que pueden ser añadidos a una ruta turística para crearla. Para poder acceder a estos puntos, antes el sistema debe calcular la localización en la que se encuentra un usuario. Una vez calculada esta localización, mediante peticiones HTTP, se puede acceder a aquellos datos que son necesarios en la creación de rutas turísticas. En esta sección, se muestran las peticiones a Foursquare que permiten obtener los siguientes elementos: Los datos personales de un usuario, los datos correspondientes a un punto cercano al usuario y las categorías a las que pertenecen estos puntos.

- **users/self?oauth\_token=""**: Permite obtener los datos de un usuario en función del AccessToken de ese usuario. Devuelve los siguientes datos en formato JSON:
  - **Id.**
  - **Nombre y apellidos.**
  - **Foto** subida por el usuario la cual se usa como avatar. Si no tiene muestra una foto por defecto.
  - **Genero del usuario** (hombre o mujer).
  - **Ciudad de origen.**
  - **Tipo de usuario.**
  - **Contacto** compuesto por teléfono y correo electrónico.

- Listado de **medallas** obtenidas en Foursquare
- Listado de **Check-ins** realizados en Foursquare
- El usuario es **alcalde** de un lugar. El alcalde es aquella persona que ha realizado más Check-ins en ese lugar.
- Listado de **amigos** que tiene el usuario en Foursquare.
- **Otros usuarios** que siguen a este usuario.
- **Peticiones** realizadas por este usuario.

Como se puede observar, Foursquare proporciona muy pocos datos personales del usuario, mostrando más información de los datos de su red que va creando este a medida que va interactuando con ella. De todos estos datos, los que se utilizan en la aplicación son:

- **Nombre y apellidos**, utilizado para mostrar una mayor información del usuario.
  - **Foto del usuario**, permitiendo mostrarla a modo de avatar para identificar más fácilmente a un usuario.
  - **Ciudad de origen**, como información adicional.
  - **Genero del usuario**, como información adicional.
  - **Correo electrónico**, a modo de identificador de usuario a la hora de buscar información.
- **venues/search**: Devuelve un conjunto de puntos en función de la ubicación actual del dispositivo. Devuelve los siguientes datos:
    - **Id**
    - **Nombre**
    - **Verificado**. Es un booleano que muestra si la información de este punto es correcta.
    - **Contacto**, ya sea por teléfono o por Twitter.
    - **Localización**, que es un objeto JSON que contiene los siguientes campos:
      - ✓ **Dirección**

- ✓ **Calles** con las que cruza
- ✓ **Ciudad**
- ✓ **Estado o provincia**
- ✓ **Código postal**
- ✓ **País**
- ✓ **Latitud**
- ✓ **Longitud**
- ✓ **Distancia** del punto a nuestra localización
- **Categoría**, que es un objeto compuesto por:
  - ✓ **Id**
  - ✓ **Nombre** de la categoría
  - ✓ **Nombre** de la categoría en plural
  - ✓ **Icono**
  - ✓ **Categoría padre**
- **Datos especiales** que pueden tener algunos lugares
- **Datos adicionales** sobre la gente que se encuentra actualmente en el lugar, por ejemplo, el número de personas que actualmente se encuentran en un bar con la aplicación de Foursquare abierta.

Como se puede observar en los datos devueltos, Foursquare presta una mayor atención a las localizaciones que componen su red que a sus usuarios en cuanto a cantidad y calidad de información. De estos datos, los utilizados han sido:

- **Id** de la localización para identificarla en la base de datos local.
- **Nombre** de la localización a modo de información.
- **Ciudad**.
- **Latitud** y **longitud** del punto, para poder representarlo en el mapa.
- **Nombre de la categoría**, para poder filtrar los puntos.
- **Icono**, para dibujarlo en el mapa.

Esta información también es devuelta por el sistema de Foursquare cuando se crea un nuevo punto, devolviendo aquellos datos que han sido rellenados por el usuario.

- **venues/categories:** Devuelve todas las categorías registradas en la base de datos de Foursquare, agrupadas por tipos. Los datos devueltos son:
  - **Id**
  - **Nombre de la categoría**
  - **Nombre de la categoría** en plural
  - **Icono**
  - **Categorías hijas**, compuesto por un conjunto de categorías.

## 5.2. Conclusiones

En este capítulo se ha mostrado la estructura y funcionalidad del sistema desarrollado en este trabajo. La estructura se basa en una arquitectura de tres capas, compuesta por: La capa de **presentación**, la capa de **lógica** y la capa de **persistencia**.

La capa de **presentación** permite interactuar a los usuarios con el sistema. Viene dada por unas interfaces gráficas definidas en el dispositivo móvil. Para definir estas interfaces, se han creado distintos archivos en formato *xml* que posteriormente han sido incluidos en las actividades del sistema.

La capa de **lógica** define toda la funcionalidad del sistema. Esta funcionalidad está compuesta por tres procesos: **Creación de rutas turísticas**, **finalización y publicación de las rutas en un Servicio Web** y **descarga y seguimiento de las rutas turísticas**.

La capa de **persistencia** proporciona todos los datos del sistema, así como los accesos que permiten acceder a ellos. Los datos están distribuidos en tres bases de datos distintas: (1) La **base de datos del dispositivo móvil**, que almacena las rutas de viaje creadas por un usuario y aquellas que descarga del servicio Web, (2) **Foursquare**, que proporciona todos los puntos cercanos a la localización de un usuario, (3) el **servicio Web**, que permite almacenar las rutas turísticas para su futura descarga.

Esta arquitectura ha supuesto la mayor parte del trabajo realizado en el desarrollo de esta aplicación, destacando como mayor complejidad la comprensión del funcionamiento del Content Provider que permite el acceso a los datos del dispositivo móvil.



## Capítulo 6

# Manual de Usuario

---

La aplicación desarrollada en este trabajo proporciona a los usuarios una aplicación que permite crear rutas turísticas. En particular, los tres procesos o funcionalidades principales de la aplicación son: (1) la creación de rutas turísticas, (2), su finalización y publicación en un servidor Web, y (3) la descarga y el seguimiento punto a punto de rutas de este servidor Web, las cuales han sido realizadas por otros usuarios.

El proceso encargado de la **creación de rutas turísticas** proporciona al usuario funcionalidades y operaciones que se relacionan con la creación de una ruta nueva, y su composición a través de la inclusión de los puntos, sitios o lugares que la componen. Con esta finalidad, la aplicación, en primer lugar debe calcular las coordenadas físicas del usuario, es decir, la longitud y latitud de la localización del usuario. Una vez calculadas, éstas coordenadas son enviadas a Foursquare para que se calculen y busquen aquellos lugares y establecimientos cercanos. Posteriormente, cuando Foursquare ha calculado estos lugares, éstos son enviados a la aplicación móvil, que los procesa y muestra al usuario. Finalmente, el usuario selecciona aquellos puntos a añadir a su ruta turística incluyendo como información su valoración personal o sus comentarios.

Tras la creación de una ruta turística, el siguiente paso u operación a realizar consiste en confirmar la ruta como finalizada, **operación que a través de un servicio Web envía la ruta a un servidor Web en el que se almacenan las rutas de los usuarios**. Con este fin, cuando el usuario indica la ruta como finalizada, el sistema obtiene la fecha actual y la incluye como información en la ruta a publicar. Una vez incluida esta fecha, se genera una cadena en formato JSON, la cual contiene todos los datos referentes a la ruta (ej. puntos que se incluyen, valoración y comentario del usuario, etc.). El servicio Web recibe estos datos, los procesa y los almacena en una base de datos dentro de un servidor Web.

La publicación de las rutas en un servidor Web da lugar a que otros usuarios puedan descargar, visualizar y recorrer punto a punto una ruta ya realizada. **Esta es la tercera funcionalidad principal de la aplicación**. Para cubrir dicha funcionalidad, la aplicación proporciona al usuario un formulario en el que se solicita información que se utiliza para buscar y filtrar rutas turísticas en el servidor, información como puede ser el nombre de una localización. Una vez rellenado, esta información es enviada al servicio Web, que procesa estos datos, filtra las rutas y las devuelve para que el usuario pueda visualizarlas.

Para describir y detallar estas tres funcionalidades principales, en este capítulo se muestra un manual de usuario que indica la interacción del usuario con la aplicación.

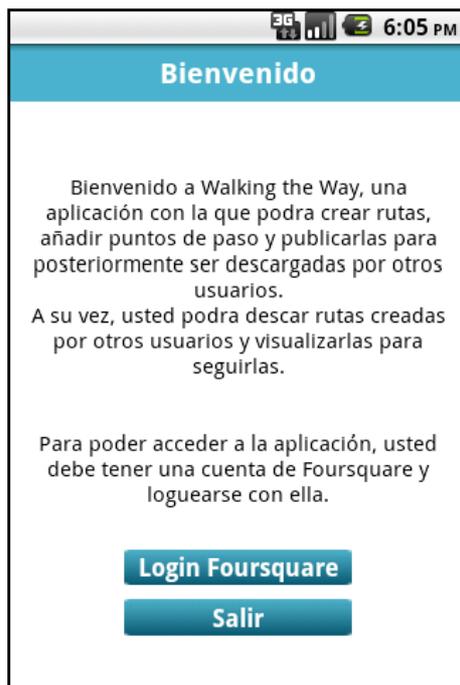
### 6.1.1. Creación de Rutas Turísticas

Este proceso permite crear rutas turísticas añadiendo lugares y establecimientos visitados por el usuario. Primero el usuario debe crear una nueva ruta turística. Una vez creada, se calcula la localización actual en la que se encuentra y envía esta localización a Foursquare. Cuando Foursquare recibe estos datos, calcula todos los lugares y establecimientos cercanos. Estos lugares y establecimientos son enviados al dispositivo móvil para que el usuario pueda añadirlos a su ruta turística.

Esta subsección muestra los pasos necesarios para crear una ruta turística. En primer lugar el usuario debe **acceder a la aplicación**, mediante sus credenciales de Foursquare. Posteriormente, debe rellenar un formulario con los datos que compondrán la **nueva ruta turística**. Seleccionada esta ruta turística, el usuario podrá **calcular su localización y añadir puntos** a la ruta. Si al añadir un punto a la ruta, el usuario se ha equivocado, puede **editar** este punto o **eliminarlo**. El usuario también cuenta con la posibilidad de **eliminar** esta ruta o cualquier otra que tenga almacenada en el dispositivo móvil.

#### Acceso a la aplicación

Previamente a acceder a la aplicación, el usuario debe identificarse con su cuenta de Foursquare. Para realizar esto, debe pulsar el botón *Login Foursquare* de la pantalla de bienvenida. Esta pantalla puede observarse en la figura 37. Si el usuario pulsa el botón *salir*, se cerrará la aplicación.



**Figura 37. Pantalla de bienvenida**

Una vez pulsado el botón *Login Foursquare*, la aplicación abre una ventana de navegador que accede a la página Web de Foursquare, concretamente al formulario que permite identificarse a sus usuarios. Esta ventana del navegador puede observarse en la figura 38. En ella se puede observar como Foursquare permite crear una nueva cuenta de usuario si no se dispone de una.

**Figura 38. Pantalla de acceso a Foursquare**

Cuando se ha realizado la autenticación en Foursquare, el sistema almacena los datos de la sesión del usuario. Esto evita que se deban volver a introducir las credenciales, recuperándolas e introduciéndolas el mismo.

### Creación de Nueva Ruta Turística

Cuando el usuario se ha identificado con sus credenciales de Foursquare en la aplicación, el sistema muestra un listado de las rutas turísticas creadas por el usuario. Si la aplicación ha sido ejecutada por primera vez, este listado no contendrá ninguna ruta.

Para crear una nueva ruta, el usuario debe desplegar el menú de la aplicación, y pulsar la opción *Nueva Ruta*, tal y como se muestra en la figura 39. La figura 40 muestra el formulario a rellenar por el usuario para la creación de la ruta turística. Los campos **Nombre de la ruta** y **Ciudad de origen** deben ser rellenados obligatoriamente.



Figura 39. Nueva ruta

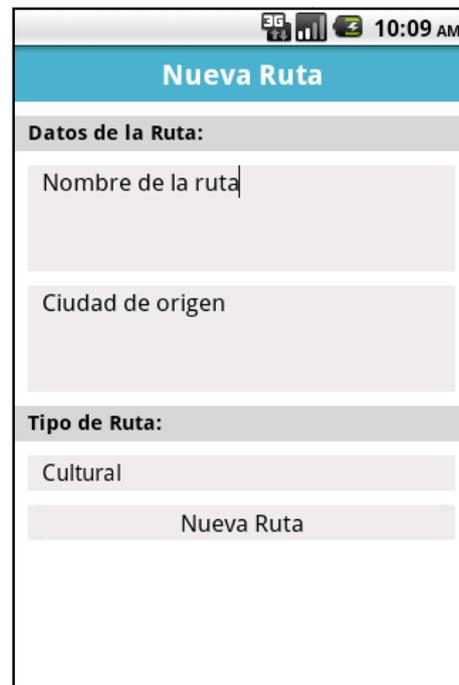


Figura 40. Formulario nueva ruta

### Añadir Puntos a la Ruta

La operación que permite añadir puntos a una ruta turística se inicia en el listado de rutas. En este listado, el usuario debe seleccionar la ruta que desea continuar. En la parte inferior de este listado se muestra la ruta actualmente seleccionada. Si la ruta que esta seleccionada ya ha sido finalizada, no se podrá añadir un nuevo punto.

Una vez seleccionada la ruta, el usuario debe acceder a la pestaña *Mapa*. Al acceder a ella, el sistema calculará su localización y los puntos cercanos al usuario. Una vez calculados estos puntos, si el usuario quiere añadir un punto concreto a su ruta turística,

debe pulsar sobre él. Al pulsar, se le desplegará un menú contextual que podemos ver en la figura 41. En esta figura se puede observar que se permite añadir un nuevo punto, mediante la opción **Check-in**, u obtener información adicional del lugar mediante la opción **Información**. Por otra parte, el título del menú, en la imagen **FreeWok**, indica el nombre del establecimiento que se desea añadir a la ruta turística. Una vez pulsada la opción **Check-in**, el sistema muestra al usuario un formulario para que añada información adicional al punto. Este formulario se puede observar en la figura 42.

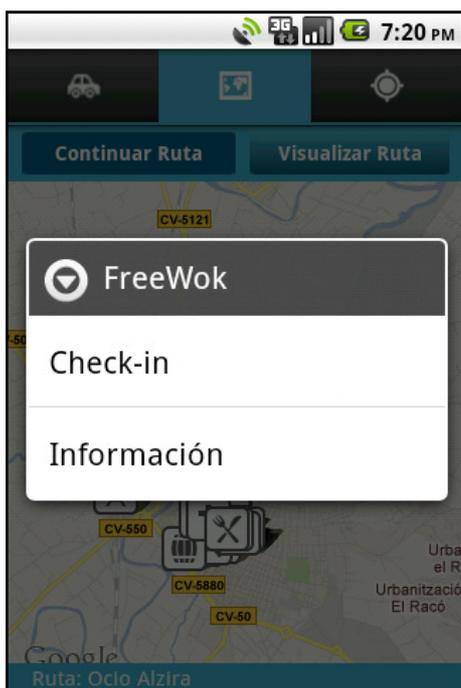


Figura 41. Menú contextual de un punto

A screenshot of a mobile application interface showing a form titled 'Realizar Check-in'. The form is divided into several sections. The first section is 'Datos del Lugar:', which contains two text input fields: 'FreeWok' and 'Asian Restaurant'. The second section is 'Valoración del Lugar:', which contains two input fields: 'Precio por persona:' with a value of '0€' and 'Valoración:' with a value of '0'. Below these is a large text area for 'Comentario'. At the bottom of the form is a 'Check-in' button.

Figura 42. Formulario añadir punto

En algunas ocasiones, puede ocurrir que el usuario desee añadir un punto que no se encuentra registrado en la red social Foursquare. En estos casos, para realizar esta operación, el usuario debe acceder al menú de la pestaña *Mapa* y pulsar la opción de **añadir localización** tal como podemos ver en la figura 43. Esto mostrara un formulario al usuario, el cual podemos ver en la figura 44, que permite introducir los datos de la nueva localización.



Figura 43. Añadir localización

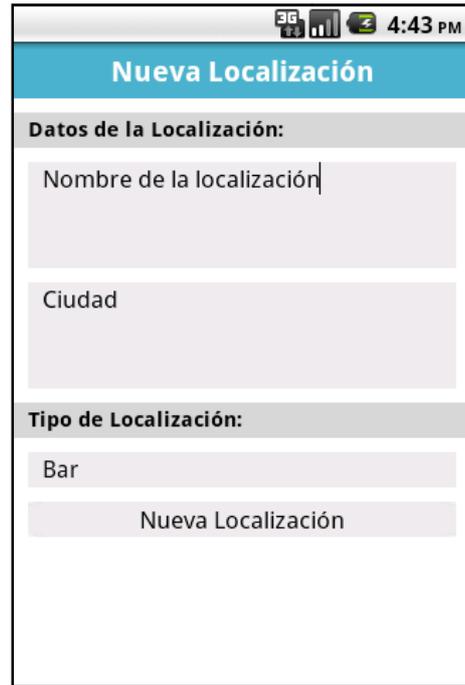


Figura 44. Formulario nueva localización

Otra opción que permite el sistema es filtrar los puntos cercanos. Para ello el usuario debe acceder al menú de la pestaña *Mapa* y pulsar la opción **filtrar puntos**, como se puede observar en la figura 45. Una vez pulsada esta opción, el sistema mostrará el menú contextual expuesto en la figura 46 que ofrece al usuario la posibilidad de filtrar los puntos mostrados en base a su categoría o tipología.



Figura 45. Filtrar Puntos

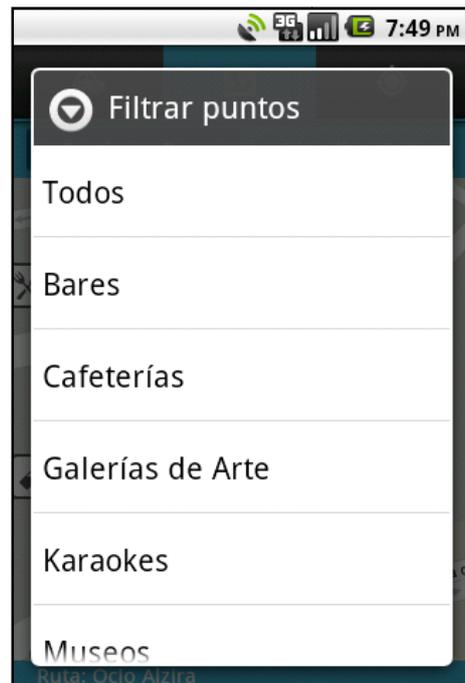


Figura 46. Menú contextual filtrar puntos

Asimismo, si el usuario cambia de posición, la aplicación permite recalculare la localización y en consecuencia los puntos cercanos. Para ello, en el menú de la pestaña *Mapa*, debe pulsar la opción **puntos cercanos** mostrada en la figura 47. Una vez pulsada esta opción, el sistema calculará todos los puntos cercanos y los mostrará en un mapa como se muestra en la figura 48.



Figura 47. Puntos cercanos



Figura 48. Puntos obtenidos

### Editar y eliminar un punto

Cabe la posibilidad que el usuario añada un punto que posteriormente no considere que deba ser incluido en la ruta turística o que al introducir los datos referentes a este punto se haya equivocado. En estos dos casos, el usuario puede **eliminar** un punto o **editarlo**.

Para eliminar un punto, el usuario debe acceder a la pestaña *Puntos*. En ella, mediante el menú, podrá acceder al listado de los puntos a eliminar, tal y como se muestra en la figura 49. Para eliminar un punto se debe seleccionar en el listado de la figura 50 y pulsar el botón **eliminar**. Se puede eliminar más de un punto seleccionado y se pueden seleccionar todos los puntos mediante el botón **seleccionar todo**. Si una ruta está finalizada, los puntos no podrán ser eliminados.



Figura 49. Eliminar Puntos



Figura 50. Listado eliminar puntos

Para editar un punto, el usuario debe pulsar sobre un punto del listado de puntos que componen la ruta. Al pulsar en el, el sistema muestra toda la información del punto. En esta pantalla, si se muestra el menú, el usuario podrá acceder al formulario que permite editar el punto, mostrado en la figura 52, o bien, eliminarlo, como se muestra en la figura 51.



Figura 51. Editar punto



Figura 52. Formulario editar punto

## Eliminar rutas

Puede darse el caso de que el usuario desee borrar rutas que ya no utilice o que hayan sido descartadas por cualquier motivo. Para realizar esto, el usuario debe acceder a la pestaña *Rutas*. Una vez se encuentre en esta pestaña, al acceder al menú, debe seleccionar la opción **eliminar rutas**, mostrada en la figura 53. Esto muestra al usuario un listado de elementos seleccionable, expuesto en la figura 54, que permite seleccionar aquellas rutas que desea eliminar y borrarlas mediante el botón **eliminar**. Se puede seleccionar más de una ruta y se puede seleccionar todas las rutas mediante el botón **seleccionar todo**.



Figura 53. Eliminar rutas

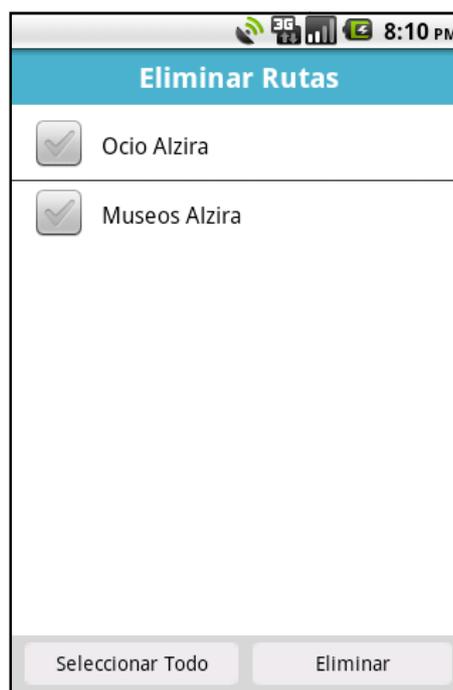


Figura 54. Listado eliminar rutas

### 6.1.2. Finalizar ruta

Este proceso permite finalizar una ruta y ser publicada en un servidor Web. Para ello, el usuario debe indicar que quiere finalizar la ruta. Una vez finalizada, la ruta no podrá ser editada.

Si un usuario desea finalizar una ruta, debe acceder a la pestaña de *Mapa*. Una vez se encuentre en esta pestaña y el sistema haya obtenido tanto la localización del usuario como los puntos cercanos a él, el usuario puede acceder al menú y pulsar la opción **finalizar ruta**. Previa confirmación, la ruta será finalizada y enviada mediante el servicio Web para que ésta sea almacenada.

### 6.1.3. Descargar rutas

Este proceso permite al usuario acceder a las rutas del servidor web y descargarlas para recorrerlas punto a punto. Para ello, el usuario debe rellenar un formulario con los requisitos de las rutas que quiera buscar.

Esta sección muestra los pasos necesarios para **descargar** la ruta turística y realizar su **seguimiento**.

#### Descargar rutas turísticas

Para descargar las rutas turísticas del servicio Web, el usuario debe acceder a la pestaña *Rutas* y pulsar el botón **otras rutas**. Esto muestra el listado de rutas que tiene almacenadas el dispositivo y que no han sido creadas por el usuario. Al desplegar el menú de este listado, mostrado en la figura 55, el usuario puede acceder a la opción **descargar rutas**. Al acceder a ella, el sistema mostrara un formulario con los campos a rellenar de las prioridades de un usuario a la hora de buscar rutas turísticas. Este formulario puede observarse en la figura 56.



Figura 55. Descargar rutas

The screenshot shows a mobile application interface for searching routes. The title bar at the top says 'Buscar Rutas'. Below the title bar, there are two sections: 'Datos del Usuario:' and 'Datos de la Ruta:'. Under 'Datos del Usuario:', there is a label 'Sexo:' and a dropdown menu with 'Hombre' selected. Under 'Datos de la Ruta:', there is a label 'Creador:' and a dropdown menu with 'Otros Usuarios' selected. Below that, there is a label 'Ciudad:' and a text input field. At the bottom, there is a label 'Tipo de Ruta:' and a dropdown menu with 'Cultural' selected. At the very bottom, there is a button labeled 'Buscar Rutas'.

Figura 56. Formulario descargar rutas

#### Seguimiento de una ruta turística

Cualquier ruta turística puede ser seguida por un usuario. Para realizar este proceso, el usuario debe seleccionar la ruta que desea seguir. Una vez seleccionada, debe ir a la

pestaña *Mapa* y pulsar el botón **visualizar ruta**. Al realizar esto, el usuario es posicionado en la localización del primer punto. Si despliega el menú, podrá acceder a las siguientes opciones: **primer punto**, que lo posiciona en el primer punto de la ruta, **siguiente punto**, que lo desplaza al siguiente punto añadido a la ruta, **último punto**, que lo posiciona en el último punto de la ruta. Este menú puede observarse en la figura 57.

Si el usuario pulsa sobre cualquier punto de esta ruta, puede observar los datos introducidos por la persona que creó la ruta turística, referentes a este punto.



**Figura 57. Seguimiento de una ruta**



## Capítulo 7

# Conclusiones

---

El trabajo realizado en esta tesis presenta una aplicación móvil para la creación de rutas turísticas cuya complejidad reside en su integración con diversas tecnologías.

En primer lugar, la aplicación móvil presentada, la cual se ha desarrollado bajo la plataforma Android, **se integra con una red social basada en geolocalización: Foursquare**. A través de la API proporcionada por esta red social, la aplicación ofrece locales, sitios y lugares turísticos, los cuales pueden ser añadidos a la ruta turística en creación.

En segundo lugar, otro de los puntos fuertes de la aplicación es la **integración de Geolocalización** dentro de la aplicación. La Geolocalización se integra dentro de esta aplicación con el fin de representar y mostrar al usuario aquellos puntos verdaderamente relevantes en la creación de una ruta, es decir, aquellos puntos cercanos a la localización actual del usuario.

En tercer lugar, la tercera pieza que integra esta aplicación es **el almacenamiento de las rutas turísticas creadas en un servidor Web** con el cual, la aplicación se comunica a través de un servicio web REST. Con este almacenamiento, las rutas almacenadas, junto a la información relacionada con los comentarios o valoraciones se hacen públicas y accesibles a otros usuarios.

La realización de esta aplicación demuestra los beneficios de las redes sociales basadas en geolocalización en un ámbito concreto: el de las aplicaciones móviles relacionadas con el turismo. Así, este trabajo provee una pieza que puede formar parte de una plataforma más completa para la construcción, publicación y descarga de rutas turísticas.

En el siguiente punto se describen varios posibles trabajos futuros que podrían completar la aplicación realizada o que podrían componer otras piezas que dieran lugar a la plataforma online que se ha citado para la construcción, publicación y descarga de rutas turísticas.

## 7.1. Trabajo Futuro

El trabajo realizado en esta tesis supone una primera aproximación de aplicación móvil para la creación de rutas turísticas. Esta aplicación puede mejorarse y completarse mediante funcionalidades concretas que se describen a continuación:

- **Integración con la red social Facebook.** Esta integración podría complementar la información almacenada sobre una ruta con determinada información del usuario como puede ser el sexo, la edad o sus aficiones. De esta forma, las búsqueda de rutas no solo podría realizarse en base a una localidad determinada, sino en base al perfil del usuario que realiza la búsqueda. Asimismo, la integración con esta red social permitiría publicar y mostrar a los contactos del usuario una ruta realizada.
- **La creación y composición de una ruta a partir de rutas ya existentes.** Esta funcionalidad permitiría al usuario componer una ruta de una localidad en base a diferentes rutas de dicha localidad. De esta forma, el usuario puede adaptar la ruta a sus gustos.
- **Almacenamiento del recorrido realizado en una ruta mediante seguimiento GPS.** Esta funcionalidad permitiría al usuario saber el lugar concreto en el que se encontraba en un determinado momento, visualizando dicha ruta como un trazado sobre un mapa.
- **La introducción de fotografías y videos en la creación de una ruta.** De esta forma, las rutas pueden enriquecerse a través de información multimedia.



# Referencias

1. **Facebook.** Facebook. [Online] <http://es-es.facebook.com/>.
2. **Gowalla.** Gowalla. [Online] <http://gowalla.com/>.
3. **Foursquare.** Fousquare. [Online] <https://foursquare.com/>.
4. **Brightkite.** Brightkite. [Online] <http://brightkite.com/>.
5. **Loopt.** Loopt. [Online] <https://www.loopt.com/>.
6. **Yelp.** Yelp. [Online] <http://www.yelp.com/>.
7. **Google.** Google Latitude. [Online] <https://www.google.com/latitude/b/0>.
8. **Trip Journal.** Trip Journal. [Online] <http://www.trip-journal.com/>.
9. **Triplt.** Triplt. [Online] <http://www.tripit.com/>.
10. **Minube.** Minube. [Online] <http://www.minube.com/>.
11. **TripAdvisor.** TripAdvisor. [Online] <http://www.tripadvisor.es/>.
12. *Como funciona el sistema GPS, en cinco pasos lógicos.* **Trimble Navigation.** 1996.
13. **Chandra, Dr A. M.** *Higher Surveying.* s.l. : New Age International, 2002.
14. **Google.** Android Developers. [Online] <http://developer.android.com/index.html>.
15. —. LocationManager class. [Online] <http://developer.android.com/reference/android/location/LocationManager.html>.
16. **Yarow, Jay.** Business Insider. [Online] <http://www.businessinsider.com/chart-of-the-day-foursquare-users-2011-3>.
17. **OAuth Community.** OAuth Community. [Online] <http://oauth.net/>.
18. **Foursquare.** My OAuth Consumers. [Online] <https://es.foursquare.com/oauth/>.
19. —. API Foursquare. [Online] <https://developer.foursquare.com/>.
20. **Generalitat Valenciana.** Moskiitt. *Moskiitt.* [Online] <http://www.moskiitt.org/>.
21. **Google.** Google Code. [Online] <http://code.google.com/intl/es-ES/apis/maps/index.html>.
22. **Yang, James.** What is n-Tier Architecture? *developerFusion.* [Online] <http://www.developerfusion.com/article/3058/boosting-your-net-application-performance/2/>.