



**UNIVERSIDAD DE JAÉN**  
*Escuela Politécnica Superior (Jaén)*

## Proyecto Fin de Carrera

# **APLICACIÓN MÓVIL PARA LA RECOMENDACIÓN DE RESTAURANTES EN JAÉN**

**Alumno: Carmen María Morillo Arias**

Tutor: Dr. D. Luis Martínez López  
Dª. Rosa María Rodríguez Domínguez  
Dpto: Informática

**Septiembre, 2013**





Universidad de Jaén  
Escuela Politécnica Superior de Jaén  
Departamento de Informática

Dr. D. Luis Martínez López y D<sup>a</sup> Rosa María Rodríguez Domínguez, tutores del Proyecto Fin de Carrera titulado: APLICACIÓN MÓVIL PARA LA RECOMENDACIÓN DE RESTAURANTES EN JAÉN que presenta D<sup>a</sup> Carmen María Morillo Arias, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre 2013.

El alumno:

D<sup>a</sup> Carmen María Morillo  
Arias

Los tutores:

Dr.D. Luis Martínez  
López

D<sup>a</sup> Rosa María Rodríguez  
Domínguez



# APLICACIÓN MÓVIL PARA LA RECOMENDACIÓN DE RESTAURANTES EN JAÉN

Alumno: Carmen María Morillo Arias

Tutores: Luis Martínez López  
Rosa María Rodríguez Domínguez

Departamento: Informática

Área: Lenguajes y Sistemas Informáticos



Universidad de Jaén  
Escuela Politécnica Superior de Jaén  
Departamento de Informática

## ***Agradecimientos***

Este trabajo no hubiese sido posible de no ser por el apoyo que he recibido de todas aquellas personas que, de una forma u otra, han colaborado para que este proyecto llegase a buen fin.

Por ello, quisiera mostrar mi agradecimiento, en primer lugar, a mi hijo Emil, por ser el motor y el motivo de todo mi esfuerzo, y a mis padres y hermanas, por estar siempre a mi lado.

A mi amigo Antonio Araque, por ser mi mentor.

A mis compañeros, por todos los ratos que hemos pasado juntos, y por su ayuda incondicional.

Y por último, a mis tutores, Luis y Rosa, por ser mi guía a lo largo del desarrollo de este proyecto, por la disponibilidad que me han ofrecido siempre y el buen trato que he recibido de ellos.

Gracias a todos.

## Contenido

---

Contenido .....	v
Índice de ilustraciones .....	vii
Índice de tablas .....	ix
Índice de gráficas.....	x
1 Introducción .....	1
1.1 Contexto y visión general.....	2
1.2 Propósito y objetivos.....	3
1.3 Resultados esperados .....	3
1.4 Investigaciones relacionadas.....	4
1.5 Planificación del proyecto.....	4
2 Sistemas de recomendación .....	9
2.1 Introducción .....	10
2.2 Elementos de un sistema de recomendación y esquema general de funcionamiento ....	11
2.3 Clasificación de los sistemas de recomendación .....	13
2.4 Sistemas de recomendación basados en contexto .....	14
2.5 REJA: un sistema de recomendación de restaurantes en Jaén .....	18
3 Android y Geolocalización .....	21
3.1 Android .....	22
3.2 Geolocalización.....	29
4 Ingeniería del Software .....	31
4.1 Introducción .....	32
4.2 Análisis.....	33
4.3 Diseño .....	44

4.4 Implementación y pruebas .....	71
5 Conclusiones .....	79
Bibliografía .....	83
Anexo I. Manual de instalación .....	87
Anexo II. Guía de usuario .....	93
Anexo III. Importación de la librería Google Maps en un proyecto Android .....	105

## Índice de ilustraciones

---

Ilustración 1. Esquema de división del trabajo.....	5
Ilustración 2. Diagrama de Gantt.....	7
Ilustración 3. Esquema general de funcionamiento de un sistema de recomendación.....	12
Ilustración 4. Matriz multidimensional en modelado contextual .....	16
Ilustración 5. Modelos para la incorporación del contexto en los sistemas de recomendación ..	18
Ilustración 6. Esquema de funcionamiento del sistema de recomendación REJA .....	19
Ilustración 7. Tasa de distribución para las versiones de Android .....	24
Ilustración 8. Arquitectura de Android.....	25
Ilustración 9. Jerarquía de directorios en de una aplicación para Android.....	27
Ilustración 10. Directorio res y sus subdirectorios en un proyecto Android .....	28
Ilustración 11. Google Maps para un dispositivo móvil.....	30
Ilustración 12. Diagrama frontera.....	37
Ilustración 13. Diagrama para el caso de uso "Recomendaciones".....	38
Ilustración 14. Diagrama para el caso de uso "Detalles de un restaurante" .....	38
Ilustración 15. Diagrama para el caso de uso "Contexto" .....	39
Ilustración 16. Diagrama para el caso de uso "Datos de usuario" .....	39
Ilustración 17. Diagrama de clases .....	45
Ilustración 18. Icono para la metáfora "Restaurantes" .....	47
Ilustración 19. Icono para la metáfora "Desplegable" .....	48
Ilustración 20. Icono para la metáfora "Localización" .....	48
Ilustración 21. Icono para la metáfora "Perfil de usuario" .....	48
Ilustración 22. Icono para la metáfora "Ir atrás" .....	49
Ilustración 23. Icono para la metáfora "Estrellas de valoración" .....	49

---

Ilustración 24. Icono para la metáfora "Editar" .....	49
Ilustración 25. Icono para las metáforas "Caminar" y "Coche" respectivamente .....	49
Ilustración 26. Icono para la metáfora "Tick" .....	50
Ilustración 27. Icono para la metáfora "Atención" .....	50
Ilustración 28. Pantalla "Identificación" .....	51
Ilustración 29. Pantalla "Inicio".....	52
Ilustración 30. Pantalla "Listado de restaurantes" .....	53
Ilustración 31. Pantalla "Detalles de un restaurante" .....	54
Ilustración 32. Pantalla "Localización" .....	55
Ilustración 33. Pantalla "Perfil de usuario" .....	56
Ilustración 34. StoryBoard para “Identificación” .....	57
Ilustración 35. StoryBoard para “Obtener la lista de restaurantes” .....	58
Ilustración 36. StoryBoard para “Obtener detalles de un restaurante” .....	59
Ilustración 37. StoryBoard para “Valorar restaurante” .....	60
Ilustración 38. StoryBoard para “Obtener los detalles de localización” .....	61
Ilustración 39. StoryBoard para “Cambiar la configuración de la localización” .....	63
Ilustración 40. Storyboard para “Obtener el perfil de usuario” .....	64
Ilustración 41. Storyboard para “Modificar el perfil de usuario”.....	66
Ilustración 42. StoryBoard para “Desvincular cuenta de usuario” .....	68
Ilustración 43. Tema Holo Light with dark action bars de Android.....	69
Ilustración 44. Icono de la aplicación .....	70
Ilustración 45. Arquitectura cliente - servidor.....	71
Ilustración 46. Activación de opciones de instalación en el dispositivo .....	88
Ilustración 47. Copiar el archivo REJApp.apk desde REJApp/bin .....	89
Ilustración 48. Copiar el archivo REJApp.apk en la tarjeta de memoria SD del dispositivo.....	89

Ilustración 49. Aplicación REJApp en la memoria SD del dispositivo.....	90
Ilustración 50. Mensajes de confirmación para instalar el archivo REJA.apk .....	90
Ilustración 51. Aplicación REJApp instalada en el dispositivo.....	91
Ilustración 52. Identificación en REJApp.....	94
Ilustración 53. Inicio en REJApp .....	95
Ilustración 54. Listado de recomendaciones en REJApp .....	95
Ilustración 55. Detalles de un restaurante en REJApp .....	96
Ilustración 56. Llamada a un restaurante en REJApp .....	97
Ilustración 57. Valorar un restaurante en REJApp .....	98
Ilustración 58. Localización en REJApp .....	99
Ilustración 59. Modificar distancia en REJApp.....	100
Ilustración 60. Inicio en modo horizontal en REJApp .....	101
Ilustración 61. Perfil de usuario en REJApp .....	101
Ilustración 62. Modificación del perfil de usuario en REJApp .....	102
Ilustración 63. Desvincular cuenta de usuario en REJApp .....	103
Ilustración 64. Android SDK Manager .....	106
Ilustración 65. Importar las librerías de Google Maps .....	107
Ilustración 66. Añadir las librerías de Google Maps a nuestro proyecto .....	108

---

## Índice de tablas

---

Tabla 1. Versiones del sistema operativo Android.....	23
Tabla 2. Resultados obtenidos para los casos de test .....	78

## Índice de gráficas

---

Gráfica 1. Comparativa de la cuota de mercado para las distintas plataformas móviles [18]..... 22

# 1 Introducción

---

## 1.1 Contexto y visión general

Internet es una herramienta de fácil acceso a cantidades masivas de información. En el ámbito del turismo, existen multitud de portales que ofertan hoteles, spas, restaurantes... y todo tipo de rutas por los puntos de interés de ciudades o pueblos, actividades, parques temáticos... etc. A pesar de ello, toda esta información no siempre resulta de utilidad para el usuario, que puede sentirse saturado ante la multitud de opciones que se le ofrecen.

Como herramienta de ayuda a solventar este tipo de problemas surgen los sistemas de recomendación [4]. Un sistema de recomendación se encarga de filtrar la información, mostrando al usuario aquella que le es relevante de acuerdo con algunos parámetros personalizables, como podrían ser, por ejemplo, su cultura, gustos o preferencias, interés mostrado en otros productos con anterioridad o perfil similar al de otros usuarios del sistema.

En función del tipo de parámetros que se tienen en cuenta a la hora de llevar a cabo una recomendación, podemos distinguir entre sistemas de recomendación demográficos, basados en contenido, basados en conocimiento, de filtrado colaborativo, basados en utilidad e híbridos. Estos últimos consisten en combinar varios de los tipos mencionados de forma conjunta a la hora de hacer una predicción. Distintos ejemplos los podemos encontrar en [1] , [2], trabajos en los que se propone una arquitectura para un sistema de recomendación que combina el filtrado colaborativo y el filtrado basado en conocimiento.

Entre las líneas de investigación más actuales en el campo de los sistemas de recomendación se encuentra la utilización del contexto a la hora de realizar recomendaciones. Un sistema de recomendación basado en contexto tiene en cuenta nuevos aspectos, que influyen a la hora de realizar una recomendación con éxito por el sistema, tales como el lugar donde el usuario se encuentra, el clima actual, nivel de experiencia en el uso del dispositivo... etc.

Por tanto, muy ligados a ésta nueva tendencia se encuentran los dispositivos móviles, por poseer unas características propias que hacen que se pueda obtener mayor información acerca del entorno del usuario, e inferir ciertos factores que pueden ser de gran utilidad a la hora integrar contexto al llevar a cabo una recomendación. En el sistema presentado en [3] se toman en cuenta variables tales como la ubicación, velocidad y trayectoria que sigue un usuario de un dispositivo móvil para implementar un sistema de recomendación basado en contexto, capaz de calcular el área y puntos de interés de acuerdo con los parámetros capturados por el dispositivo.

Aprovechando dicho potencial, y centrándonos en la localización o posición física del usuario en cada momento, obtenida a través del dispositivo móvil, nuestro propósito será desarrollar una aplicación cliente dedicada al turismo, que permita obtener recomendaciones acerca de restaurantes en la ciudad de Jaén, teniendo en cuenta la localización actual del usuario a partir de un sistema de recomendación web [1] que procesará recomendaciones iniciales atendiendo a los gustos y/o preferencias del usuario.

Esta aplicación se desarrollará para ser ejecutada en dispositivos que utilicen el sistema operativo Android. Android es un sistema operativo basado en Linux [14] desarrollado por la compañía Google Inc. [15] cuyas características han hecho que se convierta en la plataforma dominante en cuanto a cuota de mercado actualmente [18] (véase apartado 3.1).

## 1.2 Propósito y objetivos

### *Propósito:*

Diseño y desarrollo de una aplicación Android de recomendación de restaurantes en la ciudad de Jaén, basada tanto en preferencias del usuario como en el contexto.

### *Objetivos:*

- Estudiar el desarrollo de aplicaciones para sistemas Android.
- Estudiar el sistema de recomendación a utilizar y los modelos para integrar información de contexto.
- Definir la arquitectura y parámetros que permitan conectar el sistema de recomendación con la aplicación móvil [1].
- Estudiar la API Android Maps, que integra Google Maps en una aplicación Android.
- Implementar la aplicación.
- Completar la memoria.

## 1.3 Resultados esperados

Se espera como resultado:

- Aplicación móvil que implementa un sistema de recomendación geolocalizado basado en contexto de restaurantes en Jaén.

- Memoria del proyecto.

## 1.4 Investigaciones relacionadas

En [1] se presenta una visión general de los distintos sistemas de recomendación, ventajas y problemática que presentan. Asimismo, se realiza una pequeña introducción a los sistemas de recomendación basados en contexto y a las características que presentan los dispositivos móviles aplicadas en los mismos. El documento propone la arquitectura para un sistema de recomendación híbrido, que combina el filtrado colaborativo y el filtrado basado en conocimiento, adaptando las recomendaciones al contexto de la localización o ubicación del usuario que ejecuta la aplicación en el dispositivo móvil. Como cliente de esta arquitectura, se propone una interfaz para el sistema iOS.

La arquitectura descrita en la parte servidor será utilizada en el desarrollo de la aplicación que se pretende realizar a lo largo de esta propuesta, desarrollando ahora un cliente para el sistema Android.

En [3] se hace una introducción a los sistemas de recomendación basados en contexto para dispositivos móviles, atendiendo a la ubicación, velocidad y trayectoria que sigue el dispositivo. El documento especifica los parámetros a través de los cuales el cliente hace peticiones acerca de las recomendaciones al servidor, y como éste es capaz de calcular el área y punto de interés dentro de la misma en tiempo real atendiendo, además, a los parámetros de movimiento que realiza el usuario que porta el dispositivo.

Finalmente en [2], se hace un breve repaso a la arquitectura del servidor de recomendaciones que se explica en [1], acerca de los módulos de recomendación basada en el contexto de la localización y la reordenación de la lista final de recomendaciones que se muestran al usuario dentro de un área especificada.

## 1.5 Planificación del proyecto

Un primer paso en la realización del proyecto consiste en identificar las distintas tareas que han de llevarse a cabo para lograr alcanzar los objetivos marcados en 1.2 y planificar cómo y cuándo estas tareas han de realizarse. Para ello hemos de establecer un orden en la realización de las mismas y estimar el tiempo necesario para completar cada una de ellas.

### 1.5.1 Actividades

A continuación vamos a listar todas las actividades que necesitamos completar a lo largo de la realización de este proyecto:

1. Inicio.
2. Búsqueda bibliográfica.
3. Estudio de la arquitectura del sistema de recomendación a utilizar.
4. Revisión bibliográfica.
5. Estudiar el desarrollo de aplicaciones para el sistema operativo Android.
6. Estudiar la API de Google Maps.
7. Definir los parámetros de comunicación entre cliente y servidor.
8. Construir el diagrama de interacción.
9. Construir el diagrama de clases.
10. Codificar.
11. Escribir la memoria.
12. Fin.

### 1.5.2 Esquema de división del trabajo

A continuación vamos a presentar un esquema de división del trabajo que será necesario llevar a cabo para alcanzar los objetivos de este proyecto, basándonos en las actividades descritas en el apartado anterior.

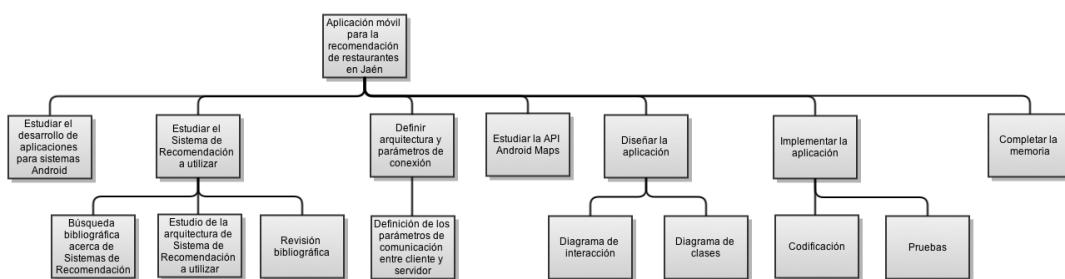


Ilustración 1. Esquema de división del trabajo

### 1.5.3 Diagrama de Gantt

Una vez obtenidas todas las actividades, hemos de hacer una asignación temporal para saber lo que debemos de hacer en cada jornada de desarrollo del proyecto.

Un diagrama de Gantt representa las actividades que componen un proyecto junto con la duración de cada una de ellas. Cada nodo representa a una actividad y su longitud la duración de la misma.

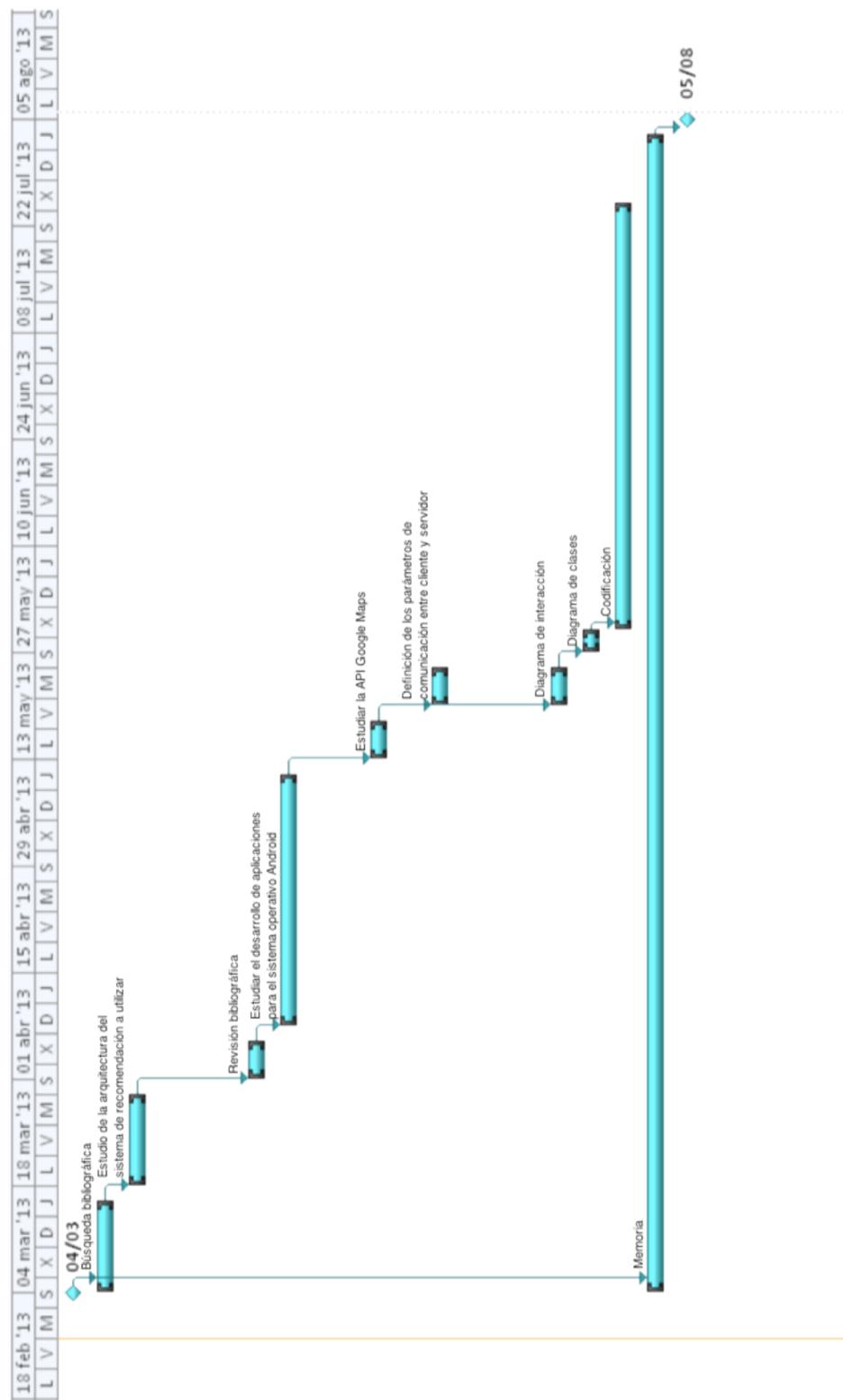


Ilustración 2. Diagrama de Gantt



## 2 Sistemas de recomendación

---

A lo largo de este capítulo haremos una introducción acerca de qué son los sistemas de recomendación y cuáles son los principales tipos de estos sistemas, haciendo especial hincapié en los sistemas de recomendación basados en contexto por su utilidad en este proyecto. De forma particular se presentará REJA, el sistema de recomendación utilizado por la aplicación cliente que vamos a desarrollar.

## 2.1 Introducción

El boom de Internet, la web 2.0 o, más recientemente, la aparición de nuevos dispositivos móviles con conexión a Internet, han propiciado que cada vez existan más usuarios conectados a la red durante más tiempo, y que empresas, organismos públicos y todo tipo de organizaciones estén presentes ofreciendo todo tipo de productos y servicios, de tal forma que sea difícil pensar que exista cualquier cosa que no sea posible obtener a través de la red.

Pese a la situación económica actual, uno de los campos que se encuentra en auge es el del comercio electrónico. Sólo en España, el volumen de ventas aumentó en un 27% durante 2012 [8]. Es obvio que, el hecho de que el comercio electrónico crezca de forma tan rápida se debe a que hay multitud de empresas presentes en la red y multitud de usuarios dispuestos a comprar a través de ella, lo que significa que cada vez encontramos más bienes, productos o servicios de todo tipo en Internet.

Esta cantidad de información a veces satura al usuario, bien porque no conoce todas las alternativas que se le ofrecen, o bien porque no tiene la experiencia suficiente para filtrar la información irrelevante. Por ello, es frecuente que busque la opinión de terceros acerca de un producto o servicio como respaldo para la toma de decisiones antes de decidir que alternativa es la que más le conviene.

Por otro lado, las empresas buscan dar un mejor servicio a sus clientes, y suplir de algún modo el trato que se les da en un negocio físico, y que no es posible dar a través de una tienda virtual.

Dentro de este contexto, surgen los sistemas de recomendación que se presentan como la solución que cumple tanto las expectativas de las empresas como las de los usuarios, permitiendo a las primeras recomendar productos a los clientes en función de sus preferencias, y ayudar a los segundos a entender mejor la información que se les ofrece.

Los sistemas de recomendación son un conjunto de herramientas software y técnicas basadas en inteligencia artificial para proporcionar sugerencias acerca de ítems que puedan ser de utilidad

para un usuario [4] en un espacio sobrecargado de información. Se trata, por tanto, de realizar un filtrado de la información de la que se dispone que permita obtener aquellos ítems en los que cada usuario en particular podría estar interesado. Para ello, el sistema de recomendación puede contrastar la información del perfil del usuario y filtrar toda la información que le es irrelevante a través del cálculo de valores que predicen la valoración que el usuario haría de un ítem. Finalmente, el sistema recomendará una serie de ítems a dicho usuario en particular, que serán aquellos que mejor puntuación han obtenido tras la predicción.

Un referente en la utilización de este tipo de sistemas en el comercio online es Amazon [5], que utiliza un sistema con recomendaciones acerca de multitud de productos diferentes (libros, telefonía móvil, pcs...) a sus usuarios.

Además de su utilización en el ámbito del comercio electrónico, los sistemas de recomendación, son utilizados comúnmente en los siguientes dominios [4]:

- Entretenimiento: recomendaciones de películas, música...
- Contenidos: noticias, documentos, páginas web, e-learning, filtros para e-mail.
- Servicios: viajes, consultoría, alquileres, webs de citas.

Las razones que motivan la aparición de este tipo de sistemas junto con la variedad de campos en los que resulta útil su aplicación, hacen que nos encontremos ante una técnica en continuo desarrollo y con grandes expectativas de futuro.

## 2.2 Elementos de un sistema de recomendación y esquema general de funcionamiento

Aunque existen muchos tipos de sistemas de recomendación distintos en función de la información y técnicas que utilizan a la hora de realizar la recomendación propiamente, podemos decir que, en general, existen tres elementos comunes que están presentes en todos ellos [4]:

- **Ítems:** un ítem es todo aquello susceptible de ser ofrecido o recomendado por el sistema. Cada ítem contará con una serie de características propias que pueden ser utilizadas durante el proceso de recomendación, y cuya complejidad dependerá en muchos casos del ítem en sí. Por ejemplo, una película o un teléfono móvil.

- **Usuarios:** representan a personas reales. Recordemos que el objetivo de un sistema de recomendación es el de ofrecer ítems que resulten interesantes a cada usuario en concreto. Por ello, la información de la que el sistema dispone acerca de cada usuario es fundamental.
- **Transacciones:** hacen referencia a cada interacción entre un usuario y el sistema de recomendación. Contienen información relevante acerca de las opiniones de los usuarios sobre los ítems.

El conjunto de la información que proporcionan estos tres elementos puede ser procesada de una forma u otra por un sistema, de modo que, en función de las características que se utilizan a la hora de generar una recomendación, se distinguirán los distintos tipos de sistemas de recomendación. Sin embargo, existe un proceso general que es llevado a cabo por todos los sistemas de recomendación integrando estos tres componentes (Veáse Ilustración 3).

Dado un usuario en concreto para el cual se desea realizar una recomendación, llamado formalmente *usuario activo*, y utilizando la información almacenada acerca de usuarios, ítems, y transacciones, se trata de predecir qué ítems del conjunto serían valorados positivamente por el usuario. Tales ítems con mejor valoración serán los recomendados al usuario activo.

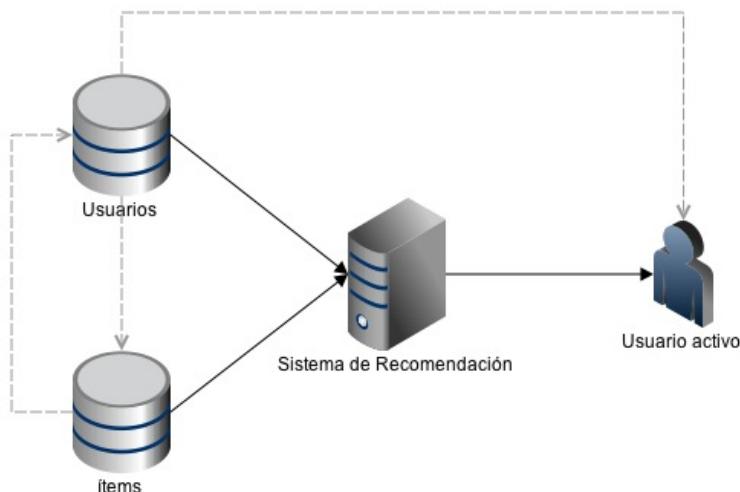


Ilustración 3. Esquema general de funcionamiento de un sistema de recomendación.

## 2.3 Clasificación de los sistemas de recomendación

Un sistema de recomendación puede ser implementado utilizando diferentes técnicas, basándose en distintas características tanto del usuario como del ítem, o siguiendo un modo u otro de funcionamiento.

En general, los sistemas de recomendación tienen en cuenta una de las siguientes fuentes de información [1]:

- Preferencias del usuario entre distintas alternativas.
- Preferencias del usuario sobre las características del ítem.
- Preferencias u opiniones de otros usuarios.
- Juicios realizados por expertos.
- Características personales que ayuden a predecir preferencias del usuario.

En base a todo ello, se pueden distinguir seis tipos de sistemas de recomendación [4]:

- **De filtrado colaborativo:** calculan una medida de similitud entre usuarios en función del historial de ítems en los que cada usuario mostró interés (por ejemplo, productos que el usuario ha adquirido o en base a una puntuación o valoración del usuario hacia el producto). El sistema recomienda entonces los ítems que les gustaron a otros usuarios con gustos similares al usuario activo. Este tipo de sistemas es uno de los más utilizados actualmente ya que toman en cuenta las valoraciones que los usuarios hacen de los ítems y no las características propias de los mismos. Su utilización resulta muy eficaz cuando se manejan grandes bases de datos con multitud de perfiles de usuario e ítems [6].
- **Basados en contenido:** inicialmente utilizaban técnicas de recuperación de información que modelan espacios vectoriales, en los que cada vector representa las características de ítems o usuarios. El sistema calcula una medida de similitud existente entre los ítems que se ofrecen basándose en las características propias de dichos ítems a través del análisis de metadatos. A la hora de llevar a cabo la recomendación al usuario activo, se escogerán los ítems similares a otros que el usuario ha valorado positivamente con anterioridad. Este enfoque resulta muy útil cuando no se dispone de gran número de usuarios registrados y, por tanto, de esa “inteligencia colectiva” que aporta el análisis de su comportamiento [6] utilizada en los sistemas de filtrado colaborativo.

- **Demográficos:** en función del perfil demográfico del usuario (edad, sexo, etc.), se crean grupos sociales, es decir, grupos de usuarios similares de acuerdo a la información que aportan en su perfil. Las recomendaciones están basadas en estos grupos [1]; para un usuario activo, se recomendarán los ítems en los que mostraron interés aquellos usuarios de su grupo.
- **Basados en conocimiento:** este tipo de sistemas realizan recomendaciones basadas en las necesidades del usuario y en las características de los ítems, es decir, cómo ciertas características del ítem cubren las necesidades o preferencias del usuario.
- **Sociales:** este tipo de sistemas se apoyan en la idea de que, en el mundo real, las personas toman más en cuenta las opiniones de sus amigos que la de las personas que, aún teniendo un perfil similar al suyo, son desconocidas. Se trata, por tanto, de recopilar información acerca del ambiente social del usuario activo, y recomendar los ítems en función de la valoración que las personas de su entorno o conocidas han hecho de cada uno de ellos. Este tipo de sistemas han llegado a ser muy populares gracias al uso de las redes sociales.
- **Híbridos:** este tipo de sistemas utilizan una combinación de los expuestos anteriormente, de forma que las debilidades que pueda presentar un tipo de sistema de recomendación en concreto puedan ser corregidas a través de la utilización de otro enfoque.

## 2.4 Sistemas de recomendación basados en contexto

Además de los tipos de sistemas de recomendación expuestos en el apartado 2.3, actualmente están surgiendo nuevos métodos que permiten incorporar otro tipo de información al proceso de recomendación de los sistemas anteriores. Dentro de estas nuevas tendencias se encuentran los sistemas de recomendación basados en contexto.

Los sistemas de recomendación tradicionales, como hemos explicado en el apartado 2.2, realizan el proceso de recomendación basándose únicamente en dos tipos de entidades: usuarios e ítems. Sin embargo, en muchas aplicaciones resulta muy útil incorporar cierta información relativa al contexto en el proceso de recomendación con la finalidad de recomendar ítems a los usuarios de acuerdo con ciertas circunstancias [4].

A modo de ejemplo, supongamos que tenemos una aplicación para la recomendación de restaurantes: si el usuario activo se encuentra en Londres, no tiene sentido recomendarle un restaurante en París (aunque este restaurante pudiese gustarle) [3]. De igual modo, en un sistema de recomendación de viajes debería de tener en cuenta la estación del año a la hora de recomendar un viaje. Otro ejemplo: la radio interactiva Sourcetone [9] tiene en cuenta el estado de ánimo del oyente a la hora de recomendarle una canción. De este modo, existen muchos casos en los que la información contextual resulta determinante a la hora de realizar una buena recomendación.

Un sistema de recomendación basado en contexto se define en [7] como aquel que toma en consideración distintos factores (temporales, de lugar, nivel de experiencia del usuario, dispositivo que utiliza en el momento de realizar la recomendación...) para inferir en el contexto en que se encuentra el usuario y adaptar la recomendación a estas circunstancias.

#### **2.4.1 El contexto en los sistemas de recomendación**

Según la R.A.E, el contexto se puede definir como el entorno físico o de situación, ya sea político, histórico, cultural o de cualquier otra índole, en el cual se considera un hecho. El contexto es, por tanto, un concepto complejo que en función del entorno en que se considera (comercio electrónico, bases de datos, localización y dispositivos móviles...) puede abarcar multitud de factores o aspectos propios de cada uno de ellos.

Para poder incorporar toda esta información al proceso de recomendación será necesario construir un modelo que permita, de algún modo, organizar o abstraer todos esos atributos que conforman el entorno o del contexto del sistema de recomendación. En [4] se proponen dos enfoques distintos para llevar a cabo esta tarea:

- **Modelado en árboles:** la información contextual se representa mediante una jerarquía en árboles de niveles. A más profundidad en el árbol, más detallada es la información.
- **Modelado multidimesional:** se trata de una organización jerárquica en matrices multidimensionales, en las que dos dimensiones representan al conjunto de los usuarios y de los ítems, y el resto al conjunto de atributos que definen un contexto en particular (véase ilustración 4).

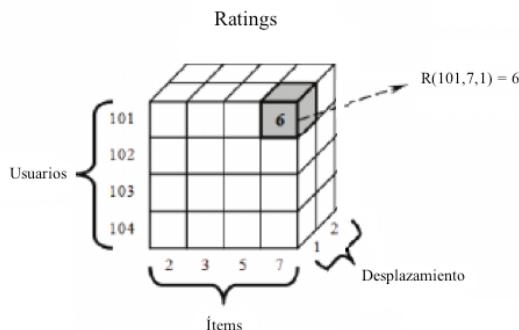


Ilustración 4. Matriz multidimensional en modelado contextual

#### 2.4.2 Métodos para incorporar el contexto en sistemas de recomendación

Como ya hemos definido, los sistemas de recomendación basados en contexto son aquellos que incorporan características relativas al entorno en el proceso de recomendación. Para ello, existen tres enfoques principales en la literatura especializada y que están gráficamente descritos en la Ilustración 5 [4] que pasamos a describir, en líneas generales, a continuación:

- **Prefiltrado contextual:** en este caso, la información que aporta el contexto se utiliza para extraer un subconjunto dentro del conjunto total de los ítems de los que se disponen, formado por todos aquellos que cumplen con las restricciones que establece dicho contexto. Una vez hecho esto, se aplica el modelo general de funcionamiento para los sistemas de recomendación descrito en 2.2 sobre este subconjunto. Esto quiere decir que el modelo de predicciones está construido sobre un contexto específico, ya que todos los ítems que lo componen se han seleccionado previamente de acuerdo a ese contexto [Ilustración 5.a].

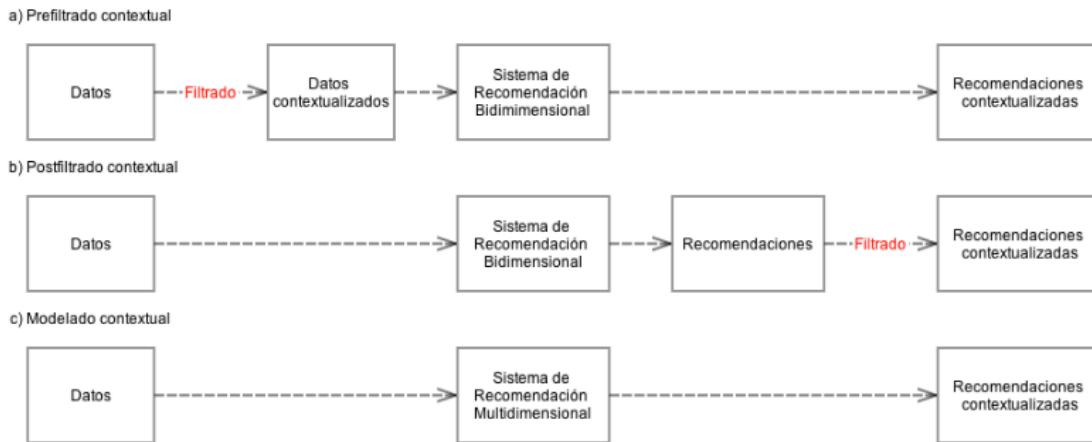
A modo de ejemplo, consideremos un sistema para la recomendación de restaurantes, en el que el usuario indica que se desplaza a pie y en el que disponemos de la localización del usuario en el momento de llevar a cabo la recomendación. En tal caso, el prefiltrado contextual basándose en la localización del usuario, realizaría un filtrado para excluir todos aquellos restaurantes lo suficientemente alejados del lugar en que se encuentra el usuario para que este se pueda desplazar hacia ellos a pie. Una vez hecho esto, el sistema calculará los restaurantes a recomendar únicamente sobre el subconjunto de aquellos que cumplen con los requisitos impuestos por el contexto.

- **Postfiltrado contextual:** este enfoque sigue el proceso inverso al anterior. En primer lugar, el contexto es ignorado y se construye el conjunto de recomendaciones siguiendo el esquema tradicional [Ilustración 3] en el que sólo se utiliza información sobre usuarios e ítems. El paso siguiente consiste en contextualizar ese conjunto. Se trata de realizar un ajuste del listado de recomendaciones para cada usuario, descartando aquellas que no cumplen con los requisitos que establece el contexto [Ilustración 4.b]. Este proceso puede llevarse a cabo bien filtrando cada recomendación y excluyendo aquellas que no sean relevantes debido al contexto, o bien reajustando el ranking de recomendaciones en la lista.

Siguiendo con el ejemplo expuesto en el apartado anterior, en este caso, el sistema realizará el proceso de recomendación previamente y, una vez obtenido el listado de restaurantes a recomendar, se realizará un filtrado para obtener únicamente aquellos a los que el usuario puede desplazarse a pie desde el punto en que se encuentra ubicado.

- **Modelado contextual:** a diferencia de los dos enfoques presentados anteriormente, que realizaban un filtrado basado en el contexto previo o posterior al proceso de recomendación, este pretende incluir la información contextual disponible en el proceso de cálculo de predicciones. Este planteamiento, por tanto, rompe con los dos anteriores, y propone un modelo multidimensional, en el que se calculan predicciones en base a usuarios, ítems y contexto [Ilustración 5.c].

En este caso, siguiendo con el ejemplo sobre la recomendación de restaurantes de los apartados anteriores, la localización del usuario y el hecho de que se desplace a pie serán dos parámetros a tener en cuenta durante el proceso de cálculo de predicciones. Una forma de representación en una matriz multidimensional sería el mostrado en la ilustración 4, en la que se presenta una matriz de *Usuario x Ítem x Desplazamiento*, y en la que desplazamiento tiene dos dimensiones que se corresponden con los valores *a pie* o *en coche*, respectivamente.



**Ilustración 5. Modelos para la incorporación del contexto en los sistemas de recomendación**

Los tres enfoques descritos no han de ser considerados excluyentes, sino que se pueden combinar entre sí, además de con los sistemas de recomendación tradicionales como ya hemos visto para formar sistemas contextuales híbridos.

## 2.5 REJA: un sistema de recomendación de restaurantes en Jaén

### 2.5.1 Sistema actual

Cada vez más, los usuarios acuden a la red a la hora buscar ofertas turísticas y de ocio. Sin embargo, como ya hemos expuesto a lo largo de este trabajo, la cantidad de información disponible resulta difícil de explorar y es complicado encontrar aquello que nos satisface.

REJA es un sistema de recomendación de restaurantes para la provincia de Jaén que pretende solventar este problema. Se trata de ofrecer restaurantes que el usuario aún no ha visitado y que puedan ser de su interés, basándose en puntuaciones que el mismo usuario realizó acerca de otros restaurantes en cuanto a calidad y servicio a través de la web [27].

REJA fué desarrollado por el grupo de investigación Sinbad2 [26] de la Universidad de Jaén, y actualmente dispone de un cliente dividido en dos partes: una aplicación web para la recomendación de restaurantes y una aplicación web para la georreferenciación y geocodificación de restaurantes [27].

### 2.5.2 Aplicación móvil para introducir contexto en REJA

Tratando de aprovechar el potencial que ofrecen los dispositivos móviles cada día más utilizados, sería conveniente que REJA incorporase al proceso de recomendación características relativas al contexto que permitan realizar predicciones más acertadas. Estas características tienen que ver con la localización y medio de desplazamiento del usuario.

REJA es un sistema de recomendación híbrido que combina los modelos colaborativo y basado en conocimiento. La información contextual relativa a la localización del usuario podría ser incorporada a través de dos enfoques distintos de sistemas de recomendación basados en contexto a implementar en dos módulos [1]:

- 1- **Prefiltrado contextual:** se incorpora tanto en el módulo de recomendación colaborativo como en el módulo basado en conocimiento. Como se ha descrito en el apartado 2.4.2, previo al cálculo de recomendaciones se realiza un filtrado para aquellos restaurantes que se encuentran dentro del área considerada.
- 2- **Postfiltrado contextual:** una vez que se obtiene el listado de recomendaciones para el usuario activo, este se reorganiza según la distancia física del usuario a cada restaurante.

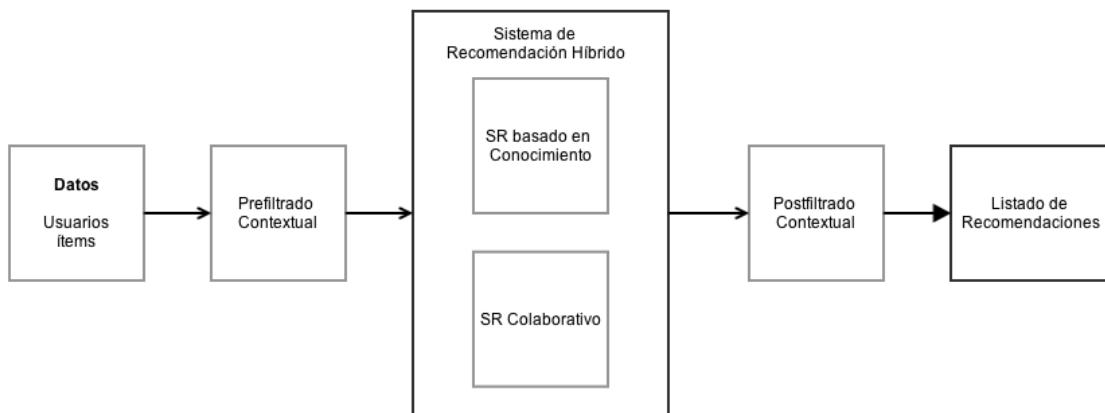


Ilustración 6. Esquema de funcionamiento del sistema de recomendación REJA

En la ilustración 6 podemos ver un esquema de cómo sería sistema de recomendación REJA basado en contexto. Con la utilización de un sistema híbrido a la hora de elaborar una lista de recomendaciones se pretenden solventar los problemas que puedan tener cada uno de los sistemas de forma particular.

En este caso, cabe destacar que los sistemas de recomendación de filtrado colaborativo presentan las siguientes limitaciones [1]:

- **Problema de la oveja gris:** ocurre cuando, al formar los grupos de usuarios interesados en ítems similares, algunos de esos usuarios no se corresponden exactamente con ningún grupo, y por ello no reciben buenas recomendaciones.
- **Histórico de datos:** cuando el histórico de datos (transacciones) que se tiene es muy pequeño, es complicado que el sistema pueda generar buenas predicciones.
- **Problema de arranque en frío:** ocurre con los usuarios que se acaban de registrar en el sistema o llevan poco tiempo utilizando, de los que se dispone de poca o ninguna información que permita compararlos con otros usuarios y proporcionarles así recomendaciones.

Cuando alguna de estas situaciones ocurre, REJA puede aportar recomendaciones a través del módulo de recomendación basado en conocimiento.

Para acceder al servicio de recomendación de REJA a través de una aplicación móvil existe un programa llamado *rejamovil.php*, que recibe los parámetros de nombre de usuario, latitud y longitud relativos a la localización, de modo de desplazamiento (andando o en coche) y devuelve un archivo .xml con la respuesta.

Un ejemplo de petición de recomendaciones sería

<http://serezade.ujaen.es/~reja/rejamovil/rejamovil.php?id=Manolo&lat=37.0000&long=-3.5000&aoc=C>

en el que se indica a *Manolo* como usuario, latitud *37.0000*, longitud *3.5000* y desplazamiento en coche *C*.

Este servicio será el que utilicemos a la hora de obtener las recomendaciones para un usuario de la aplicación Android REJApp que queremos desarrollar en esta memoria.

## 3 Android y Geolocalización

---

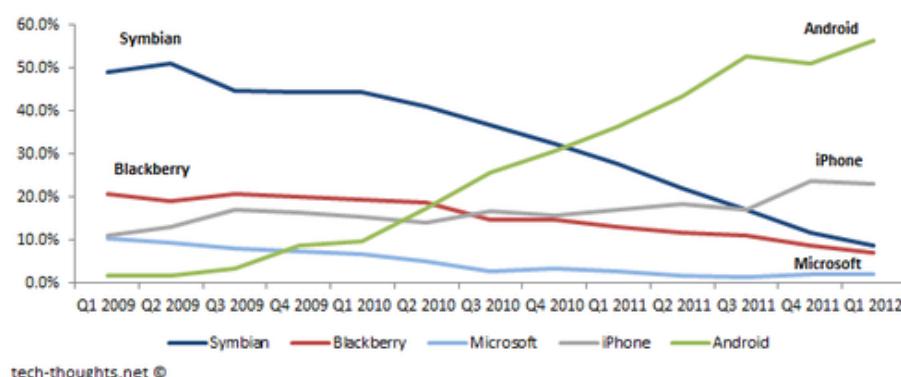
A lo largo de este capítulo vamos a describir qué es el sistema operativo Android incluyendo una descripción de las características más destacadas que hemos de conocer acerca del mismo para entender el desarrollo de la aplicación que realizamos en este proyecto. También se hará una breve introducción a la geolocalización como concepto utilizado para inferir contexto a través de nuestra aplicación a la hora de realizar recomendaciones por el sistema REJA basado en contexto.

### 3.1 Android

Android es un sistema operativo basado en Linux [14] para dispositivos móviles creado por Andy Rubin. En 2005 fué adquirido por Google Inc. [15] y Andy Rubin pasó a formar parte de la compañía como director de plataformas móviles para Google.

En el año 2007, cerca de 100 grandes compañías de varios sectores dentro de las telecomunicaciones, dispositivos móviles, semiconductores, desarrollo software y comercialización, formaron la Open Handset Alliance [16] con el objetivo de desarrollar estándares abiertos para móviles. El sistema operativo Android es una pieza fundamental de esta alianza, y parte de su código se encuentra liberado bajo la licencia Apache.

El primer lanzamiento de Android se llevó a cabo en noviembre de 2007, y en octubre del año siguiente se abrió Android Market para la descarga de aplicaciones. A partir de entonces, y debido a las características de código abierto, portabilidad y adaptación a cualquier tipo de hardware, optimización en el consumo de memoria o alta definición para gráficos y sonido [14], Android ha experimentado un gran crecimiento, hasta ser la plataforma dominante en cuanto a cuota de mercado según podemos observar en la gráfica 1:



Gráfica 1. Comparativa de la cuota de mercado para las distintas plataformas móviles [18]

Todas las características descritas anteriormente, así como la consolidación de Android como la plataforma líder para dispositivos móviles y la relativa facilidad a la hora desarrollar aplicaciones en cuanto a información disponible desde webs oficiales para desarrolladores han hecho que Android haya sido elegida como la plataforma idónea para la realización de este proyecto.

### ***Evolución de Android: versiones y tasa de distribución***

En la tabla 1 podemos observar, desde su lanzamiento en 2007, la evolución en cuanto a versiones y nivel de API de la plataforma Android [14, 17]. Cada versión tiene un nombre comercial que hace referencia a un postre, siguiendo un orden alfabético.

Version	Nombre	Logo	API	Distribución
<b>Android 1.0</b>	Apple Pie		1	< 0.1%
<b>Android 1.1</b>	Banana Bread (Petit Four)		2	< 0.1%
<b>Android 1.5</b>	Cupcake		3	< 0.1%
<b>Android 1.6</b>	Donut		4	0.1%
<b>Android 2.0 / 2.1</b>	Éclair		7	1.2%
<b>Android 2.2</b>	Froyo		8	2.5%
<b>Android 2.3</b>	Gingerbread		10	33%
<b>Android 3.0</b>	Honeycomb		13	0.1%
<b>Android 4.0</b>	Ice Cream Sandwich		15	22.5%
<b>Android 4.1</b>	Jelly Bean		16	34%

Tabla 1. Versiones del sistema operativo Android

En cuanto a la tasa de distribución, como podemos observar en la ilustración 7 [17], Jelly Bean, Ice Cream Sandwich y Ginger Bread son las más utilizadas. Esta información resulta de gran utilidad a la hora de desarrollar una aplicación, ya que es necesario indicar la versión mínima que nuestra aplicación puede soportar: esto significa que la aplicación no podrá ser instalada en dispositivos con versiones inferiores a la versión mínima escogida.

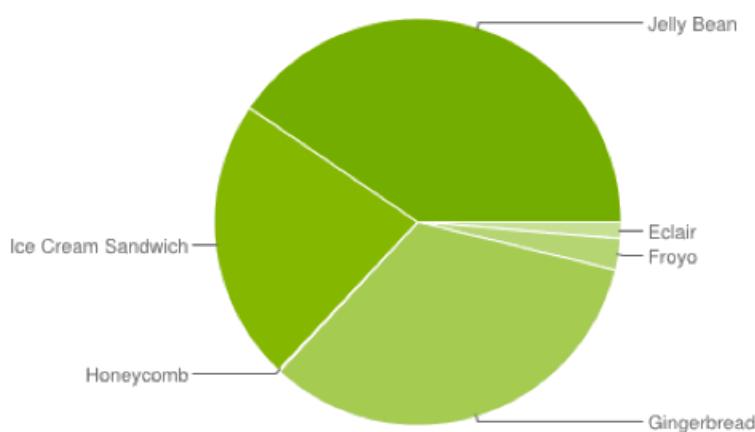


Ilustración 7. Tasa de distribución para las versiones de Android

### 3.1.1 Arquitectura

Como podemos ver en la ilustración 8, Android tiene una arquitectura dividida en cuatro capas [14]:

- **Núcleo:** está formado por el sistema operativo Linux v2.6. Actúa, por tanto, como interfaz entre la aplicación y el dispositivo hardware, y proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos. Es de código abierto bajo licencia GPL v2 [19].
- **Runtime:** basado en el concepto del máquina virtual de Java (lenguaje utilizado para el desarrollo de aplicaciones para Android), Google desarrolló una nueva máquina virtual acorde a las limitaciones de los dispositivos móviles, llamada Dalvik.
- **Librerías nativas:** son un conjunto de librerías en C/C++ compiladas en el código nativo del procesador. Estas librerías proporcionan acceso a las funcionalidades del sistema a través de la capa que explicaremos a continuación.

- **Entorno de la aplicación:** pensada como una capa de acceso a las distintas APIs que permita reutilizar los componentes que la forman fácilmente, lo que simplifica el desarrollo de aplicaciones.
- **Aplicaciones:** conjunto de aplicaciones instaladas en un dispositivo Android. Comprende tanto las instaladas en el dispositivo por defecto (tales como un cliente de correo electrónico, el calendario o la agenda de contactos) como las realizadas por otros desarrolladores disponibles que pueden ser descargadas a través de la plataforma Google Play.

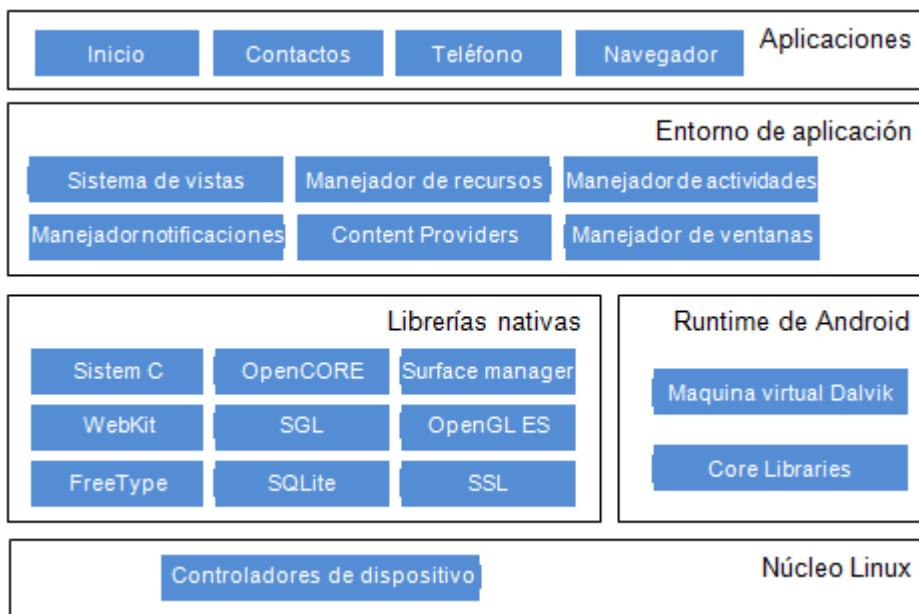


Ilustración 8. Arquitectura de Android

### 3.1.2 Componentes

Los componentes son elementos clave en el desarrollo de aplicaciones para Android. Cada componente representa un punto a través del cual el sistema puede acceder a la aplicación, y algunos de ellos también implementan la lógica de la aplicación tras la interacción del usuario con la interfaz.

Existen cuatro tipos de componentes en Android [14, 20]:

- **Actividades:** una actividad implementa la lógica para una pantalla de la interfaz de usuario. Una aplicación, por tanto, estará formada por un conjunto de actividades independientes, aunque pueden interactuar entre ellas e intercambiar información.

- **Servicios:** un servicio es un componente que se ejecuta en segundo plano y que permite realizar operaciones internas de la aplicación no asociadas a la interfaz de usuario.
- **Proveedores de contenido:** permiten almacenar y compartir el conjunto de la información que manejan las aplicaciones.
- **Receptor de anuncios:** permiten responder a anuncios de tipo broadcast. Estos anuncios pueden ser generados por el sistema o bien por una aplicación. No constan de interfaz, se ejecutan en segundo plano y suelen mostrar una notificación para anunciar la ocurrencia del evento.

Los componentes utilizados para el desarrollo de una aplicación han de ser declarados en el archivo `AndroidManifest.xml`. Cada aplicación tiene el suyo propio. El archivo `AndroidManifest.xml` permite al sistema operativo conocer toda la información esencial acerca de una aplicación antes de ejecutarla. Algunas de las que se incluye, además de los componentes, son los permisos con los que cuenta la aplicación, librerías, versiones soportadas, etc.

### 3.1.3 Seguridad

El sistema operativo Android en sí está basado en privilegios, y cada aplicación que se ejecuta en el sistema, está identificada bajo un identificador de usuario y un identificador de grupo. Además, todas las aplicaciones para Android han de estar firmadas a través de un certificado que permite identificar a su autor.

En cuanto a las aplicaciones en concreto, Android define un nivel de seguridad más que les permite el acceso a las distintas funcionalidades del sistema. Para que una aplicación puede hacer uso de ciertas características protegidas del dispositivo, es necesario declarar los permisos pertinentes dentro del archivo `AndroidManifest.xml` [21].

### 3.1.4 Almacenamiento de información

Android proporciona distintas opciones para el almacenamiento de información. En función de las necesidades de nuestra aplicación hemos de considerar una u otra opción. Vamos a describir algunas de ellas [22]:

- **Preferencias compartidas:** es una clase que permite el almacenamiento de las preferencias del usuario en nuestra aplicación, permitiendo almacenar los valores en pares de tipo clave/identificador y valor asociado a cada clave.
- **Almacenamiento interno:** almacenamiento de datos en la memoria del dispositivo. Estos datos son de carácter privado, lo que significa que sólo nuestra aplicación puede acceder a ellos.
- **Almacenamiento externo:** los dispositivos compatibles con el sistema operativo Android soportan almacenamiento externo. Los archivos guardados en este tipo de almacenamiento son accesibles desde fuera de nuestra aplicación y pueden ser modificados.
- **Bases de datos:** Android proporciona soporte para bases de datos SQLite. Todas las bases de datos creadas desde nuestra aplicación son accesibles desde cualquier clase de la misma, pero no por otras aplicaciones.

### 3.1.5 Organización de un proyecto en Android

Un proyecto de una aplicación para Android está formado por una jerarquía de directorios y archivos que conforman el esqueleto básico de la aplicación (Véase ilustración 9).

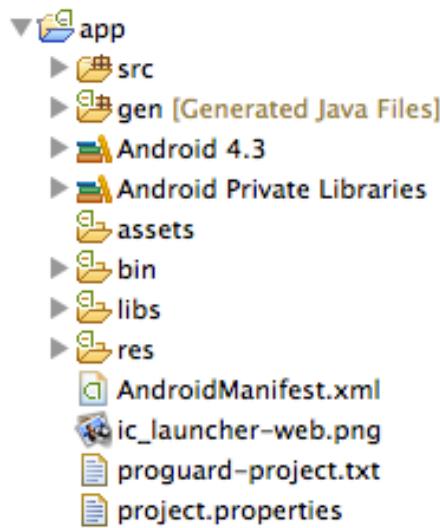


Ilustración 9. Jerarquía de directorios en de una aplicación para Android

A continuación, vamos a introducir brevemente la utilidad y el contenido de cada uno de los distintos archivos y directorios:

- **src:** contiene el código fuente que describe la lógica de la aplicación. Todas las clases Java que la componen se almacenan en este directorio, agrupadas en distintos paquetes o dentro del paquete por defecto que contiene.
- **gen:** la información que contiene este directorio es generada automáticamente por el compilador, y no ha de ser modificada por el programador bajo ningún caso. Almacena ficheros con información relativa a las opciones de configuración del proyecto o de referencias que permiten conectar los elementos entre los distintos directorios.
- **Android 4.3:** contendrá una librería android.jar que permitirá dar acceso a las APIs según las versiones elegidas.
- **Android Private Libraries:** contiene varios archivos .jar que serán las librerías que hemos importado en nuestro proyecto.
- **assets:** en este directorio es un repositorio de archivos en el que podemos incluir cualquier tipo de fichero externo que pueda ser utilizado por la aplicación en un momento determinado.
- **res:** está compuesto por varios subdirectorios que almacenan distintos tipos de recursos como se describe en la ilustración 10:

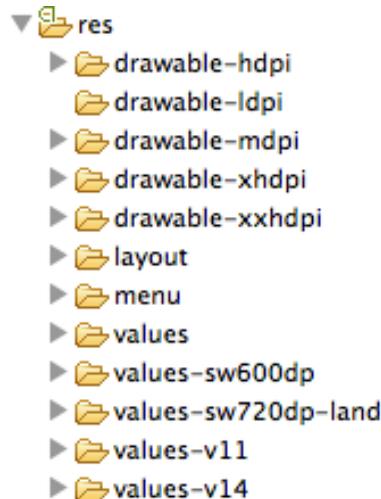


Ilustración 10. Directorio res y sus subdirectorios en un proyecto Android

- **drawable:** contiene a los iconos utilizados en la aplicación. Cada una de los directorios drawable hace referencia a una densidad o tamaño de la pantalla del dispositivo que ejecuta la aplicación.

- **layout:** contiene los archivos .xml en los que se han definido las distintas pantallas o interfaces de la aplicación.
- **menu:** contiene los archivos .xml en los que se define la interfaz de las barras de menú utilizadas en la aplicación.
- **values:** contiene los archivos dimens.xml, strings.xml y styles.xml que hacen referencia a dimensiones, cadenas de texto y estilos de diseño utilizados por la aplicación respectivamente.
- **AndroidManifest.xml:** descrito en el apartado 3.1.2.
- **proguard-project.txt, project.properties:** archivos de configuración generados automáticamente por el compilador que describen ciertas propiedades o características del proyecto. No deben ser modificados por el programador.

### 3.2 Geolocalización

La geolocalización, también llamada georreferenciación, consiste en relacionar cierta información con una posición geográfica [10]. Esta información puede hacer referencia a cualquier objeto: un restaurante, un teatro, una universidad... e incluso a personas. La posición de cada objeto se conoce según un sistema de coordenadas que permite su ubicación en el mapa. Actualmente, existen herramientas muy desarrolladas y que resultan de gran utilidad, como Google Maps [11], un servidor de aplicaciones de mapas en la web a través del cual es posible localizar cualquier tipo de negocio en un área específica para cualquier región del mundo, determinar la ubicación en la que se encuentra un usuario en cierto momento, así como calcular rutas entre coordenadas (véase ilustración 11).



Ilustración 11. Google Maps para un dispositivo móvil

Con los cada día más utilizados dispositivos móviles de última generación, que incorporan funciones de geolocalización del propio dispositivo a través de diferentes medios (GPS, redes WiFi, ...), y, por tanto, del usuario que los porta, los sistemas de geolocalización se presentan como una herramienta de gran potencial en el futuro próximo.

### 3.2.1 La API de Google Maps para Android

Como hemos descrito en 2.3, REJA es un sistema de recomendación que incorpora la localización como contexto a la hora de llevar a cabo una recomendación. Una funcionalidad interesante de nuestra aplicación será, por tanto, poder representar tanto la ubicación del usuario como la de los restaurantes que se le recomiendan en un mapa.

Para ello, haremos uso de la API que Google proporciona para la incorporación de mapas en aplicaciones para Android. Todos los detalles relativos a la incorporación de Google Maps a nuestro proyecto están descritos en el Anexo III.

## 4 Ingeniería del Software

---

## 4.1 Introducción

Una vez que hemos presentado el propósito y los objetivos que se pretenden alcanzar con la realización de este proyecto, e introducido los conceptos teóricos previos acerca de qué son los sistemas de recomendación, sistema operativo en el que se basará la aplicación y definido qué es la geolocalización, vamos a detallar el proceso seguido para el desarrollo y posterior implementación de la aplicación a través de técnicas de ingeniería del software.

La ingeniería del software significa la construcción de software de calidad con un presupuesto limitado y una fecha de entrega en contextos de cambio continuo. Otra definición complementaria define el concepto como el establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales.

El proceso de ingeniería del software comprende varias fases y en nuestro proyecto se ha estructurado de la siguiente forma:

- Análisis:

Consiste en realizar un análisis acerca de las tareas específicas que el software ha de poder realizar, así como el modo de funcionamiento o respuesta de cada una de ellas durante la interacción con el usuario. Este análisis se llevará a cabo a través de:

  - Especificación de requerimientos.
  - Casos de uso.
- Diseño:

Consiste en realizar un diseño tanto estructural, es decir, desde el punto de vista de la implementación y codificación, como del desarrollo de un prototipo visual de la aplicación, es decir, en cuanto a la interfaz y su modo de en que el usuario interactúa con ella. El diseño comprende a su vez otras fases, que son:

  - Diagrama de clases de diseño.
  - Diseño de la interfaz.
- Implementación y pruebas:

Consiste en traducir el modelo que hemos obtenido en las fases de análisis y diseño a una aplicación real ejecutable sobre una máquina y pruebas que se han realizado sobre la misma una vez implementada para asegurar que cumple con objetivos marcados al inicio del proyecto. Todo este proceso queda descrito a través de:

- Arquitectura
- Tecnologías utilizadas.
- Pruebas de caja negra.

## 4.2 Análisis

### 4.2.1 Especificación de requerimientos

En esta fase se procede a describir todo aquello que el software a desarrollar requiere para cumplir con el propósito del proyecto. Por tanto, durante la especificación de requerimientos han de establecerse claramente tanto aquellas propiedades que el sistema ha de tener como las restricciones a las que está sometido.

#### *Requerimientos funcionales*

Los requerimientos funcionales son aquellos que se refieren específicamente al funcionamiento del sistema. Para este proyecto, se han considerado los siguientes:

- **Requerimiento funcional 1: Loguear usuario.**

Consiste en identificar al usuario en el sistema a través de su nombre de usuario y contraseña. Al realizar esta acción los datos personales del usuario quedarán vinculados o almacenados en el dispositivo.

- **Requerimiento funcional 2: Representar en el mapa todos los restaurantes recomendados en función de la localización del dispositivo.**

Consiste en representar a cada uno de los restaurantes que se recomiendan al usuario en un mapa de modo que pueda observarse su situación geográfica.

- **Requerimiento funcional 3: Obtener un listado de todos los restaurantes recomendados en base a la localización del dispositivo.**

Consiste en ofrecer un listado en el que se puedan visualizar todos los restaurantes que se recomiendan al usuario. Para cada uno de ellos, se mostrará:

- Nombre del restaurante.
- Tipo de restaurante.
- Valoración media.

- **Requerimiento funcional 4: Obtener la información detallada acerca de un restaurante.**

Consiste en mostrar información detallada para un restaurante en concreto. Esta información consiste en:

- Valoración actual del restaurante en cuanto a calidad y servicio.
  - Tipo de restaurante.
  - Tipo de comida que ofrece.
  - Rango de precio en el que se encuentran sus platos.
  - Dirección postal.
  - Ciudad o municipio en que está situado.
  - Número de teléfono y posibilidad de realizar una llamada al mismo. [Véase requerimiento funcional 5]
  - Valorar el restaurante. [Véase requerimiento funcional 6]
- **Requerimiento funcional 5: Llamar al restaurante.**  
Consiste en realizar una llamada telefónica a un restaurante en concreto.
  - **Requerimiento funcional 6: Valorar un restaurante.**  
Consiste hacer una valoración para un restaurante en concreto en cuanto a calidad y servicio ofrecidos.
  - **Requerimiento funcional 7: Determinar el área de obtención de recomendaciones en base a la localización y medio de transporte del usuario.**  
Permite inferir el contexto en que se encuentra el usuario para obtener recomendaciones basadas en el mismo a través de:
    - Localización: que se obtiene a través del dispositivo móvil.
    - Medio de transporte: permite al usuario determinar si se desplaza a pie o con un vehículo.
  - **Requerimiento funcional 8: Obtener el valor de los parámetros relativos al contexto en que se encuentra el usuario.**  
Consiste en visualizar los parámetros de localización, es decir, la latitud y longitud en que se encuentra el usuario en un momento dado, así como la velocidad a la que se desplaza.
  - **Requerimiento funcional 9: Obtener los datos personales del usuario.**  
Permite visualizar los datos personales del usuario. Tales son:
    - Nombre de usuario.
    - Contraseña.

- Email.

Algunos de estos datos son editables por el usuario (véase requerimiento funcional 10).

- **Requerimiento funcional 10: Modificar los datos personales del usuario.**

Permite modificar la contraseña y email en cuanto a los datos personales del usuario.

- **Requerimiento funcional 11: Desvincular la cuenta del usuario de la aplicación.**

Consiste en eliminar los datos personales del usuario identificado (véase requerimiento funcional 1) en el sistema, de modo que al acceder de nuevo a la aplicación a través del mismo dispositivo sea necesario volver a identificarse.

### ***Requerimientos no funcionales***

Los requerimientos no funcionales son los referidos no al funcionamiento estricto del sistema, sino a otros factores externos. Estos factores tienen que ver con las restricciones o limitaciones que han de tenerse en cuenta para que la aplicación pueda ejecutarse correctamente, como por ejemplo, el dispositivo electrónico que se utiliza y sus características. Dichos factores son los siguientes:

- El dispositivo ha de tener instalada una versión del sistema operativo Android entre la 2.2 y 4.3.
- El dispositivo ha de tener instalado Google Play Services para poder hacer uso de la API de Google Maps. En caso de no tenerlo, la aplicación facilitará su descarga e instalación.
- El dispositivo ha de tener activadas las opciones que permitan la utilización de redes y satélites GPS para determinar la ubicación del dispositivo.
- Para poder instalar la aplicación en un dispositivo tal y como se indica en el anexo I, éste ha de tener activada la opción *Depuración de USB*.

#### **4.2.2 Casos de uso**

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto, los casos de uso representan las funciones que un sistema puede ejecutar. Un caso de uso estará formado por los siguientes elementos:

- **Nombre:** identifica al caso de uso de forma única.

- **Actores participantes:** entidades externas que se comunican con el sistema. Describen a un tipo de usuario y tienen un nombre que los identifica. En nuestro caso estos son:
  - *Servidor de recomendaciones*: servidor de la Universidad de Jaén que consta de un motor de recomendación y una base de datos, al cual se conecta nuestra aplicación para hacer peticiones de información.
  - *Usuario registrado*: usuario con datos personales y perfil de usuario almacenados en la base de datos del servidor de recomendaciones.
- **Condiciones de entrada:** o precondiciones que han de cumplirse para poder iniciar el caso de uso.
- **Flujo de eventos:** eventos y orden en que se suceden a lo largo de la interacción entre el/los actor/es y el sistema para un caso de uso concreto.

Previo a la descripción de cada uno de los casos de uso vamos a representar una serie de diagramas. En primer lugar representaremos el diagrama frontera, que describe completamente la funcionalidad del sistema y su entorno. El diagrama frontera (véase ilustración 12) está compuesto por cinco casos de uso principales que recogen todo lo expuesto en la especificación de requerimientos. Estos son:

- Identificación [Requerimiento funcional 1].
- Recomendaciones [Requerimientos funcionales 2, 3].
- Detalles de un restaurante [Requerimientos funcionales 4, 5, 6].
- Contexto [Requerimientos funcionales 7, 8].
- Datos de usuario [Requerimientos funcionales 9, 10, 11].

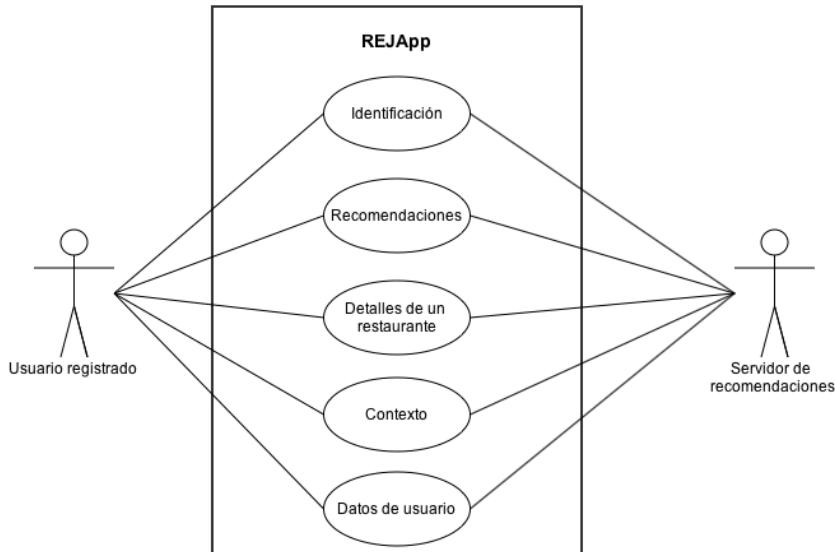


Ilustración 12. Diagrama frontera

Por ser el diagrama frontera una representación general del sistema, para cada aquellos casos de uso vamos a realizar un diagrama que nos permita profundizar y conseguir un mayor nivel de detalle acerca de su comportamiento. Para ello, se pueden emplear dos tipos de relaciones:

- <<extend>>: es una relación cuya dirección va hacia el caso de uso a detallar, y representa comportamientos excepcionales del caso de uso.
- <<include>>: es una relación cuya dirección es contraria a la relación <<extend>> y representa un comportamiento común del caso de uso.

A continuación vamos a exponer cada uno de los diagramas de caso de uso considerados para los casos de uso *recomendaciones* (véase ilustración 13), *detalles de un restaurante* (véase ilustración 14), *contexto* (véase ilustración 15) y *datos de usuario* (véase ilustración 16).

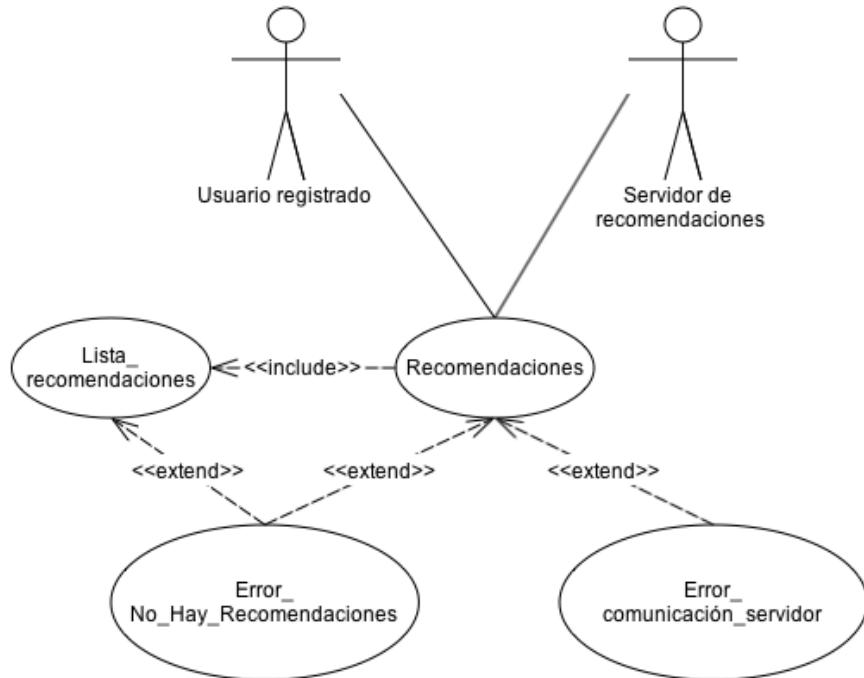


Ilustración 13. Diagrama para el caso de uso "Recomendaciones"

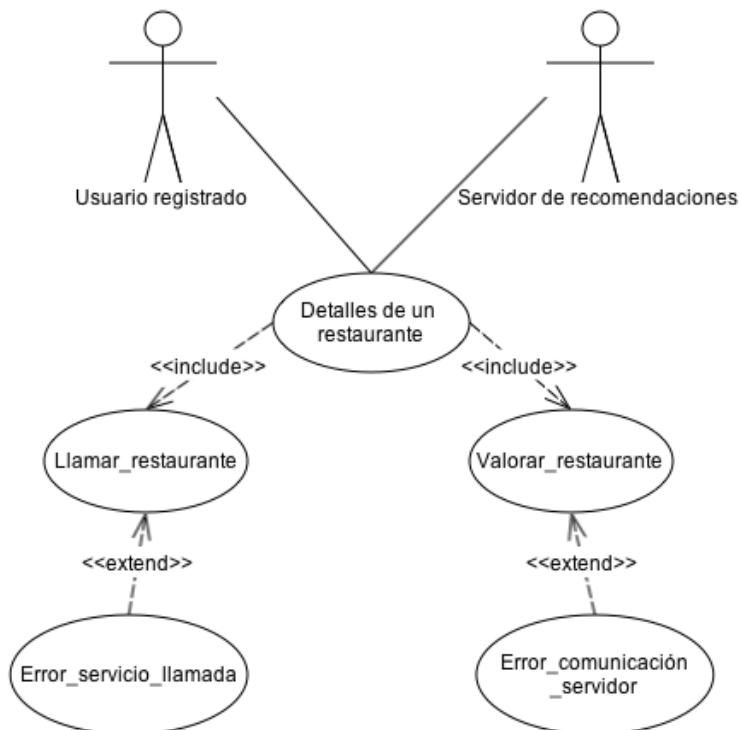


Ilustración 14. Diagrama para el caso de uso "Detalles de un restaurante"

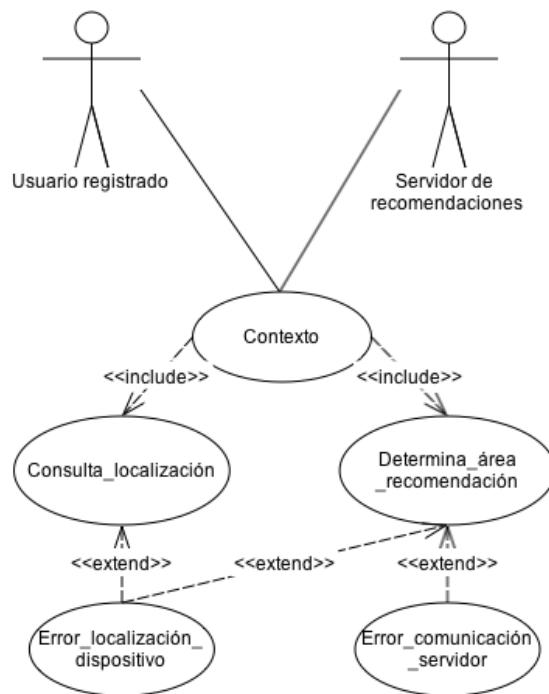


Ilustración 15. Diagrama para el caso de uso "Contexto"

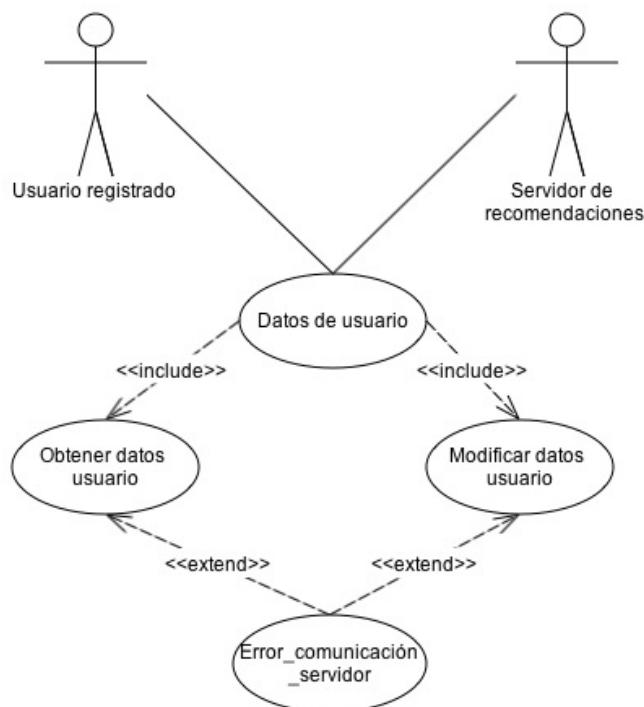


Ilustración 16. Diagrama para el caso de uso "Datos de usuario"

Por último y basándonos en cada uno de los requerimientos funcionales del apartado 4.2.1, vamos a describir verbalmente cada uno de los casos de uso considerados.

### ***Narrativa para el caso de uso “Identificación”***

- **Actor:** usuario registrado.
- **Precondiciones:** la aplicación no está vinculada a ninguna cuenta.
- **Objetivos:** iniciar sesión en la aplicación.
- **Camino principal:**
  1. El usuario introduce su identificador y contraseña.
  - 2.a El sistema valida los datos introducidos e introduce al usuario en el sistema. La cuenta del usuario queda vinculada a la aplicación.
- **Camino alternativo:**
  - 2.b El sistema no valida los datos introducidos. Volver al paso 1.

### ***Narrativa para el caso de uso “Recomendaciones”***

- **Actor:** usuario registrado.
- **Precondiciones:** la sesión está iniciada.
- **Objetivos:** obtener el conjunto de todos los restaurantes que se recomiendan al usuario.
- **Camino principal:**
  1. El usuario accede a la aplicación y solicita visualizar las recomendaciones.
  - 2.a El sistema obtiene el conjunto de restaurantes que se recomiendan.
- **Camino alternativo:**
  - 2.b El sistema no puede obtener recomendaciones. Informar al usuario del error. Volver al paso 1.

***Narrativa para el caso de uso “Detalles de un restaurante”***

- **Actor:** usuario registrado.
- **Precondiciones:** la sesión está iniciada y se han obtenido recomendaciones.
- **Objetivos:** obtener los detalles de un restaurante en concreto.
- **Camino principal:**
  1. El usuario solicita visualizar la información detallada de un restaurante.
  - 2.a El sistema muestra todos los detalles almacenados en la base de datos para ese restaurante.
  3. El usuario solicita realizar una llamada al restaurante.
  - 4.a El sistema inicia una llamada al restaurante.
  5. El usuario valora al restaurante.
  - 6.a El sistema envía la valoración al servidor.
- **Camino alternativo:**
  - 2.b El sistema no puede acceder a los detalles del restaurante. Informar del error y volver al paso 1.
  - 4.b El sistema no puede realizar la llamada. Informar del error y volver al paso 2.a .
  - 6.b El sistema no puede enviar la valoración. Informar del error y volver al paso 2.a .

### ***Narrativa para el caso de uso “Contexto”***

- **Actor:** usuario registrado.
- **Precondiciones:** la sesión está iniciada.
- **Objetivos:** obtener los detalles relativos a la localización del usuario así como activar las opciones que le permiten inferir contexto a la hora de obtener recomendaciones.
- **Camino principal:**
  1. El usuario solicita visualizar las opciones y datos relativos a su localización.
  - 2.a El sistema muestra las opciones y datos relativos a la localización.
  3. El usuario determina el área de obtención de recomendaciones.
  - 4.a El sistema incluye los datos de contexto en la petición de recomendaciones al servidor.
- **Camino alternativo:**
  - 2.b El sistema no puede obtener los datos de localización. Volver al paso 1.
  - 4.b El sistema no puede comunicarse con el servidor. Informar del error y volver al paso 1.

### ***Narrativa para el caso de uso “Datos de usuario”***

- **Actor:** usuario registrado.
- **Precondiciones:** la sesión está iniciada.
- **Objetivos:** obtener y modificar los datos personales del usuario.
- **Camino principal:**
  1. El usuario solicita visualizar sus datos personales.
  - 2.a El sistema obtiene los datos personales del usuario.
  3. El usuario realiza una modificación de sus datos personales.
  - 4.a El sistema envía los datos al servidor.
- **Camino alternativo:**
  - 2.b El sistema no puede obtener los datos del usuario. Informar del error. Volver al paso 1.
  - 4.b El sistema no puede enviar los datos al servidor. Informar del error. Volver al paso 1.

## 4.3 Diseño

Una vez que hemos definido los requisitos que nuestra aplicación ha de cumplir y el comportamiento que ha de tener, podemos continuar con la fase de diseño. El objetivo aquí será el de crear la estructura básica del sistema sobre la que se llevará a cabo posteriormente la implementación. Para ello hemos de realizar dos tareas:

- **Diseño del diagrama de clases:** en este diagrama se incluirán todas aquellas clases que han de implementarse para obtener la aplicación final, junto con sus atributos y métodos, así como las relaciones existentes entre ellas.
- **Diseño de la interfaz:** esta parte del diseño describe cómo es la interfaz con a través de la cual el usuario interactúa con la aplicación: metáforas utilizadas, iconos, estructura de las pantallas, colores utilizados, etc. Es decir, todos aquellos componentes visuales.

### 4.3.1 Diagramas de clases de diseño

Según se explicó en el apartado 3.1.5, a la hora de desarrollar una aplicación, Android hace una separación entre lógica, datos e interfaz de usuario. Se puede decir que el patrón que sigue es un Modelo – Vista – Controlador o MVC [25]. El MVC es un patrón o modelo de abstracción de desarrollo del software que separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos.

A continuación vamos a mostrar un diagrama UML de todas las clases que se han implementado como solución junto con sus atributos y los métodos implementados, así como las librerías que se han utilizado (véase ilustración 17).

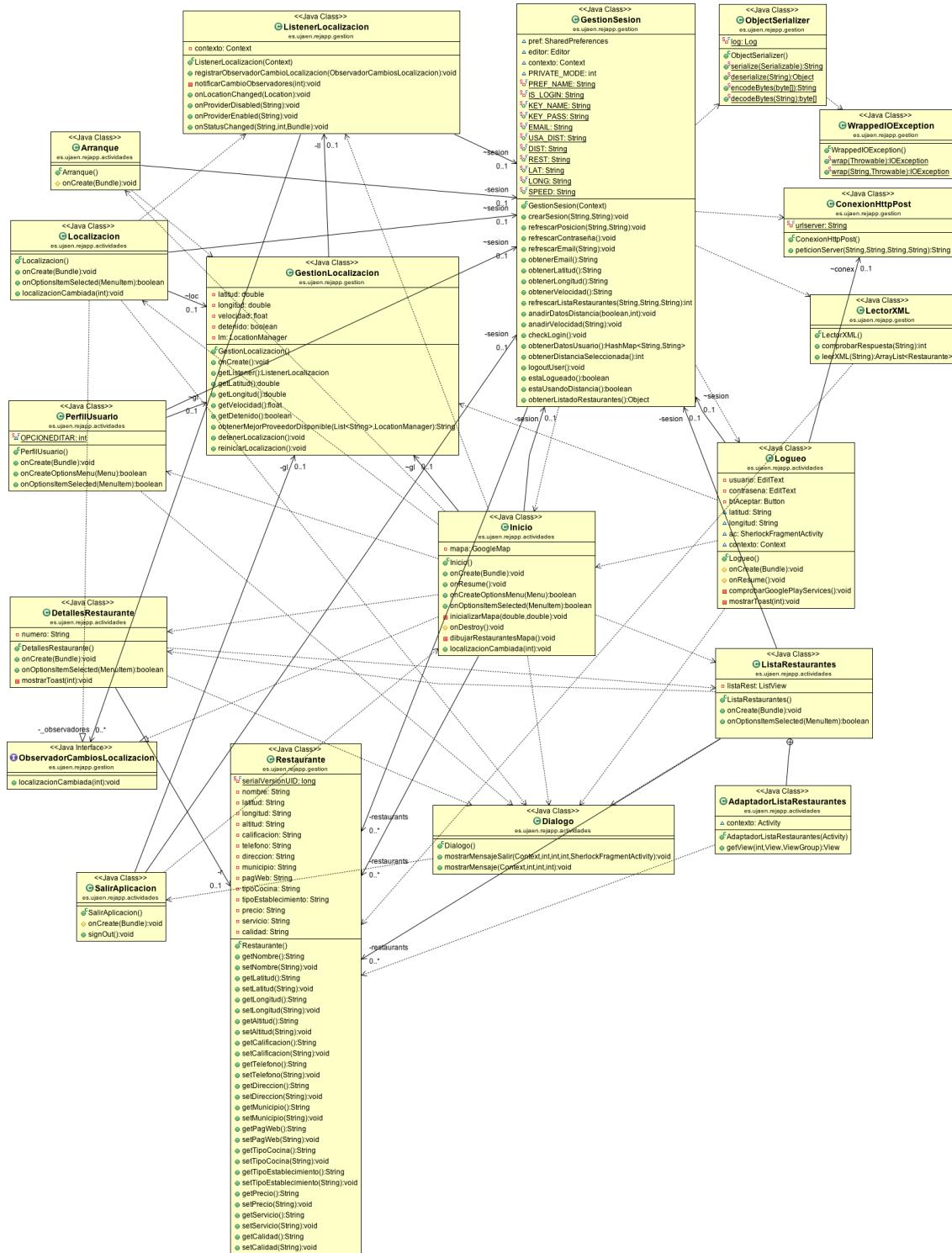


Ilustración 17. Diagrama de clases

Paquete *es.ujaen.rejapp.actividades*:

- **Arranque**: es la clase llamada al iniciar la aplicación. En función de si una cuenta de usuario está vinculada al dispositivo o no, conducirá al usuario a la pantalla de logueo o a la de inicio.
- **DetallesRestaurante**: muestra los detalles de un restaurante.
- **Dialogo**: muestra los dialogos de error o mensajes informativos.
- **Inicio**: muestra la pantalla de inicio y da acceso a todas las demás de la aplicación.
- **ListaRestaurantes**: muestra el listado de restaurantes recomendados y permite acceder a los detalles de cada uno de ellos.
- **Localizacion**: muestra la localización del dispositivo y permite activar la opción de usar distancia para obtener recomendaciones dentro de un área.
- **Logueo**: da acceso al usuario a la aplicación.
- **PerfilUsuario**: permite modificar la información del perfil de usuario y desvincular la cuenta del dispositivo.
- **SalirAplicacion**: permite eliminar los datos del usuario y detener el servicio de localización cuando el usuario decide desvincular sus datos de la aplicación.

Paquete *es.ujaen.rejapp.gestion*:

- **ConexionHttpPost**: realiza la conexión con el servidor.
- **GestionLocalizacion**: activa la localización del dispositivo, realizando una nueva petición cada 30 segundos.
- **GestionSesion**: gestiona todos los datos de la aplicación: cuenta de usuario, lista de restaurantes recomendados y opciones de configuración marcadas por el usuario.
- **LectorXML**: realiza el procesamiento del fichero .xml que se obtiene como respuesta del servidor.
- **ListenerLocalizacion**: listener encargado de refrescar la lista de restaurantes cada vez que la localización del dispositivo cambia.
- **ObjectSerializer**: clase para el almacenamiento de tipos de datos no simples en la clase GestionSesion. Tiene licencia Apache Software 2.0.

- **ObservadorCambiosLocalizacion:** interfaz que implementan las clases Localizacion e Inicio y que les permite refrescar la vista cada vez que la localización del dispositivo cambia y, por tanto, la lista de restaurantes recomendados al realizar una nueva petición al servidor.
- **Restaurante:** clase que representa a un restaurante.

Librerías utilizadas

- **Google Play Services Lib:** permite incluir los mapas de Google Maps en nuestra aplicación.
- **ActionBarSherlock:** permite utilizar la barra de acciones en todas las versiones de Android, ya que esta fué incluida a partir de Android 3.0 y nuestra aplicación, por los motivos expuestos en el apartado 3.1, tiene como versión mínima la 2.2.
- **Acra:** permite el envío de informes vía email cuando ocurre un fallo en la aplicación.

#### 4.3.2 Diseño de la interfaz

### **Metáforas**

La metáfora consiste en describir una idea, concepto u objeto a través de su semejanza con otra cosa. En el diseño de interfaces gráficas resultan de gran utilidad, pues son elementos que describen visualmente las tareas que la interfaz permite desarrollar. En el caso de las aplicaciones diseñadas para dispositivos móviles cobran aún mayor importancia, pues permiten representar toda la funcionalidad que la aplicación ofrece en pantallas de reducido tamaño.

La mayoría de las metáforas utilizadas en este proyecto pertenecen al conjunto de iconos predefinidos para el diseño de aplicaciones para Android definidos en la guía de estilo.

### **Restaurantes**

Para hacer referencia a los restaurantes, utilizaremos la metáfora Restaurantes (Véase Ilustración 18), con un cuchillo y un tenedor, por ser bastante representativa.



**Ilustración 18. Icono para la metáfora "Restaurantes"**

### Desplegable

Es el ícono utilizado por todas las aplicaciones para Android cuando no hay más espacio disponible en la barra de acciones, indicando al usuario que existen más tareas que se pueden realizar dentro de la aplicación. Al pulsarlo, se mostrará un desplegable en el que se pueden seleccionar dichas acciones (Véase Ilustración 19).

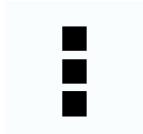


Ilustración 19. Ícono para la metáfora "Desplegable"

### Localización

Usaremos el ícono utilizado por las aplicaciones Android para indicar la localización (Véase Ilustración 20), ya que todos los usuarios de los dispositivos basados en este sistema están familiarizados con el mismo.



Ilustración 20. Ícono para la metáfora "Localización"

### Perfil de usuario

Es el ícono utilizado por las aplicaciones Android que indica que se puede acceder a los datos de un usuario (Véase Ilustración 21).



Ilustración 21. Ícono para la metáfora "Perfil de usuario"

### Ir atrás

Es el ícono definido para el desarrollo de aplicaciones para Android que permite la navegación hacia atrás, es decir, pasar desde una pantalla a su predecesora en la secuencia de pantallas que sigue la aplicación (Véase Ilustración 22).



Ilustración 22. Ícono para la metáfora "Ir atrás"

### Estrellas de valoración

Para permitir al usuario valorar un restaurante, utilizaremos la metáfora Estrellas de valoración (Véase Ilustración 23), ya que su uso está estandarizado y es conocida por la mayoría de los usuarios. Para valorar un restaurante, basta con seleccionar un número de estrellas, que cambiarán su color a amarillo como indicación.



Ilustración 23. Ícono para la metáfora "Estrellas de valoración"

### Editar

Es el ícono utilizado por las aplicaciones Android que indica edición (Véase Ilustración 24).



Ilustración 24. Ícono para la metáfora "Editar"

### Caminar, Coche

Utilizaremos el ícono Caminar junto con el ícono Coche (Véase Ilustración 25) en las opciones de localización, indicando al usuario que la elección de la distancia máxima a la que se deben situar los restaurantes debería estar basada en función del medio en que nos movemos.



Ilustración 25. Ícono para las metáforas "Caminar" y "Coche" respectivamente

### **Tick**

Es el ícono utilizado por las aplicaciones Android que indica *Aceptar* (Véase Ilustración 26) al realizar una operación. En esta aplicación lo utilizaremos para la pantalla edición del perfil de usuario, así como en los diálogos para indicar que una acción se ha realizado correctamente.



Ilustración 26. Ícono para la metáfora "Tick"

### **Atención**

Utilizaremos el ícono Atención (Véase Ilustración 27) para indicar la ocurrencia de un problema en la aplicación, así como de la elección de una acción irreversible.



Ilustración 27. Ícono para la metáfora "Atención"

## **Pantallas**

A continuación vamos a mostrar el diseño de las pantallas que compondrán nuestra interfaz. Una pantalla es un boceto que determina a través de qué tipo de componentes se realizará la interacción, así como su organización y ubicación dentro de la misma.

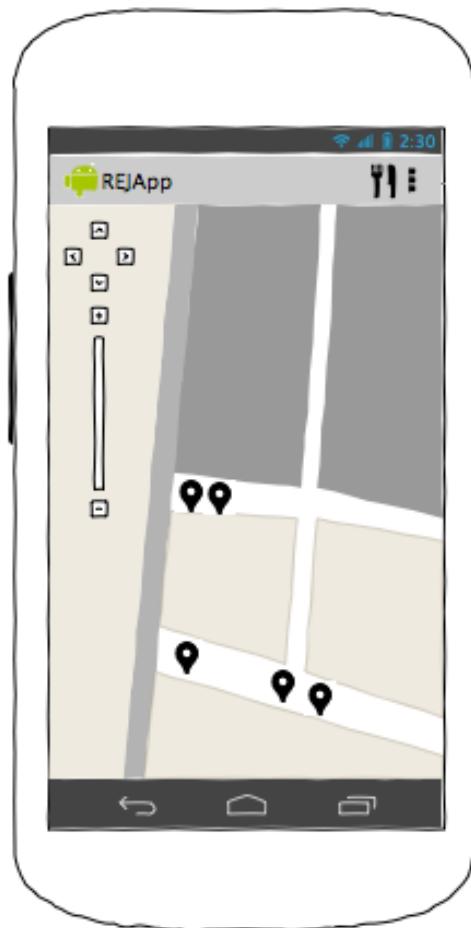
### ***Identificación***



**Ilustración 28. Pantalla "Identificación"**

En la pantalla *Identificación* (Véase ilustración 28), el usuario debe de introducir su nombre de usuario y contraseña, para poder conectarse al sistema de recomendación y acceder al resto de la funcionalidad que ofrece la aplicación.

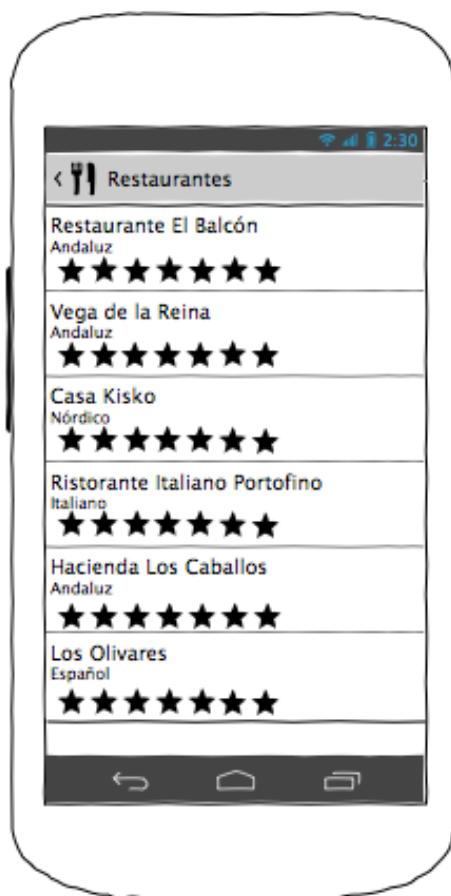
### **Inicio**



**Ilustración 29. Pantalla "Inicio"**

La pantalla de *Inicio* (Véase ilustración 29) será la pantalla principal de la aplicación. En la parte superior se sitúa la barra de acciones, en la que se muestran el icono y nombre de la aplicación a la izquierda, y los iconos que permiten acceder a toda la funcionalidad a la derecha.

El resto de la pantalla muestra el mapa en el que se situarán los puntos correspondientes a los restaurantes.

***Listado de restaurantes*****Ilustración 30. Pantalla "Listado de restaurantes"**

Esta pantalla (Véase ilustración 30) nos muestra una lista de todos los restaurantes recomendados de acuerdo a nuestra localización. En la barra de acciones situada en la parte superior, encontramos el ícono *Ir atrás*, así como el ícono *Restaurantes*, que indica la pantalla en la que nos encontramos actualmente. El listado muestra, para cada restaurante, su nombre y valoración media de los usuarios a través de las *Estrellas de valoración*.

Todo el listado de restaurantes se podrá visualizar haciendo scrolling tanto hacia abajo como hacia arriba, en el caso en que sea más largo que el alto de la pantalla del dispositivo.

**Detalles de un restaurante**



Ilustración 31. Pantalla "Detalles de un restaurante"

En la pantalla *Detalles de un restaurante* (Véase ilustración 31) mostraremos toda la información almacenada en la base de datos acerca de un restaurante. En la barra de acciones, encontramos el ícono *Ir atrás*, junto con el ícono *Restaurantes* y el nombre del restaurante del cual se muestran sus detalles.

El resto de la pantalla se puede considerar dividida en tres zonas: la primera comprende la información más relevante relativa al restaurante en sí, tal como la valoración de los usuarios en cuanto a calidad y servicio, tipo de restaurante y comida que ofrece, y precio medio de sus menús. En la segunda zona, mostramos la información de contacto. Por último, el usuario podrá puntuar al restaurante seleccionando a través de la metáfora *Estrellas de valoración*.

### Localización

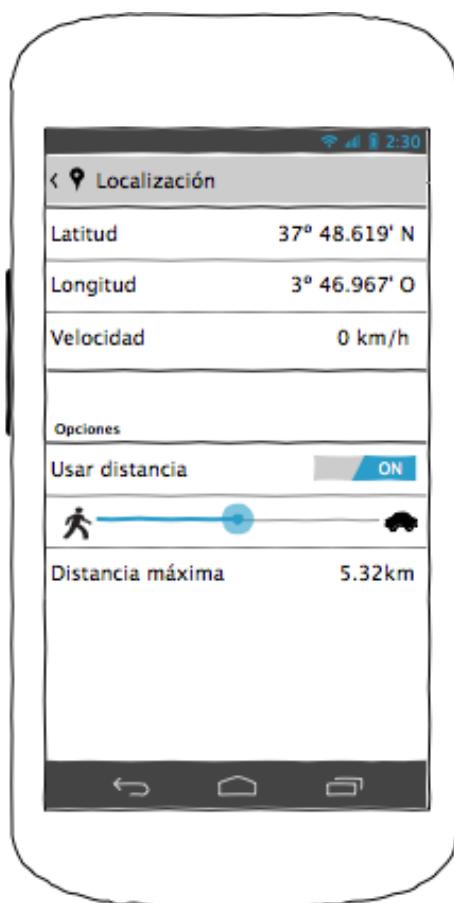


Ilustración 32. Pantalla "Localización"

En la pantalla *Localización* (Véase ilustración 32) encontramos toda la información relativa a la situación del dispositivo, así como las opciones que permiten seleccionar la distancia máxima a la que están localizados los restaurantes recomendados dentro de un radio desde el punto en que se encuentra el usuario.

En la barra de acciones, encontramos el ícono *Ir atrás* seguido del ícono *Localización*. Después, latitud, longitud y velocidad, seguidas de las opciones de configuración descritas en el párrafo anterior.

### Perfil de usuario

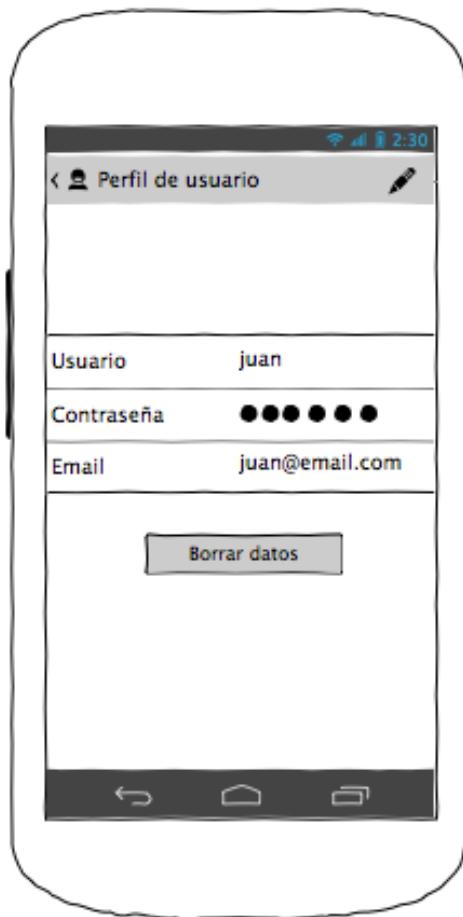


Ilustración 33. Pantalla "Perfil de usuario"

En la pantalla *Perfil de usuario* (Véase ilustración 33) se muestran todos los datos del usuario<sup>1</sup>. En la barra de acciones, a la izquierda encontramos el ícono *Ir atrás* junto con el ícono *Usuario*. A la derecha, el ícono *Editar* que permite acceder a la edición de los datos del usuario. El resto de la pantalla muestra nombre de usuario, contraseña y email. Por último, se muestra el botón *Borrar datos*, que permite al usuario desvincular su cuenta de la aplicación en el dispositivo.

<sup>1</sup> Nótese que Perfil de usuario no hace referencia al perfil de un usuario dentro en el contexto de los sistemas de recomendación, sino a los datos personales del usuario. Se ha utilizado esta notación por convención debido a que los usuarios están familiarizados con estos términos a la hora de hacer referencia a sus datos personales, pues son ampliamente utilizados en redes sociales y páginas web con registro para el acceso a ciertas funcionalidades.

## Storyboards

Un StoryBoard es un conjunto de pantallas mostradas en secuencia con el objetivo de servir de guía para entender una historia, previsualizar una animación o seguir la estructura de una película antes de realizarse o filmarse [23]. En el proceso de diseño de una interfaz se usan secuencias de pantallas, que permiten analizar una posible interacción concreta. En una interfaz la secuencia de acciones a realizar no será lineal únicamente, sino que podrá suceder un árbol de ejecución con varias ramas posibles.

### Identificación

Para poder acceder a la aplicación, el usuario debe loguearse. Para ello, ha de seguir la siguiente secuencia de pasos (Véase ilustración 34):

- Al abrir la aplicación, aparecerá la pantalla *Identificación*, en la que el usuario debe de introducir su nombre de usuario y contraseña.
- Una vez situado sobre uno de dichos campos, aparecerá un teclado virtual para rellenarlos. Después, el usuario pulsará el botón *Aceptar*.
- La aplicación se comunicará entonces con el servidor para autenticar los datos introducidos.
- La aplicación accede entonces a la pantalla de *Inicio*.



Ilustración 34. StoryBoard para “Identificación”

### ***Obtener la lista de restaurantes***

La secuencia de pasos a seguir para visualizar el listado de restaurantes es la siguiente (Véase ilustración 35):

- Pulsar sobre el ícono *Restaurantes* situado en la parte derecha de la barra de acciones.
- Se accederá a la pantalla de listado de restaurantes. El usuario podrá navegar por el listado haciendo scroll hacia arriba y hacia abajo.
- Para volver a la pantalla de inicio, bastará con pulsar sobre el ícono *Ir atrás* situado a la izquierda de la barra de acciones.

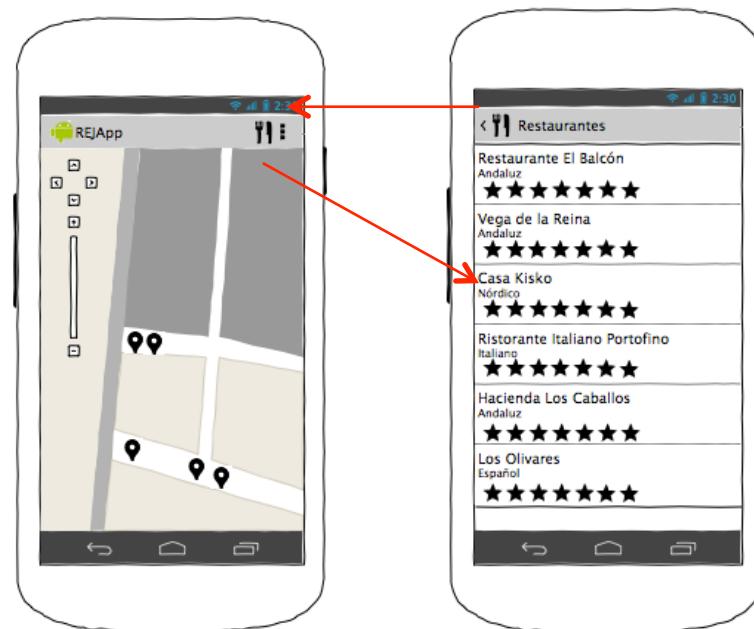


Ilustración 35. StoryBoard para “Obtener la lista de restaurantes”

### Obtener detalles de un restaurante

Para obtener los detalles de un restaurante (Véase ilustración 36) se pueden seguir dos caminos:

- Acceder al listado de restaurantes de la forma en que se indica en el apartado anterior (Véase ilustración 35) y, una vez que la podemos visualizar, pulsar sobre cualquiera de los restaurantes que aparecen en el listado.
- En la pantalla de *Inicio*, pulsar sobre el ícono de *Localización* que indica la situación del restaurante dentro del mapa.
- Pulsando sobre el texto *Llamar* o sobre el número de teléfono en el listado se iniciará una llamada al restaurante a través de la pantalla de llamadas del dispositivo. Esta opción estará disponible únicamente para aquellos restaurantes de los que se dispone de dicha información.

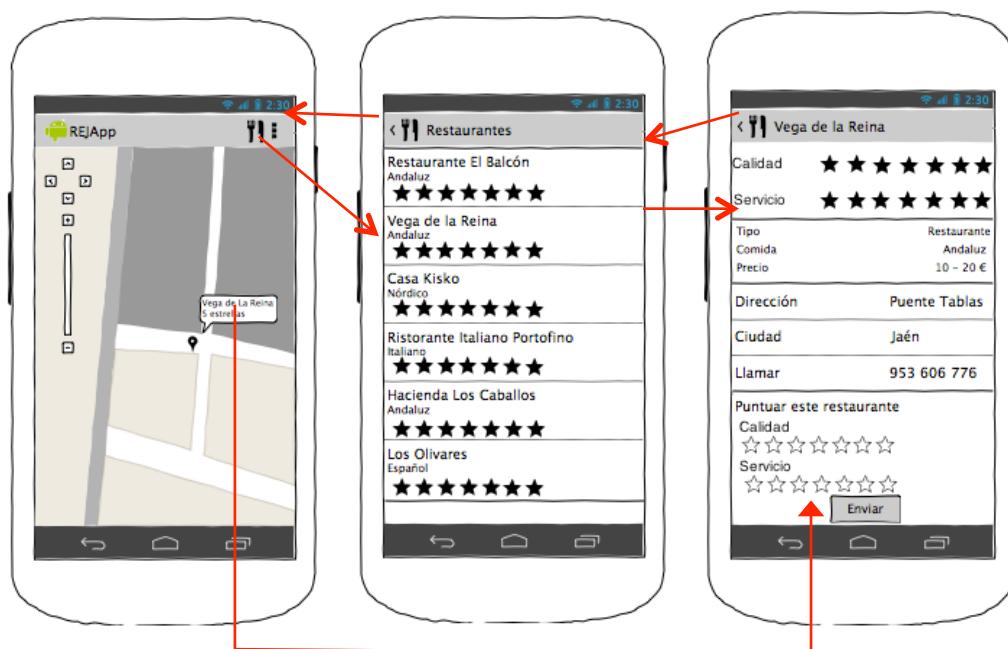


Ilustración 36. StoryBoard para “Obtener detalles de un restaurante”

### Valorar restaurante

Para valorar un restaurante, primero debemos acceder a la pantalla en la que se muestra la información detallada de dicho restaurante, a través de alguno de los dos caminos posibles que se explican en el apartado anterior (Véase ilustración 36). Después (Véase ilustración 37):

- Seleccionar las *Estrellas de valoración* pulsando sobre el ícono.
- Pulsar el botón *Enviar* situado al lado de las estrellas.
- Aparecerá un mensaje de confirmación si la puntuación ha sido almacenada correctamente. Al pulsar el botón *OK* de dicho mensaje, volveremos a visualizar la pantalla de detalles del restaurante.

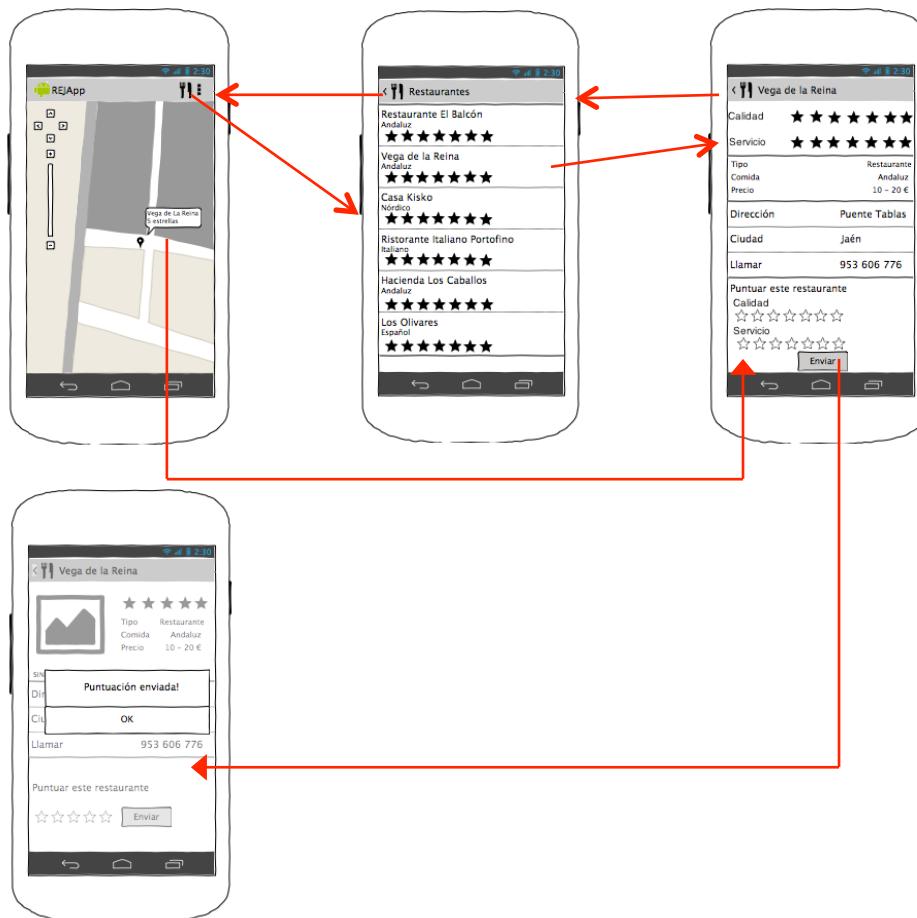


Ilustración 37. StoryBoard para “Valorar restaurante”

### Obtener los detalles de localización

Para obtener los detalles relativos a la localización hemos de hacer lo siguiente (Véase ilustración 38):

- En la pantalla de *Inicio*, pulsar sobre el ícono *Desplegable*.
- Seleccionar la opción *Localización* en el desplegable que aparece.
- Una vez que podemos visualizar la pantalla de *Localización*, podemos volver a la pantalla de Inicio a través del ícono *Ir atrás*.



Ilustración 38. StoryBoard para “Obtener los detalles de localización”

### ***Cambiar la configuración de la localización***

Para modificar las preferencias relativas a la distancia máxima del radio en el que se pueden situar las recomendaciones, haremos lo siguiente (Véase ilustración 39):

- Pulsar sobre el botón *Desplegable* en la barra de acciones de la pantalla de *Inicio*.
- Seleccionar la opción *Localización* en el desplegable que se nos mostrará.
- Activar la opción *Usar distancia*, deslizando el dedo del switch hacia la derecha (*ON*). En versiones inferiores a la 4.0 (API 14) se utilizará un checkbox.
- Una vez activada dicha opción, se activará la barra de desplazamiento que permite seleccionar la distancia máxima deseada. Dicha distancia se mostrará de igual modo en formato numérico justo debajo de la barra.

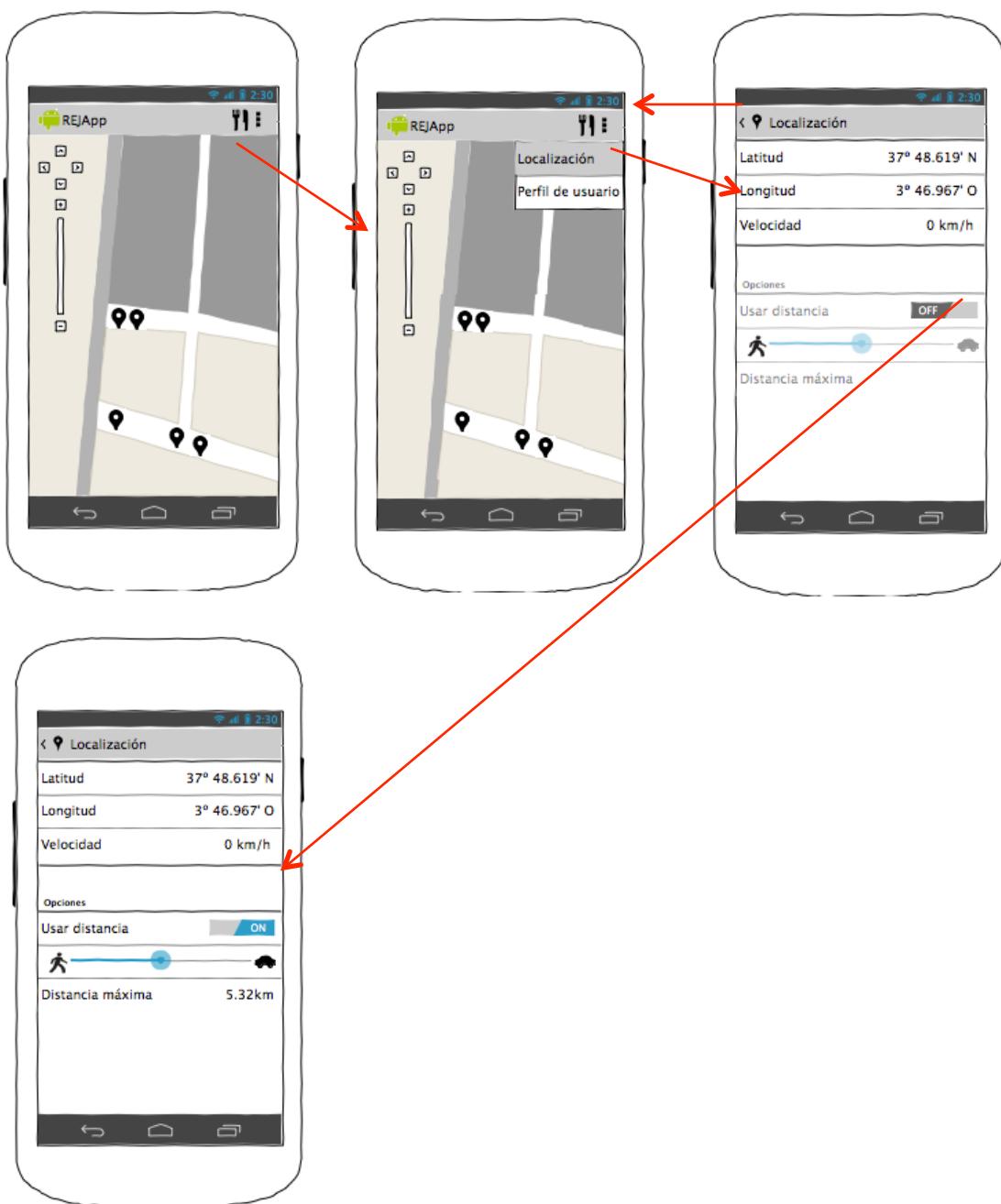


Ilustración 39. StoryBoard para “Cambiar la configuración de la localización”

### Obtener el perfil de usuario

El perfil de usuario se puede obtener a través de la siguiente secuencia (Véase ilustración 40):

- Desde la pantalla de *Inicio*, pulsamos sobre el ícono *Desplegable*.
- Seleccionamos la opción *Perfil de usuario* del desplegable que aparece.
- Accedemos a la pantalla con los datos del perfil de usuario. Para volver a la pantalla de *Inicio*, basta con pulsar sobre el ícono *Ir atrás*.



Ilustración 40. Storyboard para “Obtener el perfil de usuario”

### ***Modificar el perfil de usuario***

Para modificar los datos de la cuenta de un usuario, en primer lugar debemos acceder a su perfil de la forma en que se ha explicado en el apartado anterior (Véase ilustración 40). Después (Véase ilustración 41):

- Pulsaremos sobre el ícono *Editar*.
- Se accede a una nueva pantalla que permite la edición de la contraseña y e-mail del usuario. Basta con pulsar sobre los campos para activar la edición.
- Una vez modificados los datos, podemos guardarlos pulsando en el ícono *Tick* situado en la parte izquierda de la barra de acciones. Los datos entonces quedarán modificados y la aplicación nos devolverá a la pantalla *Perfil de usuario*.

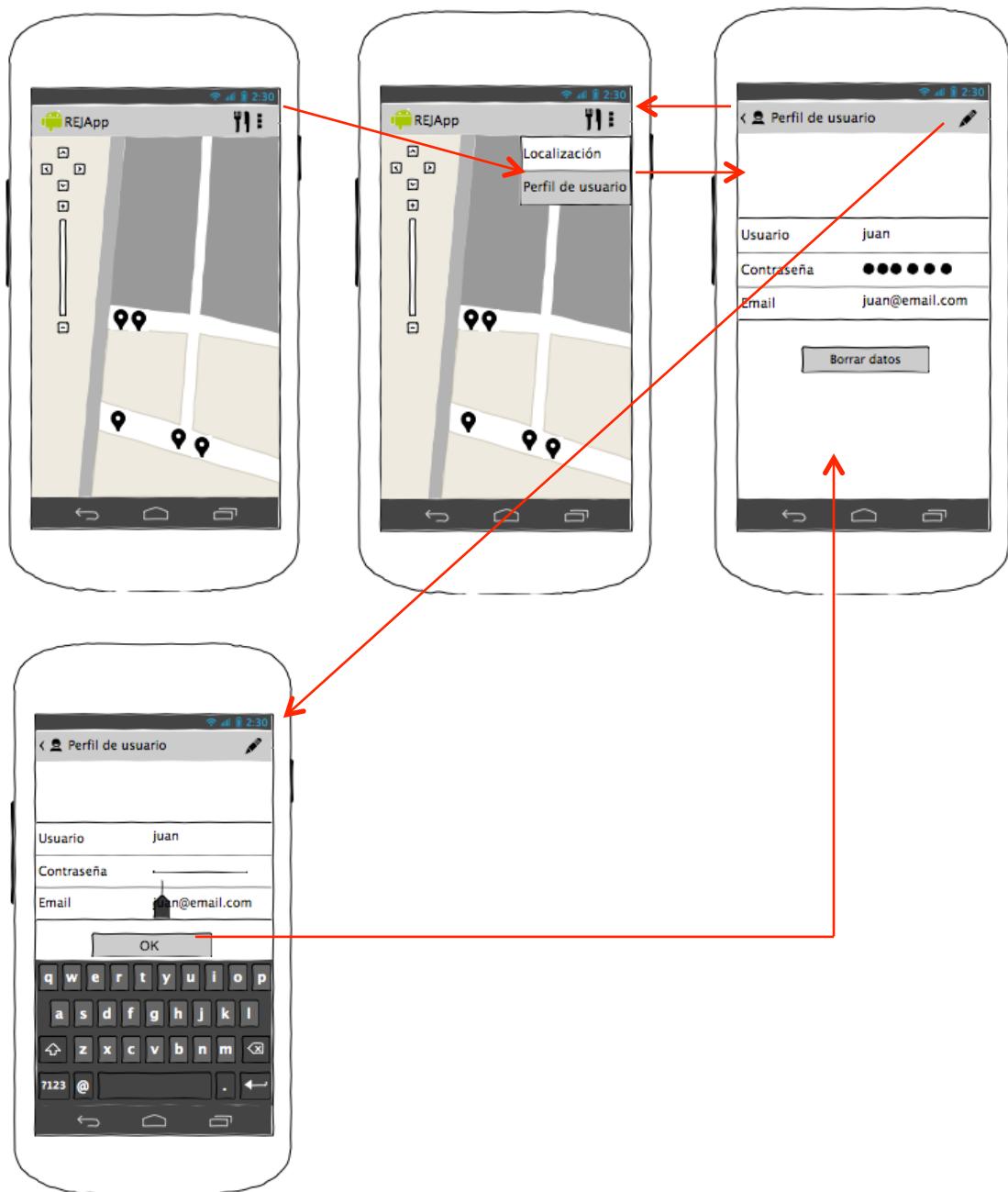


Ilustración 41. Storyboard para “Modificar el perfil de usuario”

***Desvincular cuenta de usuario***

Para desvincular la cuenta de usuario de la aplicación en el dispositivo, debemos acceder al perfil de usuario siguiendo la secuencia de pasos indicada en la sección *Obtener perfil de usuario* (Véase ilustración 40). Después (Véase ilustración 42):

- Pulsar sobre el botón *Borrar datos*.
- Aparecerá un mensaje de confirmación. Si pulsamos *OK*, la cuenta quedará desvinculada y volveremos a la pantalla de *Logueo*. Si pulsamos sobre el botón cancelar, la operación no se llevará a cabo y la aplicación nos devolverá a la pantalla de *Perfil de usuario*.

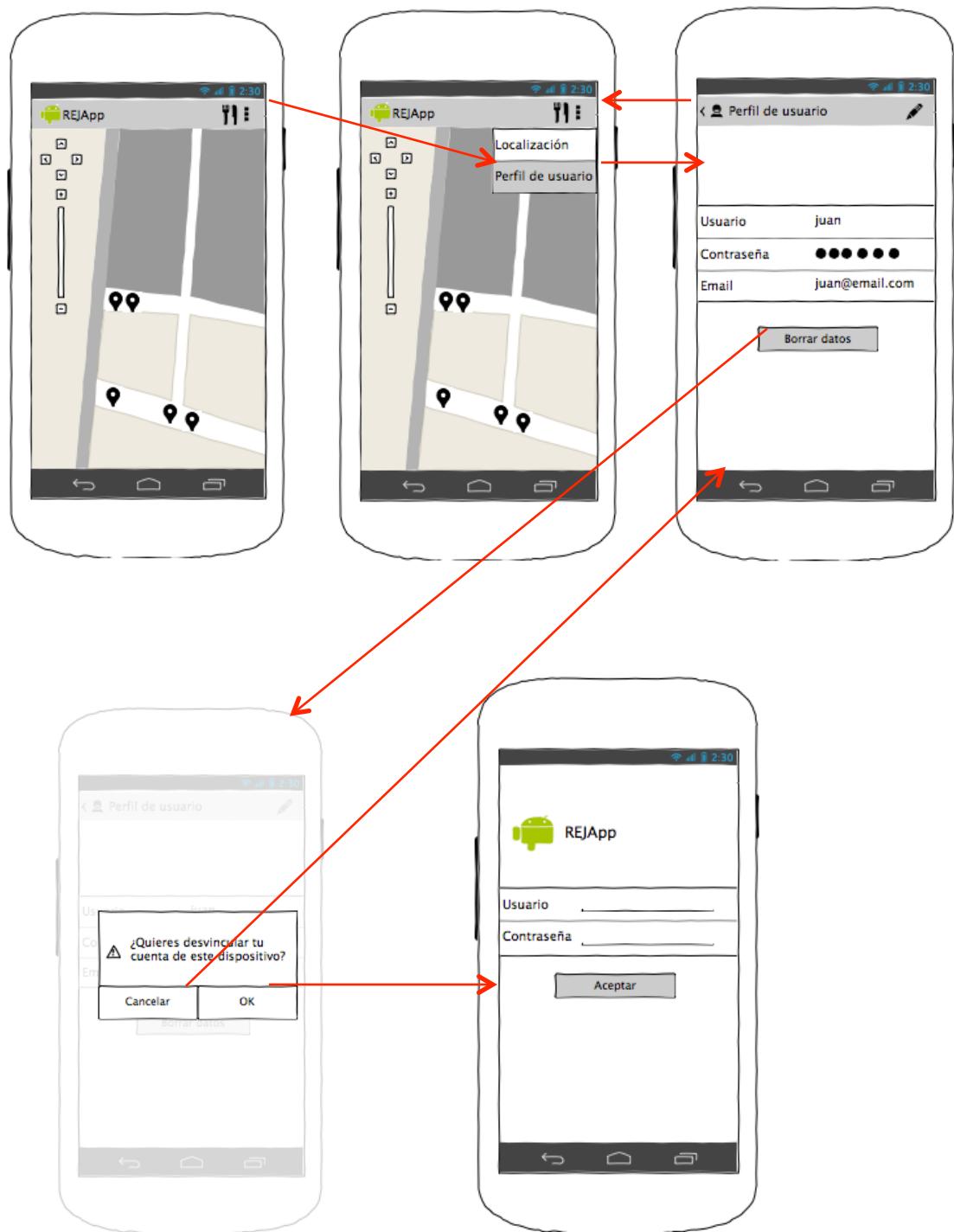


Ilustración 42. StoryBoard para “Desvincular cuenta de usuario”

## Estilo

Todo el estilo de la aplicación se ha realizado siguiendo la guía de estilo que Google propone para diseño de aplicaciones para Android en [24].

### Diseño estructural y colores corporativos

Cada una de las pantallas de la aplicación está compuesta por dos componentes: la barra de acciones, en la que se muestran todas las opciones que se pueden realizar en la aplicación, y la pantalla en sí, que muestra información e implementa funcionalidades propias de su contexto.

Para dar consistencia a la multitud de aplicaciones que se desarrollan para Android, Google proporciona tres temas del sistema que pueden ser utilizados. Para el desarrollo de REJApp se ha utilizado uno de ellos: *Holo Light with Dark action bars* (Véase ilustración 43).

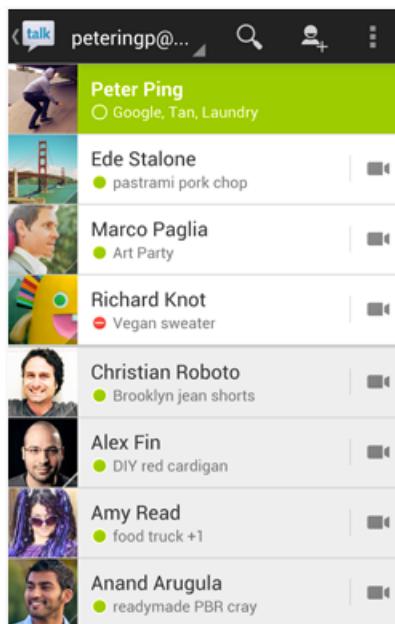


Ilustración 43. Tema Holo Light with dark action bars de Android

### ***Logotipo***

Debido a la cantidad de dispositivos distintos que utilizan el sistema operativo Android, cada uno con características propias tales como resolución de pantalla, tamaño, etc... y para facilitar el diseño de iconos que puedan visualizarse correctamente en cada uno de ellos, Android ha puesto a disposición de los desarrolladores una herramienta llamada *Android Asset Studio*, que permite el diseño y creación de iconos fácilmente integrables en nuestra aplicación.

Para el diseño de nuestro ícono hemos hecho uso de esta herramienta, integrada en el entorno de desarrollo. El ícono (Véase ilustración 44) es sencillo siguiendo las directrices de Android, y muestra elementos relacionados con la restauración.



**Ilustración 44. Ícono de la aplicación**

### ***Tipografía***

El tipo de letra utilizado es *Roboto*, también definida en la guía de estilo de Android. El tamaño utilizado en nuestra aplicación es *Medium*, que está marcado como *18sp*.

## 4.4 Implementación y pruebas

### 4.4.1 Arquitectura

Nuestra aplicación es un cliente dentro de la arquitectura cliente – servidor. Según esta arquitectura, el cliente es la parte encargada de interactuar con el usuario y por tanto, de realizar peticiones al servidor. El servidor es la parte encargada de procesar la información y enviar una respuesta al cliente de acuerdo a la petición que realizó.

En nuestro caso, el cliente solicita recomendaciones al servidor para un usuario en concreto, de acuerdo a su localización y radio de movimiento. El servidor, con los datos que le ha proporcionado el cliente procesa la petición para obtener una lista de recomendaciones. Esta lista será enviada al cliente en un documento de formato .xml, que será procesado por este para mostrar finalmente la información al usuario.

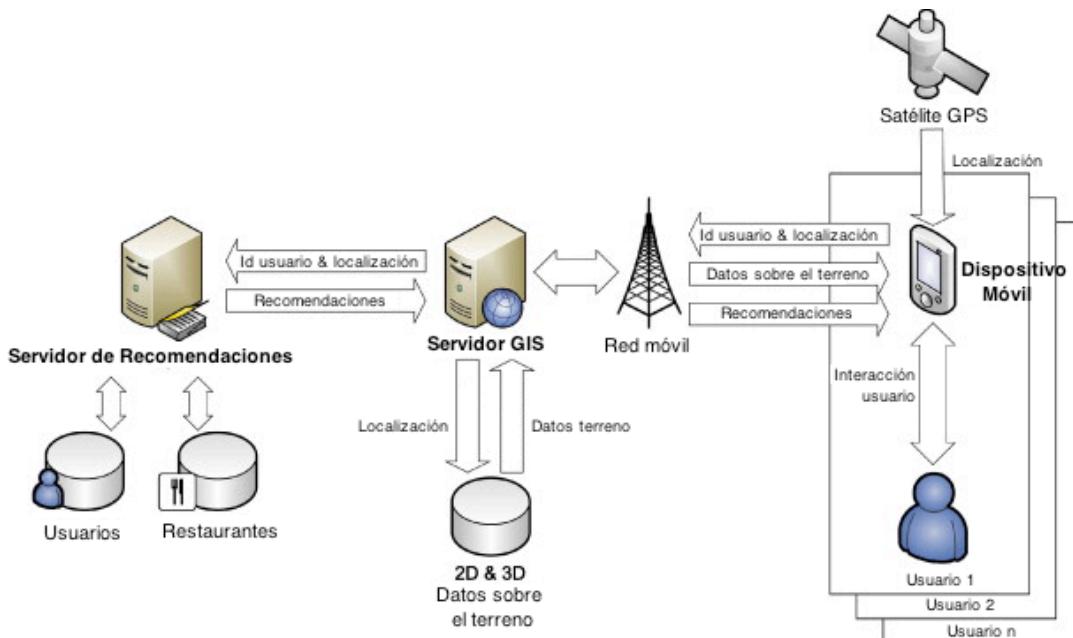


Ilustración 45. Arquitectura cliente - servidor

#### **4.4.2 Tecnologías utilizadas**

Para el desarrollo de este proyecto se han utilizado las siguientes tecnologías:

- **Java:** es un lenguaje orientado a objetos cuyo potencial reside en la compilación a código intermedio o bytecode de las aplicaciones desarrolladas en este lenguaje. Esto permite que una aplicación pueda ejecutarse en múltiples plataformas y de manera independiente al hardware a través de una máquina virtual java.  
Java es el lenguaje utilizado para desarrollar la lógica de las aplicaciones para Android. Como hemos explicado en el apartado 3.1.2, utiliza la máquina virtual Dalvik para la ejecución en sistemas operativos Android.
- **XML:** es un lenguaje de marcas para documentos que permite almacenar información de forma estructurada. Android lo utiliza como estándar para el desarrollo de las interfaces de usuario.

#### **4.4.3 Recursos hardware y software**

- Desde el punto de vista del desarrollador:
  - Software:
    - Sistema Operativo MacOs Leopard 10.6.8
    - Entorno Eclipse ADT Bundle para el desarrollo de aplicaciones para Android
    - Microsoft Office 2011
    - WireframeScketcher Studio 3.0.9
  - Hardware:
    - MacBook Pro 2.4 GHz Intel Core i5
- Desde el punto de vista del usuario final:
  - Dispositivo móvil con el sistema Android versión 2.2 hasta 4.3

#### **4.4.4 Pruebas de caja negra**

El objetivo de esta fase es realizar un conjunto de pruebas que permitan garantizar la calidad del software que hemos desarrollado.

Dichas pruebas están basadas en los requerimientos funcionales que hemos definido en 4.2.1.

## Casos de test

- Test 1: Logueo correcto del usuario. [Requerimiento funcional 1]

Precondiciones	El usuario debe estar registrado en el sistema de recomendación.  El usuario cerró la sesión si la inició con anterioridad.
Acción	El usuario introduce su nombre y contraseña*.
Checkpoint 1	La aplicación valida los datos y muestra al usuario la pantalla de inicio.

- Test 2: Logueo incorrecto del usuario. [Requerimiento funcional 1]

Precondiciones	El usuario debe estar registrado en el sistema de recomendación.  El usuario cerró la sesión si la inició con anterioridad.
Acción	El usuario introduce su nombre y contraseña*.
Checkpoint 1	La aplicación no valida los datos y muestra al usuario un mensaje de error.

- Test 3: Representación en el mapa de los restaurantes recomendados. [Requerimiento funcional 2]

Precondiciones	El usuario debe de estar logueado en el sistema.
Acción	El usuario accede a la pantalla de inicio.
Checkpoint 1	La aplicación muestra una serie de restaurantes en el mapa según su ubicación.

- Test 4: Obtener un listado con todos los restaurantes recomendados. [Requerimiento funcional 3]

Precondiciones	El usuario debe de estar logueado en el sistema.
Acción	El usuario accede a la pantalla de listado de restaurantes.
Checkpoint 1	La aplicación muestra todos los restaurantes recomendados en un listado.

- Test 5: Obtener la información detallada acerca de un restaurante. [Requerimiento funcional 4]

Precondiciones	El usuario debe de estar logueado en el sistema.
Acción	El usuario accede a la pantalla de detalles de un restaurante.
Checkpoint 1	La aplicación muestra los detalles de un restaurante.

- Test 6: Llamar a un restaurante. [Requerimiento funcional 5]

Precondiciones	El usuario debe estar logueado en el sistema.  El número de teléfono del restaurante seleccionado está almacenado en la base de datos.
Acción	El usuario selecciona la opción de llamar de la pantalla detalles de un restaurante.
Checkpoint 1	La aplicación realiza una llamada al número del restaurante.

- Test 7: Valorar un restaurante. [Requerimiento funcional 6]

Precondiciones	El usuario debe de estar logueado en el sistema.
Acción	El usuario selecciona un número de estrellas para valorar la calidad y el servicio de un restaurante.
Checkpoint 1	La aplicación envía los datos al servidor ** y muestra un mensaje de confirmación.

- Test 8: Determinar el área de obtención de recomendaciones en base a la localización y medio de transporte del usuario. [Requerimiento funcional 7]

Precondiciones	El usuario debe de estar logueado en el sistema.
Acción	El usuario activa la opción de usar distancia y selecciona una distancia máxima.
Checkpoint 1	La aplicación realiza una nueva petición de recomendaciones en base a los parámetros seleccionados por el usuario.

- Test 9: Obtener el valor de los parámetros relativos a la localización y velocidad del dispositivo.[Requerimiento funcional 8]

Precondiciones	El usuario debe de estar logueado en el sistema.
Acción	El usuario accede a la pantalla de localización.
Checkpoint 1	La aplicación muestra la latitud, longitud y velocidad del dispositivo.

- Test 10: Obtener el perfil de usuario. [Requerimiento funcional 9]

Precondiciones	El usuario debe de estar logueado en el sistema.
Acción	El usuario accede a la pantalla de perfil de usuario.
Checkpoint 1	La aplicación muestra el nombre de usuario, contraseña y email* del usuario.

- Test 11: Modificar el perfil de usuario.[Requerimiento funcional 10]

Precondiciones	El usuario debe de estar logueado en el sistema.
Acción	El usuario selecciona la opción de editar perfil de usuario.
Checkpoint 1	La aplicación permite editar la contraseña y email del usuario y envía los nuevos datos al servidor **.

- Test 12: Desvincular la cuenta del usuario de la aplicación. [Requerimiento funcional 11]

Precondiciones	El usuario debe de estar logueado en el sistema.
Acción	El usuario selecciona la opción de borrar datos en el perfil de usuario.
Checkpoint 1	La aplicación desvincula los datos del usuario de la aplicación y vuelve a la pantalla de logueo.

\* Esta versión de la aplicación es un prototipo para ser usado con usuarios registrados en el sistema, que carecen de contraseña y email.

\*\* La versión de la aplicación está desarrollada a modo de prototipo, por lo que algunas funcionalidades incluidas en la aplicación no están aún operativas por no estar implementadas en el servidor.

## ***Resultados obtenidos***

Test 1	
Checkpoint 1	Ok
Test 2	
Checkpoint 1	Ok
Test 3	
Checkpoint 1	Ok
Test 4	
Checkpoint 1	Ok
Test 5	
Checkpoint 1	Ok
Test 6	
Checkpoint 1	Ok
Test 7	
Checkpoint 1	Ok
Test 8	
Checkpoint 1	Ok
Test 9	
Checkpoint 1	Ok
Test 10	
Checkpoint 1	Ok
Test 11	
Checkpoint 1	Ok

Test 12	
Checkpoint 1	Ok

**Tabla 2. Resultados obtenidos para los casos de test**

## 5 Conclusiones

---

El sector de turismo está en pleno crecimiento: cada día más gente se anima a viajar e Internet es una herramienta clave a la hora de buscar información acerca de los destinos a visitar. Como hemos expuesto a lo largo de este trabajo, los sistemas de recomendación son aplicables a un amplio abanico de posibilidades, entre las que se encuentra, por supuesto, este sector. El uso de este tipo de sistemas constituye una mejora notable en todos los segmentos de mercado a los que se aplica, ya que transmiten confianza y seguridad a los usuarios, ofreciendo productos y servicios que realmente les interesan entre la multitud de ofertas presentes en la web.

Las características propias de Internet como red de información siempre disponible se han visto potenciadas con la aparición de los dispositivos móviles, que posibilitan el acceso a ella desde cualquier lugar. Esto permite que los usuarios se conviertan no sólo en consumidores, sino también en potenciales productores de información con la aparición de la llamada web 2.0.

Con la realización de este proyecto se intenta aprovechar el potencial conjunto de estas dos tecnologías en auge: los sistemas de recomendación y los dispositivos móviles como herramienta para agregar características del contexto en que se encuentra el usuario en el proceso de recomendación. Como hemos explicado a lo largo de esta memoria, los sistemas de recomendación basados en contexto son una de las últimas tendencias dentro de este campo, pues permiten adaptar las recomendaciones a una o a un conjunto de circunstancias, lo que en determinados casos dota al conjunto de ítems que se recomiendan de una mayor utilidad para el usuario. Del aprovechamiento de estas tecnologías surgirán en un futuro próximo aplicaciones con grandes posibilidades.

REJApp no es más que un prototipo de aplicación que se comunica con un servidor de recomendaciones en tiempo real para ofrecer restaurantes del interés de usuario en función del contexto en que este se encuentra, en este caso, de su localización y medio de transporte.

Para poder tener una versión completamente operativa de este prototipo que pueda exponerse al público a través de una plataforma de descarga de aplicaciones para el sistema operativo Android, sería necesario implementar algunas funcionalidades que se han contemplado en la aplicación pero que no están aún operativas en el lado del servidor.

En primer lugar, deberían de incluirse los campos email y contraseña, para añadir un nivel de seguridad a las cuentas de usuario. Además, habrían de añadirse módulos de envío de información desde el cliente al servidor, que permitan la actualización de datos en el servidor

por parte del usuario, estos son modificación del perfil, valoración de restaurantes y, por supuesto, registro de nuevos usuarios en el sistema.

Este prototipo es también extensible a restaurantes localizados en cualquier lugar del mundo, y podría ofrecerse a empresarios del sector como una opción interesante tanto a la hora de darse a conocer como para saber de primera mano la opinión de los clientes en base a la valoración que estos hacen de su restaurante.

Durante el desarrollo de este proyecto, además de reforzar ciertos conocimientos acerca de los sistemas de recomendación en general, así como profundizar en el funcionamiento, características y aplicación de los sistemas de recomendación basados en contexto, he aprendido a implementar una aplicación para la plataforma Android. Por tratarse en ambos casos de tecnologías actuales, en auge y desarrollo, todo lo aprendido me será de gran utilidad para el desarrollo de trabajos posteriores a lo largo de mi carrera profesional.



# Bibliografía

---

- [1] José M. Noguera, Manuel J. Barranco, Rafael J. Segura, Luis Martínez: “A mobile 3D-GIS hybrid recommender system for tourism”. *Information Sciences*, vol. 215, pp. 37-52, 2012.
- [2] José M. Noguera, Manuel J. Barranco, Rafael J. Segura, Luis Martínez: “A Location-aware Tourism Recommender System based on Mobile Devices”. The 10th International FLINS Conference on Uncertainty Modeling in Knowledge Engineering and Decision Making, Instanbul (Turkey), August 26-29th, 2012.
- [3] Manuel J. Barranco, José M. Noguera , Jorge Castro, Luis Martínez: “A Context-aware Mobile Recommender System based on Location and Trajectory”. *Management Intelligent Systems, Advances Systems and Computing*, vol. 171, pp. 153-162. Springer, 2012
- [4] Francesco Ricci, Lior Rokach, Bracha Shapira: “Recommender Systems Handbook”. Springer, Enero 2011.
- [5] Amazon, <http://www.amazon.es/>
- [6] Francisco Carrero, Jose Carlos Cortizo. <http://loogic.com/sistemas-de-recomendacion/> . Último acceso: Agosto 2013
- [7] E. Peis; J. M. Morales-del-Castillo (Universidad de Granada); J. A. Delgado-López. <http://www.upf.edu/hipertextnet/numero-6/recomendacion.html#Sistemas-contexto> . Último acceso: Agosto 2013
- [8] Artículo Puromarketing.com [http://www.puromarketing.com/76/16807/cinco-claves-basicas-esenciales-futuro-commerce.html#\\_](http://www.puromarketing.com/76/16807/cinco-claves-basicas-esenciales-futuro-commerce.html#_) . Último acceso: Agosto 2013
- [9] Sourcetone, [www.sourcetone.com](http://www.sourcetone.com)
- [10] Linda L. Hill: “Georeferencing: the geographic associations of information.” MIT Press, 2009.
- [11] Google Maps, [https://maps.google.es/\\_](https://maps.google.es/_) . Último acceso: Agosto 2013
- [12] <http://developer.android.com/tools/publishing/app-signing.html> . Último acceso: Agosto 2013
- [13] <https://code.google.com/apis/console/> . Último acceso: Agosto 2013
- [14] Jesús Tomás Gironés: “El gran libro de Android”, Marcombo 2013.
- [15] Wallace Jackson: “Android Apps for absolute beginners”, Apress 2012.

- [16] Open Handset Alliance, <http://www.openhandsetalliance.com/>
- [17] <http://developer.android.com/about/dashboards/index.html> . Último acceso: Agosto 2013
- [18] Sameer Singh: “Global smartphone market trends: Android, Iphone lead, Windows Phone struggles”, <http://www.tech-thoughts.net/2012/07/global-smartphone-market-share-trends.html#.UgjA4uD8E1g>
- [19] <http://source.android.com/source/licenses.html> . Último acceso: Agosto 2013
- [20] <http://developer.android.com/guide/components/fundamentals.html> . Último acceso: Agosto 2013
- [21] <http://developer.android.com/guide/topics/security/permissions.html> . Último acceso: Agosto 2013
- [22] <http://developer.android.com/guide/topics/data/data-storage.html> . Último acceso: Agosto 2013
- [23] Wikipedia, <https://es.wikipedia.org/wiki/Storyboard> . Último acceso: Agosto 2013
- [24] <http://developer.android.com/design/index.html> . Último acceso: Agosto 2013
- [25] <http://www.fdi.ucm.es/profesor/jpavon/poo/2.14.mvc.pdf> . Último acceso: Agosto 2013
- [26] Sinbad2, <http://sinbad2.ujaen.es/>
- [27] REJA, <http://serezade.ujaen.es/~reja/reja1526/>



## Anexo I. Manual de instalación

---

A continuación vamos a describir cada uno de los pasos que hemos de seguir para instalar nuestra aplicación en un dispositivo. Para ello, en primer lugar, hemos de comprobar que la versión del sistema operativo Android del dispositivo se encuentra entre la 2.2 y la 4.3.

En el cd que se adjunta existe una carpeta llamada REJApp que contiene el código fuente de la aplicación y todos los archivos necesarios para su instalación.

Pasos a seguir:

1. Activar la opción *Fuentes desconocidas* dentro de Ajustes > Aplicaciones de nuestro dispositivo. Tras ello nos pedirá la confirmación a través de una ventana y aceptaremos.

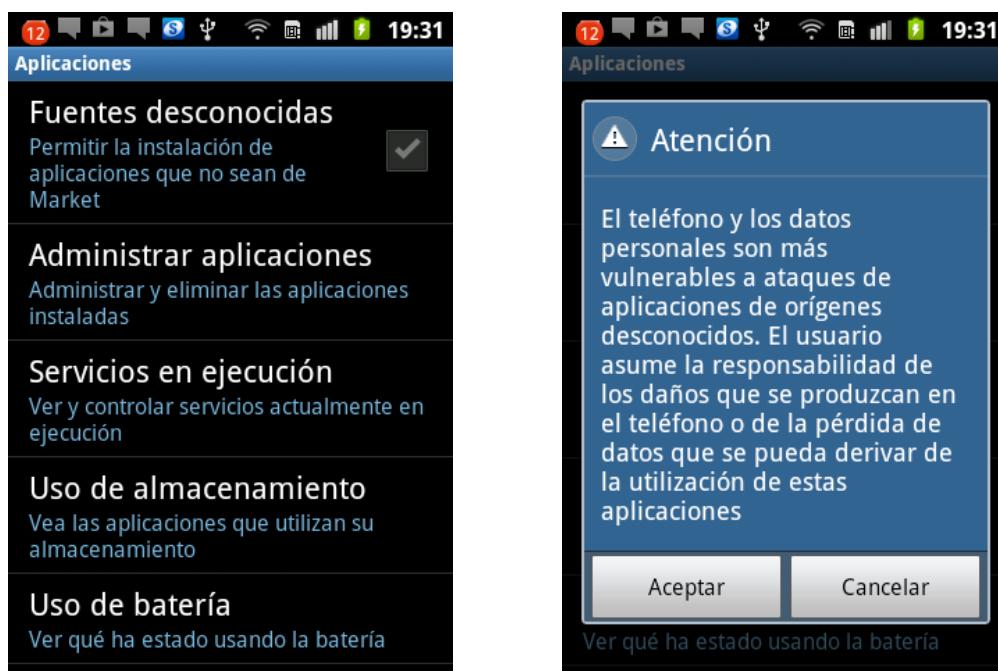


Ilustración 46. Activación de opciones de instalación en el dispositivo

2. Acceder al directorio REJApp/bin y copiar el archivo REJApp.apk. En Android, los archivos con esta extensión son archivos comprimidos que contienen toda la información necesaria para poder ejecutar una aplicación en un dispositivo.

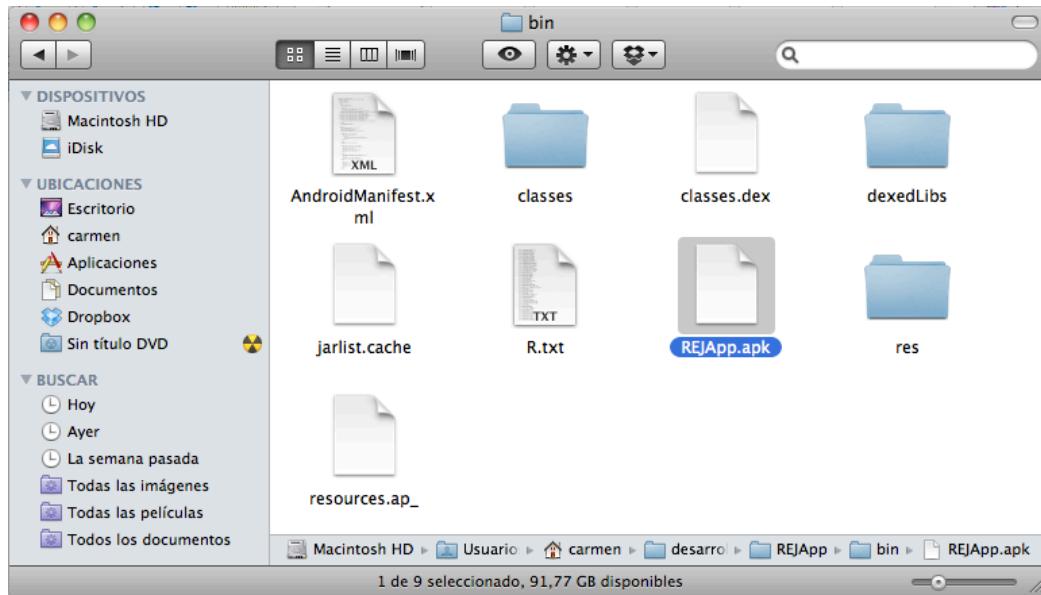


Ilustración 47. Copiar el archivo REJApp.apk desde REJApp/bin

3. Conectar el dispositivo mediante cable USB al ordenador.
4. Acceder a la tarjeta de memoria SD del dispositivo y pegar el archivo REJApp.apk.

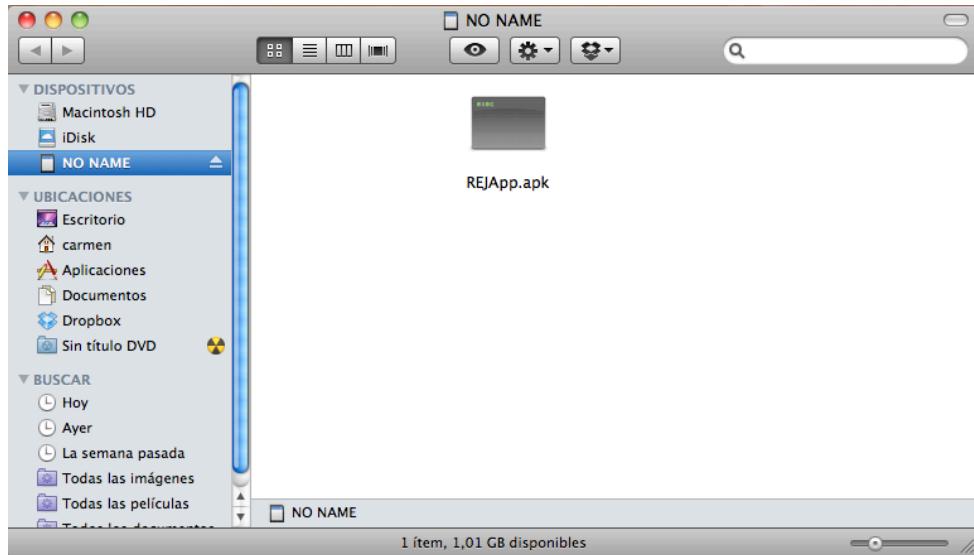


Ilustración 48. Copiar el archivo REJApp.apk en la tarjeta de memoria SD del dispositivo

5. Una vez que el archivo se haya transferido al dispositivo bastará con seleccionarlo para poder instalar la aplicación.

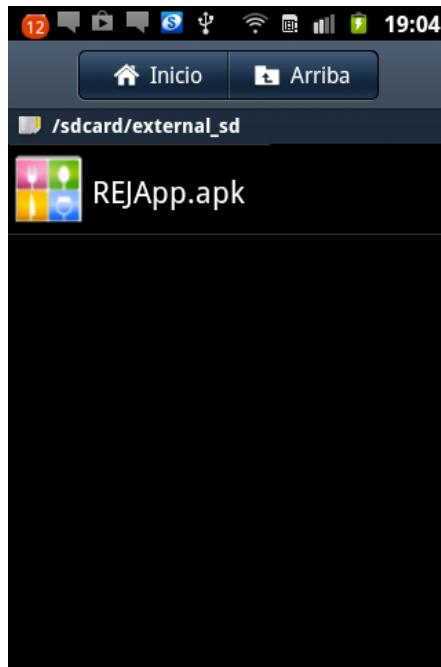


Ilustración 49. Aplicación REJApp en la memoria SD del dispositivo

Tras ello, aparecerá un mensajes de advertencia que hemos de confirmar:

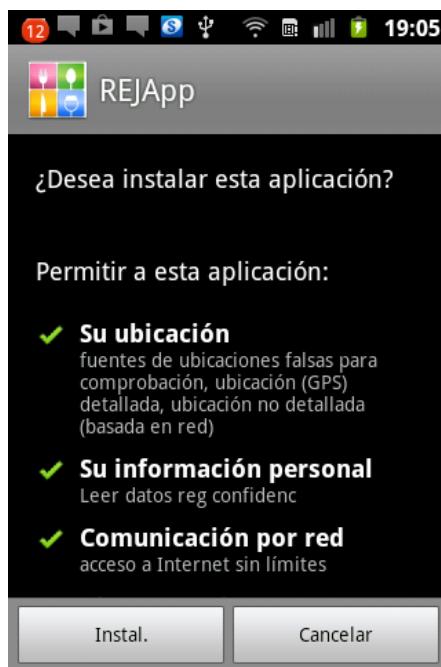


Ilustración 50. Mensajes de confirmación para instalar el archivo REJA.apk

6. Una vez que hemos aceptado se procederá a la instalación y podremos acceder a la aplicación.

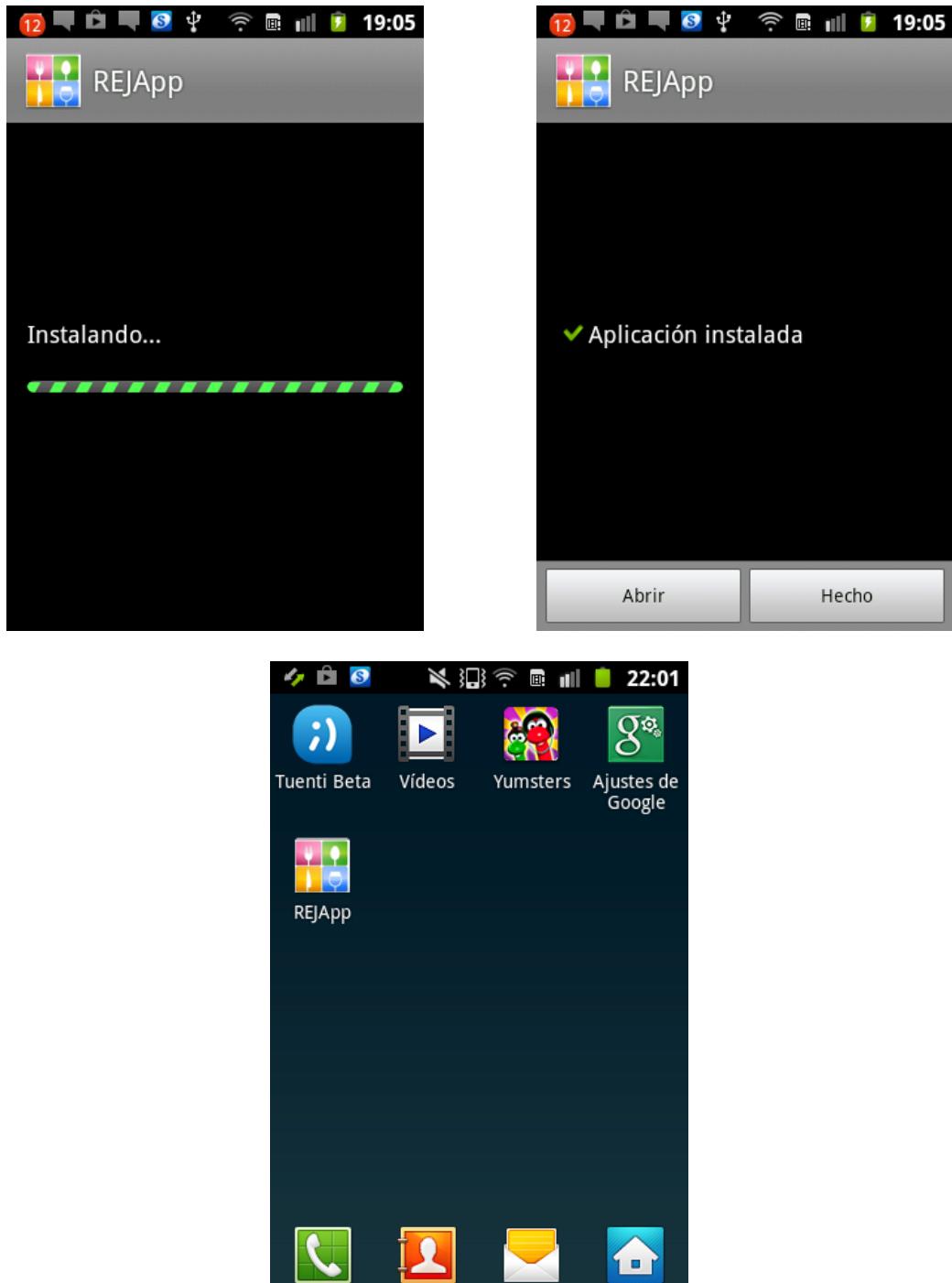


Ilustración 51. Aplicación REJApp instalada en el dispositivo



## Anexo II. Guía de usuario

---

A lo largo de este manual vamos a describir cómo acceder a toda la funcionalidad de la aplicación para permitir realizar un uso correcto de la misma.

Para ello, vamos a seguir un orden de ejecución de acuerdo al orden en que se han colocado las distintas funciones dentro de la pantalla principal, de más usadas a menos usadas.

### **Identificación**

Para acceder a todo lo que REJApp ofrece hemos de disponer de una cuenta de usuario. Una vez la tengamos, bastará con escribir nuestro nombre de usuario correctamente y pulsar el botón *Aceptar* en la pantalla de identificación tal y como se indica en la ilustración 52. La aplicación nos mostrará la pantalla de *Inicio* (véase ilustración 53), que muestra un mapa con una serie de puntos o marcadores: el punto de color azul hace referencia a la localización del usuario, y los marcadores en rojo a cada uno de los restaurantes que el sistema nos recomienda.



**Ilustración 52. Identificación en REJApp**

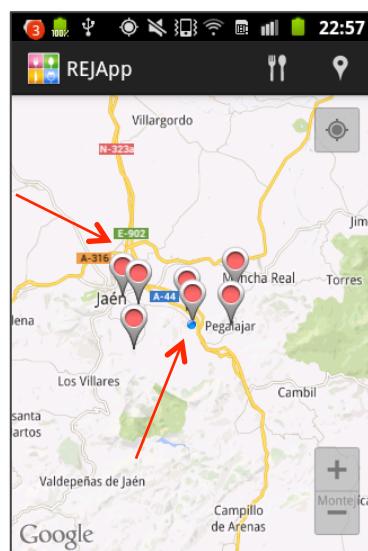


Ilustración 53. Inicio en REJApp

#### *Visualizar la lista de restaurantes recomendados*

Podemos acceder a la lista de restaurantes a través del ícono *Restaurantes* desde la pantalla de *Inicio*, como podemos ver en la ilustración 54.

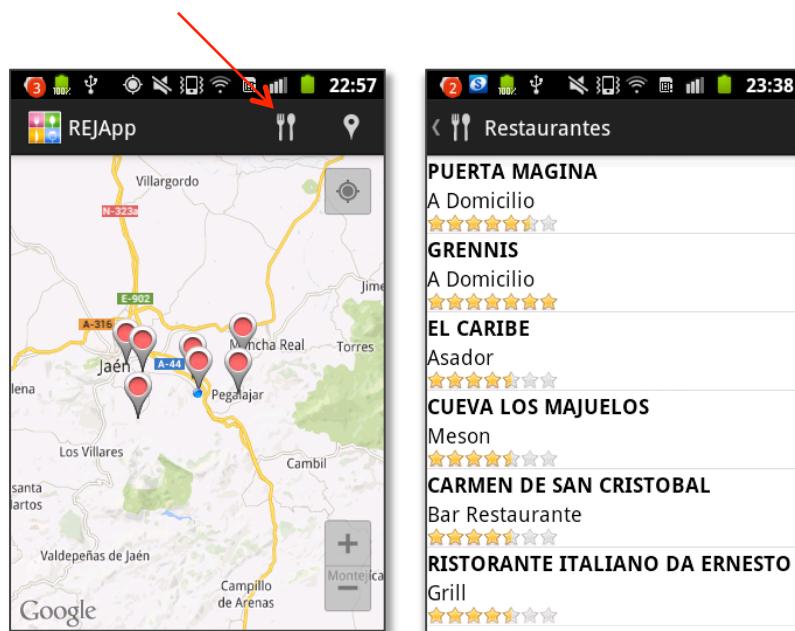


Ilustración 54. Listado de recomendaciones en REJApp

### Visualizar los detalles de un restaurante

En este caso disponemos de dos caminos distintos (véase ilustración 55):

- Desde la lista de restaurantes, pulsando sobre el restaurante elegido.
- Desde la pantalla de *Inicio*, pulsado sobre el marcador del restaurante y de nuevo sobre la descripción que aparece con su nombre.

Podemos visualizar todos los detalles de un restaurante haciendo scroll en la pantalla.

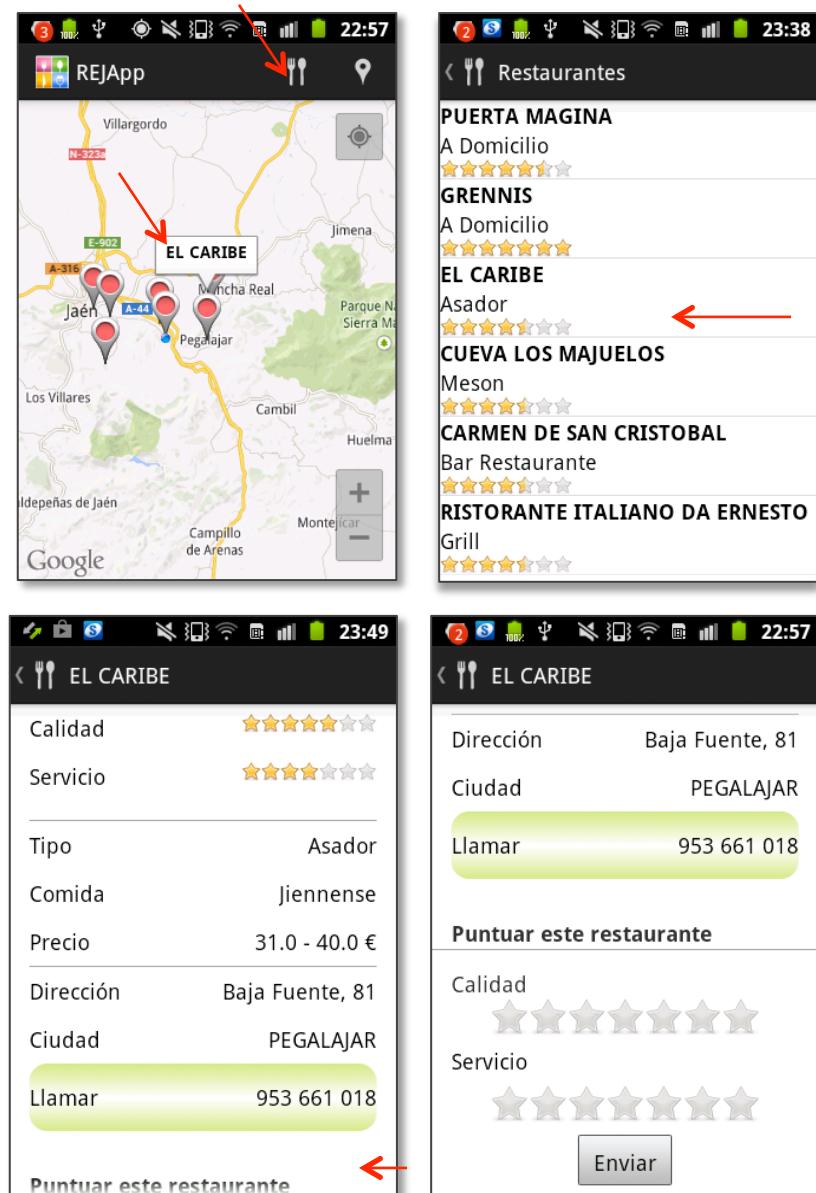


Ilustración 55. Detalles de un restaurante en REJApp

***Llamar a un restaurante***

En la pantalla de detalles para un restaurante aparecerá el número de teléfono del mismo si está almacenado en la base de datos. Para realizar una llamada, bastará con pulsar sobre el número de teléfono, como podemos observar en la ilustración 56.



Ilustración 56. Llamada a un restaurante en REJApp

### Valorar un restaurante

Desde la pantalla de detalles de un restaurante, seleccionaremos las estrellas convenientes para calidad y servicio, y pulsaremos el botón *Enviar*. La aplicación mostrará un mensaje de confirmación, tal y como muestra la ilustración 57.



Ilustración 57. Valorar un restaurante en REJApp

**Acceder a los detalles de localización**

Desde la pantalla de *Inicio*, pulsamos sobre el ícono *Localización*, tal y como se muestra en la ilustración 58.

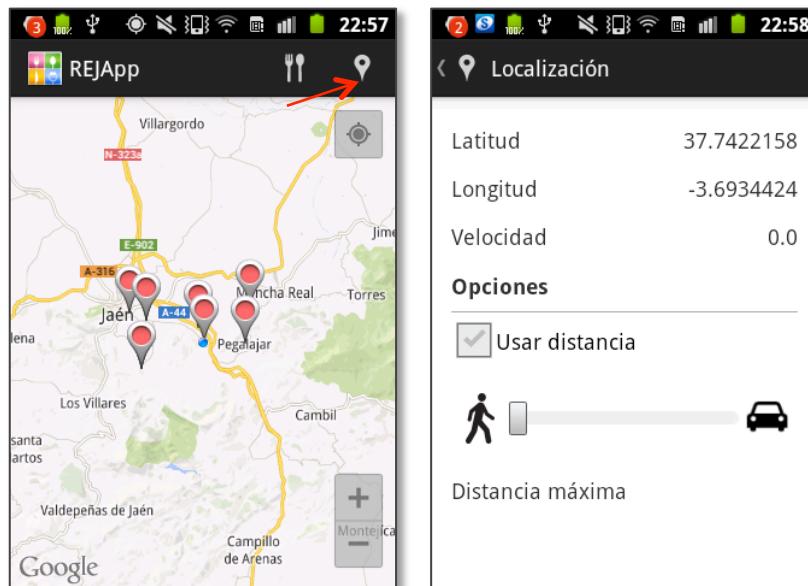


Ilustración 58. Localización en REJApp

**Modificar la distancia máxima de los restaurantes que se recomiendan**

En la pantalla de *Localización*, marcamos la opción *Usar distancia*. La barra de desplazamiento se activará y podremos elegir una distancia, que figurará de forma numérica abajo (véase ilustración 59).



Ilustración 59. Modificar distancia en REJApp

### Acceder al perfil de usuario

Desde la pantalla de *Inicio*, pulsamos sobre el ícono *Perfil de usuario*. Este ícono aparecerá en pantalla en función del ancho de la misma. Si es pequeña y el dispositivo utilizado consta del botón de menú, entonces podemos acceder pulsando sobre el mismo. Si no, aparecerá como una opción del desplegable. Si visualizamos la pantalla en modo horizontal, aparecerá al final, tal y como se muestra en la ilustración 60.

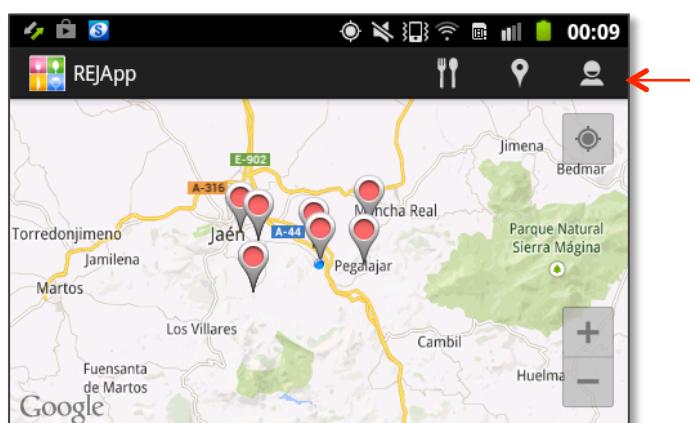


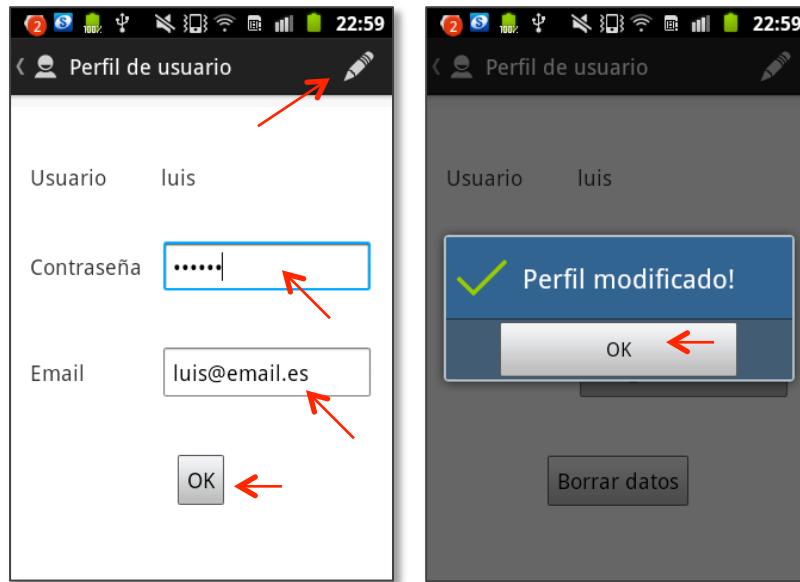
Ilustración 60. Inicio en modo horizontal en REJApp



Ilustración 61. Perfil de usuario en REJApp

### ***Modificar el perfil de usuario***

En la pantalla de perfil de usuario, pulsamos sobre el ícono *Editar*. Los campos contraseña y email se activarán y podemos editarlos pulsando sobre ellos. Para guardar los cambios, pulsaremos sobre el botón *Ok*, tal y como muestra la ilustración 62.



**Ilustración 62. Modificación del perfil de usuario en REJApp**

**Desvincular la cuenta de usuario**

En la pantalla de perfil de usuario, pulsamos sobre el botón *Borrar datos*, y aparecerá un mensaje de confirmación. Pulsamos *Aceptar* y la cuenta quedará desvinculada.

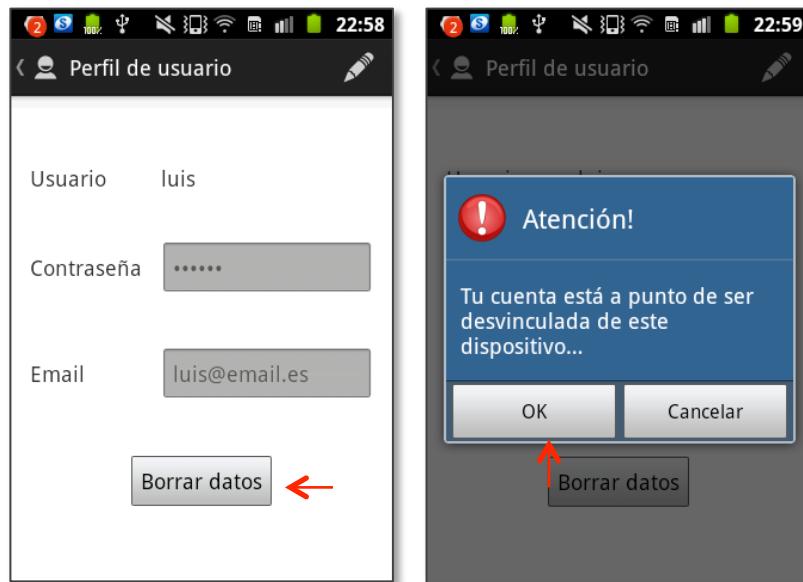


Ilustración 63. Desvincular cuenta de usuario en REJApp



## **Anexo III. Importación de la librería Google Maps en un proyecto Android**

---

Vamos a utilizar la biblioteca de Google Maps para la creación de mapas en nuestra aplicación. Para ello, es necesario seguir una serie de pasos:

1. Instalar el paquete Google Play Services a través de Android SDK Manager.

En el entorno de desarrollo Eclipse ADT Bundle, podemos acceder al SDK Manager a través de *Window > Android SDK Manager*.

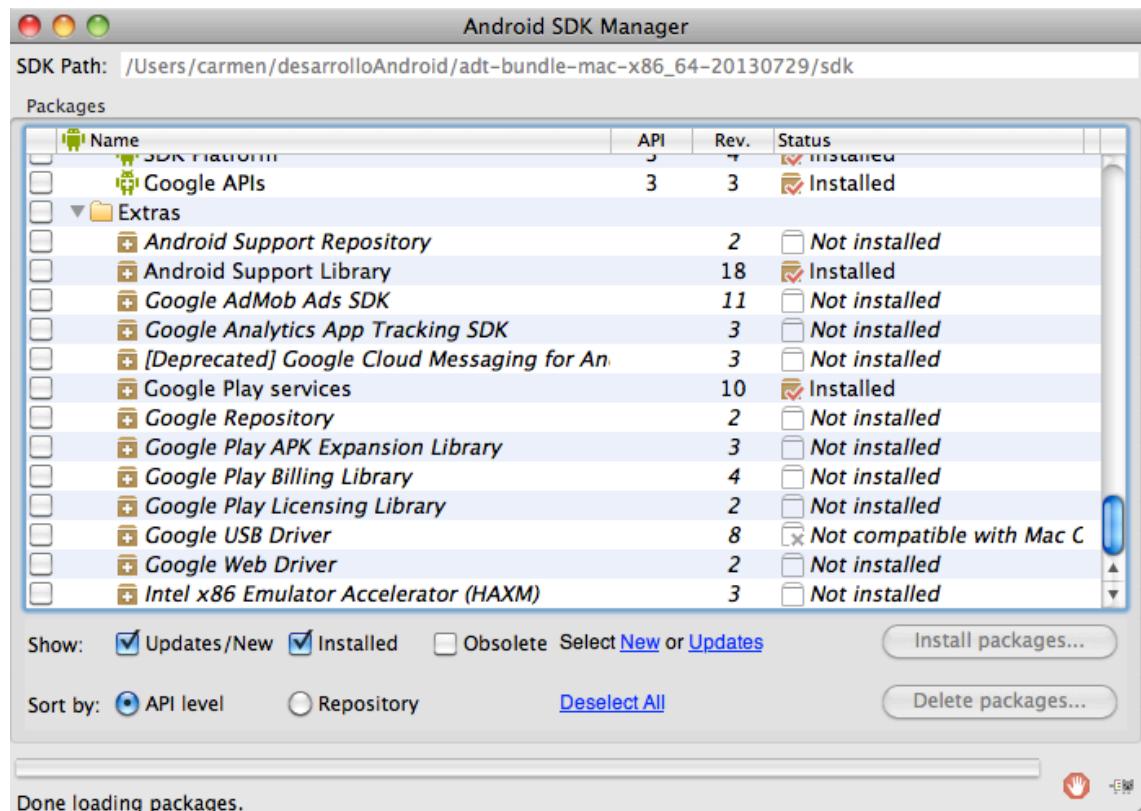


Ilustración 64. Android SDK Manager

El paquete se encuentra dentro del directorio *Extras*. Si no se encuentra instalado, bastará con marcarlo para su instalación y pulsar el botón *Install packages...* .

2. Importar al espacio de trabajo el proyecto con las librerías de Google Maps.

Desde el menú *File > Import* seleccionaremos la opción *Existing Android Code Into Workspace*. Ahora podemos navegar hacia el proyecto que se encuentra dentro del directorio de instalación de nuestro entorno, siguiendo la ruta */sdk/extras/google/google\_play\_services*.

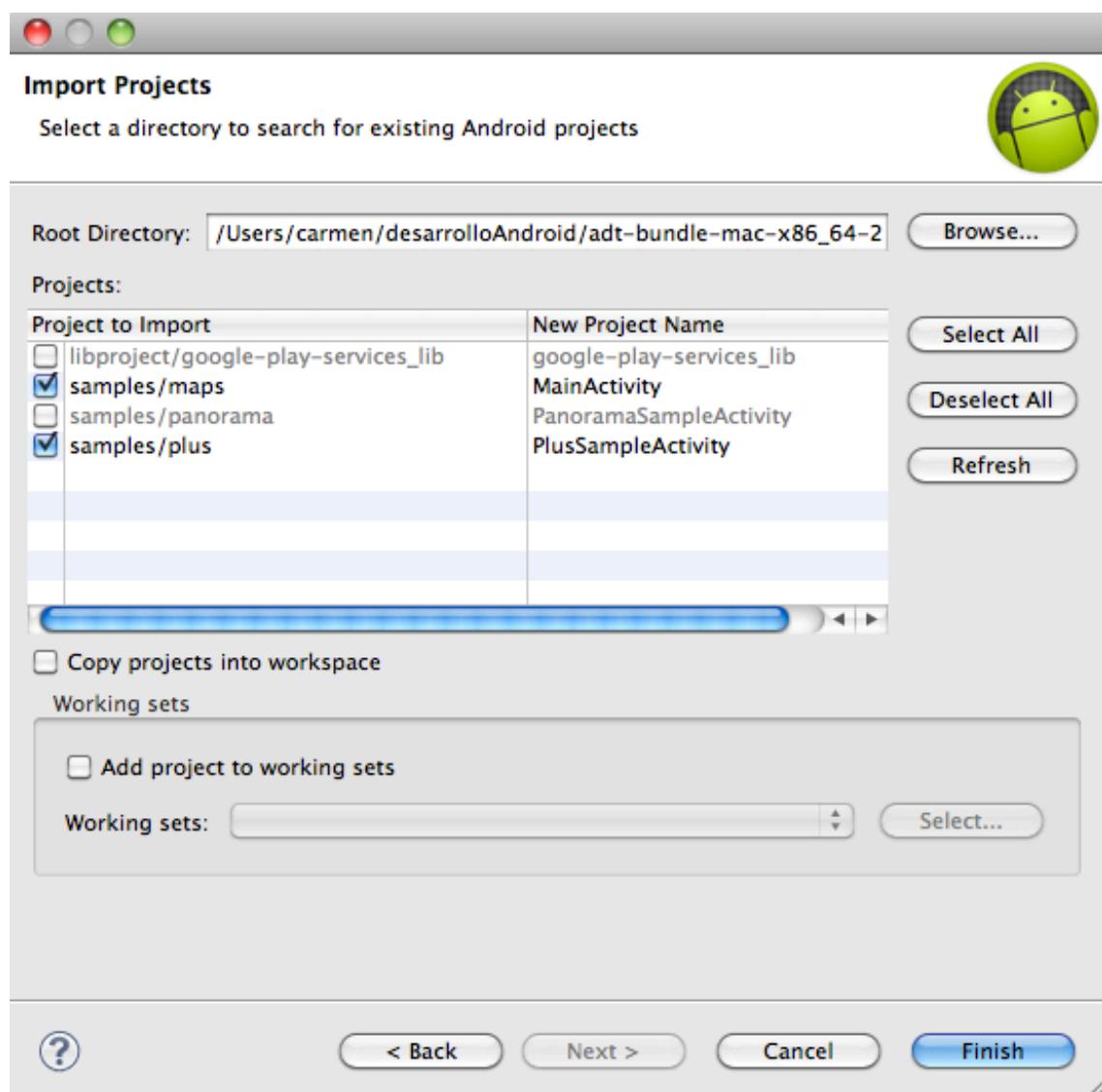


Ilustración 65. Importar las librerías de Google Maps

### 3. Añadir las librerías de Google Maps a nuestro proyecto.

Con el cursor sobre el proyecto, pulsamos el botón derecho y seleccionamos la opción *Properties > Android*. En la sección de librerías podemos añadir el proyecto que acabamos de importar pulsando el botón *Add...*.

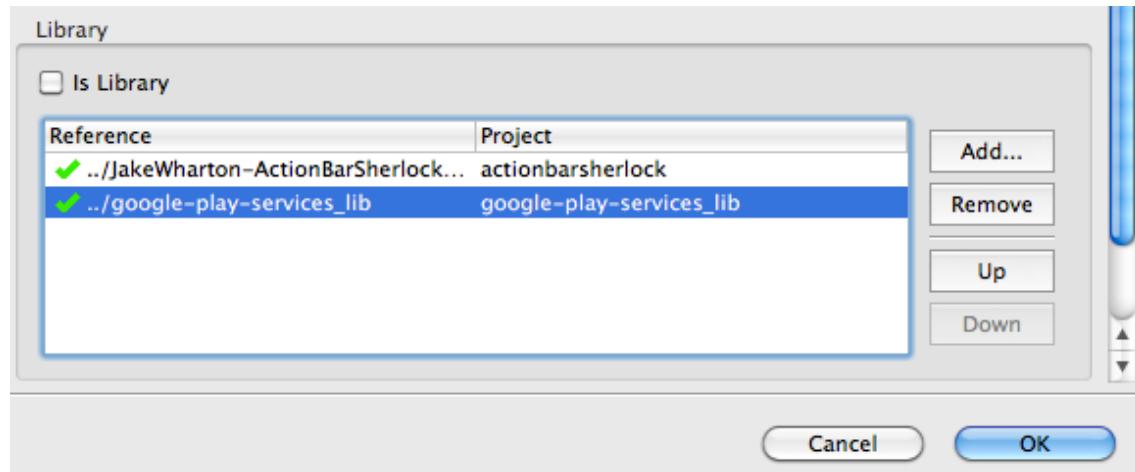


Ilustración 66. Añadir las librerías de Google Maps a nuestro proyecto

4. Obtener una clave de API de Google Maps que nos permita utilizar el servicio.

Para ello, en primer lugar debemos extraer la huella digital SHA1 del fichero *debug.keystore* a través del programa *Keytool* (el proceso detallado está en [12]). Después, a través de la página web [13] registramos nuestra aplicación y marcamos la utilización del servicio *Google Maps Android API v2*. Ahora sólo necesitamos obtener nuestra API key introduciendo la huella digital obtenida anteriormente.

5. Añadir la API Key a nuestro proyecto.

Para ello incluimos la siguiente información en el archivo *AndroidManifest.xml* de nuestra aplicación, dentro de las etiquetas *<application></application>*:

```
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyDw0LIAKUlElGuphMe_ytHmlCth6kV7AJM" />
```

6. Añadir los permisos necesarios para la visualización de mapas a nuestro proyecto.

```
<permission  
    android:name="es.ujaen.rejapp.permission.MAPS_RECEIVE"  
    android:protectionLevel="signature"/>  
<uses-permission  
    android:name="es.ujaen.rejapp.permission.MAPS_RECEIVE"/>  
<uses-permission  
    android:name="com.google.android.providers.gsf.permission.READ_G  
SERVICES"/>  
  
<uses-permission android:name="android.permission.INTERNET" />  
  
<uses-permission  
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
  
<uses-permission  
    android:name="android.permission.ACCESS_FINE_LOCATION" />  
<uses-permission  
    android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"  
/>  
<uses-permission  
    android:name="android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission  
    android:name="android.permission.ACCESS_MOCK_LOCATION" />
```

7. Indicar en nuestro proyecto que la Google Maps requiere OpenGL

Para ello incluimos la siguiente información en el archivo AndroidManifest.xml de nuestra aplicación

```
<uses-feature  
    android:glEsVersion="0x00020000"  
    android:required="true" />
```

