



## Agenda móvil para ferias y exposiciones

**Nombre Estudiante: Francisco de Asís Olmo Ortiz**

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

**Nombre Consultor/a: Francesc Giralt Queralt**

**Profesor/a responsable de la asignatura: Carles Garrigues Olivella**

02/01/2019



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Agenda móvil para ferias y exposiciones</i>
<b>Nombre del autor:</b>	<i>Francisco de Asís Olmo Ortiz</i>
<b>Nombre del consultor/a:</b>	<i>Francesc Giralte Queralt</i>
<b>Nombre del PRA:</b>	<i>Carles Garrigues Olivella</i>
<b>Fecha de entrega (mm/aaaa):</b>	01/2019
<b>Titulación:</b>	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave:</b>	<i>Agenda, ferias, eventos</i>
<b>Resumen del Trabajo</b>	
<p>El presente proyecto tiene como finalidad la creación de una aplicación móvil que sirva como agenda electrónica a los asistentes a ferias de muestras y exposiciones. Se ha desarrollado con tecnología nativa de la plataforma Android, haciendo uso de las últimas tendencias tanto en el diseño de interfaz de usuario, como en la implementación del código fuente.</p> <p>Toda la lógica de la aplicación se ha basado en el patrón modelo vista vista modelo, haciendo uso de la tecnología Architecture Components del SDK de Android.</p> <p>Como sistema de gestión de base de datos, se ha optado por la plataforma en la nube Firebase, a través de su sistema Real time database, así como del servicio Storage, para el almacenamiento de archivos multimedia.</p> <p>El resultado obtenido es una aplicación de última generación compacta y rápida que ofrece una buena experiencia de usuario, resultante de la combinación del diseño Material design, la arquitectura Architecture Components y la base de datos en tiempo real Firebase.</p> <p>El uso de estas recientes tecnologías, supone un gran paso adelante en el desarrollo de aplicaciones nativas para Android, por la enorme simplificación de procesos tanto en el diseño como la programación, lo que redundará en plazos más cortos de desarrollo y en aplicaciones más compactas, rápidas y seguras.</p>	

## **Abstract in English**

The purpose of this project is to create a mobile application that serves as an electronic agenda for attendees at trade shows and exhibitions. It has been developed with native technology from the Android platform, making use of the latest trends in both the user interface design and the implementation of the source code.

All the logic of the application has been based on the pattern model view view model, making use of the Architecture Components technology of the Android SDK.

As a database management system, we have opted for the Firebase cloud platform, through its Real time database system, as well as the Storage service, for the storage of multimedia files.

The result obtained is a compact and fast last generation application that offers a good user experience, resulting from the combination of Material Design, Architecture Components and the Firebase real-time database.

The use of these recent technologies is a great step forward in the development of native applications for Android, due to the enormous simplification of processes both in design and programming, which results in shorter development times and more compact applications, faster and safer.

# Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	4
1.4 Planificación del Trabajo.....	6
1.5 Breve resumen de productos obtenidos.....	8
1.6 Breve descripción de los otros capítulos de la memoria.....	8
2. Diseño de la aplicación.....	9
2.1 Usuarios y contexto de uso. Análisis.....	9
2.2 Diseño.....	10
3. Programación y desarrollo.....	17
3.1 Diseño técnico de la aplicación.....	17
3.2 Implementación del proyecto.....	25
3.3 Pruebas de la aplicación.....	36
4. Conclusiones.....	38
5. Bibliografía.....	39
6. Anexos.....	41

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

Se trata de una agenda móvil para los asistentes a ferias de muestras o exposiciones que les permita mantenerse informados sobre los eventos que tendrán lugar, así como suscribirse a aquellos que deseen y ser avisados con antelación. También ofrecería información sobre los expositores y las noticias que vayan aconteciendo.

La aplicación propuesta supondría una herramienta muy útil a la hora de organizar la agenda de eventos a los que asistir, así como para avisar con antelación de la hora y lugar de los mismos.

Los clientes potenciales a los que se dirigiría la aplicación, serían ferias de muestra que suelen disponer de presupuestos pequeños y que por tanto, no pueden permitirse invertir en una aplicación creada exclusivamente para ellos. Por tanto, la presente propuesta puede ser muy interesante. Se define a continuación.

Se trataría de un servicio que se usaría solo durante los días en los que tiene lugar la expo, estaría personalizado con sus logotipos y colores distintivos y disponible para dispositivos Android e iOS, si bien en este proyecto solo se desarrollaría la versión de Android. Dicho servicio constaría de la aplicación móvil y de un servicio web back-end desde el cual se gestionarían todos los contenidos y se enviarían los avisos a los usuarios de la app. Esto permitiría a dichas ferias disponer de este tipo de herramientas por un coste muy inferior al de un desarrollo propio y además, cada vez que se lleve a cabo la expo, tendrían sus contenidos actualizados.

Tras realizar un estudio de mercado, en busca de soluciones que ofrezcan diferentes alternativas a la hora de desarrollar una aplicación móvil para exposiciones, se pueden encontrar dos posibilidades que si bien consisten ambas en la obtención de una app propia para la expo, suponen no obstante, distintos niveles de calidad:

- A.** Desarrollo de aplicaciones creadas de forma exclusiva para una exposición. En este caso, las distintas ofertas tanto económicas como de calidad, se basan en el ya habitual debate entre el desarrollo nativo frente al desarrollo híbrido, basado en tecnologías web.
- B.** Creación de una aplicación para la feria o expo, basada en la adaptación de plantillas o prototipos ya diseñados y desarrollados. En los casos analizados, se ha hecho uso del desarrollo híbrido, con la finalidad de conseguir un diseño uniforme cualquiera que sea la plataforma móvil, así como poder ofrecer un mejor precio. Algunos ejemplos son las apps de la Feria del Mueble de Zaragoza [1] o la

Feria Internacional de la Maquinaria Agrícola [2], creadas por Zimaltec Soluciones. Otro servicio de similar naturaleza, es ofrecido por la empresa Xeria, para eventos como Basquisite [3] o InteriHotel [4].

El producto que aquí se presenta, iría un paso más adelante, pues no se incluye en ninguna de las soluciones anteriores, sino que propone un modelo de negocio alternativo. Se trataría de alquilar una plataforma completa de servicio cloud más la aplicación móvil, con las ventajas del desarrollo nativo, adaptando el diseño a la imagen corporativa del evento, pero sin tener que pagar por una aplicación propia.

Ideas similares, se encuentran en plataformas como Meetmaps [5] o EvenTwo [6], para congresos y eventos. Se trata de sistemas de suscripción anual cuyo coste ronda alrededor de los 500€ y que están muy centradas en construir una red social en torno a los asistentes a un congreso. La organización del evento cuenta con un backend para gestionar los contenidos de la app móvil.

El servicio que se pretende desarrollar en el presente proyecto, estaría más enfocado a exposiciones y ferias de muestra que a congresos y su utilidad principal sería servir de agenda personal a los asistentes, durante los días de duración. Tampoco obligaría a una suscripción anual, sino exclusivamente al periodo de duración del evento.

## 1.2 Objetivos del Trabajo

Descripción de los requerimientos funcionales de la aplicación:

- **Información de la expo**  
Ofrecerá información sobre la expo o feria, promotores, entidades colaboradoras, ubicación y acceso mediante el uso de Google Maps y contacto a través de email, teléfono, Twitter y Facebook.
- **Listado de expositores**  
Mostrará a todas las empresas que exponen sus productos o servicios. En la lista aparecerán el logotipo y marca y una breve descripción de cada empresa. Al seleccionar una empresa, se pasa a la pantalla de detalle, donde se mostrará la información completa, así como la posibilidad de contactar a través de teléfono, email, Twitter o Facebook, si dispone de estas redes sociales. También se tendrá acceso a visitar su web, en caso de que exista.

Esta lista de expositores, estará sujeta a cambios que serían comunicados al usuario mediante notificación push. El usuario al pulsar sobre el aviso de la notificación, abrirá directamente la ficha del expositor actualizado, o nuevo o bien, en caso de baja, mostrará directamente el listado.

- **Agenda de eventos**

Listado de todos los eventos programados durante la expo o feria. Se mostrará el título, breve descripción, día, hora y lugar de cada evento de la agenda. Al seleccionar un evento de la lista, se mostrará la pantalla con su información de detalle, incluyendo la posibilidad de compartir en las redes sociales. En dicha pantalla de detalle, se mostrará sobre Google Maps, la ubicación donde tendrá lugar el evento. Finalmente, el usuario podrá suscribirse a dicho evento, para poder recibir avisos que le recuerden la fecha y hora de celebración.

Al igual que la lista de expositores, la agenda de eventos estará sujeta a cambios que serían comunicados al usuario mediante notificación push. El usuario al pulsar sobre el aviso de la notificación, abrirá directamente la ficha del evento actualizado o nuevo. En caso de baja se mostraría directamente el listado.

- **Subscripción a eventos**

Los eventos a los que el usuario se haya suscrito, se mostrarán en una la lista de favoritos. La lista de eventos suscritos y el detalle del evento, mostrarán la siguiente información:

- En el listado de eventos suscritos, en cada ítem de la lista se ofrecerá un botón que permitirá eliminar la subscripción.
- En el detalle del evento suscrito, aparecerá un menú de opciones que permitirá al usuario configurar la forma en que recibirá los avisos: sin sonido y vibración, con sonido y vibración, así como cuántos avisos desea recibir. Una vez se haya celebrado el evento, el usuario dispondrá de un formulario simple que le permitirá enviar su opinión sobre el mismo, al servicio web.

- **Noticias de la expo**

Se ofrecerá información de actualidad, con la posibilidad de fotos o vídeo, lógicamente, si se dispone de dicho material.

En cuanto a los **requerimientos no funcionales** o atributos de calidad, los objetivos se centrarían en conseguir la mejor experiencia de usuario, en relación al contexto de uso: alto nivel de ruido, asistentes que suelen cargar con folletos, bolsas y demás objetos que imposibilitan el manejo certero del móvil, distracción constante por lo atractivo del entorno, etcétera. Todo esto, requiere la máxima atención en los siguientes aspectos:

- **Usabilidad**

Interfaz manejable con una sola mano, siempre que sea posible. Estricto cumplimiento del principio de diseño KISS, es decir, mantener el funcionamiento en todo caso, lo más simple posible. Uso de elementos de interfaz nativos del sistema, con cuyo funcionamiento, los usuarios están familiarizados.



- **Eficiencia**  
Novedades comunicadas desde Firebase mediante mensajes push que abrirán notificaciones locales nativas, evitando que los usuarios tengan que buscarlas manualmente. Uso de alarmas del sistema, para recordar al usuario la proximidad de los eventos a los que se haya suscrito. Para aumentar la eficacia de estas comunicaciones, la aplicación usará la vibración, dado el contexto de alto ruido ambiental citado anteriormente.
- **Seguridad**  
Uso del sistema de permisos del SDK de Android, para el acceso a recursos locales, mapas, alarmas del sistema y notificaciones, así como para la conexión con Firebase vía Internet.

### 1.3 Enfoque y método seguido

#### A. Metodología de diseño y desarrollo del proyecto

La naturaleza inherente a un proyecto de fin de estudios, como el que aquí se presenta, hace pensar que la metodología más apropiada sería un desarrollo en cascada o waterfall. Se entiende así, al tratarse de un proyecto consistente en la entrega de un producto final completo, acompañado de su memoria y presentación.

No obstante, el objeto del proyecto no es sino el diseño y desarrollo de una aplicación móvil, para lo cual se recomienda la implementación de metodologías ágiles [7] que fueron creadas expresamente para este tipo de trabajos.

Ambas realidades, hacen recomendable optar por una combinación de las dos metodologías, cuyos principios serían aplicables de la siguiente manera:

#### ***Waterfall:***

- El proyecto está estructurado en una secuencia de fases definidas por el cliente (la Universidad), con fechas de entrega establecidas desde el principio.
- No existe un equipo de desarrollo, sino que el estudiante asume el rol de la empresa desarrolladora.
- El producto final obtenido, será evaluado por la Universidad, una vez terminado su desarrollo completo.

### **Agile Scrum:**

- El desarrollo del proyecto se estructuraría en etapas o sprints que definen entregas del producto claramente definidas, en cuanto que pueden ser testeadas y evaluadas por sí mismas y que permiten avanzar a la siguiente fase con la seguridad del trabajo afianzado.
- El consultor de la asignatura asume los roles de Scrum Máster y Director del Proyecto.
- Si bien el equipo de desarrollo está formado solamente por el estudiante, éste asumiría todos los perfiles profesionales del Scrum Team, pues se entiende que su formación le permite afrontar este reto y abordar en solitario el desarrollo del proyecto.
- La Universidad, asumiría el rol del Product Owner o cliente, al ser quien establece los requerimientos del proyecto y a su vez, ser el receptor del producto final.
- Los sprints y sus conjuntos de tareas, quedan establecidos en el diagrama de Gantt del punto siguiente.

### **B. Estrategia de desarrollo de la aplicación móvil**

Al afrontar el desarrollo de una aplicación móvil, siempre surge el debate del desarrollo web frente al nativo. Ambas estrategias ofrecen una serie de pros y contras que básicamente pueden resumirse en:

- El desarrollo nativo ofrece los mejores resultados en cuanto a usabilidad y experiencia de usuario. No obstante, su coste es elevado al tener que programar la misma aplicación para diferentes plataformas.
- El desarrollo web no proporciona el nivel de experiencia de usuario de las aplicaciones nativas, pero ofrece una alternativa más económica para escenarios con menos exigencias o con menor presupuesto.

En el caso del producto que se pretende desarrollar en el presente proyecto, se ha considerado el desarrollo nativo como la mejor opción. Los motivos de esta elección se basan en los criterios siguientes:

- Ofrecer a las ferias y exposiciones, una aplicación de máxima calidad que les asegure una crítica positiva por parte de los usuarios. Tal como se ha comentado en la introducción, este no sería un producto exclusivo para una feria. Por el contrario, se trataría de una aplicación fácil y rápidamente personalizable para quienes la contraten, mediante un modelo de negocio basado en la suscripción temporal del servicio. De esta manera, la organización de la feria podría disponer de un producto profesional sin realizar un desembolso privativo. En cuanto a la empresa desarrolladora, su éxito comercial se basaría, lógicamente, en poder captar el mayor número posible de eventos.

- Una segunda razón, aunque de menor importancia, para optar por un desarrollo nativo (en este caso para Android), sería la oportunidad de poner en práctica todo lo aprendido sobre el desarrollo de aplicaciones para esta plataforma y poder contar con una aplicación publicada en Google Play, como aportación de gran valor para el currículum profesional.

### C. Estrategia de diseño de la aplicación móvil

En cuanto a la estrategia de diseño, se optaría claramente por un enfoque consistente, pues el objetivo es ofrecer la misma experiencia de usuario, sea cual sea el dispositivo que se use, manteniendo el mismo contenido, flujo, estructura y función.

## 1.4 Planificación del Trabajo

**Recursos.** Se describen a continuación los recursos que serán necesarios para el desarrollo del proyecto que aquí se presenta.

TIPO DE RECURSO	NOMBRE
<b>Humano</b>	<ul style="list-style-type: none"> <li>▪ Diseñador gráfico</li> <li>▪ Diseñador de aplicaciones multiplataforma</li> <li>▪ Desarrollador de aplicaciones Android</li> <li>▪ Diseñador de bases de datos</li> </ul>
<b>Material</b>	<ul style="list-style-type: none"> <li>▪ Lápices, papel, bolígrafos y reglas</li> </ul>
<b>Software</b>	<ul style="list-style-type: none"> <li>▪ Diseño mockup (NinjaMock)</li> <li>▪ Diseño gráfico (Fireworks, Photoshop)</li> <li>▪ Prototipado (Axure)</li> <li>▪ Android Studio</li> <li>▪ Base de datos y notificaciones push (Firebase)</li> <li>▪ Navegador Google Chrome</li> </ul>
<b>Hardware</b>	<ul style="list-style-type: none"> <li>▪ PC de desarrollo</li> <li>▪ Móviles de varios tamaños y versiones de Android</li> <li>▪ Cables de conexión USB</li> </ul>

**Planificación**. Diagrama de Gantt

Trabajo final de máster DADM. PLANIFICACIÓN DEL TRABAJO		DURACIÓN (horas)	TEMPORALIZACIÓN
<b>PEC2. Diseño</b>		L = lectivas F = festivas	OCTUBRE
			11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1	Análisis. Usuarios y contexto de uso	5 L + 4 F = 9	12 13 14
2	Diseño conceptual. Arquitectura de la información	12 F	13 14 18
3	Diseño visual. Prototipos de baja y alta fidelidad	5 L + 12 F = 17	19 20 21 22 27 28 29
4	Evaluación de la usabilidad	4 F	20 21
5	Diseño técnico. Casos de uso	5 L + 4 F = 9	26 27 28
6	Diseño técnico. UML de modelo de datos	12 F	27 28 29
<b>Hito 1. Entrega PEC2</b>			30
<b>PEC3. Implementación</b>			NOVIEMBRE
			1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12
1	Creación de base de datos Firebase + mensajes push	5 L + 16 F = 21	2 3 4 5 16 17 18
2	Codificación de la aplicación	25 L + 80 F = 105	9 10 11 16 17 18 23 24 25 30 31 1 2 3 7 8 9
<b>Hito 2. Entrega PEC3</b>			12
<b>PEC4. Entrega Final</b>			DICIEMBRE
			13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2
1	Codificación. Testeo y pruebas	10 L + 32 F = 42	14 15 16 21 22 23
2	Publicación en Google Play	5 L	28
3	Preparación de la memoria final	6 F	29 30
4	Creación de la presentación en vídeo	10 F	30 31
<b>Hito 3. Entrega PEC4</b>			1

Para el desarrollo del presente proyecto, se han elegido los días viernes, sábado y domingo de cada semana. Los días viernes, tendrían una dedicación de 5 horas por las tardes, consideradas horas lectivas.

Sábados y domingos, considerados festivos, tendrían 4 horas de dedicación en la mañana y otras 4 en la tarde. Se marcan en verde los días lectivos y en naranja los festivos. Así mismo, los hitos corresponden con las fechas de entrega y se muestran en rojo.

## 1.5 Breve resumen de productos obtenidos

El producto que se pretende obtener de la realización del proyecto, es fundamentalmente una aplicación nativa para dispositivos Android y más específicamente para teléfonos, pues se requerirá de una conexión 3G/4G que no suele estar disponible en la mayoría de tablets de este sistema operativo.

Adicionalmente al producto principal, este proyecto pondrá en marcha un servicio cloud para entrada y salida de datos, mediante sistema Firebase.

Junto con los anteriores productos y siguiendo las exigencias establecidas en la normativa de los trabajos de final de máster de la UOC, se entregarán una memoria del trabajo y una presentación virtual que resuma y presente de forma audiovisual, el producto obtenido.

## 1.6 Breve descripción de los otros capítulos de la memoria

De acuerdo con las etapas del desarrollo de aplicaciones Android, establecidas en la web oficial Developers [8], los capítulos que conformarían la memoria serían los siguientes:

- **Diseño de la aplicación.** Basado en los principios del diseño centrado en el usuario, es decir: análisis, diseño, prototipo y evaluación de la usabilidad.
- **Programación y desarrollo.** Modelo de datos para base de datos local (SQLite) y remota (Firebase), así como uso de los patrones de arquitectura modelo vista controlador o vista modelo, para el desarrollo de la app.
- **Tests y pruebas.** Pruebas del sistema y corrección de errores.
- **Publicación de la aplicación.** Publicar la aplicación en Google Play.

## 2. Diseño de la aplicación

En este segundo capítulo del proyecto, se abordan las distintas fases que conlleva el diseño de la aplicación. Dicho proceso se ajustará a la filosofía del Diseño Centrado en el Usuario [9], dado que se trata en definitiva de la creación de una aplicación interactiva y por tanto, el objetivo del diseño deberá centrarse en todo momento en ofrecer la mejor experiencia al usuario de la app.

Se describen a continuación, las etapas del diseño.

### 2.1 Usuarios y contexto de uso. Análisis

En esta primera fase, se trata de identificar a los distintos tipos de usuarios que podrían hacer uso de la aplicación, así como conocer sus necesidades y el contexto en el que habrán de interactuar con la misma.

Para ello se ha seleccionado a un grupo de personas que acostumbran a asistir a ferias de muestras locales, a las cuales se les ha realizado una entrevista de tipo abierto, a partir de la pregunta "***¿cómo sería la app móvil que te gustaría usar en una expo, a modo de agenda?***".

Un primer análisis cuantitativo de los usuarios objetivo de la aplicación, muestra dos tipos principales:

- Empresarios del sector que visitan las ferias para conocer nuevas técnicas o tendencias, así como para asistir a seminarios, talleres y conferencias que les resultan de gran interés.
- Personas particulares interesadas o consumidores frecuentes, de los productos o servicios que se exponen. Estos suelen asistir en busca de nuevas ideas o de novedades de los productos ya conocidos. Este tipo, sería el mayoritario.

En ambos casos, el contexto de uso de la aplicación sería el mismo: un ambiente con mucho público y por tanto con bastante ruido, eventos que pueden desarrollarse en lugares diferentes de las instalaciones de la expo, sobreinformación de los expositores en forma de catálogos y folletos en papel. Todo ello, hace que al usuario le resulte difícil recordar los horarios y lugares de los citados eventos. Para cubrir estas necesidades, se propone una aplicación que ayude a organizar los eventos a los que se desee asistir, así como avisar cuando dichos eventos vayan a tener lugar.

De toda la información recogida en esta primera fase de análisis, se procede a continuación al modelado de usuarios, ya en la fase de diseño.

## 2.2 Diseño

### 2.2.1 Modelado de usuarios

Con la información obtenida en la fase de análisis, se procede a modelar los perfiles de usuario, a través de la elaboración de fichas de persona y la descripción de escenarios de uso. Esto permite al diseñador, trabajar con una idea más concreta y mejor definida de las necesidades de dichos usuarios.

- Tipo de usuario: *Empresarios del sector*

### Ficha de persona

## Juan García Liaño



**Biografía**

Juan es natural de Sevilla y trabaja asociado con otros tres compañeros, en una microempresa dedicada a la restauración de muebles. Es especialista en tallado de madera. Les ha llegado información sobre una feria de muebles artesanales, en la cual va a desarrollarse un taller de tallado en madera de olivo.

Juan usa un teléfono Android de 5.5", con la versión 7. Ha instalado la app en su móvil y ha programado su asistencia al taller de talla y además a una conferencia sobre cómo comercializar productos de artesanía.

**Objetivos**

- Recibir info desde la app de forma automática
- Organizar su agenda con los eventos de interés
- Programar en su móvil avisos para dichos eventos

**Frustraciones**

- Perderse eventos de interés por olvidar la hora
- No enterarse de todas las actividades de la expo
- Tener que consultar una aplicación para estar informado

**Tecnología**

Internet

Software

Aplicaciones móviles

Redes sociales

**Marcas que usa**



**Personalidad**

Extrovertido Introvertido

Sensible Intuitivo

Cerebral Sentimental

Juicioso Espontáneo

Amigable Previsor Vital

**Escenario de uso**

Juan se encuentra asistiendo al taller de talla, junto a un reducido grupo de profesionales como él. El recinto es pequeño y todos están atentos a las explicaciones del maestro tallista. La aplicación le avisa en ese momento de que en dos horas tendrá lugar la conferencia. Afortunadamente se acordó de programar esa alarma con solo vibración y evita así, molestar a sus compañeros de taller con el sonido de dicha alarma.

Ha terminado el taller y Juan se encuentran conversando con sus colegas. La alarma vuelve a avisarle de que en media hora empieza la conferencia. Pulsa sobre la notificación y abre la ficha del evento, para consultar la ubicación sobre el mapa. También opta por compartir el evento del taller en Facebook. Dada la cercanía del nuevo evento, se dirige a pie junto a otros dos compañeros.

- Tipo de usuario: *Personas particulares*

**Ficha de persona**

## María Fernández Pons



“ Me gusta organizar mi agenda para la expo y no perder el tiempo consultando folletos y catálogos”

**EDAD:** 32

**OCUPACIÓN:** Profesora de secundaria

**ESTADO:** Soltera

**RESIDENCIA:** Murcia

**NIVEL:** Usuario medio

**ARQUETIPO:** La Metódica

Alegre

Organizada

Paciente

### Biografía

María vive en Murcia y acaba de comprar una casa de campo para pasar sus días de descanso. Busca mobiliario y elementos decorativos de estilo rústico para su recién adquirida propiedad. Decide asistir a una feria de muestras de artesanía, donde se exponen muebles elaborados en madera de olivo.

Es usuaria de un móvil Android de 6" y usa con asiduidad las redes sociales, especialmente Facebook, donde suele compartir sus experiencias. María ha instalado en su terminal la aplicación de la feria, para estar informada.

### Objetivos

- Recibir información de los expositores de la expo
- Asistir a eventos relacionados con decoración
- Organizar su agenda para la feria de muestras

### Frustraciones

- Tener que consultar tantos catálogos en papel
- Perderse eventos por no recibir información
- Olvidarse de los horarios de actividades de interés

### Tecnología

Internet

Software

Aplicaciones móviles

Redes sociales

### Marcas que usa






**Escenario de uso**

María ha configurado su agenda para asistir a una demostración de un decorador profesional donde espera tomar algunas ideas para su chalet. Son las doce del medio día y la aplicación le ha avisado que en media hora va a tener lugar el evento programado.

En ese momento se encuentra visitando expositores y a esa hora la concurrencia a las instalaciones de la expo es muy elevada, con lo cual el ruido le impide escuchar el sonido de la alarma. Afortunadamente, ha configurado dicha alarma para que use también use la vibración, con lo cual se percata del aviso.

Una vez consultada la dirección en la app, decide tomar un taxi rápidamente para asistir a su evento.

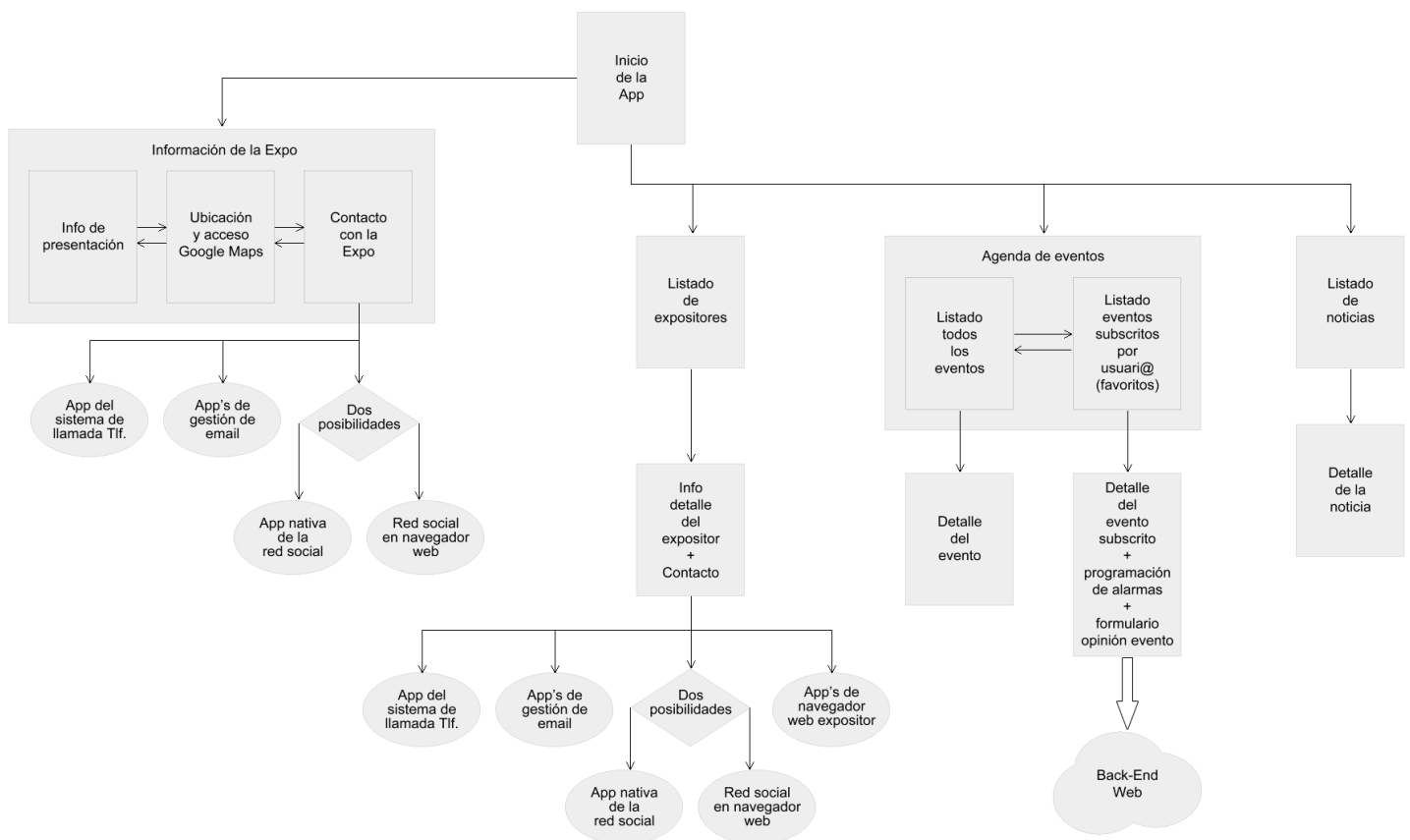
Las fotografías que se muestran son de uso libre y gratuito, seleccionadas del sitio web PEXELS [10]



## 2.2.2 Diseño conceptual

El diseño conceptual se refiere a la arquitectura de la información de la aplicación. En esta fase se procede a la organización y estructura de los contenidos y al etiquetado de sus títulos, categorías y definiciones. Así mismo, se define la estructura de navegación a través de las distintas pantallas que integrarán la aplicación. En este caso, se ha establecido una jerarquía, a partir de un menú principal de opciones.

El resultado de este proceso se plasma en el siguiente árbol de navegación:



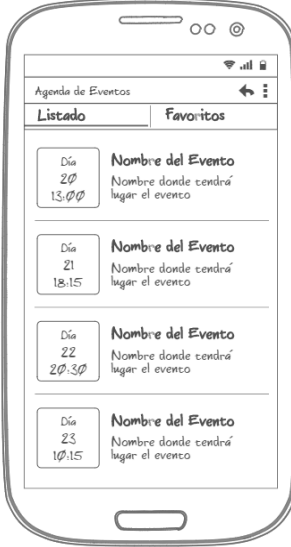

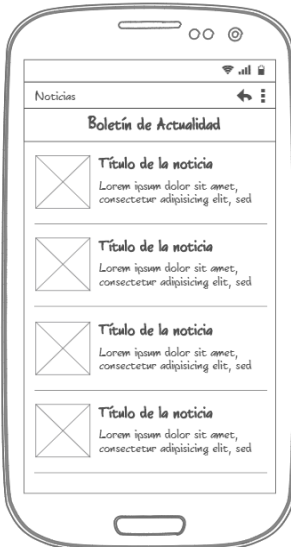


Las figuras rectangulares representan las pantallas de la aplicación y las circulares aplicaciones externas que serán abiertas explícita o implícitamente desde nuestra app.

Las figuras rectangulares que en su interior contienen otras figuras rectangulares, representan pantallas clásicas del sistema Android basadas en *tabs* o pestañas que son intercambiables con el *gesture swipe*, sin tener que cambiar de pantalla.

## 2.2.3 Prototipo

Con los resultados obtenidos del diseño conceptual, se procede a la creación de los prototipos de la aplicación o lo que es lo mismo, al diseño de la interfaz de usuario y de la interacción. Estos prototipos se diseñan en dos etapas, una primera donde se dibujan esbozos a lápiz o con software para este uso y una segunda fase, donde se evolucionan estos esbozos hasta obtener los diseños definitivos o prototipos de alta fidelidad.

Prototipos de baja fidelidad		
		
<p>Pantalla de inicio</p>	<p>Tab Presentación en Info Expo</p>	<p>Listado de eventos</p>
		<p><i>Algunas muestras de prototipos de baja fidelidad.</i></p> <p><i>Diseños realizados con el software NinjaMock® [11]</i></p>
<p>Detalle de un expositor</p>	<p>Listado de noticias</p>	

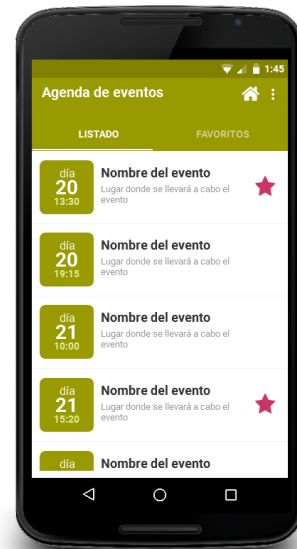
# Prototipos de alta fidelidad



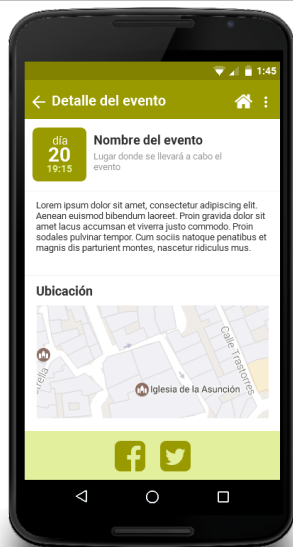
Pantalla de inicio



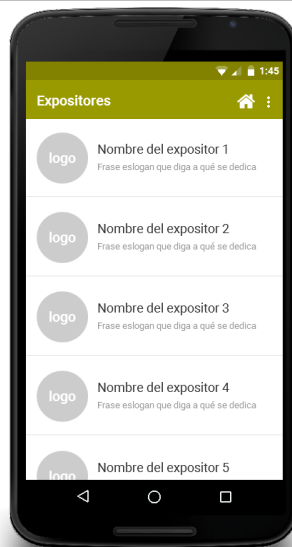
Tab Presentación en Info Expo



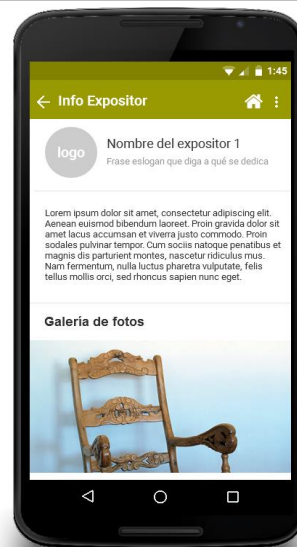
Listado de eventos



Detalle de un evento



Listado de expositores



Detalle de un expositor



Listado de noticias



Detalle de una noticia

*Prototipos de alta fidelidad de la aplicación.*

*Diseños realizados con el software Axure® [12]*

A modo de resumen, la aplicación propuesta se basa en un diseño para smartphones con sistema operativo Android. Por el contexto de uso previsto, se ha estimado conveniente que la app funcionase solo en modo portrait o vertical, para evitar giros indeseables de pantalla que fuesen en detrimento de la experiencia de usuario.

Con estas premisas, las soluciones de diseño e interacción finalmente aplicadas, se basan en los principios indicados por Google y conocidos como Material Design [13]. Si bien este concepto aparece a partir de la versión 5 de Android, gracias a la política de compatibilidad con versiones anteriores, se hace posible que la apariencia de las app's tenga patrones de diseño comunes, a partir prácticamente de la versión 4.0.

Los elementos que caracterizan a las aplicaciones diseñadas bajo estos conceptos, son básicamente los siguientes:

- Status bar, con iconos para la hora, batería, señal de telefonía y wi-fi, etc
- Toolbar para título, iconos y menú desplegable
- Menú desplegable en la esquina superior derecha
- Icono de flecha atrás y animación tipo swipe para regresar de lista al detalle
- Uso de pestañas o tabuladores para pasar entre varias vistas de contenidos, con animación tipo swipe, integradas en la misma pantalla, gracias al uso de los fragments

Material Design, recomienda también el uso de colores vivos, a partir de una paleta propuesta, para jugar con variaciones de brillo del tono elegido. Para este diseño se ha escogido una combinación de color verde, por adecuarse a la temática de una feria de artesanía en madera de olivo, tomada como ejemplo para esta ocasión.

Finalmente, en cuanto al uso de la tipografía, Google propone el tipo Roboto para los diseños de aplicaciones Android que obviamente, ha sido el usado para este prototipo.

Como conclusión, todos los elementos gráficos de la interfaz, son widgets clásicos de una app nativa Android que además, es lo que el usuario de este sistema espera y que garantizará la mejor experiencia de usuario.

## 2.2.4 Evaluación de la usabilidad

Como última fase del Diseño Centrado en el Usuario, se establece la estrategia para realizar la evaluación de la aplicación. No obstante, es un procedimiento que ha de hacerse en cada etapa del diseño y no solo cuando ya se ha obtenido el prototipo de alta fidelidad. De esta manera, se podrá ir avanzando en el diseño, con la seguridad de haber testeado con resultado positivo, la etapa anterior.

Para la evaluación de la usabilidad, se ha decidido hacer uso de los siguientes métodos:

- ***Evaluación heurística.***

Se analiza el diseño de interfaz propuesto, en base a los principios de la usabilidad o heurísticas. En este caso, sería el rol de *diseñador de aplicaciones multiplataforma*, quien se encargaría de llevarlo a cabo.

- ***Pruebas con usuarios.***

Se seleccionaría un grupo de usuarios, formado por personas ajenas al desarrollo del proyecto y cuyo nivel de conocimiento se limite al del usuario habitual de cualquier dispositivo móvil. Estos usuarios interactuarían con la app, en un contexto similar al definido en los escenarios de los perfiles de usuario, realizando una navegación que permita acceder a todas las pantallas y opciones de la aplicación.

Los usuarios dispondrían de un formato de test, mediante el cual se puntuarían del 1 al 10, las siguientes cuestiones sobre la aplicación:

- Las etiquetas de títulos y opciones son comprensibles
- Iconos, menús y navegación, son los habituales de la plataforma
- La navegación permite conocer en todo momento, dónde se está
- La interfaz informa adecuadamente sobre la utilidad de la pantalla
- Es fácil realizar la tarea que se pretende
- Los mensajes de respuesta informan correctamente
- Se han detectado errores o fallos de funcionamiento
- En general, es fácil aprender a utilizar la aplicación
- La aplicación resulta útil para los fines que se pretenden
- Recomendaría la aplicación a otras personas interesadas

Los datos obtenidos deberán ser analizados y tenidos en cuenta, para estimar o desestimar qué opciones se incorporan finalmente al diseño de la interfaz.

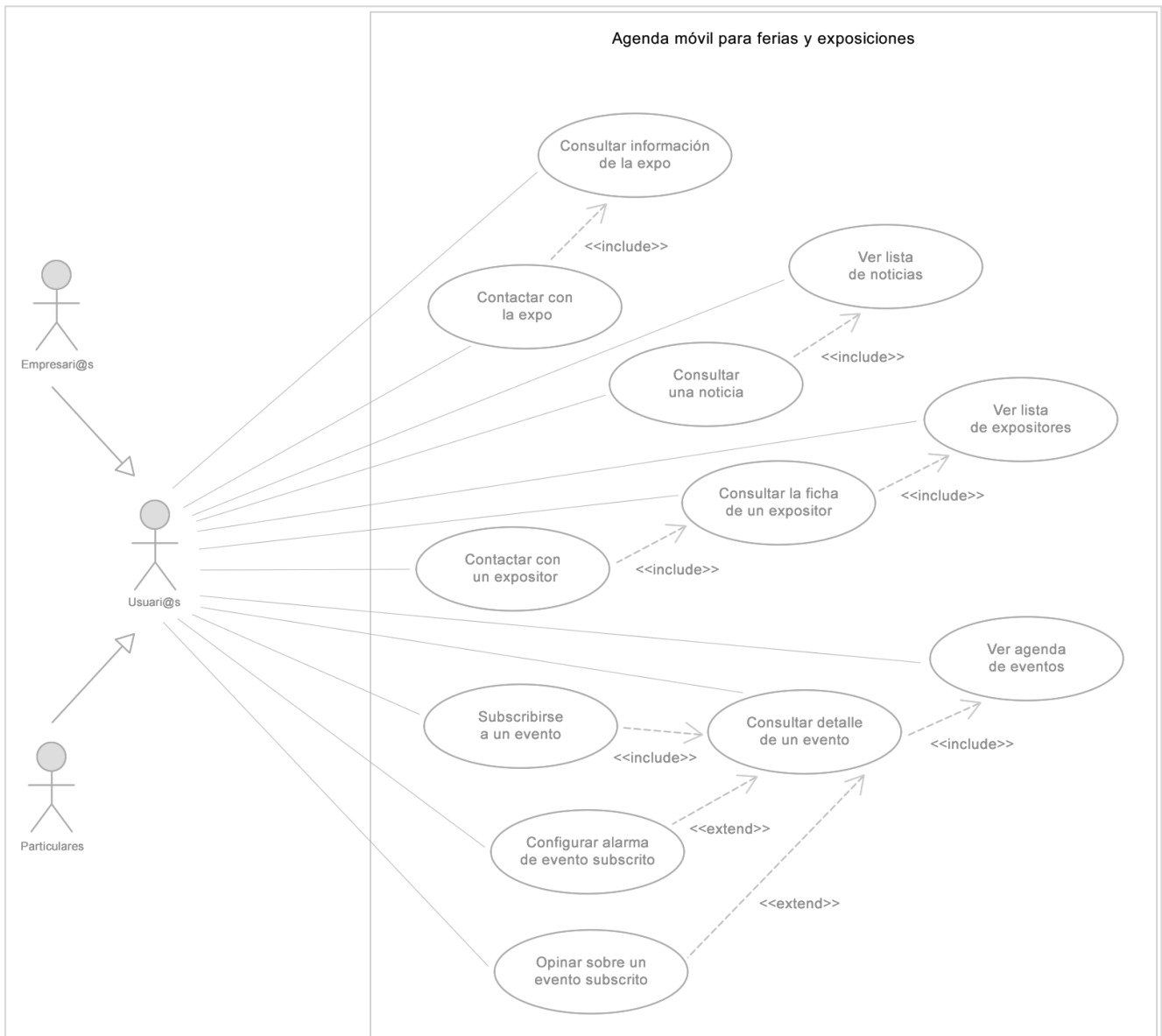
# 3. Programación y desarrollo

## 3.1 Diseño técnico de la aplicación

En este primer apartado, se procede a la definición formal de los casos de uso de la aplicación, así como a establecer su modelo de datos y diseño arquitectónico.

- **Definición de los casos de uso**

Se muestra en primer lugar el diagrama UML que identifica de forma gráfica a los actores y su relación con los casos de uso del sistema.



A continuación, se listan todos los casos de uso de la aplicación, mediante su correspondiente plantilla de descripción.

<b>Caso de uso 1</b>	Consultar información de la expo
<b>Objetivo</b>	Acceder a la información que la app contiene acerca de la expo
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario se encuentra en el menú principal y puede acceder desde la opción &lt;&lt;información&gt;&gt;</li> <li>2. El usuario puede encontrarse en cualquier pantalla y acceder desde el menú de la esquina superior derecha o desde el menú lateral deslizante (<i>navigation drawer</i>)</li> </ol>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se muestra la pestaña de info de presentación de la expo</li> <li>2. Se muestra la pestaña de Google Maps de ubicación y acceso</li> <li>3. Se muestra la pestaña de info de contacto con la expo</li> </ol>
<b>Incluye</b>	---
<b>Extiende</b>	---
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de info de presentación</li> <li>2. Mediante <i>swipe gesture</i>, se desplaza a derecha a la pantalla de ubicación y acceso</li> <li>3. Mediante <i>swipe gesture</i>, se desplaza a derecha a la pantalla de contacto con la expo</li> <li>4. Con <i>swipe gesture</i>, se desplaza a izquierda a las pantallas anteriores</li> </ol>

<b>Caso de uso 2</b>	Contactar con la expo
<b>Objetivo</b>	Contactar con la expo, por los distintos medios disponibles
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	El usuario se encuentra en la pantalla de Información de la Expo, en la pestaña de <<Contacto con la Expo>>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. El sistema ha accedido a la app de llamadas del móvil</li> <li>2. El sistema ha accedido a la app de email del móvil</li> <li>3. El sistema ha accedido a la app de la red social correspondiente</li> </ol>
<b>Incluye</b>	Consultar información de la expo
<b>Extiende</b>	---
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario accede mediante la opción de contacto telefónico, a la aplicación de llamadas de su teléfono</li> <li>2. El usuario accede mediante la opción de contacto por email, a la opción que prefiera, para gestionar su correo electrónico</li> <li>3. Mediante sus correspondientes opciones, el usuario accede a las aplicaciones de las redes sociales de la expo o bien a sus respectivas webs, en caso de no tener instaladas en el móvil dichas aplicaciones</li> </ol>

<b>Caso de uso 3</b>	Ver lista de noticias
<b>Objetivo</b>	Consultar el listado de noticias acaecidas durante la expo
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario se encuentra en el menú principal y puede acceder desde la opción &lt;&lt;noticias&gt;&gt;</li> <li>2. El usuario puede encontrarse en cualquier pantalla y acceder desde el menú de la esquina superior derecha o desde el menú lateral deslizante (<i>navigation drawer</i>)</li> </ol>
<b>Postcondiciones</b>	El sistema muestra la pantalla del listado de noticias
<b>Incluye</b>	---
<b>Extiende</b>	---
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario, mediante scroll vertical, consulta el listado de noticias</li> <li>2. El usuario, podrá acceder al detalle de la noticia que desee, mediante <i>touch gesture</i> sobre el ítem correspondiente de la lista</li> </ol>

<b>Caso de uso 4</b>	Consultar una noticia
<b>Objetivo</b>	Consultar el detalle de una noticia, del listado de noticias de la expo
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	El usuario se encuentra en la pantalla del listado de noticias
<b>Postcondiciones</b>	El sistema muestra la pantalla del detalle de la noticia elegida
<b>Incluye</b>	Ver lista de noticias
<b>Extiende</b>	---
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario consulta la información sobre la noticia elegida que se muestra en pantalla</li> <li>2. El usuario podrá regresar al listado de noticias, mediante el icono de flecha atrás o mediante las opciones correspondientes de los menús</li> </ol>

<b>Caso de uso 5</b>	Ver lista de expositores
<b>Objetivo</b>	Consultar el listado de expositores, asistentes a la expo
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario se encuentra en el menú principal y puede acceder desde la opción &lt;&lt;expositores&gt;&gt;</li> <li>2. El usuario puede encontrarse en cualquier pantalla y acceder desde el menú de la esquina superior derecha o desde el menú lateral deslizante (<i>navigation drawer</i>)</li> </ol>
<b>Postcondiciones</b>	El sistema muestra la pantalla del listado de expositores
<b>Incluye</b>	---
<b>Extiende</b>	---
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario, mediante scroll vertical, consulta el listado de expositores</li> <li>2. El usuario, podrá acceder a la info de un expositor, mediante <i>touch gesture</i> sobre el ítem correspondiente de la lista</li> </ol>



<b>Caso de uso 6</b>	Consultar la ficha de un expositor
<b>Objetivo</b>	Consultar la información de uno de los expositores asistentes a la expo
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	El usuario se encuentra en la pantalla del listado de expositores
<b>Postcondiciones</b>	El sistema muestra la pantalla del detalle del expositor elegido
<b>Incluye</b>	Ver lista de expositores
<b>Extiende</b>	---
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario consulta la información del expositor elegido que se muestra en pantalla</li> <li>2. Podrá regresar al listado de expositores, mediante el icono de flecha atrás o mediante las opciones correspondientes de los menús</li> </ol>

<b>Caso de uso 7</b>	Contactar con un expositor
<b>Objetivo</b>	Contactar con uno de los expositores asistentes a la expo
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	El usuario se encuentra en la pantalla de detalle del expositor elegido
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. El sistema ha accedido a la app de llamadas del móvil</li> <li>2. El sistema ha accedido a la app de email del móvil</li> <li>3. El sistema ha accedido a la app de la red social correspondiente</li> </ol>
<b>Incluye</b>	Consultar la ficha de un expositor
<b>Extiende</b>	---
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario accede mediante la opción de contacto telefónico, a la aplicación de llamadas de su teléfono</li> <li>2. El usuario accede mediante la opción de contacto por email, a la opción que prefiera, para gestionar su correo electrónico</li> <li>3. Mediante sus correspondientes opciones, el usuario accede a las aplicaciones de las redes sociales del expositor o bien a sus respectivas webs, de no tener instaladas en el móvil dichas apps</li> </ol>

<b>Caso de uso 8</b>	Ver agenda de eventos
<b>Objetivo</b>	Consultar la agenda de los eventos programados para la expo
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario se encuentra en el menú principal y puede acceder desde la opción &lt;&lt;agenda&gt;&gt;</li> <li>2. El usuario puede encontrarse en cualquier pantalla y acceder desde el menú de la esquina superior derecha o desde el menú lateral deslizante (<i>navigation drawer</i>)</li> </ol>
<b>Postcondiciones</b>	El sistema muestra la pantalla del listado de todos los eventos
<b>Incluye</b>	---
<b>Extiende</b>	---
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario, mediante scroll vertical, consulta el listado de eventos</li> <li>2. Mediante <i>swipe gesture</i>, se desplaza a derecha a la pantalla del listado de eventos suscritos</li> <li>3. Podrá acceder al detalle del evento que desee, mediante <i>touch gesture</i> sobre el ítem correspondiente de la lista</li> </ol>

<b>Caso de uso 9</b>	Consultar detalle de un evento
<b>Objetivo</b>	Consultar la información de uno de los eventos programados para la expo
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	1. El usuario se encuentra en la pantalla del listado de todos los eventos 2. El usuario se encuentra en la pantalla de la lista de eventos suscritos
<b>Postcondiciones</b>	El sistema muestra la pantalla del detalle del evento elegido
<b>Incluye</b>	Ver agenda de eventos
<b>Extiende</b>	---
<b>Flujo de eventos</b>	1. El usuario consulta la información del evento elegido que se muestra en pantalla 2. Podrá regresar a la agenda de eventos, mediante el icono de flecha atrás o mediante las opciones correspondientes de los menús

<b>Caso de uso 10</b>	Subscribirse a un evento
<b>Objetivo</b>	Añadir un evento a la lista de favoritos del usuario, con el fin de confeccionar su propia agenda, además de poder programar avisos y emitir opinión sobre los eventos a los que ha asistido
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	El usuario se encuentra en la pantalla del detalle de un evento
<b>Postcondiciones</b>	1. El evento quedará marcado en base de datos como suscrito por el usuario 2. El evento suscrito aparecerá marcado con un icono en la lista de todos los eventos 3. El evento suscrito aparecerá en la lista de eventos favoritos
<b>Incluye</b>	Consultar detalle de un evento
<b>Extiende</b>	---
<b>Flujo de eventos</b>	El usuario activa la opción de suscribirse al evento cuyo detalle está consultando
<b>Flujo alternativo</b>	En caso de ya estar suscrito, el usuario podrá anular dicha suscripción, desactivando el botón de suscripción

<b>Caso de uso 11</b>	Configurar alarma de evento suscrito
<b>Objetivo</b>	Establecer alarma para avisar de la proximidad en fecha y hora de la celebración de un evento suscrito
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	El usuario se encuentra en la pantalla del detalle de un evento suscrito
<b>Postcondiciones</b>	1. El sistema registra una nueva alarma asociada al evento suscrito 2. El sistema avisará en fecha y hora mediante sonido y vibración
<b>Incluye</b>	---
<b>Extiende</b>	Consultar detalle de un evento, ya que es condición para este caso de uso que el evento esté suscrito por el usuario
<b>Flujo de eventos</b>	1. El usuario pulsa la opción de crear alarma para el evento suscrito 2. Podrá establecer la antelación que desee, a la fecha y hora de celebración del evento 3. Así mismo, podrá seleccionar las opciones de sonido y/o vibración
<b>Flujo alternativo</b>	1. El usuario podrá eliminar la alarma del evento, en caso de que ya exista, pulsando la opción de eliminar alarma 2. El evento ya tuvo lugar. La app ya no muestra las opciones de alarma

<b>Caso de uso 12</b>	Opinar sobre un evento suscrito
<b>Objetivo</b>	Recibir las opiniones de los usuarios que han asistido a un evento, con el fin de evaluar el nivel de éxito y aceptación de dicho evento
<b>Actores</b>	Usuari@
<b>Precondiciones</b>	El usuario se encuentra en la pantalla del detalle de un evento suscrito
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. El formulario de opinión ha sido rellenado por el usuario y enviado al backend del sistema</li> <li>2. En la pantalla de detalle del evento, quedará la opinión enviada a modo de consulta</li> </ol>
<b>Incluye</b>	---
<b>Extiende</b>	Consultar detalle de un evento, ya que es condición para este caso de uso que el evento esté suscrito por el usuario
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario completa el formulario de opinión y pulsa el botón enviar</li> <li>2. La app envía esta opinión al backend del sistema</li> <li>3. Se notifica del resultado del envío y queda constancia de la información enviada en la pantalla, a modo de consulta</li> </ol>

### 3.1.1 Diseño de la arquitectura

#### ▪ Base de datos de la aplicación

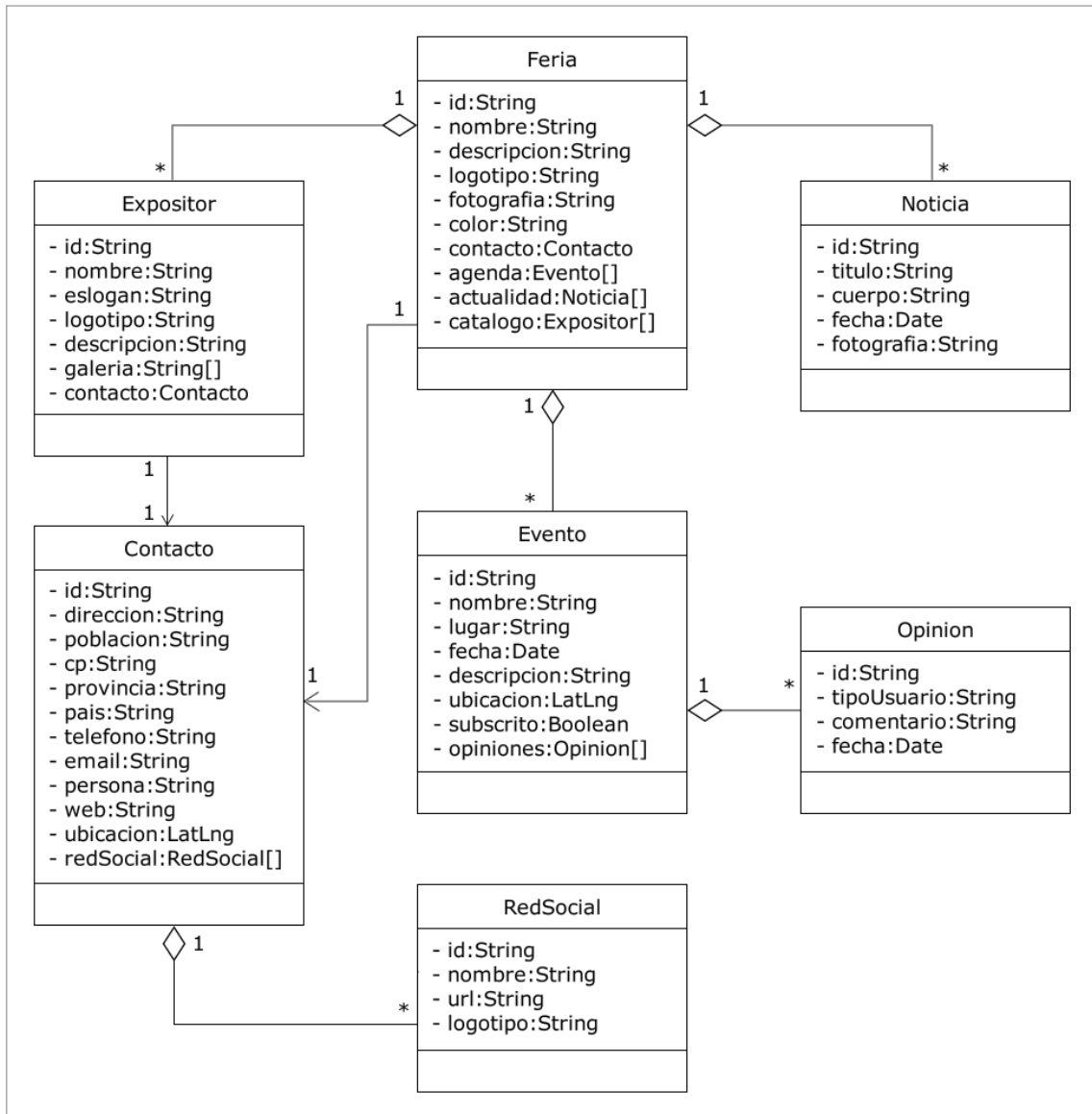
Se ha optado por hacer uso del sistema de bases de datos en tiempo real Firebase [14], para proveer a la app móvil de su correspondiente backend.

Firebase permite el trabajo sin conexión, habilitando un sistema de persistencia en disco, el cual almacena en caché los datos de una consulta, para poder ser usados offline. Esta característica permite no tener que hacer uso de una base de datos SQLite local.

Así mismo, al tratarse de un sistema de bases de datos NoSQL, su estructura no se basará en un esquema entidad-relación, sino que lo hará en formato JSON, en base al modelo de datos requerido para proveer de forma óptima a las vistas de la aplicación.

Dicho modelo de datos habrá de coincidir, por tanto, con la estructura de clases que se define a continuación, a través de su correspondiente diagrama UML.

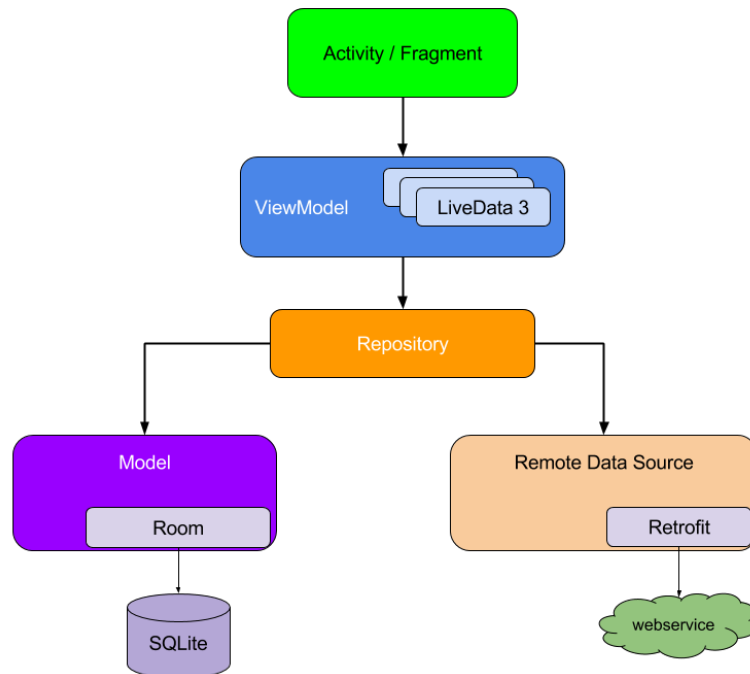
▪ Diagrama de clases UML



El presente diagrama UML, muestra el conjunto de clases que formarán el modelo de datos de la aplicación. Se muestran también las propiedades de dichas clases, así como la relación entre ellas.

## ▪ Patrón de arquitectura del sistema

Para el desarrollo de la aplicación propuesta, se ha optado por el patrón arquitectónico Modelo Vista Vista Modelo (MVVM), de acuerdo con las recomendaciones oficiales de Android Developers. Para su implementación, se han creado una serie de clases específicas para esta plataforma, basadas en el nuevo estándar Android Architecture Components [15]. A continuación, se muestra el esquema que aparece en la documentación oficial.



Básicamente se trata de conseguir el máximo nivel de separación entre las vistas de la aplicación y su modelo de datos. Para ello y tal como muestra el esquema, cada clase solo se relacionará con su inmediata inferior, de acuerdo al flujo indicado por las flechas.

De esta manera, las vistas (activity / fragment) contendrán instancias del ViewModel, subscribiéndose a las clases LiveData que actuarán de acuerdo al patrón de diseño observer. Estos observadores mostrarán en las vistas, cualquier cambio que ocurra en el modelo de datos. Se introduce por tanto, el paradigma de programación reactiva.

Finalmente, el modelo de datos, ya sea local o remoto, será gestionado por una interface o API que estaría incluida en el repositorio que muestra el esquema (Repository). Su misión sería abstraer toda la complejidad de dicha gestión y dejar que los ViewModel tengan como única responsabilidad, las peticiones de entrada y salida de datos que demanden las vistas.

## 3.2 Implementación del proyecto

Una vez concluida la fase de diseño, se procede al desarrollo de la aplicación, aprovechando los frameworks, librerías y APIs que ofrece el SDK oficial de Android.

En primer lugar, es necesario configurar el equipo de desarrollo, con los componentes de software requeridos para la programación de aplicaciones Android. En concreto serán necesarias al menos, las siguientes aplicaciones:

- Entorno de desarrollo **Java SE Development Kit** [16], para el uso de este lenguaje en la implementación del código fuente de la app.
- Entorno integrado de desarrollo **Android Studio** [17], plataforma oficial de desarrollo de aplicaciones de este sistema operativo.
- Sistema de control de versiones, como por ejemplo **GIT** [18], el cual será configurado desde Android Studio para mantener de forma local o remota, un control efectivo de las versiones de desarrollo de las apps.

A continuación, se detallan los pasos a seguir para el correcto desarrollo de la aplicación.

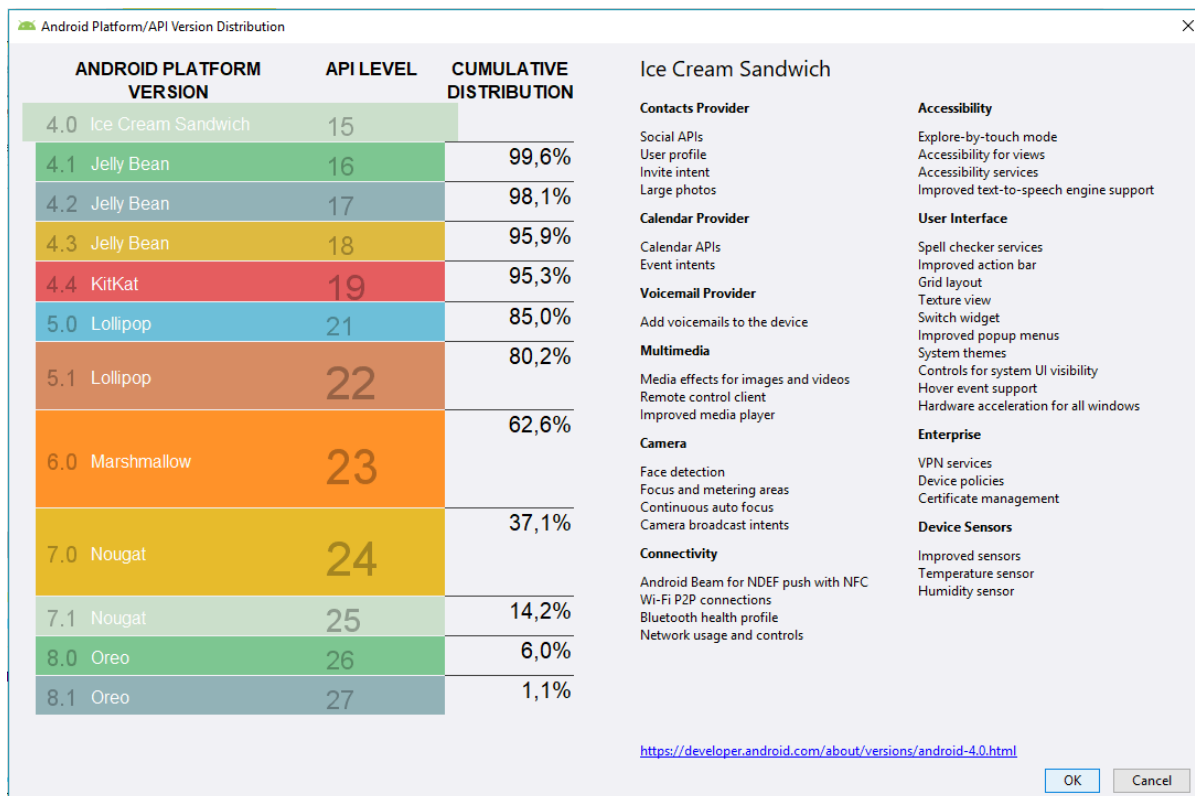
### 3.2.1 Desarrollo de la aplicación

Tras comprobar el correcto funcionamiento del entorno de desarrollo, se ejecuta la aplicación Android Studio, escogiendo la creación de un nuevo proyecto para teléfonos y tabletas.

De forma guiada, se irá introduciendo la información que solicita la plataforma, tal como nombre de la aplicación, dominio de la empresa, nombre del paquete principal y si se requiere el uso de lenguajes C++ y el más reciente Kotlin.

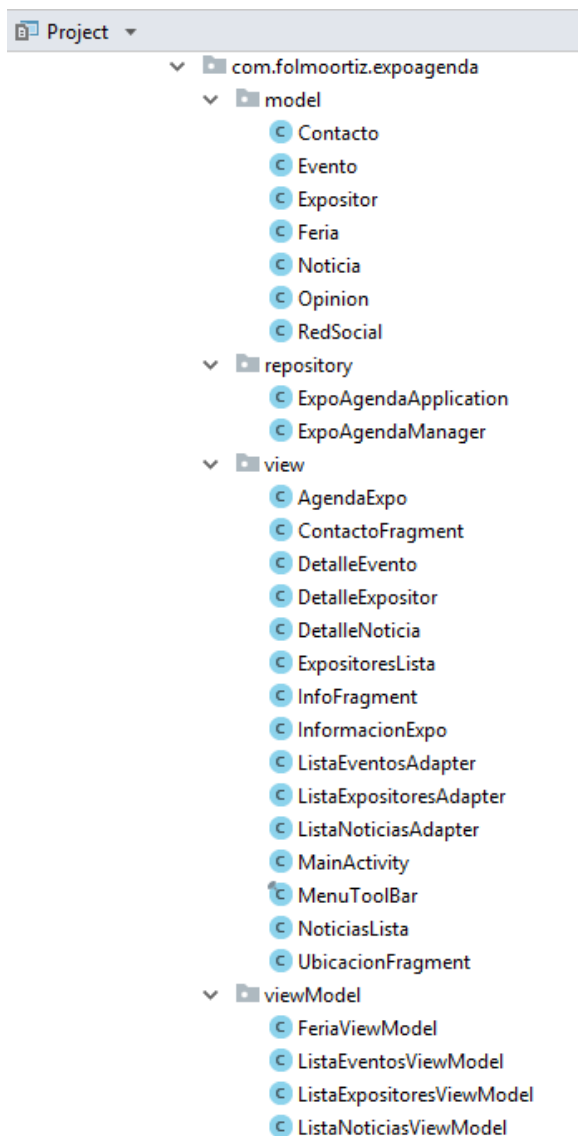
De especial interés es la pantalla en que se solicita la versión mínima de Android que será compatible con la app a desarrollar. Esto garantizará la mayor cobertura posible del parque de dispositivos que actualmente se encuentran en uso, debido al principal problema que afronta este sistema operativo: la fragmentación.

En la imagen que se muestra a continuación, puede comprobarse la diversidad de versiones de Android que aún se hallan operativas. Se muestra también, información sobre el porcentaje de compatibilidad que se obtendrá en función de la versión mínima de la que se parta.



Una vez finalizada la fase de creación del nuevo proyecto, Android Studio muestra una configuración básica con una pantalla o Activity y los elementos clásicos de una aplicación nativa, contenidos en sus respectivas carpetas.

### 3.2.1.1. Estructura de clases del proyecto



Tal como se ha descrito anteriormente, para el desarrollo de este proyecto, se ha optado por el patrón arquitectónico Modelo Vista Vista Modelo (MVVM).

Por tanto, partiendo del paquete base creado con la aplicación, se procede a crear la estructura de paquetes que se muestra en la imagen contigua y que lógicamente, se corresponde con el citado patrón arquitectónico.

En los siguientes apartados, se irá detallando el contenido de cada una de las capas que conforman la aplicación.



### 3.2.1.2. La capa Modelo

La aplicación que se desarrolla en este proyecto, basará su modelo de datos en la estructura de clases definida anteriormente en el diagrama UML. En cuanto a la base de datos que contendrá la información, tal como también se ha comentado ya, se implementará a través del sistema en la nube Firebase.

Por tanto, como se aprecia en la imagen anterior, dentro del paquete Modelo, se han creado todas las clases definidas en el esquema UML, con sus correspondientes propiedades o atributos. Se trata de clases simples que quedan perfectamente definidas como POJO's o "Plain Old Java Object", cuya única función será soportar el modelo de datos.

Estas clases, se completan con sus respectivos métodos *getter* y *setter*, para entrada y salida de datos. Sirva como ejemplo la clase Noticia:

```
public class Noticia implements Serializable {

    private String id;
    private String titulo;
    private String cuerpo;
    private String fecha;
    private String fotografia;

    public Noticia() {}

    public String getId() {return id;}

    public void setId(String id) {this.id = id;}

    public String getTitulo() {return titulo;}

    public void setTitulo(String titulo) {this.titulo = titulo;}

    public String getCuerpo() {return cuerpo;}

    public void setCuerpo(String cuerpo) {this.cuerpo = cuerpo;}

    public String getFecha() {return fecha;}

    public void setFecha(String fecha) {this.fecha = fecha;}

    public String getFotografia() {return fotografia;}

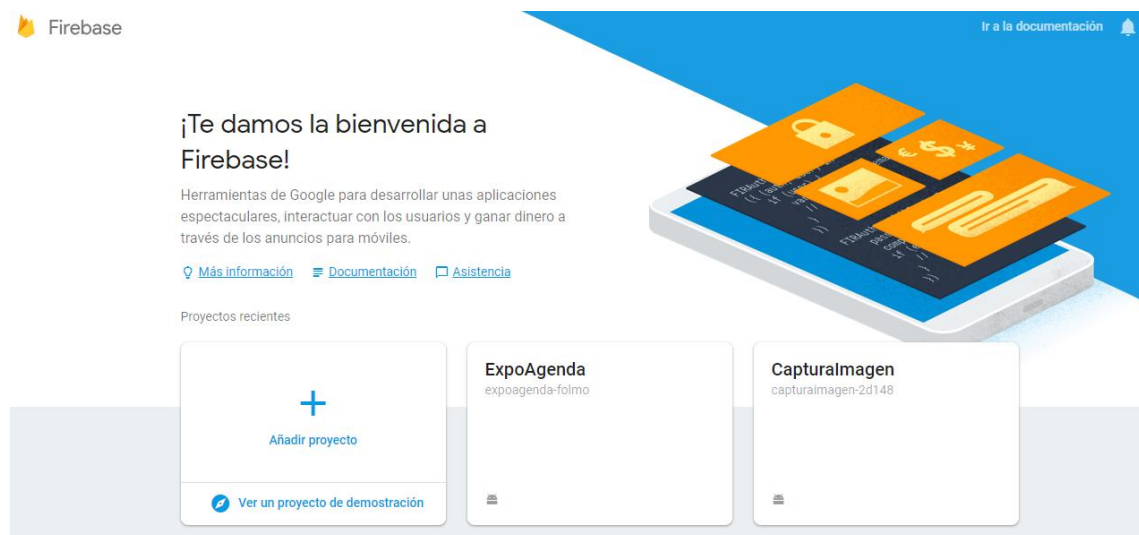
    public void setFotografia(String fotografia) {
        this.fotografia = fotografia;
    }
}
```

Como puede verse en la declaración de la clase, tanto este ejemplo como el resto de clases del modelo, implementan la interface *Serializable*, nativa de Java. Su finalidad es pasar objetos completos como parámetros, de las pantallas de listas a sus pantallas de detalle.

Toda esta infraestructura de clases del modelo, tiene su implementación en la plataforma Firebase, a través de su estructura de datos en formato JSON.

Para la carga de los datos y de los archivos de imágenes, se ha creado un fichero en dicho formato JSON, a través de un editor de código. Una vez comprobada su estructura correcta, se ha importado en Firebase, quedando de esta manera configurada tanto la estructura de clases, como los registros de datos. Este fichero JSON se adjunta como documento de texto anexo a la memoria del proyecto.

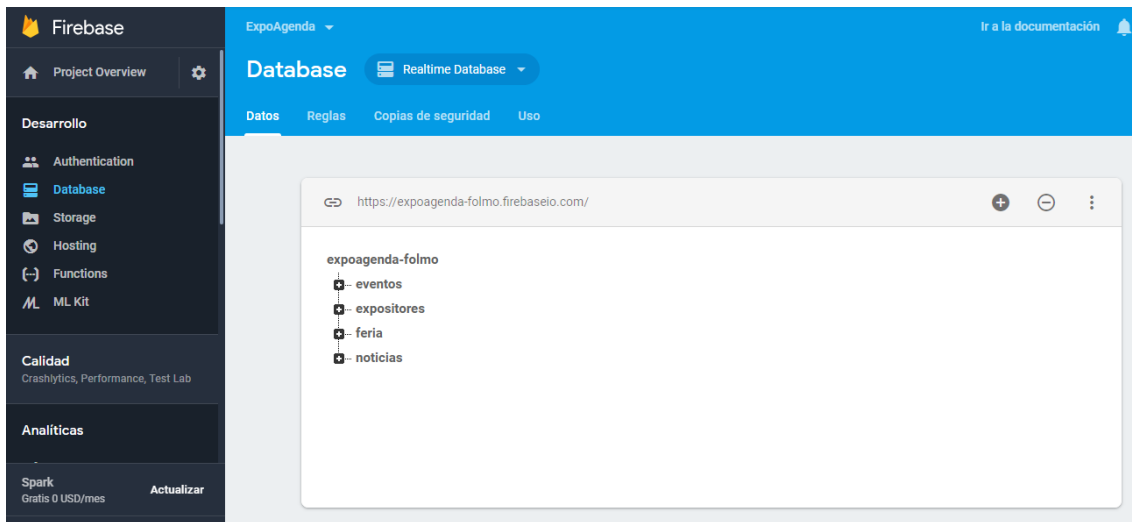
A continuación, se muestran capturas de pantallas, de todo el proceso de implementación de la base de datos Firebase.



El primer paso para hacer uso de Firebase, es crear un nuevo proyecto. En este caso, se ha creado el proyecto ExpoAgenda.

En segundo lugar, se muestra la estructura de nodos creados a partir del nodo base. Estos nodos, han sido creados en este orden, para hacerlos coincidir con las vistas que han de consumir sus datos, haciendo este proceso más rápido y eficiente.

En la página siguiente, se expone el gráfico con la estructura principal de la base de datos.

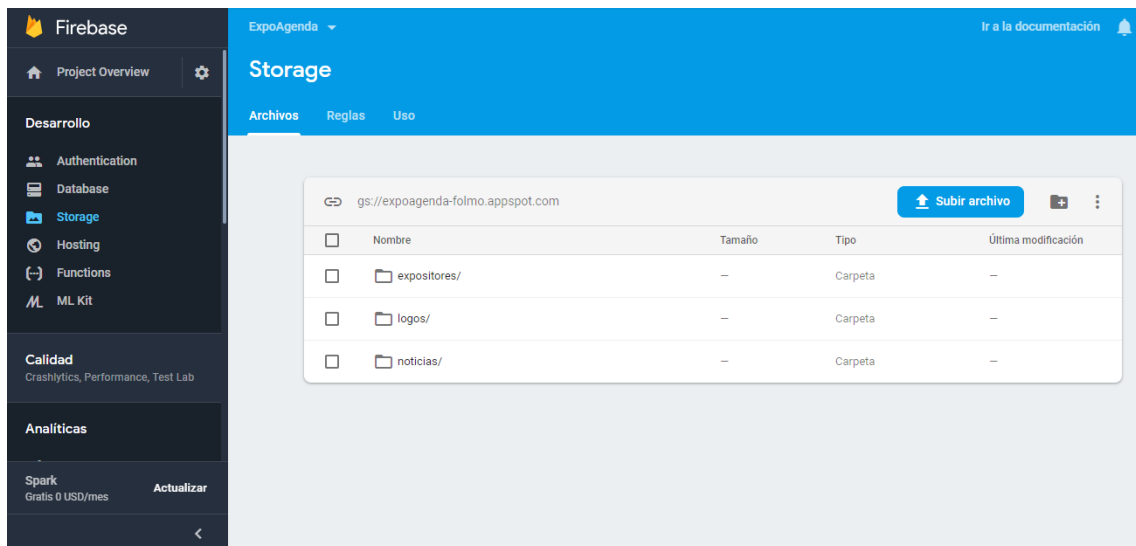


En la imagen, se aprecian los cuatro nodos creados a partir del nodo principal de la base de datos.

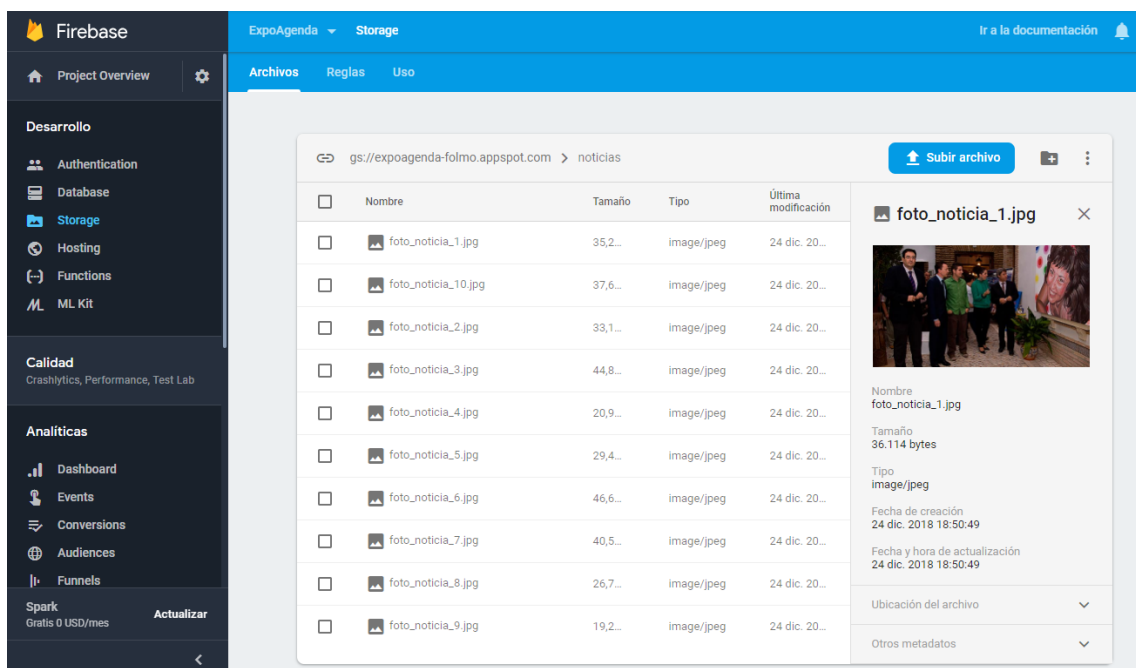


Se observan en la captura de pantalla, dos registros del nodo noticias, con sus atributos y datos correspondientes, tal como se han definido en el modelo de datos de la aplicación.

A continuación, se muestran pantallas con la configuración del servicio Storage, para subir los ficheros de las imágenes de la aplicación.



Puede apreciarse la estructura de carpetas que se ha creado, para albergar los contenidos multimedia de las tres clases que se corresponden con sus nombres.



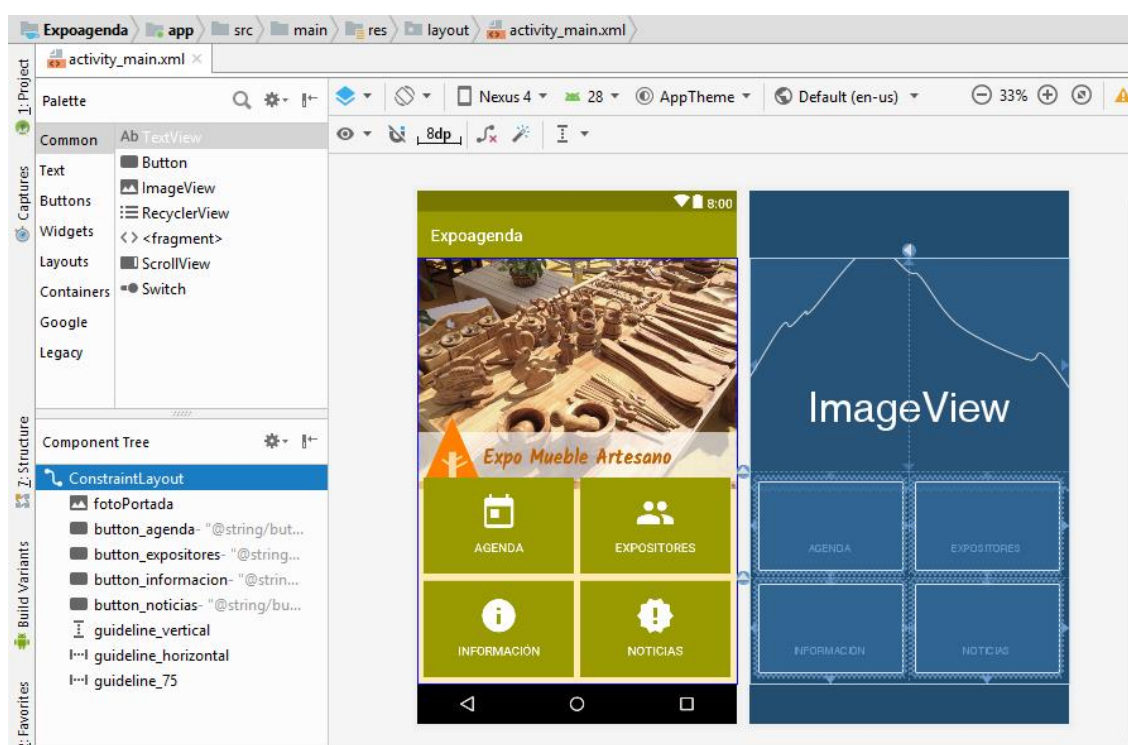
Listado de la carpeta noticias, con el detalle de la información de uno de los ficheros.

### 3.2.1.3. La capa Vista

Para la implementación de las vistas, se ha hecho uso de componentes de última generación, que ofrezcan al usuario la experiencia que espera de una aplicación profesional de Android.

Como ya se comentó en la sección de diseño, se han seguido las directrices y conceptos de la especificación Material design.

Se muestran a continuación capturas de pantallas, donde se pueden apreciar los componentes UI usados para la construcción de las vistas que en la plataforma Android se configuran en Activities y Fragments.



Maquetación del layout activity\_main.xml, haciendo uso del nuevo ConstraintLayout que permite elaborar diseños responsivos, de forma totalmente visual.



Layout que incorpora la App bar en la parte superior, donde se incluye el menú desplegable a la derecha y la flecha de navegación atrás a la izquierda. Todos los iconos utilizados pertenecen al catálogo Material icons y están en formato vectorial svg.

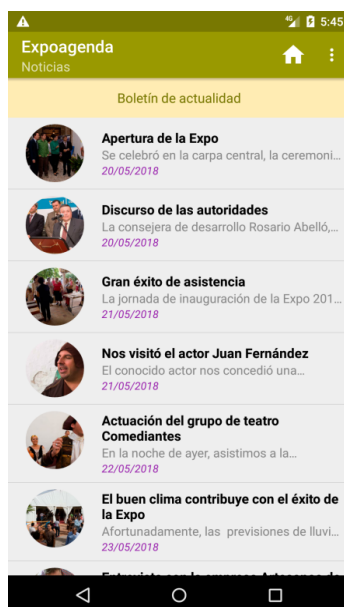
En la imagen se muestra uno de los compones UI de última generación, la Card View, que con formato de tarjeta, engloba toda la zona central de la vista. Se aprecia también la propiedad elevation que permite proyectar sombra sobre el fondo.



Otro de los componentes nativos de las apps modernas, son los Tab Layouts que permiten mostrar diferentes vistas en la misma pantalla, mediante el gesture swipe a derecha e izquierda o bien tocando alguna de sus pestañas.

El componente contenedor, es un View Pager y mediante un Fragment Manager, se encarga de sustituir en la vista, un fragmento por otro.

Finalmente, la imagen muestra un fragment que incluye un elemento Google Maps para ubicar a la expo.



Por último, se destaca el uso de componentes Recycler View para gestionar los listados. Se completan con la clase View Holder, cuya función consiste en cargar en pantalla solo los ítems que se muestran, dotando a las listas de gran agilidad y suavidad en la interacción.

### 3.2.1.4. La capa Vista Modelo

La última de las capas del patrón arquitectónico Modelo Vista Vista Modelo, se ha implementado haciendo uso de los Architecture Components. En concreto, utiliza las clases ViewModel o AndroidViewModel y la clase LiveData que como ya se comentó con anterioridad, sigue el patrón de diseño observador, para permitir a la vista subscribirse a la fuente de datos y actualizarse automáticamente cuando haya cambios.

La forma de operar de los LiveData consiste en dotar a los objetos de las clases del modelo, de una capa que los convierte en observables por las vistas. A continuación, se muestran ejemplos de cómo funciona la conexión de los ViewModel con las vistas.

```
public class ListaNoticiasViewModel extends ViewModel {  
  
    private ExpoAgendaManager expoAgendaManager;  
  
    public ListaNoticiasViewModel() {  
        super();  
        expoAgendaManager = ExpoAgendaManager.getInstance();  
    }  
  
    // devuelve la lista de noticias  
  
    public MutableLiveData<ArrayList<Noticia>> getListaNoticias() {  
        return expoAgendaManager.getListaNoticias();  
    }  
}
```

Este simple código, solicita a las clases de la capa Repository que obtenga la lista de noticias de la base de datos y la devuelve envuelta en un objeto LiveData, el cual será observado por la vista donde se instancie el ViewModel.

```
// inserta el ViewModel y obtiene la lista de noticias  
  
        listaNoticiasViewModel = ViewModelProviders.of(this)  
        .get(ListaNoticiasViewModel.class);  
  
        listaNoticiasViewModel.getListaNoticias().observe(this,  
new Observer<ArrayList<Noticia>>() {  
  
            @Override  
            public void onChanged(@Nullable ArrayList<Noticia>  
noticias) {  
                inicializaRecyclerView(noticias);  
            }  
        }  
});
```

Este código se inserta en el evento onCreate de la vista, creando una instancia del ViewModel y declarando un observador suscrito a LiveData del ArrayList<Noticia> el cual será devuelto siempre que ocurra un cambio, mediante el evento onChanged(). Finalmente, se observa como este ArrayList se pasa al Adapter del RecyclerView para dotar de datos a la lista.

### 3.2.1.5. La capa Repository

Esta capa no contiene clases propias, se trata solo de una recomendación de Android Developers y su finalidad es abstraer toda la lógica de entrada y salida de datos entre la aplicación y la base de datos, sea local o remota.

De esta manera, los ViewModel se desentienden completamente de la fuente de datos, siendo al final su única función, solicitar la información requerida por las vistas. Esto permite que los cambios que pudieran surgir en las bases de datos, no afecten en absoluto al resto de capas de la aplicación, pues todos los cambios serían realizados en las clases del repositorio.

Estas clases se crean siguiendo el patrón de diseño Singleton y mediante métodos estáticos, solicitan la información a la base de datos. A continuación, se muestra un ejemplo de código.

```
// devuelve la info de la feria

    public static MutableLiveData<Feria> getFeria() {

        feria = new MutableLiveData<>();

        DatabaseReference dataBaseFeria =
firebaseDatabase.getReference().child("/feria");

        dataBaseFeria.addValueEventListener(new
ValueEventListener() {

            @Override
            public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                feria.setValue(dataSnapshot.getValue(Feria.class));
            }

            @Override
            public void onCancelled(@NonNull DatabaseError
databaseError) {
                feria = null;
            }
        });
        return feria;
    }
}
```

Se observa en el código que el método getFeria() devuelve un objeto Feria envuelto en un LiveData. Para ello, crea una referencia a la base de datos Firebase y conecta con el nodo feria, mediante la instrucción .child("/feria"). Esta referencia queda a la espera de datos agregando el listener dataBaseFeria.addValueEventListener().

Como puede comprobarse, el uso combinado de Firebase y de Architecture components, dota al desarrollo Android de gran facilidad y agilidad a la hora de implementar la lógica de las aplicaciones.

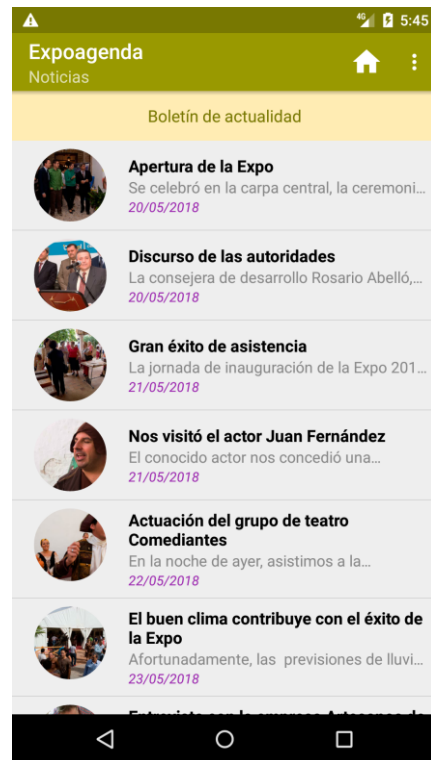


### 3.3 Pruebas de la aplicación

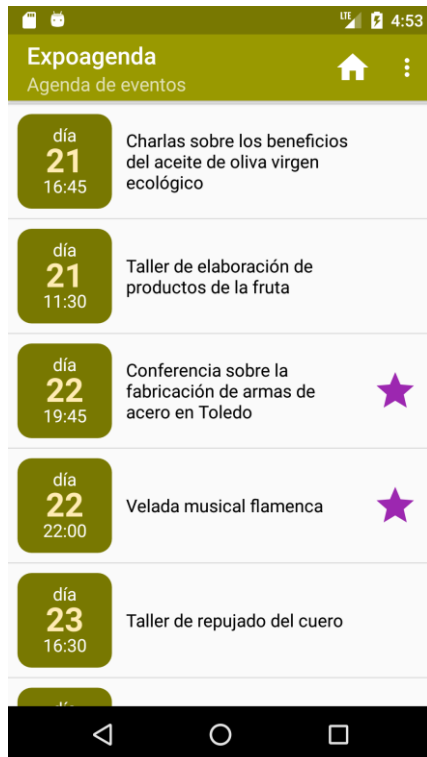
- Versiones del sistema operativo donde se han realizado pruebas



KitKat 4.4



Lollipop 5.1



Marshmallow 6.0

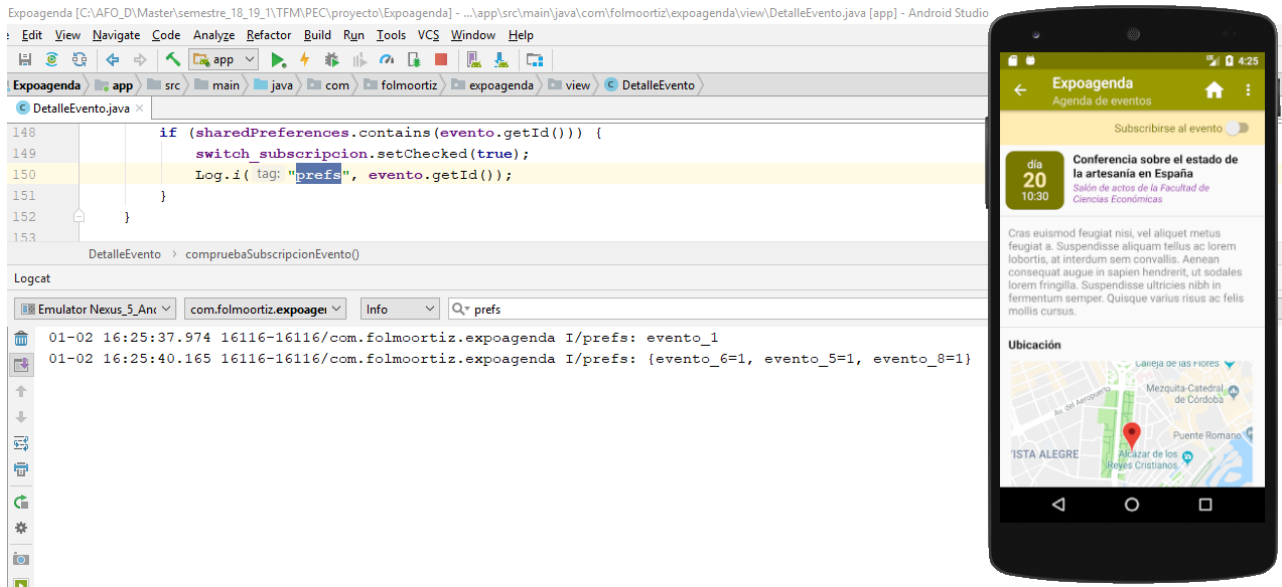


Oreo 8.1

Estas versiones se han probado sobre dispositivos virtuales, a través del AVD de Android Studio 3.2.1

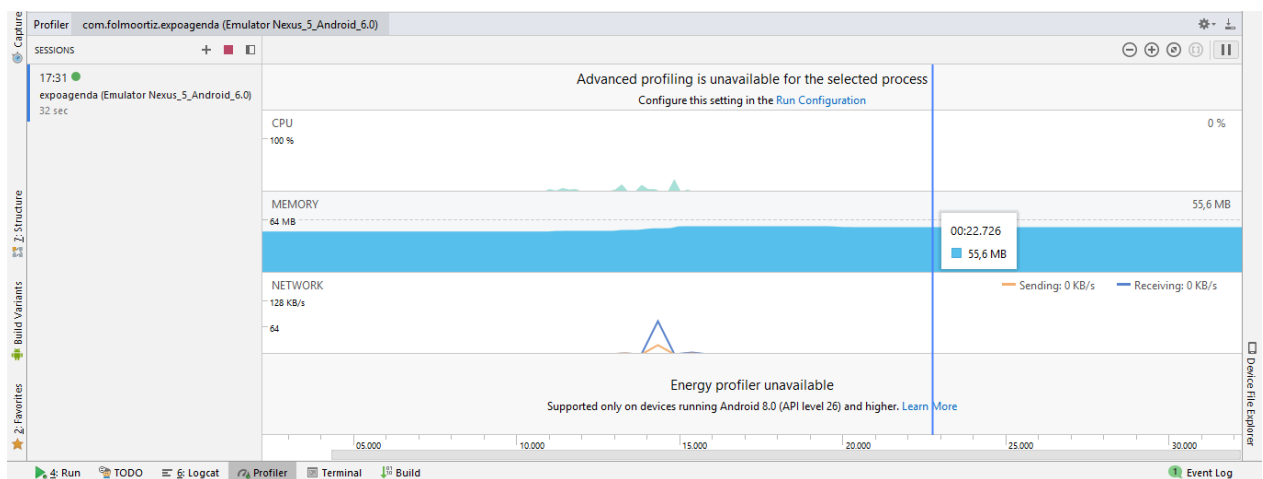
Las pruebas sobre dispositivos reales, se han realizado con terminales con versiones Nougat 7.0 y Pie 9.0

- Uso de la herramienta Logcat



Ejemplo de uso de Logcat, para comprobar la creación y eliminación de subscripción a eventos de la agenda.

- Testeo de funcionamiento con Profiler



Comprobación del rendimiento de la aplicación, midiendo el consumo de CPU, memoria y uso de la red de conexión.

## 4. Conclusiones

La realización del presente trabajo fin de máster ha supuesto todo un reto, ya que tenía como uno de sus objetivos implementar la nueva arquitectura propuesta oficialmente por Google y denominada Architecture components.

Esto ha traído consigo tener que reaprender muchos de los procesos ya adquiridos de proyectos anteriores, basados en el clásico modelo vista controlador. No obstante, el resultado final ha sido muy satisfactorio, al poder comprobar cómo se simplifican procesos anteriormente complejos y difíciles.

Todo ello, unido a la combinación de los Architecture components con las bases de datos en tiempo real de Firebase, supone un conjunto de conocimientos absolutamente actualizados y que permite afrontar con garantía cualquier desarrollo futuro.

En cuanto al conjunto de objetivos planteados en el proyecto, tan solo ha quedado sin realizar el apartado relativo a la publicación de la app en Play Store. Esta decisión se ha tomado a la vista de la posibilidad de agregar algunas utilidades más al proyecto y poder usarlo como base para una futura aplicación comercial. Por tanto, no se ha visto conveniente exponer aún el proyecto al público, hasta implementar los nuevos requisitos funcionales.

Finalmente, el desarrollo del proyecto se ha llevado a cabo de acuerdo a la planificación, sin mayores incidencias. Tan solo destacar que la primera entrega del proyecto no pudo completarse satisfactoriamente, por los imprevistos laborales surgidos y por coincidir con la entrega de trabajos de otras asignaturas del máster.

Aun así, la conclusión acerca de la metodología seguida en el desarrollo del trabajo, es que ha sido la más adecuada dada la naturaleza propia de un trabajo fin de master.

## 5. Bibliografía

### 5.1 Enlaces

- [1] Feria del Mueble de Zaragoza 2018  
Visitado el 29/09/2018:  
<https://play.google.com/store/apps/details?id=es.zimaltec.feriazaragoza.feriamueble2018>
  
- [2] Feria Internacional de la Maquinaria Agrícola FIMA 2018  
Visitado el 29/09/2018:  
<https://play.google.com/store/apps/details?id=es.zimaltec.feriazaragoza.fima2018>
  
- [3] Basquisite  
Visitado el 29/09/2018:  
<https://play.google.com/store/apps/details?id=es.xeria.basquisite>
  
- [4] InteriHotel Madrid  
Visitado el 29/09/2018:  
<https://play.google.com/store/apps/details?id=es.xeria.interihotelmadrid>
  
- [5] Meetmaps. Plataforma de engagement para eventos  
Visitado el 30/09/2018: <https://meetmaps.com>
  
- [6] EvenTwo  
Visitado el 30/09/2018: <https://eventwo.com/es/producto/>
  
- [7] OBS Business School. Metodologías ágiles  
Visitado el 05/10/2018: <https://www.obs-edu.com/es/blog-project-management/scrum/metodologias-agiles-scrum-y-kanban-y-xp>
  
- [8] Web oficial Android Developers  
Visitado el 06/10/2018: <https://developer.android.com>
  
- [10] PEXELS  
Visitado el 12/10/2018: <https://www.pexels.com>
  
- [11] NinjaMock  
Visitado el 19/10/2018: <https://ninjamock.com>
  
- [12] AXURE  
Visitado el 21/10/2018: <https://www.axure.com>
  
- [13] MATERIAL DESIGN  
Visitado el 21/10/2018: <https://material.io/guidelines>
  
- [14] Firebase  
Visitado el 22/10/2018: <https://firebase.google.com/>

- [15] Android Architecture Components  
Visitado el 24/10/2018:  
<https://developer.android.com/topic/libraries/architecture/>
- [16] Java SE Development Kit  
Visitado el 26/11/2018:  
<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- [17] Android Studio  
Visitado el 26/11/2018:  
<https://developer.android.com/studio/#downloads>
- [18] Sistema de control de versiones GIT  
Visitado el 28/11/2018:  
<https://git-scm.com/>

## 5.2 Material docente de la UOC

- [9] **Patricia Gil, E.; de Lera Tatjer, E.; Monjo Palau, A.** (2009).  
*"Usuarios y sistemas interactivos"* (1.ª ed.). Barcelona: FUOC

## 6. Anexos

### Anexo I. Fichero “folmoortiz\_datos\_expoagenda.json”

Este fichero contiene toda la base de datos, preparada para poder importarla desde Firebase, mediante la opción “importar JSON”.

### Anexo II. Fichero “adjunto leeme\_sha1\_release.txt”

Dentro de la carpeta **certificado\_apk\_release** adjunta, se encuentran los ficheros *expoagenda.jks* y *leeme\_sha1\_release.txt*. El primero es un certificado para firmar la aplicación. Toda la información de claves e instalación, se encuentra en el fichero txt.

### Anexo III. API de Google Maps

Para el uso de mapas en la app, se ha creado un proyecto en Google Cloud Platform. Estos son los datos:

- . Proyecto: Expo Agenda
- . ID proyecto: expo-agenda
- . Clave: AlzaSyBT26yi3Ht4GFLb9ykRpEcUrTgEr7p3WXc

Como puede apreciarse en la imagen adjunta, no se han aplicado restricciones de aplicación, para poder usar sin problemas la Clave de API generada.

The screenshot shows the Google Cloud Platform interface for an API key. At the top, there's a navigation bar with 'Google Cloud Platform' and 'Expo Agenda'. Below that, a breadcrumb trail shows 'Clave de API' with options to 'VOLVER A GENERAR CLAVE' or 'ELIMINAR'. The main content area explains that the API key can be used in the project and with any compatible API. It lists the creation date as '30 dic. 2018 18:35:43' and the creator as 'folmoortiz@gmail.com (tú)'. The API key itself is displayed in a text box: 'AlzaSyBT26yi3Ht4GFLb9ykRpEcUrTgEr7p3WXc'. Below the key, the name is set to 'Clave de API 1'. Under the 'Restricciones de clave' section, it states that restrictions are used to prevent unauthorized use and quota theft. It shows 'Restricciones de aplicación: Ninguna' and 'Restricciones de API: Maps SDK for Android'. At the bottom, under 'Restricciones de aplicación', the 'Ninguna' option is selected, with other options like 'URLs de referencia HTTP (sitios web)', 'Direcciones IP (servidores web, tareas cron, etc.)', 'Aplicaciones para Android', and 'Aplicaciones para iOS' being unselected.

#### **Anexo IV.** Implementación de Firebase

Se muestran los datos con los que se ha creado el proyecto Firebase para su uso con la aplicación:

- . Nombre del proyecto: ExpoAgenda
- . ID del proyecto: expoagenda-folmo
- . País: España

En el Anexo I se encuentra la información necesaria para crear la base de datos de este proyecto.

La configuración necesaria para el uso de Firebase, es un proceso guiado paso a paso, a través de la herramienta **Tools > Firebase**, de Android Studio 3.2.1. Desde las opciones Realtime Database y Storage, se indican todas las acciones requeridas para la implementación de ambos servicios.

#### **Anexo V.** Manual de usuario

El uso de la app es extremadamente sencillo, pues se trata de un servicio de consulta de información de una feria de muestras.

El uso e interactividad, es el habitual de cualquier aplicación moderna para Android, mediante menús, iconos y tabs.

Desde el detalle de un evento, el usuari@ podrá suscribirse al mismo, mediante el gadget de interruptor de la esquina superior derecha. Al entrar a la lista de eventos, podrá ver una estrella morada, como indicativo de que ese evento lo marcó como favorito.