



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FINAL DE GRADO

---

Aplicación Android de Eventos y Congresos

---

*Autor:*

FERNANDO GÓMEZ APARICIO

*Supervisor:*

SERGIO AGUADO GONZÁLEZ

*Tutor académico:*

M. ÁNGELES LÓPEZ MALO

Fecha de lectura: 26 de junio de 2017

Curso académico: 2016/2017

## **Resumen**

Este documento explica el proceso de desarrollo de una aplicación móvil para el sistema operativo Android. A partir de una idea y unas capturas de pantalla con la apariencia de la aplicación, se ha realizado el análisis, el diseño, la implementación y las pruebas usando una metodología ágil. El resultado es una aplicación Android totalmente funcional y que cumple todos los requisitos exigidos.

La aplicación se ha creado durante la estancia en prácticas en la empresa *Soluciones Cuatroochenta*, compañía especializada en la programación de aplicaciones para móviles.

Este documento junto con la aplicación constituyen el Trabajo de final de grado (TFG) del Grado de Ingeniería Informática de la *Universitat Jaume I*.

## **Palabras clave**

Android, metodologías ágiles, teléfono inteligente, Firebase, NoSQL.

## **Keywords**

Android, agile methodologies, smartphone, Firebase, NoSQL

# Índice general

1. Introducción.....	11
1.1 Contexto y motivación del proyecto .....	11
1.1.1 Contexto: Soluciones Cuatroochenta .....	11
1.1.2 Motivación.....	12
1.2 Objetivos del proyecto .....	13
2. Descripción del proyecto .....	15
2.1 Descripción general.....	15
2.2 Funcionalidad.....	15
2.3 Alcance .....	16
2.4 Tecnologías .....	16
2.5 Control de versiones .....	18
3. Planificación del proyecto .....	19
3.1 Metodología .....	19
3.1.1 ¿Por qué utilizar una metodología ágil en este proyecto? .....	20
3.1.2 Especificación de la metodología.....	20
3.1.3 Historias de usuario .....	21
3.1.4 Puntos de historia .....	22
3.1.5 Pila del producto.....	22
3.2 Análisis .....	23
3.2.1 Pila del producto inicial.....	23
3.3 Estimación de recursos .....	26
3.4 Estimación de costes .....	27
3.5 Planificación .....	28
3.5.1 Planificación de los sprints.....	28

4. Diseño de software .....	29
4.1 Diseño de datos o clases.....	29
4.2 Diseño de la arquitectura.....	31
4.2.1 Estilo arquitectónico.....	31
4.2.2 Arquitectura del sistema.....	31
4.2.3 Arquitectura del sistema operativo Android.....	32
4.2.4 Ciclo de vida de una Actividad .....	33
4.3 Diseño de la interfaz de usuario.....	34
4.3.1 Interfaz gráfica.....	35
4.3.2 Jerarquía de pantallas .....	36
4.4 Patrones de diseño.....	37
5. Implementación y pruebas.....	39
5.1 Detalles de implementación .....	39
5.1.1 Módulos de software .....	39
5.1.2 Implementación de patrones de diseño.....	40
5.2 Pruebas de aceptación .....	42
6. Desarrollo del proyecto: Sprints .....	43
6.1 Sprint 1 (27/02/2017 a 12/03/2017).....	44
6.1.1 Resumen del sprint .....	44
6.2. Sprint 2 (13/03/2017 a 26/03/2017).....	44
6.2.1 Resumen del sprint .....	44
6.2.2 Pruebas de aceptación .....	46
6.2.3 Resultados .....	47
6.3. Sprint 3 (27/03/2017 a 09/04/2017).....	48
6.3.1 Resumen del sprint .....	48
6.3.2 Pruebas de aceptación .....	52

6.3.3 Resultados .....	55
6.4. Sprint 4 (10/04/2017 a 23/04/2017) .....	55
6.4.1 Resumen del sprint .....	55
6.4.2 Pruebas de aceptación .....	58
6.4.3 Resultados .....	61
6.5. Sprint 5 (24/04/2017 a 07/05/2017) .....	61
6.5.1 Resumen del sprint .....	61
6.5.2 Pruebas de aceptación .....	65
6.5.3 Resultados .....	68
6.6. Sprint 6 (08/05/2017 a 21/05/2017) .....	69
6.6.1 Resumen del sprint .....	69
6.6.2 Pruebas de aceptación .....	72
6.6.3 Resultados .....	74
6.7 Pila final del producto (21/05/2017) .....	75
7. Futuras extensiones .....	77
7.1 Futuras extensiones de la aplicación .....	77
7.2 Otras aplicaciones complementarias .....	77
8. Conclusiones .....	79
8.1 Resumen .....	79
8.2 Objetivos logrados .....	79
8.3 Valoración personal .....	80
Bibliografía .....	81
A. Modelo de datos .....	83
B. Capturas de pantalla .....	87



# Índice de ilustraciones

Ilustración 1: Estadísticas sobre el número de dispositivos en España. Fuente: WE ARE SOCIAL LTD [2] .....	12
Ilustración 2: Historia de usuario.....	22
Ilustración 3: Diagrama de clases.....	30
Ilustración 4: Organización de la tabla “Maps” en Firebase (NoSQL).....	30
Ilustración 5: Arquitectura del sistema.....	32
Ilustración 6: Arquitectura del sistema operativo Android. Fuente: © Google [5].....	34
Ilustración 7: Ciclo de vida de una Actividad Android. Fuente: © Google [6] .....	35
Ilustración 8: Jerarquía de pantallas .....	36
Ilustración 9: Patrón MVC .....	37
Ilustración 10: Patrón Singleton. Fuente: Wikipedia [8].....	38
Ilustración 11: Comunicaciones con Callbacks.....	38
Ilustración 12: Pantallas hechas en verde y empezadas en amarillo hasta el Sprint 2 ...	48
Ilustración 13: Pantallas hechas en verde y empezadas en amarillo hasta el Sprint 3 ...	55
Ilustración 14: Pantallas hechas en verde y empezadas en amarillo hasta el Sprint 4 ...	61
Ilustración 15: Pantallas hechas en verde y empezadas en amarillo hasta el Sprint 5 ...	69
Ilustración 16: Pantallas hechas en verde en el Sprint 6 .....	76
Ilustración 17: Página principal.....	87
Ilustración 18: Menú sin iniciar sesión.....	88
Ilustración 19: Formulario para iniciar sesión.....	89
Ilustración 20: Menú con sesión iniciada .....	90
Ilustración 21: Información de la inscripción.....	91
Ilustración 22: Formulario de la inscripción, parte 1.....	92
Ilustración 23: Formulario de la inscripción, parte 2.....	93
Ilustración 24: Confirmación de la inscripción .....	94

Ilustración 25: Sección de noticias .....	95
Ilustración 26: Sección de ponentes .....	96
Ilustración 27: Información del ponente.....	97
Ilustración 28: Agenda con calendario .....	98
Ilustración 29: Agenda sin calendario .....	99
Ilustración 30: Información de un acto.....	100
Ilustración 31: Lista de documentos con información.....	101
Ilustración 32: Ejemplo PDF personalizado .....	102
Ilustración 33: Lista de los lugares importantes del evento.....	103
Ilustración 34: Mapa con los servicios de la ciudad.....	104
Ilustración 35: Patrocinadores .....	105
Ilustración 36: Información de la empresa .....	106
Ilustración 37: Lista de notificaciones.....	107
Ilustración 38: Interacción con otro usuario .....	108
Ilustración 39: Lista de contactos .....	109
Ilustración 40: Chat privado .....	110
Ilustración 41: Lista de preguntas al ponente .....	111
Ilustración 42: Preguntar al ponente.....	112
Ilustración 43: Perfil .....	113
Ilustración 44: Acreditación .....	114
Ilustración 45: Vista de una pregunta con comentarios.....	115
Ilustración 46: Menú de un ponente .....	116
Ilustración 47: Lista de preguntas de un ponente .....	117
Ilustración 48: Pantalla de realidad aumentada .....	118



# Índice de tablas

Tabla 1: Pila del producto inicial.....	26
Tabla 2: Costes del proyecto .....	28
Tabla 3: Pila del producto inicial.....	45
Tabla 4: Pila del producto al inicio del Sprint 3 .....	49
Tabla 5: Pila del producto al inicio del Sprint 4 .....	56
Tabla 6: Pila del producto al inicio del Sprint 5 .....	63
Tabla 7: Pila del producto al inicio del Sprint 6 .....	70
Tabla 8: Pila del producto al finalizar el proyecto.....	76



# Capítulo 1

## Introducción

### Índice

---

1.1. Contexto y motivación del proyecto .....	11
1.1.1 Contexto: Soluciones Cuatroochenta .....	11
1.1.2 Motivación .....	12
1.2. Objetivos del proyecto .....	13

---

En este documento se explica en profundidad el proceso de creación de la aplicación **Hackathon** en la empresa *Soluciones Cuatroochenta*. Se explica en detalle y con el siguiente orden: descripción de la empresa en la que se realiza, descripción detallada del programa a implementar, cómo se ha planificado este desarrollo y mediante qué metodologías, las tecnologías utilizadas, el diseño del sistema, la implementación y los resultados finales.

## 1.1 Contexto y motivación del proyecto

### 1.1.1 Contexto: Soluciones Cuatroochenta

La empresa en la cual se realizan las prácticas externas es Soluciones Cuatroochenta S.L. Esta organización tiene su sede central en la Universidad Jaume I de Castellón de la Plana, concretamente en el edificio Espaitec II. El carácter de esta compañía, a pesar de tener departamento de marketing y periodismo, es mayoritariamente informático. Concretamente se dedica al desarrollo de aplicaciones móviles tanto para Android como para iOS. Fue fundada en noviembre de 2011 por Alfredo R. Cebrián, licenciado en Publicidad y Relaciones Públicas y CEO, y Sergio Aguado, ingeniero superior en Informática y CTO. Éste último será quien supervise la estancia en prácticas. Como curiosidad de la empresa, su nombre se debe a la resolución de la pantalla del primer iPhone que salió al mercado en 2007 (480x320) [1].

La empresa está formada por 3 departamentos: el departamento administrativo, el comercial y el informático. El primero trata los aspectos legales y de administración de

la empresa. El segundo se ocupa del marketing, los artículos periodísticos y las relaciones con los clientes. El tercero es el que contiene a la mayoría de empleados de la empresa. Se divide en diseño de interfaces de las aplicaciones, iOS, web y Android. El alumno realiza las prácticas en el subdepartamento de Android, formado por unas 15 personas.

### 1.1.2 Motivación

La labor del alumno se limita únicamente a la parte informática de la empresa. La aplicación a desarrollar no será usada por la empresa. El único fin de ésta es que el alumno aprenda una nueva tecnología. Por lo tanto, aunque el programa no será usado por la empresa, el beneficio que obtiene ésta es la formación del alumno en la programación para dispositivos Android.

El aprendizaje de la programación enfocada a dispositivos móviles es un tema de capital importancia hoy en día. Una gran parte de la población mundial posee un smartphone, han superado a los ordenadores portátiles y se sobrepasa. Por lo tanto, el público al que va dirigido las aplicaciones móviles es realmente amplio. Esto crea un mercado enorme con infinidad de posibilidades. En la ilustración 1 se pueden ver las estadísticas sobre diferentes dispositivos tecnológicos en España.

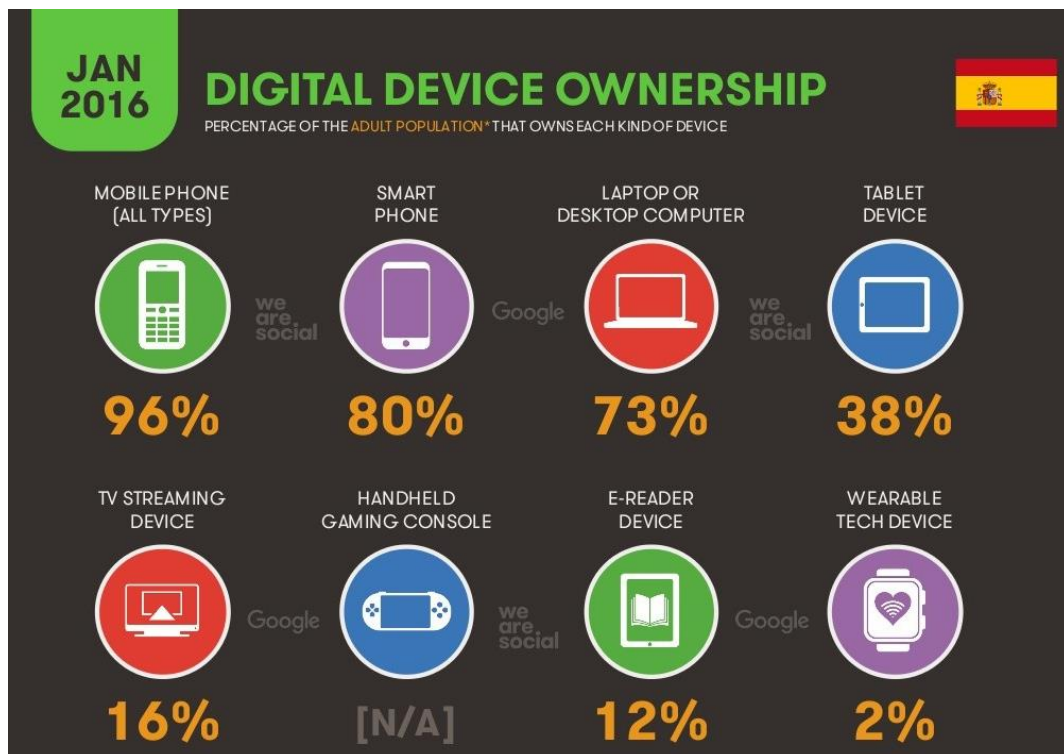


Ilustración 1: Estadísticas sobre el número de dispositivos en España. Fuente: WE ARE SOCIAL LTD [2]

Como se puede apreciar, el 80% de la población adulta en España posee un teléfono inteligente. Ya supera en un 7% a los ordenadores en un periodo de tiempo mucho más corto. Hay que tener en cuenta que en los años noventa ya se podía adquirir un ordenador a un precio razonable en España, hace casi treinta años. Sin embargo, el primer dispositivo Android se comercializó en 2008. En menos de diez años han superado a los ordenadores y la tendencia apunta a que esta diferencia seguirá creciendo.

Finalmente, otro dato de capital importancia en la motivación por la programación Android es la cuota de mercado. Se hizo público [3] que en el primer trimestre de 2013 el 92% de los smartphones vendidos en España eran Android, seguido de iOS con un 4,4%. La diferencia es abismal.

## 1.2 Objetivos del proyecto

El proyecto consiste en el desarrollo de una aplicación Android para la gestión de eventos y congresos. La aplicación debe funcionar sobre Android 4.0 y superiores. El diseño es proporcionado por Cuatroochenta.

La aplicación a desarrollar tiene como finalidad satisfacer diversas necesidades y facilitar algunos procesos. El objetivo principal es promocionar un evento (el Hackathon por ejemplo) para darlo a conocer y que los usuarios puedan acceder a la información sobre éste e interactuar con los demás participantes. Estos objetivos se pueden desglosar en los siguientes subobjetivos:

- Que el usuario sea capaz de consultar información básica sobre el evento.
- Que el usuario pueda consultar un calendario con los diferentes actos.
- Que el usuario pueda interactuar con los demás participantes y con los ponentes.
- Que el usuario se entere de las últimas noticias y cambios del evento.
- Notificar a un usuario cualquier cambio.
- Poder consultar cualquier duda que los posibles participantes tengan.
- Hacer posible que un usuario se registre en el evento y obtenga una acreditación mediante esta aplicación.

Además, el proyecto también tiene como objetivo que el estudiante adquiera una serie de conocimientos y competencias:

- Aprendizaje de la programación Android.
- Adquirir experiencia real con metodologías ágiles.
- Familiarizarse con el trabajo en equipo en el ámbito profesional.

## Capítulo 2

# Descripción del proyecto

### Índice

---

2.1. Descripción general .....	15
2.2. Funcionalidad .....	15
2.3. Alcance .....	16
2.4. Tecnologías .....	16
2.5. Control de versiones .....	18

---

### 2.1 Descripción general

La empresa *Soluciones Cuatroochenta* quiere desarrollar una aplicación que sirva para gestionar eventos y congresos. Dicho evento será el Hackathon que se realizó en Castellón, pero esta aplicación deberá poder modificarse fácilmente para adaptarse a cualquier evento con unas características similares. El programa informático servirá para que los usuarios puedan consultar información sobre el evento en cuestión. Además, podrán interactuar entre ellos.

### 2.2 Funcionalidad

La aplicación que se va a desarrollar tendrá un solo rol, el de usuario. La información que se mostrará depende de si está registrado o no. A continuación se especifica qué puede ver y qué no un usuario.

La aplicación debe permitir a todos los usuarios:

- Ver la página principal del evento, con la información básica de éste.
- Acceder a la sección con las últimas noticias del evento.
- Consultar estas noticias.
- Consultar el calendario del evento.
- Para cada acto del calendario se puede consultar información.
- Ver la lista de ponentes que asistirán al evento.

- Consultar información sobre cada uno de estos ponentes.
- Obtener los documentos del evento: inscripción, calendario e información.
- Consultar en un mapa dónde se encuentran las sedes.
- Ver los servicios cercanos disponibles (hoteles, restaurantes, ...).
- Consultar notificaciones sobre cambios.
- Ver los patrocinadores del evento.
- Consultar los aspectos legales del evento.

La aplicación debe permitir a todos los usuarios que no se han registrado:

- Inscribirse en el evento.

La aplicación debe permitir a todos los usuarios inscritos:

- Iniciar sesión y cerrar sesión.
- Consultar y modificar el perfil.
- Realizar preguntas a los ponentes.
- Interactuar con otros participantes.
- Ver su acreditación para el evento.

## 2.3 Alcance

La aplicación propuesta es la del Hackathon pero podría adaptarse a cualquier otro evento o congreso similar. El proyecto se limita a la parte del cliente, es decir, al usuario de la aplicación para que consulte información e interactúe con otros usuarios. La parte del administrador que es la que proporciona la información no está previsto que sea implementada por el alumno. Tampoco se implementa la parte del ponente. Este tipo de usuario es un invitado al evento para dar una conferencia que, mediante la aplicación, puede contestar a las preguntas que le formulen los usuarios.

## 2.4 Tecnologías

El IDE o entorno de programación será AndroidStudio. Es una herramienta que proporciona Google y que es totalmente gratuita. Está basado en IntelliJ por lo que es casi igual. AndroidStudio ofrece muchas ayudas para programar una aplicación Android como



puede ser una lista de autocompletado cuando se intenta referenciar un elemento de la interfaz gráfica (un botón por ejemplo) desde el código Java.

El lenguaje de programación principal es Java. Esto es una gran ventaja ya que es el lenguaje vehicular de la carrera y, tras tres años y medio, los conocimientos adquiridos son amplios. Por lo tanto, el lenguaje no es un problema pero hay que aprender a usar las librerías y API que Android ofrece para poder crear una aplicación completa. Los ficheros que contienen la interfaz de usuario están en formato XML. Se pueden editar de manera gráfica arrastrando botones y otros elementos. También se puede editar los ficheros directamente como ficheros de texto, esto requiere más conocimientos pero el resultado será más cercano a lo que el programador quiere.

En un principio se proponía que, en caso de tener tiempo de sobra, se crearía un backend mediante NodeJS. Teniendo en cuenta que el aprendizaje de Android ocupa casi un 25% del tiempo de las prácticas, la implementación de esta parte opcional de la aplicación parecía casi imposible. Sin embargo, el debate sobre qué base de datos utilizar nos dio la solución a este problema, usar Firebase. Esta plataforma permite el tratamiento de datos en tiempo real. Es decir, si un usuario inserta algún dato en la base de datos, los demás usuarios lo verán instantáneamente. Todo esto sin necesidad de programar un backend.

Sin embargo, en la empresa hacen hincapié en usar notificaciones PUSH. Esta función de Firebase permite notificar a un dispositivo concreto mediante un token único que se genera automáticamente, incluso si la aplicación no está abierta. Para poder lanzar dichas notificaciones es necesario un pequeño backend hecho con Node. Firebase proporciona una función llamada “Functions” que hace las veces de backend, simplemente hay que programar las funciones que se necesiten en JavaScript. El funcionamiento consiste en detectar los cambios en la base de datos y tratarlos para determinar qué tipo de notificación enviar y a quién o quiénes. Por ejemplo, una función detecta si se añade o modifica algún dato en la tabla “Notifications” y envía una notificación a todos los dispositivos que usan la aplicación (se guarda una lista con los tokens mencionados previamente).

Hay que tener en cuenta que aunque Firebase simplifique bastante el tratamiento de la información, es necesario adquirir los conceptos de bases de datos NoSQL, así como

el uso de la propia herramienta para consultar, modificar o eliminar información de la base de datos. La organización de la información tiene la misma estructura que un fichero JSON.

El almacenamiento de los datos no está orientado a relacionar las tablas mediante un JOIN, se prima la velocidad de las consultas. Por lo tanto, no se hacen consultas de varias tablas y esto hace necesario que se repita información en la base de datos. Por ejemplo, un usuario tiene una imagen de perfil. Si cuando éste comenta una publicación queremos que se muestre su imagen al lado del comentario, la imagen se guarda en la tabla usuarios y en la tabla comentarios.

Además, todas las operaciones son asíncronas. No se puede leer instantáneamente un valor y devolverlo, se escribe el correspondiente código para realizar una lectura y se le asigna un listener que será llamado cuando la operación haya terminado. Por lo tanto, si cierta información es necesaria antes de cargar una pantalla, hay que implementar un callback para que dicho listener ejecute una llamada al código que necesita dicha información para avisarle de que ya está disponible en local.

## 2.5 Control de versiones

En cuanto al control de versiones se ha usado Git. A pesar de que las operaciones se pueden realizar desde la línea de comandos, Android Studio incorpora una herramienta gráfica (la misma que IntelliJ) que hace mucho más fáciles las actualizaciones del código. En un proyecto informático es esencial el uso de control de versiones para poder volver a versiones anteriores, tener el proyecto guardado de forma segura y poder acceder a él desde cualquier ordenador.

## Capítulo 3

# Planificación del proyecto

### Índice

---

3.1. Metodología .....	19
3.1.1 ¿Por qué utilizar una metodología ágil en este proyecto? .....	20
3.1.2 Especificación de la metodología .....	20
3.1.3 Historias de usuario .....	21
3.1.4 Puntos de historia .....	22
3.1.5 Pila del producto .....	22
3.2. Análisis .....	23
3.2.1 Pila del producto inicial .....	23
3.3. Estimación de recursos .....	26
3.4. Estimación de costes .....	27
3.5. Planificación .....	28
3.5.1 Planificación de los sprints .....	28

---

## 3.1 Metodología

A lo largo de los cuatro años del grado se imparten diversas asignaturas que explican teórica y prácticamente cómo funciona la planificación de un proyecto software. Se han enseñado metodologías tradicionales y modernas. A partir de estos conocimientos se pueden comparar y determinar qué beneficios y qué inconvenientes aportan a un determinado tipo de proyecto informático.

Las metodologías tradicionales intentan detallar todos los aspectos del proyecto, planificarlo estrictamente de principio a fin. Por lo tanto, el desarrollo no está enfocado a posibles modificaciones durante la implementación. Otra desventaja es que no se consigue un producto que enseñar al cliente hasta las fases finales y no se obtiene retroalimentación de éste.

Por el contrario, todo lo anterior lo hacen posible las metodologías ágiles. Son muy flexibles, permiten añadir o eliminar funcionalidades durante el desarrollo así como completar una tarea en otro periodo al planificado inicialmente. Además, se obtiene un producto funcional desde las primeras fases.

### 3.1.1 ¿Por qué utilizar una metodología ágil en este proyecto?

El motivo principal de la elección de metodologías ágiles es el tipo de proyecto que se desea implementar. Es muy recomendable usarlas en proyectos de tamaño medio como éste. Además, la empresa *Cuatroochenta* tiene experiencia con esta metodología y la usa en todos sus proyectos por lo que cualquier problema o duda que surja podrá ser resuelto rápidamente.

Como se menciona en el apartado anterior, el software es muy cambiante y difícil de estimar. La flexibilidad en la planificación es un aspecto muy importante en este proyecto. Es de vital importancia poder añadir funcionalidades a lo largo de la implementación ya que es muy fácil olvidarse alguna en el análisis inicial.

Otra razón de peso en la elección de estas metodologías es que son muy adecuadas cuando el equipo de desarrollo no está familiarizado con las tecnologías que va a utilizar. A pesar de conocer Java, he tenido que aprender Android, Firebase y NoSQL. Por lo tanto, es mejor usar estas técnicas.

Teniendo en cuenta estas características, el uso de estas metodologías nos reportará los siguientes beneficios:

- Realizar cambios durante el desarrollo del proyecto sin causar problemas.
- Obtener un producto funcional desde el primer sprint.
- Como se tiene un producto, se puede obtener retroalimentación del cliente.
- Conocer en todo momento el estado en el que se encuentra el proyecto.

### 3.1.2 Especificación de la metodología

Hay una gran diversidad de metodologías ágiles (Extreme Programming o XP, Crystal Clear, ...). Teniendo en cuenta las características de este proyecto se ha elegido Scrum, aunque no se aplicarán todas sus características estrictamente. Los motivos de la elección de esta metodología son:

- Flexibilidad frente a los cambios.
- El proyecto es de un tamaño medio.
- Reducción del Time to Market, el cliente puede ver y utilizar ciertas funciones del producto antes de que esté terminado.
- Predicción de tiempos. Teniendo en cuenta los sprints completados se puede ver la velocidad media y, por lo tanto, hacer mejores predicciones del tiempo que costará implementar tareas.
- La empresa conoce bien la metodología.

### 3.1.3 Historias de usuario

Las metodologías ágiles utilizan las historias de usuario para realizar la especificación de los requisitos del sistema. Cada una de ellas describe una funcionalidad que el programa debe implementar de acuerdo a las necesidades del cliente. Hay que tener en cuenta que las metodologías tradicionales usan casos de uso mientras que las ágiles se basan en estas historias de usuario.

Una historia de usuario está compuesta de:

- Id: número que la identifica.
- Título: breve descripción de la historia.
- Descripción: explicación de la funcionalidad que debe implementar el programa en lenguaje del usuario. Sigue la siguiente estructura: **Como [rol], quiero [objetivo], para [beneficio].**
- Prioridad: valor que le da el cliente a esta parte de la aplicación de acuerdo al valor que dicha historia aporta al sistema.
- Puntos de historia: estimación del tiempo que costará completar la tarea.

En la ilustración 2 se muestra un ejemplo de historia de usuario con la información descrita previamente.

ID	1	Inscribirse en el evento	TÍTULO
		Como usuario quiero registrarme como participante para poder participar en el evento.	DESCRIPCIÓN
ESTIMACIÓN		Estimación: 2 puntos de historia.	Valor: 90. PRIORIDAD

Ilustración 2: Historia de usuario.

### 3.1.4 Puntos de historia

Las metodologías ágiles han introducido muchos cambios y conceptos nuevos. Los puntos de historia surgieron como una manera de dimensionar y relacionar la complejidad de las historias de usuario con respecto a otras. En vez de asignar una estimación en horas a una historia de usuario, se le asignan unos puntos de historia.

En este proyecto se ha estimado que un punto de historia equivale a un día de trabajo. Es decir, unas 5 horas de trabajo.

### 3.1.5 Pila del producto

A partir de las historias de usuario y los puntos de historia se puede generar la pila del producto.

La pila del producto es una lista ordenada de todo aquello que el propietario del producto cree que necesita este producto. Enumera todo aquello que los usuarios esperan de la aplicación, es una lista de las funcionalidades que se deben implementar. Al inicio del proyecto se ha de crear una pila del producto inicial pero esta puede ser modificada varias veces durante el desarrollo del proyecto de acuerdo a los cambios que surjan.

Para poder construir la pila del producto es necesario realizar una serie de acciones previamente:

- Crear una lista con los requisitos a partir de una reunión con el cliente. Hay que tener en cuenta que esta parte no se realiza ya que la empresa lo ha hecho previamente y da esta información al alumno.

- A partir de esta lista, crear las historias de usuario (en este caso de ha utilizado Jira).
- A cada historia de usuario asignarle sus puntos de historia.
- Finalmente, el cliente le asigna una prioridad a cada historia conjuntamente con el equipo de desarrollo.

## 3.2 Análisis

### 3.2.1 Pila del producto inicial

Tras realizar los pasos mencionados en el punto anterior, se crea la pila del producto inicial. En este proyecto, un punto de historia equivale a 5 horas de trabajo y la prioridad que se le asigna a cada historia va de 0 a 100. En la tabla 1 se muestra la pila inicial del producto ordenada por el valor de la prioridad de las historias de usuario.

<b>PILA DEL PRODUCTO INICIAL</b>	
<b>ID</b>	<b>HISTORIAS DE USUARIO</b>
1	<b>Inscribirse en el evento</b>
	Como usuario quiero registrarme como participante para poder participar en el evento.
	Estimación: 2 puntos de historia. Valor: 90.
2	<b>Menú para navegar por la aplicación</b>
	Como usuario quiero poder acceder a cualquier sección mediante un menú intuitivo para poder consultar toda la información y realizar todas las acciones deseadas.
	Estimación: 2 puntos de historia. Valor: 80.
3	<b>Interactuar con otros usuarios y tenerlos en tu lista de contactos</b>

	Como participante quiero poder interactuar con otros participantes para poder conocer a gente que participe en el evento.		
	Estimación: <b>8</b> puntos de historia.	Valor: <b>80</b> .	
<b>4</b>	<b>Consultar la agenda del evento</b>		
	Como usuario quiero poder acceder a una agenda con la planificación temporal de todos los eventos para poder asistir a ellos.		
	Estimación: <b>4</b> puntos de historia.	Valor: <b>70</b> .	
<b>5</b>	<b>Ver información sobre los ponentes del evento y hacerles preguntas</b>		
	Como participante quiero poder ver información sobre cada ponente y hacer preguntas para resolver mis dudas.		
	Estimación: <b>5</b> puntos de historia.	Valor: <b>70</b> .	
<b>6</b>	<b>Mapa con los servicios de la ciudad</b>		
	Como usuario quiero poder acceder a un mapa que tenga indicados los servicios más importantes de la ciudad (restaurantes, hoteles, ...) para poder saber dónde están.		
	Estimación: <b>3</b> puntos de historia.	Valor: <b>60</b> .	
<b>7</b>	<b>Mapa con los sitios importantes del evento</b>		
	Como usuario quiero poder acceder a un mapa que tenga indicados todos los sitios importantes del evento para poder saber dónde se realizan los actos.		
	Estimación: <b>1</b> punto de historia.	Valor: <b>60</b> .	
<b>8</b>	<b>Últimas noticias del evento</b>		
	Como usuario quiero poder acceder a una sección con las últimas noticias sobre el evento para poder estar informado de las novedades.		
	Estimación: <b>3</b> puntos de historia.	Valor: <b>50</b> .	



9	<b>Notificaciones</b>	
	Como participante quiero que se me notifique cualquier cambio en el evento o cualquier interacción de otro participante conmigo para poder estar informado instantáneamente.	
	Estimación: <b>2</b> puntos de historia.	Valor: <b>40</b> .
10	<b>Acreditación</b>	
	Como participante quiero tener una acreditación única para poder acceder a todas las actividades del evento.	
	Estimación: <b>1</b> un punto de historia.	Valor: <b>40</b> .
11	<b>Documentos informativos</b>	
	Como usuario quiero ser capaz de descargar los documentos con la información del evento en formato PDF para conocer todos los detalles.	
	Estimación: <b>2</b> puntos de historia.	Valor: <b>40</b> .
12	<b>Acceder perfil</b>	
	Como participante quiero poder ver mi perfil y modificarlo para actualizar cualquier cambio o corregir posibles errores.	
	Estimación: <b>2</b> puntos de historia.	Valor: <b>40</b> .
13	<b>Modificar perfil</b>	
	Como participante quiero poder ver mi perfil y modificarlo para actualizar cualquier cambio o corregir posibles errores.	
	Estimación: <b>2</b> puntos de historia.	Valor: <b>40</b> .
14	<b>Información sobre la inscripción</b>	
	Como usuario quiero poder acceder a la información de la inscripción para valorar si quiero inscribirme o no.	

	Estimación: <b>2</b> puntos de historia.	Valor: <b>30</b> .
15	<b>Información legal</b>	
	Como usuario quiero ser capaz de consultar la información legal del evento para conocer todas sus características.	
	Estimación: <b>1</b> punto de historia.	Valor: <b>20</b> .
16	<b>Patrocinadores</b>	
	Como usuario quiero poder ver todos los patrocinadores del evento.	
	Estimación: <b>1</b> punto de historia.	Valor: <b>20</b> .

*Tabla 1: Pila del producto inicial*

En total el proyecto consta de 40 puntos de historia. Teniendo en cuenta que un punto de historia equivale a un día (5 horas) y que las prácticas son 60 días en total (60 días x 5 horas/día = 300 horas), sobran 20 días. Éstos son empleados en las tres primeras semanas que se dedicaron al aprendizaje de Android. Como ya se ha mencionado anteriormente, este proyecto utiliza metodologías ágiles. Por lo tanto, esta pila del producto no es definitiva y se modificará durante el desarrollo.

### 3.3 Estimación de recursos

A continuación se muestran los recursos necesarios para la elaboración del proyecto divididos en apartados según sean software, hardware o humanos.

- Humanos:
  - El alumno (Fernando Gómez Aparicio), como programador, analista y tester.
  - El responsable y supervisor del alumno en la empresa para ayudar con cualquier problema que surja (Sergio Aguado González).
- Hardware:
  - Ordenador portátil Lenovo G580 (propiedad del alumno).
  - Auriculares para poder escuchar los vídeos (propiedad del alumno).

- Bq Aquaris M5, móvil en el que ejecutar las aplicaciones (propiedad del alumno).
  - Ratón y teclado (propiedad de la empresa).
  - Pantalla adicional para poder programar en una y ver otra información a la vez en la otra (propiedad de la empresa).
- Software:
- Windows 10, el sistema operativo del ordenador del alumno.
  - Android Studio, el entorno de programación (gratuito).
  - Java: JRE(Java Runtime Environment) y JDK (Java Development Kit).
  - Udemy.com, una plataforma web educativa en la que están los tutoriales necesarios para poder aprender a programar en Android.
  - JavaScript, NodeJS, XML, Firebase.

### 3.4 Estimación de costes

Los costes pueden ser calculados a partir de las horas empleadas en el proyecto. La primera tarea es el aprendizaje que consta de 70 horas. Después, está la definición de la pila del producto que cuesta 1 día según el horario de prácticas, es decir, 5 horas. A esta parte de análisis hay que sumar que la planificación de cada sprint supone una hora para cada uno de ellos. Las demás horas se invertirán en la programación del proyecto hasta completar las 300. En la Tabla 2 se desglosan todos los gastos detalladamente.

Concepto	Cálculo	Total
Ordenador	1 x 700 €	700 €
Móvil	1 x 240 €	240 €
Ratón y teclado	5 € + 20 €	25 €
Auriculares	1 x 3 €	3 €
Pantalla	1 x 40 €	40 €
Curso de aprendizaje Android	1 x 20 €	20 €

Windows 10	135 €	135 €
RRHH: aprendizaje	70 h x 10 €/h	700 €
RRHH: análisis	11 h x 25 €/h	275 €
RRHH: programación	219 h x 20 €/h	4.380 €
	Coste total €	6.518 €

Tabla 2: Costes del proyecto

## 3.5 Planificación

### 3.5.1 Planificación de los sprints

El uso de Scrum implica que el desarrollo del proyecto se dividirá en sprints quincenales (es el periodo de tiempo más adecuado). Para cada sprint se establecerá una lista de tareas a completar en este periodo. Se debe tener en cuenta la estimación temporal (puntos de historia) de cada una de éstas para poder determinar cuántas se deben asignar a un sprint. Las prácticas son 300 horas, a 25 horas por semana son 12 semanas. Por lo tanto, si los sprints son quincenales el proyecto constará de 6. A continuación se muestra una lista con estos sprints:

- Primer sprint (del 27 de febrero al 12 de marzo).
- Segundo sprint (del 13 de marzo al 26 de marzo).
- Tercer sprint (del 27 de marzo al 9 de abril).
- Cuarto sprint (del 10 de abril al 23 de abril).
- Quinto sprint (del 24 de abril al 7 de mayo).
- Sexto sprint (del 8 de mayo al 21 de mayo).

## Capítulo 4

# Diseño de software

### Índice

---

4.1. Diseño de datos o clases .....	29
4.2. Diseño de la arquitectura .....	31
4.2.1 Estilo arquitectónico .....	31
4.2.2 Arquitectura del sistema .....	31
4.2.3 Arquitectura del sistema operativo Android .....	32
4.2.4 Ciclo de vida de una Actividad .....	33
4.3. Diseño de la interfaz de usuario .....	34
4.3.1 Interfaz gráfica .....	35
4.3.2 Jerarquía de pantallas .....	36
4.4. Patrones de diseño .....	37

---

Al comienzo de un proyecto en el que se usan metodologías ágiles es necesario elaborar un diseño inicial básico y simple. A lo largo del desarrollo del proyecto, dicho diseño será refinado, mejorado y se hará más complejo hasta convertirse en el diseño final. A continuación se explica el diseño final del proyecto a todos los niveles de detalle.

### 4.1 Diseño de datos o clases

Este diseño pretende mostrar las entidades que serán necesarias y cómo se relacionarán entre ellas para que la implementación de la aplicación satisfaga todas las necesidades del cliente. En la ilustración 3 se muestra el diagrama de clases.

El diagrama no representa la estructura de la base de datos. Hay que recordar que se ha usado una base de datos NoSQL y la información se almacena de una forma bastante diferente. Las principales diferencias son:

- No hay relaciones entre las tablas, se prima la velocidad de las consultas.
- Se repite la información si es necesario.
- Gran facilidad de modificación, se pueden modificar las columnas de las tablas sin que afecte a las otras.

La ilustración 4 es un ejemplo de cómo está organizada la información en Firebase. Es igual que un fichero JSON. En el Anexo A se muestra el modelo de datos.

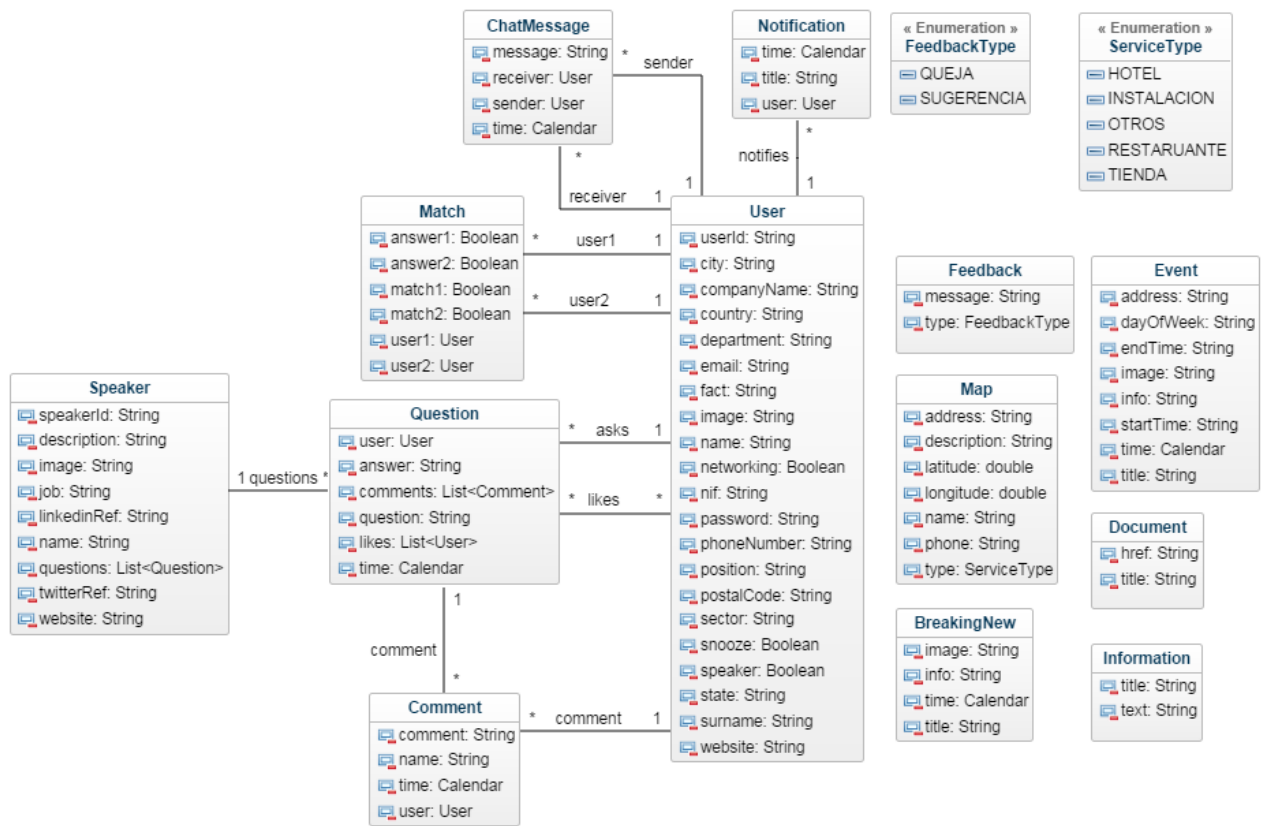


Ilustración 3: Diagrama de clases

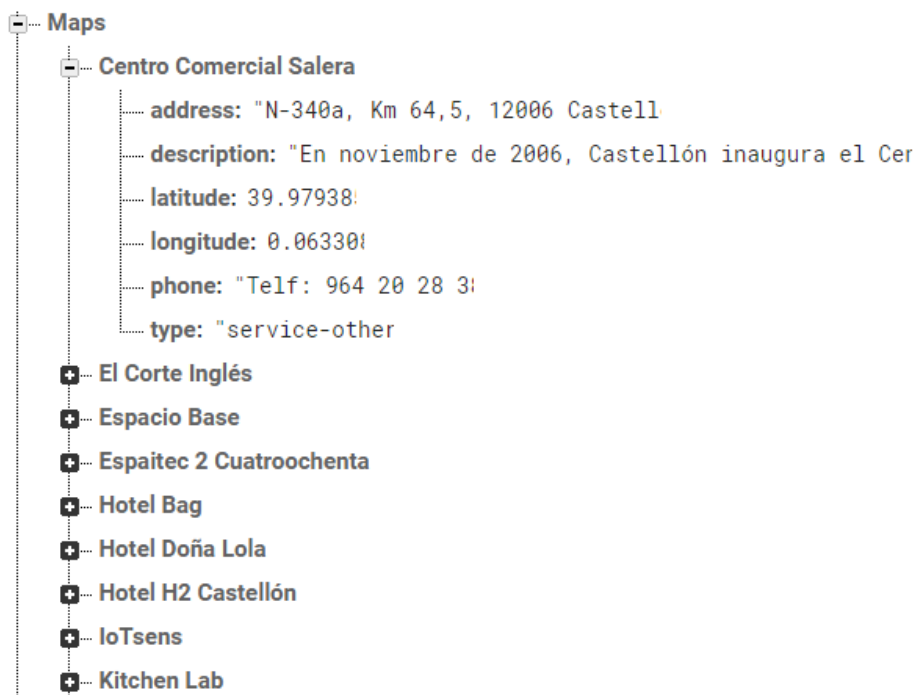


Ilustración 4: Organización de la tabla "Maps" en Firebase (NoSQL)

## 4.2 Diseño de la arquitectura

La arquitectura del software determina la estructura, funcionamiento e interacción entre las partes del sistema informático. Por lo tanto, la arquitectura del software es el diseño de más alto nivel de la estructura de un sistema [4].

Por consiguiente, un correcto diseño de la arquitectura es de capital importancia para que el proyecto cumpla las expectativas con éxito.

### 4.2.1 Estilo arquitectónico

Este proyecto es el resultado de la combinación de diversos estilos arquitectónicos, al igual que la gran mayoría de los sistemas informáticos.

En concreto, los estilos arquitectónicos que componen este sistema son tres: arquitectura orientada a objetos, arquitectura centrada en los datos y arquitectura cliente-servidor.

La arquitectura orientada a objetos está formada por componentes que encapsulan la información para poder tratarla y manipularla. La arquitectura centrada en los datos está construida alrededor de un almacén de datos, los clientes operan independientemente sobre este almacén que es el que permite que se comuniquen.

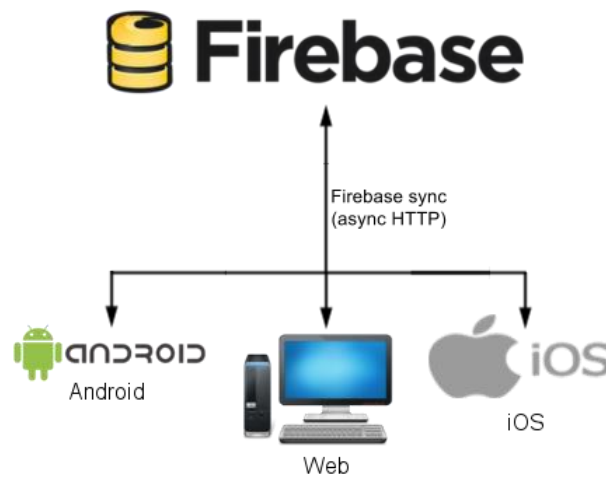
Ambas arquitecturas se combinan y complementan para formar el sistema informático. El resultado es un grupo de componentes construidos alrededor de un almacén de datos. Cada componente encapsula la información y ofrece operaciones para tratarla.

Finalmente, está la arquitectura cliente-servidor. El almacén de datos citado en el párrafo anterior se encuentra en un servidor al cual acceden todos los clientes para obtener la información que necesitan.

### 4.2.2 Arquitectura del sistema

El sistema sigue la clásica arquitectura cliente-servidor. Firebase hace posible que no sea necesario implementar un backend en este proyecto. Como ya se ha explicado en

el apartado 2.4 Tecnologías, Firebase es asíncrono y notifica cuándo hay una modificación en la base de datos. Esta funcionalidad es suficiente para las necesidades de este proyecto. Sin embargo, para aplicaciones más complejas sí puede ser necesario implementar un backend. Usando el API los clientes pueden acceder para leer o escribir información, así como recibir cualquier actualización de los datos en tiempo real. Por lo tanto, Firebase hace las funciones del servidor. Por otro lado están los clientes, que en este caso son las aplicaciones iOS, web y Android. En este proyecto solo se habla de la última ya que el sistema a implementar es una aplicación Android. En la ilustración 5 se muestra la arquitectura del sistema y cómo interactúan las partes por las que está compuesto.



*Ilustración 5: Arquitectura del sistema*

### 4.2.3 Arquitectura del sistema operativo Android

En las siguientes líneas se dará una visión global por capas de cuál es la arquitectura empleada en Android. Cada una de estas capas utiliza servicios ofrecidos por las anteriores, y ofrece a su vez los suyos propios a las capas de niveles superiores, tal como muestra la ilustración 6 (© Google) [5]:

- **Aplicaciones:** Este nivel contiene, tanto las incluidas por defecto de Android como aquellas que el usuario vaya añadiendo posteriormente. Todas estas aplicaciones utilizan los servicios, las API y librerías de los niveles anteriores.



- **Framework de aplicaciones:** Representa fundamentalmente el conjunto de herramientas de desarrollo de cualquier aplicación. Todas las aplicaciones que se desarrollan para Android utilizan el mismo conjunto de API y el mismo "framework", representado por este nivel. Entre las más importantes se encuentran Activity Manager, Window Manager, Telephone Manager, Content Provider, View System, Location Manager, Notification Manager y XMPP Service.
- **Librerías:** La siguiente capa se corresponde con las librerías utilizadas por Android. Éstas han sido escritas utilizando C/C++ y proporcionan a Android la mayor parte de sus capacidades más características. Junto al núcleo basado en Linux, estas librerías constituyen el corazón de Android.
- **Entorno de ejecución de Android:** Al mismo nivel que las librerías de Android se sitúa el entorno de ejecución. Éste lo constituyen las Core Libraries, que son librerías con multitud de clases Java y la máquina virtual Dalvik.
- **Núcleo Linux:** Android utiliza el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los dispositivos móviles. Esta capa contiene los drivers necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes. Siempre que un fabricante incluye un nuevo elemento de hardware, lo primero que se debe realizar para que pueda ser utilizado desde Android es crear las librerías de control o drivers necesarios dentro de este kernel de Linux embebido en el propio Android.

#### 4.2.4 Ciclo de vida de una Actividad

Cada proceso que se ejecuta en Android, correspondiente a una aplicación, estará formado por una o varias actividades independientes (componentes Activity) de esa aplicación. Cada uno de los componentes básicos de Android tiene un ciclo de vida bien definido, esto implica que el desarrollador puede controlar en cada momento en qué estado se encuentra dicho componente, pudiendo así programar las acciones que mejor convengan. El componente Activity, probablemente el más importante, tiene un ciclo de vida como el mostrado en la ilustración 7 [6].



Ilustración 6: Arquitectura del sistema operativo Android. Fuente: © Google [5]

### 4.3 Diseño de la interfaz de usuario

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo.

Una interfaz de usuario debe cumplir las tres reglas (Theo Mandel, 1997) [7]:

- El usuario debe tener el control.
- Reducir la carga de memoria del usuario.
- Mantener la consistencia de la interfaz.

Es imprescindible tener una idea clara de cómo será la interfaz antes de empezar a programar. En *Cuatroochenta* el equipo de diseño crea la interfaz de las pantallas, le enseña al cliente cómo quedarán y cómo se comunicarán. Si el cliente acepta, dichas pantallas se le pasan al equipo de desarrollo. Así, los programadores saben exactamente

qué tienen que programar para las interfaces de usuario y el producto será del agrado del cliente.

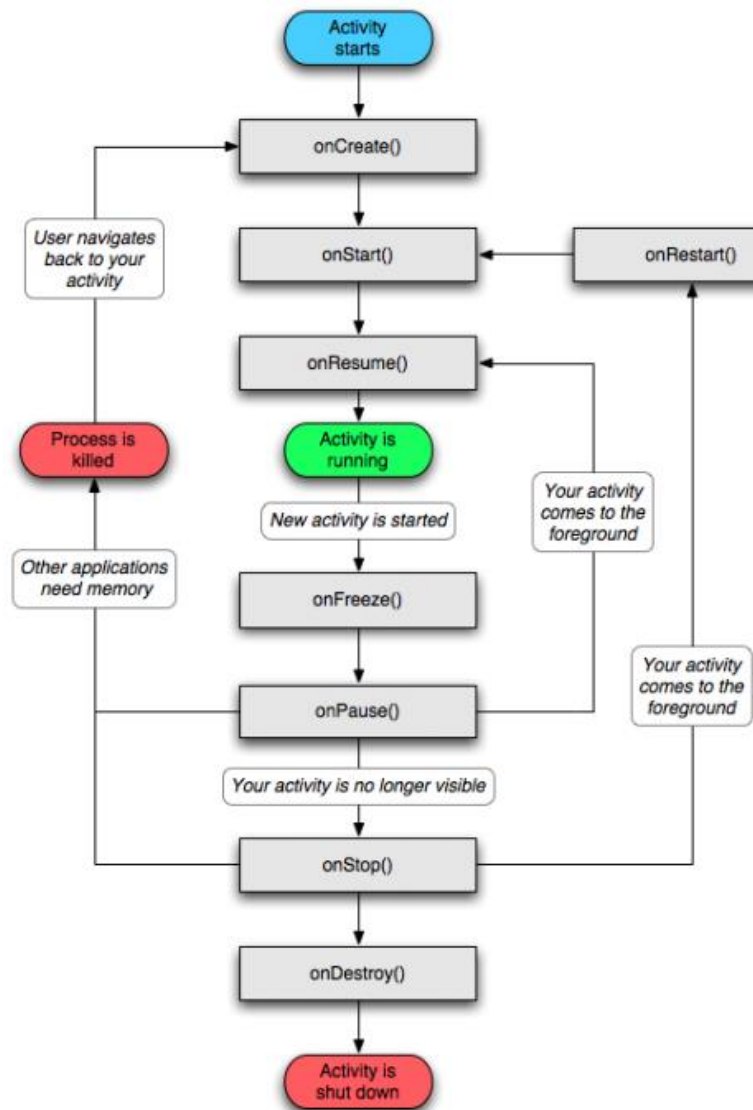


Ilustración 7: Ciclo de vida de una Actividad Android. Fuente: © Google [6]

### 4.3.1 Interfaz gráfica

A partir del diseño de la interfaz gráfica proporcionada por el equipo de diseño de la empresa se ha programado la aplicación. Durante el desarrollo el programador está en contacto con los diseñadores ya que surgen dudas sobre algunos aspectos de la interfaz, pero suele tener que ver con pequeños detalles de las pantallas.

Las pantallas diseñadas para la aplicación del Hackathon se muestran al final del documento, en el Anexo B.

### 4.3.2 Jerarquía de pantallas

En la ilustración 8 se muestra la jerarquía de pantallas, en ella se ve cómo se relacionan y a qué pantallas se puede acceder desde cada una. El número que se muestra a la izquierda de cada una de las pantallas se corresponde con el número de una ilustración del Anexo B, Capturas de pantalla.

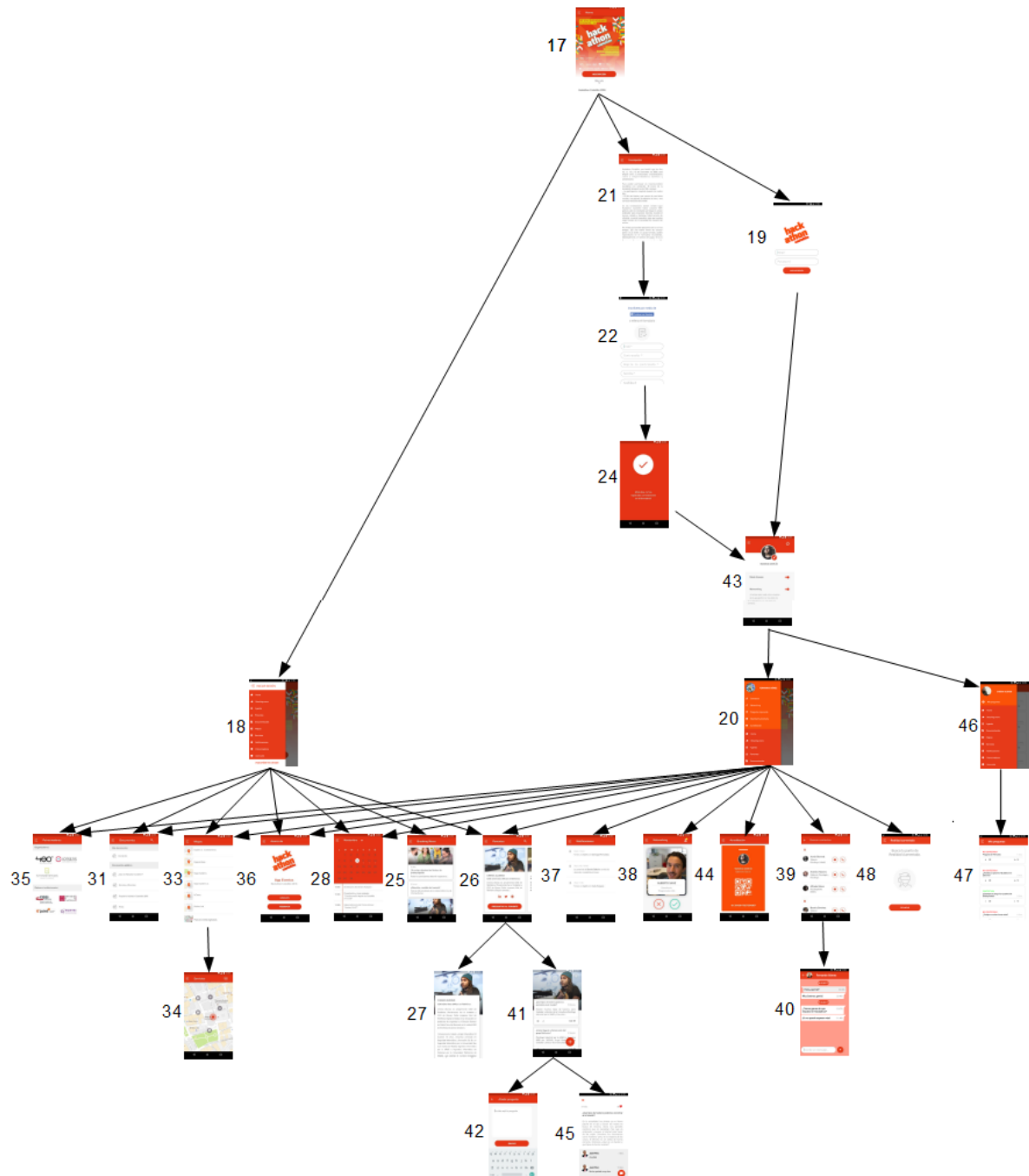


Ilustración 8: Jerarquía de pantallas

## 4.4 Patrones de diseño

De acuerdo a la estructura del proyecto respecto a la organización de la información, se prevé usar 3 patrones de diseño: MVC, Singleton y Callback.

### Modelo-Vista-Controlador:

Se ha seguido este patrón para separar los datos y la lógica de la interfaz de usuario. Es uno de los primeros patrones que se idearon en el campo de las interfaces de usuario, independientemente de la plataforma. Así, la vista está formada por todos los ficheros que se encuentran en el módulo **layout**, las pantallas que contienen la interfaz gráfica. El modelo se encuentra en el módulo **model** que contiene las clases que encapsulan la información y el módulo **dao** que obtiene la información de la base de datos y la actualiza. Finalmente, las actividades Android hacen la función del controlador. Detectan la acción del usuario, se comunican con el modelo y actualizan los datos y la vista. En la ilustración 9 se muestra el funcionamiento de este patrón.

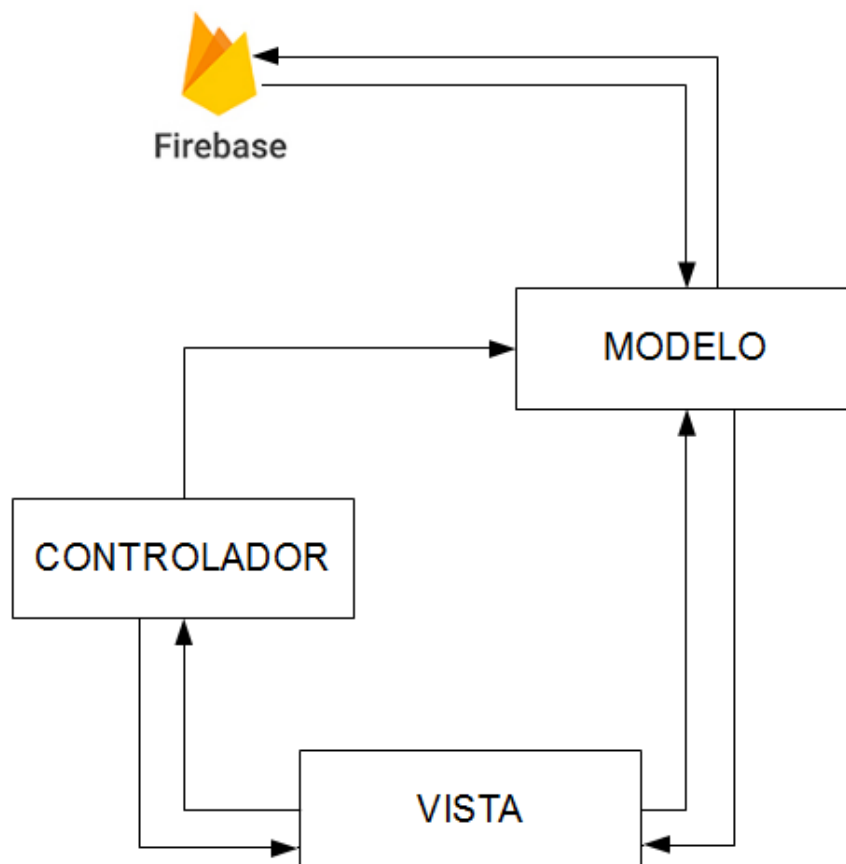


Ilustración 9: Patrón MVC

## Singleton

El patrón Singleton tiene como finalidad asegurarse de que hay una única instancia de una clase determinada. La necesidad de implementar este patrón puede deberse al alto coste de crear una instancia de la clase en cuestión.



Ilustración 10: Patrón Singleton. Fuente: Wikipedia [8]

## Callback

Como ya se comenta en el apartado 2.4 en el que se explican las tecnologías utilizadas, Firebase es asíncrono. Por lo tanto, si es necesario tener cierta información antes de que se muestre la pantalla, hay que esperar a que la información se haya leído de la base de datos. En la ilustración 11 se muestra cómo se comunican las clases mediante el patrón Callback.

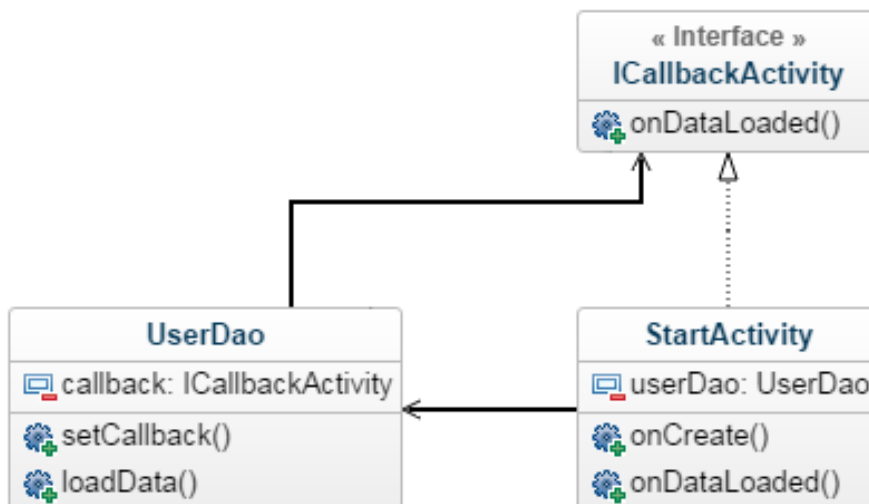


Ilustración 11: Comunicaciones con Callbacks

## Capítulo 5

# Implementación y pruebas

### Índice

---

5.1. Detalles de implementación .....	39
5.1.1 Módulos de software .....	39
5.1.2 Implementación de patrones de diseño .....	40
5.2. Pruebas de aceptación .....	42

---

## 5.1 Detalles de implementación

En este apartado se muestran los módulos en los que se ha organizado el software y los patrones de diseño que se han implementado.

### 5.1.1 Módulos de software

Por defecto Android separa los diferentes recursos de la aplicación del código Java. Sin embargo, es conveniente separar el código en diferentes paquetes de acuerdo a la funcionalidad que ofrecen. Esta técnica no es exclusiva de Android, todos los proyectos software agrupan los ficheros por funcionalidades para tener una correcta organización. El código de la aplicación de eventos y congresos ha quedado organizado de la siguiente forma:

- **src**
  - **activities**: contiene las actividades Android que forman la aplicación, éstas hacen las funciones de controlador en el patrón MVC.
  - **adapter**: contiene las clases que adaptan la información a las listas.
  - **dao**: contiene las clases que acceden a la base de datos Firebase, éstas obtienen la información, la tratan para guardarla en objetos de las clases del modelo y ofrecen operaciones para obtener la información. Hay una clase DAO por cada tabla de la base de datos, se corresponden con el Anexo A y son las siguientes:
    - **ChatDao.java**

- **DocumentDao.java**
  - **EventDao.java**
  - **FeedbackDao.java**
  - **InfoDao.java**
  - **MapDao.java**
  - **MatchDao.java**
  - **NewsDao.java**
  - **NotificationDao.java**
  - **SpeakerDao.java**
  - **TokenDao.java**
  - **UserDao.java**
  - **httpConnections**: contiene las clases que acceden a información sobre el mapa a partir de una URL.
  - **model**: contiene las clases del modelo que encapsulan la información, son las que se muestran en la ilustración 3, la del diagrama de clases.
  - **notifications**: se encarga de detectar cuándo llega una notificación y la lanza.
  - **parser**: contiene las clases que convierten información con estructura JSON al formato deseado.
- **res**
- **drawable**: contiene las imágenes, iconos y diseños personalizados (botones por ejemplo) que usa la aplicación.
  - **layout**: contiene los ficheros que implementan la interfaz de usuario, los que contienen las vistas que se muestran en el Anexo B.
  - **menu**: contiene los ficheros que conforman los menús de la aplicación.
  - **values**: contiene los ficheros que determinan estilos, colores, dimensiones y palabras o frases cortas que usa la aplicación múltiples veces.

### 5.1.2 Implementación de patrones de diseño

Un proyecto software de tamaño medio como este debe estar bien estructurado si queremos que tenga éxito. Un buen diseño nos hará mucho más fácil la implementación de nuevas funcionalidades y la modificación del proyecto en caso de ser necesario. Una



técnica de programación muy recomendada es la utilización de patrones de diseño para resolver problemas ya conocidos.

### **Modelo-Vista-Controlador**

Los detalles de implementación de este patrón ya se han explicado detalladamente en el apartado 4.4 Patrones de diseño.

### **Singleton**

En este proyecto el alto coste de creación de una clase ha ocurrido con los ficheros que cargan información de la base de datos. Queremos asegurarnos de que la información se carga una sola vez, no cada vez que se inicia una actividad al cambiar de pantalla. Aunque se ha comprobado que el tiempo que tarda no es apreciable, no sería eficiente cargar la misma información constantemente. Además, si la aplicación creciera y gestionara grandes cantidades de información, es posible que la constante carga de datos que ya se han descargado provocara retrasos que el usuario sí notaría.

### **Callback**

Por ejemplo, una llamada asíncrona ocurre al iniciar la aplicación si hay una sesión iniciada. En el menú principal se debe mostrar la imagen, nombre y apellidos del usuario. Por lo tanto, es necesario tener esta información antes de cargar el menú. Esto se consigue mediante la implementación del patrón de diseño Callback. La actividad que arranca la aplicación (StartActivity, muestra la imagen de portada hasta que se carga) implementa la interfaz ICallbackActivity. Cuando se inicia, le pasa a UserDao un objeto callback (su propia instancia) y llama al método de carga del usuario. UserDao espera a que Firebase le mande los datos y, cuando los tiene, llama al callback para indicárselo. Cuando éste recibe la llamada carga la siguiente actividad ya que los datos del usuario ya están disponibles.

Como se ve en la ilustración 11, primero se ejecuta el onCreate de StartActivity. Éste llama a setCallback(this) y luego a loadData(). Finalmente, loadData carga los datos y, cuando los tiene, llama a callback.onDataLoaded().

## 5.2 Pruebas de aceptación

El objetivo de las pruebas de aceptación es comprobar si se cumplen los requisitos especificados por el cliente. Son pruebas de alto nivel creadas usualmente por clientes y usuarios, en colaboración con analistas, desarrolladores y testers. Suelen definirse de forma precisa antes de comenzar a desarrollar y son el criterio definitivo de calidad de una función de usuario. Ya que este proyecto es individual, el supervisor Sergio Aguado hizo de cliente/usuario y el alumno de analista, desarrollador y tester.

Cada prueba de aceptación que tiene la siguiente estructura:

- **DADO (GIVEN):** un escenario,
- **CUANDO (WHEN):** ocurre una evento determinado,
- **ENTONCES (THEN):** se realiza una acción en consecuencia.

A cada historia de usuario se le debe asignar un conjunto de pruebas de aceptación que aseguren que cumple todos los requisitos como el cliente espera. Para dar una historia de usuario como completada tiene que pasar todas las pruebas de aceptación que se la hayan asignado en la planificación del sprint. Las pruebas de aceptación realizadas se muestran detalladamente en el desarrollo de cada sprint.

## Capítulo 6

# Desarrollo del proyecto: Sprints

### Índice

---

6.1. Sprint 1 (27/02/2017 a 12/03/2017) .....	44
6.1.1 Resumen del sprint .....	44
6.2. Sprint 2 (13/03/2017 a 26/03/2017) .....	44
6.2.1 Resumen del sprint .....	44
6.2.2 Pruebas de aceptación .....	46
6.2.3 Resultados .....	47
6.3. Sprint 3 (27/03/2017 a 09/04/2017) .....	48
6.3.1 Resumen del sprint .....	48
6.3.2 Pruebas de aceptación .....	52
6.3.3 Resultados .....	55
6.4. Sprint 4 (10/04/2017 a 23/04/2017) .....	55
6.4.1 Resumen del sprint .....	55
6.4.2 Pruebas de aceptación .....	58
6.4.3 Resultados .....	61
6.5. Sprint 5 (24/04/2017 a 07/05/2017) .....	61
6.5.1 Resumen del sprint .....	61
6.5.2 Pruebas de aceptación .....	65
6.5.3 Resultados .....	68
6.6. Sprint 6 (08/05/2017 a 21/05/2017) .....	69
6.6.1 Resumen del sprint .....	69
6.6.2 Pruebas de aceptación .....	72
6.6.3 Resultados .....	74
6.7 Pila final del producto (21/05/2017) .....	75

---

En este capítulo se muestra el seguimiento del proyecto, cómo se ha desarrollado y qué tareas se han hecho en cada uno de los seis sprints. Para cada uno de los sprints se presenta qué historias de usuario de la pila del producto se planea completar al principio, qué pruebas de aceptación se han creado y pasado y los resultados obtenidos en términos de la aplicación final (la jerarquía de pantallas con las que se han completado). Además, a partir del segundo sprint hay un apartado extra con la actualización de la pila del producto. A lo largo del proyecto se añaden o eliminan historias de usuario y se modifican las existentes (los puntos de historia por ejemplo).

## 6.1 Sprint 1 (27/02/2017 a 12/03/2017)

### 6.1.1 Resumen del sprint

En el primer sprint la mayoría del tiempo se dedicó a realizar tutoriales para familiarizarse con Android y aprender los conocimientos básicos necesarios para poder implementar una aplicación móvil. También se realizó la propuesta técnica del proyecto y se empezó a planificar.

## 6.2. Sprint 2 (13/03/2017 a 26/03/2017)

### 6.2.1 Resumen del sprint

En el segundo sprint se continuó con los tutoriales hasta la mitad del periodo. A continuación se definió la pila del producto inicial como se puede ver en la tabla 3.

<b>ID</b>	<b>FUNCIONALIDAD</b>	<b>ESTADO</b>
1	Acceder a la información sobre la inscripción en el evento.	✗
2	Inscribirse en el evento mediante un formulario.	✗
3	Poder navegar por la aplicación gracias a un menú.	✗
4	Poder ver las últimas noticias actualizadas sobre el evento.	✗
5	Consultar la agenda con la planificación temporal del evento.	✗
6	Ver información sobre los ponentes del evento y hacerles preguntas.	✗
7	Descargar documentos en PDF con información más detallada del evento.	✗
8	Poder acceder a un mapa en el que estén indicados todos los servicios que ofrece la ciudad (hoteles, restaurantes, ...).	✗
9	Poder acceder a un mapa en el que estén indicados todos los sitios importantes del evento.	✗

10	Recibir notificaciones sobre algún cambio en el calendario o alguna interacción con otro usuario.	✘
11	Poder consultar los patrocinadores del evento.	✘
12	Poder consultar la información legal del evento.	✘
13	Una vez creada la cuenta, acceder al perfil.	✘
14	Una vez creada la cuenta, modificar al perfil.	✘
15	Obtener una acreditación que permita participar en el evento.	✘
16	Interactuar con otros usuarios y tenerlos en tu lista de contactos.	✘

*Tabla 3: Pila del producto inicial*

Una vez se había definido la pila del producto, se eligieron unas determinadas tareas para completarlas durante este sprint. Al ser el principio del proyecto, se eligieron tareas sencillas para continuar aprendiendo y avanzar a un buen ritmo. Hay que recordar que la estimación de los puntos de historia fue de un día, o lo que es lo mismo, 5 horas. A continuación se muestra la pila del sprint.

#### **Acceder a la información sobre la inscripción del evento.**

- Crear la primera pantalla que se ve al iniciar la aplicación (50 minutos).
- Crear la pantalla principal y la Toolbar de acuerdo al diseño proporcionado (4 horas y 20 minutos).
- Crear la pantalla con la información sobre la inscripción del evento (50 minutos).

Tiempo total: 6 horas → 2 puntos de historia (10 horas).

#### **Inscribirse en el evento mediante un formulario (inacabada).**

- Crear la pantalla que contiene el formulario de inscripción del evento y recoger los datos que el usuario introduzca (3 horas y 50 minutos).

Tiempo total: 3 horas y 50 minutos → 1 punto de historia (5 horas).

**Poder navegar por la aplicación gracias a un menú (inacabada).**

- Crear la interfaz del menú genérico y su funcionalidad (3 horas y 20 minutos).

Tiempo total: 3 horas y 20 minutos → 1 punto de historia (5 horas).

**Poder consultar los patrocinadores del evento.**

- Crear la pantalla con los logos de los patrocinadores del evento (1 hora y 30 minutos).

Tiempo total: 1 hora y 30 minutos → 1/2 punto de historia (2 horas y 30 minutos).

**Poder consultar la información legal del evento.**

- Crear la pantalla que muestre la información legal sobre el evento (40 minutos).

Tiempo total: 40 minutos → 1/2 punto de historia (2 horas y 30 minutos).

En total se han asignado 5 puntos de historia, que equivalen a 5 días o a 25 horas. Aunque los sprints constan de 50 horas, hay que tener en cuenta que al principio de este aún se estaban realizando tutoriales. Por otra parte, también cabe destacar que la tarea del menú se completa a medias debido a que es necesaria para poder navegar por la aplicación pero aún faltan datos del usuario para poder implementarla al completo. La tarea de la inscripción tampoco se completará, quedarán algunos aspectos pendientes para el siguiente sprint.

## 6.2.2 Pruebas de aceptación

A continuación se muestran las pruebas de aceptación que se han pasado satisfactoriamente al final de este segundo sprint.

- **DADO** que un usuario usa la aplicación sin necesidad de estar registrado, **CUANDO** pulsa el botón para acceder a la información del evento, **ENTONCES** se le muestra dicha información.
- **DADO** que un usuario introduce sus datos básicos (nombre, apellidos, teléfono y ciudad) en el formulario de inscripción, **CUANDO** el usuario pulsa el botón para registrarse, **ENTONCES** queda registrado como un usuario de la aplicación y se guardan sus datos.
- **DADO** que un usuario usa la aplicación sin necesidad de estar registrado, **CUANDO** pulsa el botón del menú, **ENTONCES** se abre el menú genérico para los usuarios que no están registrados con posibilidad de acceso a las secciones públicas de la aplicación.
- **DADO** que un usuario abre el menú de la aplicación, **CUANDO** selecciona la opción para acceder a la pantalla con los patrocinadores del evento, **ENTONCES** se muestra la pantalla con la información de todos estos patrocinadores.
- **DADO** que un usuario abre el menú de la aplicación, **CUANDO** selecciona la opción para acceder a la pantalla con la información legal del evento, **ENTONCES** se muestra la pantalla con la información sobre los aspectos legales del evento.

### 6.2.3 Resultados

En la ilustración 12 se pueden ver las pantallas que se han completado hasta el momento rodeadas por un rectángulo verde y las que se han empezado pero no están acabadas por uno amarillo.

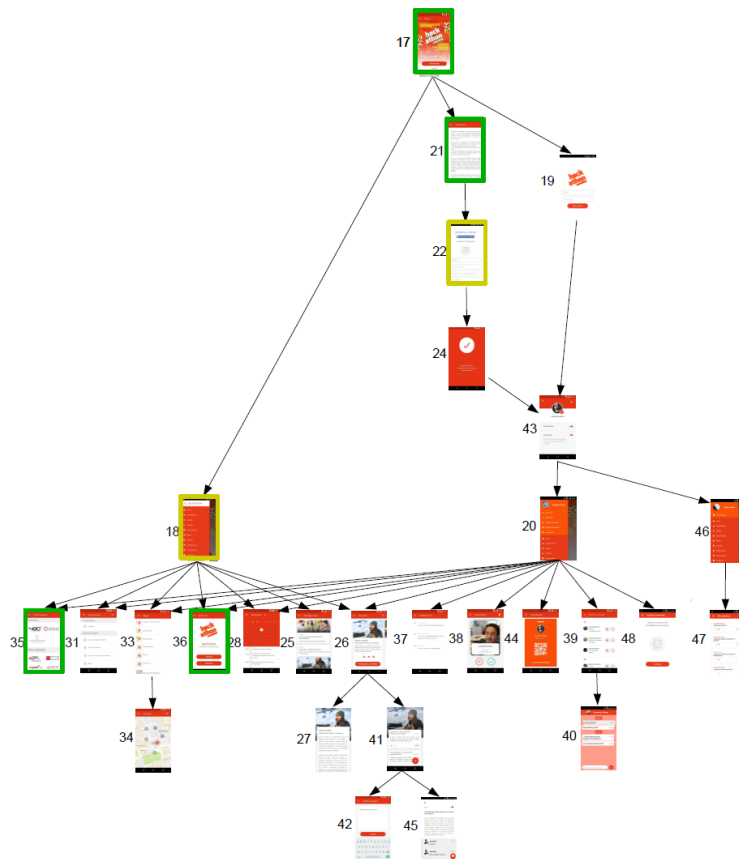


Ilustración 12: Pantallas hechas en verde y empezadas en amarillo hasta el Sprint 2

### 6.3. Sprint 3 (27/03/2017 a 09/04/2017)

#### 6.3.1 Resumen del sprint

Antes de empezar con la pila del producto, es necesario comentar los cambios que se produjeron durante este sprint. Al inicio de este periodo fue cuando se decidió que lo mejor era usar Firebase, por esto algunas tareas ya realizadas se tuvieron que modificar y se tuvo que dedicar tiempo que no estaba previsto al aprendizaje de esta tecnología.

En la tabla 4 se muestra la actualización del estado de la pila del producto al inicio de este tercer sprint.





















ID	FUNCIONALIDAD	ESTADO
1	Acceder a la información sobre la inscripción en el evento.	
2	Inscribirse en el evento mediante un formulario.	
3	Poder navegar por la aplicación gracias a un menú.	
4	Poder ver las últimas noticias actualizadas sobre el evento.	
5	Consultar la agenda con la planificación temporal del evento.	
6	Ver información sobre los ponentes del evento y hacerles preguntas.	
7	Descargar documentos en PDF con información más detallada del evento.	
8	Poder acceder a un mapa en el que estén indicados todos los servicios que ofrece la ciudad (hoteles, restaurantes, ...).	
9	Poder acceder a un mapa en el que estén indicados todos los sitios importantes del evento.	
10	Recibir notificaciones sobre algún cambio en el calendario o alguna interacción con otro usuario.	
11	Poder consultar los patrocinadores del evento.	
12	Poder consultar la información legal del evento.	
13	Una vez creada la cuenta, acceder al perfil.	
14	Una vez creada la cuenta, modificar al perfil.	
15	Obtener una acreditación que permita participar en el evento.	
16	Interactuar con otros usuarios y tenerlos en tu lista de contactos.	
17	Una vez el usuario se ha registrado, puede iniciar sesión.	
18	Dar feedback a los administradores con quejas o sugerencias.	

Tabla 4: Pila del producto al inicio del Sprint 3

A partir del estado de la pila del producto se planificaron las siguientes tareas para ser implementadas durante este tercer sprint.

**Inscribirse en el evento mediante un formulario (inacabada).**

- Arreglar los problemas con los spinners que permiten elegir país, comunidad y provincia al usuario (1 hora y 30 minutos).
- Añadir los campos email y contraseña que no constaban en el diseño proporcionado (30 minutos).
- Registrarse mediante Firebase (2 horas).

Tiempo total: 4 horas → 1 punto de historia (5 horas).

**Poder iniciar sesión.**

- Crear un interfaz en la que el usuario tenga que introducir el email y la contraseña (30 minutos).
- Realizar la autenticación mediante Firebase (30 minutos).

Tiempo total: 1 hora → 1/2 punto de historia (2 horas y 30 minutos).

**Poder navegar por la aplicación gracias a un menú.**

- Administración de las sesiones (2 horas).
- El menú muestra una información u otra dependiendo de si el usuario ha iniciado sesión o no (1 hora).

Tiempo total: 3 horas → 1 punto de historia (5 horas).

**Documentos informativos del evento (inacabada).**

- Crear la interfaz con la lista de los documentos (1 hora 30 minutos).
- Mostrar un documento cualquiera en formato PDF (1 hora 30 minutos).

Tiempo total: 3 horas → 1 punto de historia (5 horas).

### **Mapa con los servicios de la ciudad.**

- Lista con los servicios de la ciudad (3 horas).
- Mapa que muestra los servicios de la ciudad con marcadores e información de cada uno de ellos (7 horas).
- Pantalla que muestra los detalles del servicio (1 hora).
- Mostrar la ruta al servicio en el mapa desde la ubicación actual (2 horas).

Tiempo total: 13 horas → 3 puntos de historia (15 horas).

### **Mapa con los sitios importantes del evento.**

- Lista con los sitios importantes del evento (40 minutos).
- Mapa que muestra los sitios importantes del evento con marcadores e información de cada uno de ellos (1 hora y 30 minutos).
- Pantalla que muestra los detalles de la instalación (30 minutos).
- Mostrar la ruta al sitio de interés del evento en el mapa desde la ubicación actual (30 minutos).

Tiempo total: 3 horas y 10 minutos → 1 punto de historia (5 horas).

### **Acceder al perfil.**

- Crear la interfaz del perfil (2 horas).
- Mostrar la información del usuario en dicha interfaz (30 minutos).

Tiempo total: 2 horas y 30 minutos → 1/2 punto de historia (2 horas y 30 minutos).

### **Dar feedback a los administradores.**

- Crear la interfaz (30 minutos).
- Enviar la información a la base de datos para que los administradores la puedan ver (30 minutos).

Tiempo total: 30 minutos → 1/5 punto de historia (1 hora).

En total se han asignado 8,2 puntos de historia, que equivalen a 8,2 días o a 41 horas. El tiempo que falta hasta las 50 horas, además del que sobra ya que las predicciones están hechas para que sobre tiempo por si hay imprevistos, se dedica al aprendizaje de Firebase y NoSQL. Como se puede ver, la parte de creación de los sitios de interés está previsto que sea mucho más rápida que la de servicios porque se implementará después y es prácticamente idéntica a esta última.

### 6.3.2 Pruebas de aceptación

A continuación se muestran las pruebas de aceptación que se han pasado satisfactoriamente al final de este tercer sprint.

- **DADO** que un usuario quiere inscribirse, **CUANDO** introduce los datos en el formulario de inscripción (puede elegir el país, comunidad y provincia mediante un spinner y se añaden los campos email y contraseña), **ENTONCES** el usuario se registra en Firebase y se guardan todos sus datos.
- **DADO** que un usuario quiere inscribirse, **CUANDO** no rellena todos los campos obligatorios, **ENTONCES** se muestra un mensaje de error indicando el problema.
- **DADO** que un usuario quiere inscribirse, **CUANDO** no tiene conexión con Firebase (normalmente porque no hay Internet), **ENTONCES** se muestra un mensaje de error indicando que la conexión no ha sido posible y que pruebe más tarde.
- **DADO** que un usuario registrado quiere iniciar sesión, **CUANDO** introduce su email y contraseña correctamente en la pantalla de iniciar sesión, **ENTONCES** el usuario inicia sesión mediante Firebase y se muestran las secciones personales de la aplicación.
- **DADO** que un usuario registrado o no registrado quiere iniciar sesión, **CUANDO** introduce un email que no se ha registrado o una contraseña que no se corresponde, **ENTONCES** se muestra un mensaje de error con el problema y no se inicia sesión.
- **DADO** que un usuario no ha iniciado sesión, **CUANDO** intenta abrir el menú, **ENTONCES** únicamente se puede acceder a las secciones públicas.

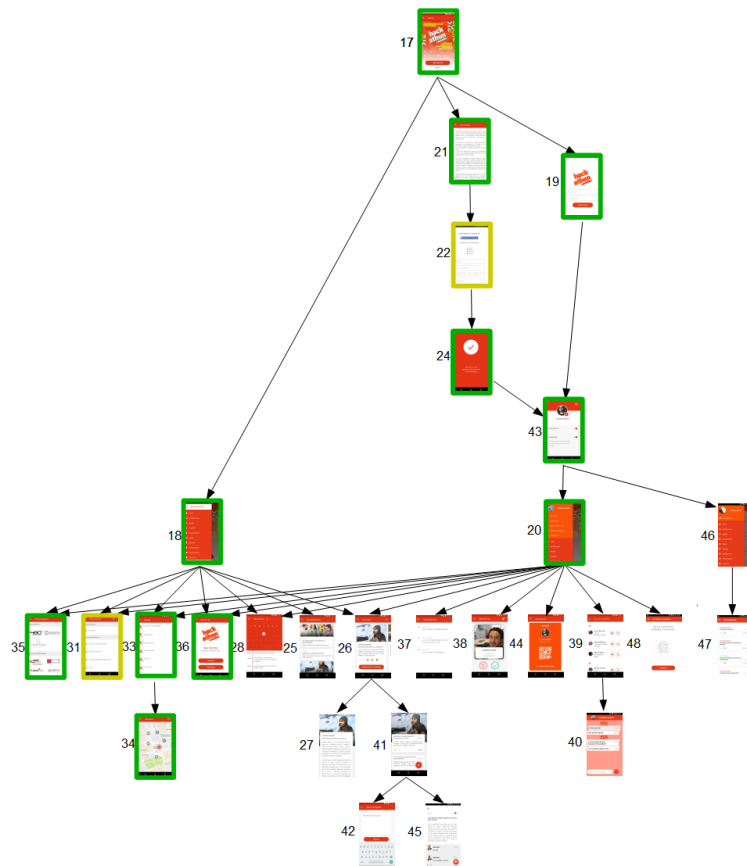
- **DADO** que un usuario ha iniciado sesión, **CUANDO** abre el menú, **ENTONCES** se muestra un menú con todas las secciones públicas y también las privadas del usuario (perfil, acreditación, ...).
- **DADO** que un usuario abre el menú, **CUANDO** accede a la sección de documentos, **ENTONCES** se muestra una lista de documentos.
- **DADO** que un usuario abre el menú, **CUANDO** intenta acceder a la lista de servicios de la ciudad del evento, **ENTONCES** se muestra una lista con todos estos servicios.
- **DADO** que un usuario se encuentra en la pantalla de la lista de servicios, **CUANDO** pulsa el icono del mapa, **ENTONCES** se muestra el mapa con todos los servicios.
- **DADO** que un usuario se encuentra en la pantalla de la lista de servicios, **CUANDO** pulsa sobre un servicio de la lista, **ENTONCES** se abre el mapa con todos los servicios centrado en el seleccionado por el usuario, éste está resaltado, se muestra su información básica y un botón para ver la información completa del servicio.
- **DADO** que se encuentra en la pantalla del mapa de servicios, **CUANDO** toca el marcador de un servicio, **ENTONCES** se muestra la información básica de dicho servicio, el marcador cambia de color y se muestra un botón para ver la información completa del servicio.
- **DADO** que un usuario se encuentra en la pantalla del mapa de servicios con uno seleccionado, **CUANDO** pulsa el botón para ver más información, **ENTONCES** se cambia a la pantalla con la información del evento que también tiene un botón que permite generar la ruta desde la ubicación actual.
- **DADO** que el usuario se encuentra en la pantalla con la información de un servicio, **CUANDO** pulsa el botón para ver la ruta y no da permisos de ubicación, **ENTONCES** se muestra un mensaje de que debe dar permisos para poder generar la ruta.
- **DADO** que el usuario se encuentra en la pantalla con la información de un servicio, **CUANDO** pulsa el botón para ver la ruta y da permisos de ubicación, **ENTONCES** se muestra un mapa en el que está la ubicación actualizada del usuario y la ruta más corta hasta el servicio.

- **DADO** que un usuario abre el menú, **CUANDO** intenta acceder a la lista de sitios de interés del evento, **ENTONCES** se muestra una lista con todos estos lugares.
- **DADO** que un usuario se encuentra en la pantalla de la lista de sitios de interés, **CUANDO** pulsa el icono del mapa, **ENTONCES** se muestra el mapa con todos los sitios de interés.
- **DADO** que un usuario se encuentra en la pantalla de la lista de sitios de interés, **CUANDO** pulsa sobre un sitio de interés de la lista, **ENTONCES** se abre el mapa con todos los sitios de interés centrado en el seleccionado por el usuario, éste está resaltado, se muestra su información básica y un botón para ver la información completa del sitio de interés.
- **DADO** que se encuentra en la pantalla del mapa de sitios de interés, **CUANDO** toca el marcador de un sitio de interés, **ENTONCES** se muestra la información básica de dicho sitio de interés, el marcador cambia de color y se muestra un botón para ver la información completa del sitio de interés.
- **DADO** que un usuario se encuentra en la pantalla del mapa de sitios de interés con uno seleccionado, **CUANDO** pulsa el botón para ver más información, **ENTONCES** se cambia a la pantalla con la información del sitio de interés que también tiene un botón que permite generar la ruta desde la ubicación actual.
- **DADO** que el usuario se encuentra en la pantalla con la información de un sitio de interés, **CUANDO** pulsa el botón para ver la ruta y no da permisos de ubicación, **ENTONCES** se muestra un mensaje de que debe dar permisos para poder generar la ruta.
- **DADO** que el usuario se encuentra en la pantalla con la información de un sitio de interés, **CUANDO** pulsa el botón para ver la ruta y da permisos de ubicación, **ENTONCES** se muestra un mapa en el que está la ubicación actualizada del usuario y la ruta más corta hasta el sitio de interés.
- **DADO** que un usuario ha iniciado sesión, **CUANDO** intenta acceder a su perfil mediante el menú, **ENTONCES** se muestra la pantalla del perfil con su información básica.
- **DADO** que un usuario accede a la sección de feedback, **CUANDO** introduce el mensaje que quiere hacer llegar a los administradores y pulsa

el botón de enviar feedback, **ENTONCES** dicha información queda almacenada en la base de datos Firebase.

### 6.3.3 Resultados

En la ilustración 13 se pueden ver las pantallas que se han completado hasta el momento rodeadas por un rectángulo verde y las que se han empezado pero no están acabadas por uno amarillo.



*Ilustración 13: Pantallas hechas en verde y empezadas en amarillo hasta el Sprint 3*

## 6.4. Sprint 4 (10/04/2017 a 23/04/2017)

### 6.4.1 Resumen del sprint

En la tabla 5 se muestra la actualización del estado de la pila del producto al inicio de este cuarto sprint.



















ID	FUNCIONALIDAD	ESTADO
1	Acceder a la información sobre la inscripción en el evento.	
2	Inscribirse en el evento mediante un formulario.	
3	Poder navegar por la aplicación gracias a un menú.	
4	Poder ver las últimas noticias actualizadas sobre el evento.	
5	Consultar la agenda con la planificación temporal del evento.	
6	Ver información sobre los ponentes del evento y hacerles preguntas.	
7	Descargar documentos en PDF con información más detallada del evento.	
8	Poder acceder a un mapa en el que estén indicados todos los servicios que ofrece la ciudad (hoteles, restaurantes, ...).	
9	Poder acceder a un mapa en el que estén indicados todos los sitios importantes del evento.	
10	Recibir notificaciones sobre algún cambio en el calendario o alguna interacción con otro usuario.	
11	Poder consultar los patrocinadores del evento.	
12	Poder consultar la información legal del evento.	
13	Una vez creada la cuenta, acceder al perfil.	
14	Una vez creada la cuenta, modificar el perfil.	
15	Obtener una acreditación que permita participar en el evento.	
16	Interactuar con otros usuarios y tenerlos en tu lista de contactos.	
17	Una vez el usuario se ha registrado, puede iniciar sesión.	
18	Dar feedback a los administradores con quejas o sugerencias.	

Tabla 5: Pila del producto al inicio del Sprint 4



A partir del estado de la pila del producto se planificaron las siguientes tareas para ser implementadas durante este cuarto sprint.

**Poder ver las últimas noticias del evento.**

- Crear la interfaz de la lista (1 hora).
- Crear la interfaz de los detalles de la noticia (1 hora).
- Mostrar las noticias en ambas interfaces y conectarlas (1 hora).

Tiempo total: 3 horas → 1 punto de historia (5 horas).

**Consultar la agenda con la planificación temporal del evento (inacabada).**

- Crear la interfaz de la lista de eventos (1 hora).
- Crear la interfaz de los detalles de un evento (1 hora).
- Crear la Toolbar con el calendario como se muestra en el diseño y actualizar la información de acuerdo a la fecha que el usuario escoja (10 horas).

Tiempo total: 12 horas → 3 puntos de historia (15 horas).

**Ver información sobre los ponentes del evento y hacerles preguntas.**

- Crear la interfaz de la lista de ponentes y mostrar la información (2 horas y 30 minutos).
- Crear la interfaz de los detalles del ponente y mostrar la información del que el usuario haya elegido (2 horas).
- Crear la lista de preguntas que se le han hecho al ponente y mostrar la información de cada una (4 horas).
- Permitir que el usuario realice una pregunta a un ponente y se almacene en la base de datos (3 horas).
- Crear la interfaz de los detalles de una pregunta: la pregunta, la contestación y los comentarios que los usuarios han hecho de la respuesta proporcionada. Además si el usuario está registrado puede dar 'like' a la respuesta (7 horas).

Tiempo total: 18 horas y 30 minutos → 4 puntos de historia (20 horas).

En total se han asignado 8 puntos de historia, que equivalen a 8 días o a 40 horas. No se han asignado las 50 horas ya que este sprint coincide con las vacaciones de Pascua y hay dos días laborables menos. Como se puede ver, la primera tarea ha sido mucho más corta que lo estimado en la pila del producto inicial. Con las partes que ya se han implementado es obvio que la tarea no es tan compleja como se pensó en un principio. Sin embargo, la tarea de las preguntas al ponente resultó larga y compleja debido a la gran cantidad de información que hay que manejar y cómo se relaciona entre sí.

#### 6.4.2 Pruebas de aceptación

A continuación se muestran las pruebas de aceptación que se han pasado satisfactoriamente al final de este cuarto sprint.

- **DADO** que un usuario tiene abierto el menú de la aplicación, **CUANDO** intente acceder a la sección de últimas noticias, **ENTONCES** se muestra un listado con todas las noticias de interés del evento y un breve resumen.
- **DADO** que un usuario se encuentra en la sección de últimas noticias, **CUANDO** pulsa una, **ENTONCES** accede a la información detallada de dicha noticia.
- **DADO** que un usuario tiene abierto el menú de la aplicación, **CUANDO** intenta acceder a la sección de la agenda, **ENTONCES** se muestra una lista de eventos que se corresponden con el mes actual.
- **DADO** que un usuario se encuentra en la sección de la agenda, **CUANDO** pulsa la Toolbar, **ENTONCES** se despliega un calendario por el que puede navegar.
- **DADO** que un usuario se encuentra en la pantalla de la agenda con el calendario desplegado, **CUANDO** selecciona una fecha o cambia de mes, **ENTONCES** se muestra una lista con los eventos que hay ese mes o un mensaje de que no hay eventos si la lista está vacía.
- **DADO** que el usuario se encuentra en la pantalla de la agenda y ha seleccionado un mes en el que hay al menos un evento, **CUANDO** pulsa sobre uno de los eventos, **ENTONCES** se muestra información detallada del evento en cuestión.

- **DADO** que un usuario tiene abierto el menú de la aplicación, **CUANDO** intenta acceder a la sección de los ponentes, **ENTONCES** se muestra una pantalla con la lista de los ponentes del evento, cada uno tiene: una imagen, su nombre, su trabajo, una breve descripción, enlaces a su perfil de LinkedIn, Twitter y a su web, y un botón para realizarle una pregunta.
- **DADO** que un usuario se encuentra en la pantalla de la lista de los ponentes, **CUANDO** toca la foto del ponente, **ENTONCES** se muestra una pantalla con la información detallada de éste.
- **DADO** que un usuario se encuentra en la pantalla de la lista de ponentes, **CUANDO** pulsa sobre el enlace de Twitter, LinkedIn o del sitio web, **ENTONCES** se abre el navegador con la URL correspondiente.
- **DADO** que un usuario se encuentra en la pantalla de la lista de ponentes, **CUANDO** toca el icono de la lupa en la Toolbar, **ENTONCES** se muestra un campo de texto mediante el que buscar ponentes.
- **DADO** que un usuario se encuentra en la pantalla de la lista de ponentes con el campo de búsqueda desplegado, **CUANDO** escribe algo, **ENTONCES** se muestran solo los ponentes que cumplen los criterios.
- **DADO** que un usuario se encuentra en la pantalla de la lista de ponentes con el campo de búsqueda desplegado, **CUANDO** borra todo lo escrito, **ENTONCES** se muestran todos los ponentes.
- **DADO** que un usuario se encuentra en la pantalla con la lista de los ponentes, **CUANDO** pulsa el botón de hacer una pregunta, **ENTONCES** se cambia a la pantalla con la lista de preguntas que se le han hecho al ponente, cada pregunta tiene: pregunta, contestación, número de comentarios y ‘likes’; también hay un botón para escribir una nueva pregunta.
- **DADO** que un usuario se encuentra en la pantalla de la lista de preguntas de un ponente, **CUANDO** pulsa el botón de añadir pregunta, **ENTONCES** se cambia a una pantalla con el formulario para enviar una pregunta al ponente.
- **DADO** que un usuario se encuentra en la pantalla con el formulario de enviar pregunta (da igual si ha iniciado sesión o no), **CUANDO** no escribe nada y pulsa el botón de enviar, **ENTONCES** se muestra un mensaje de error diciendo que tiene que escribir algo.

- **DADO** que un usuario se encuentra en la pantalla con el formulario para enviar una pregunta, **CUANDO** escribe una pregunta y pulsa el botón enviar, **ENTONCES** se envía la pregunta al ponente y queda registrada en la base de datos.
- **DADO** que un usuario se encuentra en la pantalla con la lista de preguntas que se le han hecho a un ponente, **CUANDO** pulsa sobre una de ellas, **ENTONCES** se muestra una pantalla con información detallada: la pregunta, contestación completa, número de comentarios y ‘likes’ y una lista de comentarios.
- **DADO** que un usuario se encuentra en la pantalla de los detalles de una pregunta, ha iniciado sesión y no ha dado ‘like’ a la respuesta, **CUANDO** toca el icono del corazón gris, **ENTONCES** se pone rojo y se registra que a dicho usuario le gusta la respuesta.
- **DADO** que un usuario se encuentra en la pantalla de los detalles de una pregunta, ha iniciado sesión y ha dado ‘like’ a la respuesta, **CUANDO** toca el icono del corazón rojo, **ENTONCES** se pone gris y se borra que a dicho usuario le gusta la respuesta.
- **DADO** que un usuario se encuentra en la pantalla de los detalles de una pregunta y no ha iniciado sesión, **CUANDO** toca el icono del corazón gris, **ENTONCES** no pasa nada.
- **DADO** que un usuario se encuentra en la pantalla de los detalles de una pregunta **CUANDO** toca el botón de comentar, **ENTONCES** se abre una pantalla con un formulario para enviar un comentario.
- **DADO** que un usuario se encuentra en la pantalla del formulario para enviar un comentario, **CUANDO** no ha escrito nada y pulsa enviar comentario, **ENTONCES** se muestra un mensaje de error.
- **DADO** que un usuario se encuentra en la pantalla con el formulario para comentar una respuesta y ha iniciado sesión, **CUANDO** escribe el comentario y pulsa el botón de enviar, **ENTONCES** el comentario se guarda en la base de datos con el nombre y la imagen del usuario y se mostrará en la lista de comentarios de la pregunta.
- **DADO** que un usuario se encuentra en la pantalla con el formulario para comentar una respuesta y no ha iniciado sesión, **CUANDO** escribe el comentario y pulsa el botón de enviar, **ENTONCES** el comentario se

guarda en la base de datos con el nombre “Anónimo” y la imagen del perfil por defecto y se mostrará en la lista de comentarios de la pregunta.

### 6.4.3 Resultados

En la ilustración 14 se pueden ver las pantallas que se han completado hasta el momento rodeadas por un rectángulo verde y las que se han empezado pero no están acabadas por uno amarillo.

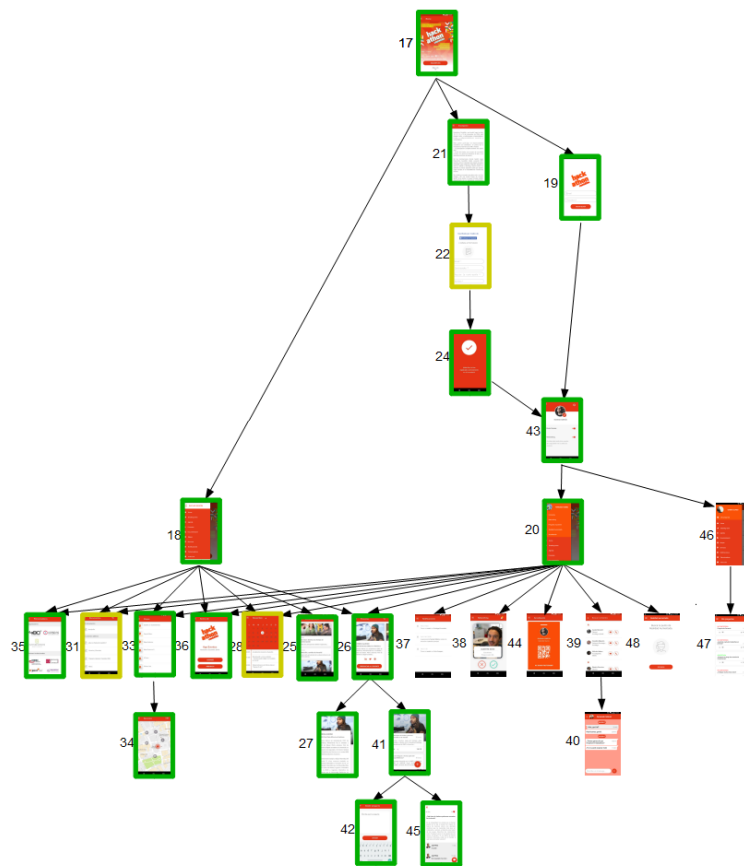


Ilustración 14: Pantallas hechas en verde y empezadas en amarillo hasta el Sprint 4
















## 6.5. Sprint 5 (24/04/2017 a 07/05/2017)

### 6.5.1 Resumen del sprint

Al inicio de este sprint Sergio Aguado nos informó de que las notificaciones debían ser implementadas mediante notificaciones PUSH de Firebase. Éstas llegan al dispositivo

incluso aunque éste no tenga abierta la notificación. Por lo tanto, el tiempo que se debe emplear en las notificaciones será mayor del esperado.

En la tabla 6 se muestra la actualización del estado de la pila del producto al inicio de este quinto sprint.

<b>ID</b>	<b>FUNCIONALIDAD</b>	<b>ESTADO</b>
1	Acceder a la información sobre la inscripción en el evento.	
2	Inscribirse en el evento mediante un formulario.	
3	Poder navegar por la aplicación gracias a un menú.	
4	Poder ver las últimas noticias actualizadas sobre el evento.	
5	Consultar la agenda con la planificación temporal del evento.	
6	Ver información sobre los ponentes del evento y hacerles preguntas.	
7	Descargar documentos en PDF con información más detallada del evento.	
8	Poder acceder a un mapa en el que estén indicados todos los servicios que ofrece la ciudad (hoteles, restaurantes, ...).	
9	Poder acceder a un mapa en el que estén indicados todos los sitios importantes del evento.	
10	Recibir notificaciones sobre algún cambio en el calendario o alguna interacción con otro usuario.	
11	Poder consultar los patrocinadores del evento.	
12	Poder consultar la información legal del evento.	
13	Una vez creada la cuenta, acceder al perfil.	
14	Una vez creada la cuenta, modificar al perfil.	
15	Obtener una acreditación que permita participar en el evento.	




16	Interactuar con otros usuarios y tenerlos en tu lista de contactos.	
17	Una vez el usuario se ha registrado, puede iniciar sesión.	
18	Dar feedback a los administradores con quejas o sugerencias.	

Tabla 6: Pila del producto al inicio del Sprint 5

A partir del estado de la pila del producto se planificaron las siguientes tareas para ser implementadas durante este quinto sprint.

### **Inscribirse en el evento mediante un formulario.**

- Actualizar la información de los spinners: país, comunidades y provincias (2 horas).
- Permitir el registro mediante la cuenta de Facebook: email, nombre, apellidos e imagen de perfil (6 horas).

Tiempo total: 8 horas → 2 puntos de historia (10 horas).

### **Consultar la agenda con la planificación temporal del evento.**

- Permitir que las noticias se puedan añadir al calendario personal con una alarma en el día y hora que corresponda (2 horas).

Tiempo total: 2 horas → 1/2 puntos de historia (2 horas y 30 minutos).

### **Descargar documentos en PDF con información más detallada del evento.**

- Poder descargar y ver los documentos (2 horas y 30 minutos).
- Generar un PDF personalizado cuando un usuario se inscribe (1 hora).

Tiempo total: 3 horas → 1 puntos de historia (5 horas).

### **Modificar perfil.**

- Permitir al usuario cambiar todos sus datos excepto el email y la contraseña, que podrá ser restablecida en el siguiente sprint (1 hora).
- Permitir al usuario modificar su imagen de perfil (1 hora).

Tiempo total: 2 horas → 1 puntos de historia (5 horas).

### **Interactuar con otros usuarios y tenerlos en tu lista de contactos.**

- Crear la interfaz en la que se mostrarán los posibles contactos (1 hora).
- Mostrar un usuario aleatorio que el usuario no tenga en contactos. También hay que tener en cuenta si el otro usuario tiene activada la opción de hacer contactos (3 horas).
- Crear pantalla de aceptación de contacto (1 hora).
- Crear pantalla de rechazo de contacto (1 hora).
- Crear pantalla de contactos interesados, cuando ambos aceptan (1 hora).
- Crear la interfaz con la lista de los contactos que el usuario ha hecho hasta el momento (2 horas).
- Permitir que el usuario pueda enviar un email o llamar a cualquiera de sus contactos (2 horas).

Tiempo total: 11 horas → 3 puntos de historia (15 horas).

### **Acreditación.**

- Crear la interfaz de la acreditación y mostrar los datos del usuario: nombre, apellidos, empresa e imagen (30 minutos).
- Generar un código QR personalizado y único (15 minutos).
- Generar un ID único para el usuario (15 minutos).

Tiempo total: 1 hora → 1/2 punto de historia (2 horas y 30 minutos).

**Recibir notificaciones sobre algún cambio en el calendario o alguna interacción con otro usuario (inacabada).**



- Buscar información y aprender a usar las notificaciones PUSH de Firebase (3 horas).
- Crear un backend programado en JavaScript usando Node que detecte cuándo hay un cambio en un evento para notificar a todos los usuarios con el mensaje correspondiente (6 horas).

Tiempo total: 9 horas → 2 puntos de historia (10 horas).

En total se han asignado 10 puntos de historia, que equivalen a 10 días o a 50 horas y 30 minutos. La tarea de inscripción se alarga más de lo esperado en un principio debido a los spinners y a la autenticación mediante Facebook que en principio no se iba a implementar. También lo ha hecho la de los documentos, ha fallado más de lo esperado. Por el contrario, la de modificar el perfil se hizo más corta ya que el formulario es igual que el de inscripción. También fue más rápida la parte de hacer contactos, es larga pero no tanto como se esperó en un principio. Finalmente, como ya se ha comentado al principio de este apartado, el cambio a las notificaciones PUSH supuso más tiempo del esperado. Se implementó parcialmente ya que las notificaciones generales son más sencillas y tienen menos condiciones que las personales. En el siguiente sprint se acabará de implementar este pequeño backend.

### 6.5.2 Pruebas de aceptación

A continuación se muestran las pruebas de aceptación que se han pasado satisfactoriamente al final de este quinto sprint.

- **DADO** que un usuario se encuentra en el formulario de inscripción, **CUANDO** selecciona un elemento del spinner, **ENTONCES** los demás se actualizan correctamente (por ejemplo, si elige la Comunidad Valencia, las provincias solo muestran Alicante, Castellón y Valencia).
- **DADO** que un usuario se encuentra en el formulario de inscripción, **CUANDO** elige la opción de registrarse mediante Facebook y acepta las condiciones, **ENTONCES** se registra con la información de su perfil de Facebook (nombre, apellidos, correo e imagen de perfil).
- **DADO** que un usuario se encuentra en la pantalla de la información de un evento de la agenda, **CUANDO** pulsa el botón de añadir al calendario pero

no da permisos, **ENTONCES** se muestra un mensaje que indica que tiene que dar permisos para poder añadirlo.

- **DADO** que un usuario se encuentra en la pantalla de la información de un evento de la agenda, **CUANDO** pulsa el botón de añadir al calendario y da permisos a la aplicación, **ENTONCES** el evento se añade al calendario personal del dispositivo con una alarma para el día y hora del evento
- **DADO** que un usuario ha iniciado sesión, **CUANDO** va a la pantalla de los documentos, **ENTONCES** se muestra el documento de su inscripción y la lista de los públicos.
- **DADO** que un usuario no ha iniciado sesión, **CUANDO** va a la pantalla de los documentos, **ENTONCES** solo se muestra la lista de documentos públicos.
- **DADO** que un usuario se encuentra en la pantalla de los documentos, **CUANDO** pulsa sobre uno de los documentos, **ENTONCES** se abre dicho documento en la aplicación que elija el usuario (Drive, Adobe, ...).
- **DADO** que un usuario se encuentra en la pantalla del formulario de inscripción con todos los datos necesarios introducidos, **CUANDO** pulsa el botón de registro, **ENTONCES** se genera un PDF personalizado con la información de su inscripción y se guarda en la base de datos.
- **DADO** que un usuario ha iniciado sesión y se encuentra en la pantalla del perfil, **CUANDO** pulsa el botón de editar perfil (un lápiz), **ENTONCES** se muestra una pantalla con un formulario muy similar al de inscripción.
- **DADO** que un usuario ha iniciado sesión y se encuentra en la pantalla con el formulario de edición del perfil, **CUANDO** no introduce todos los campos necesarios y pulsa el botón de editar perfil, **ENTONCES** se muestra el mensaje de error correspondiente.
- **DADO** que un usuario ha iniciado sesión y se encuentra en la pantalla con el formulario de edición del perfil, **CUANDO** introduce todos los campos necesarios y pulsa el botón de editar perfil, **ENTONCES** se actualiza la información en la base de datos y se le redirige a la pantalla del perfil que ya tiene la información actualizada.
- **DADO** que un usuario ha iniciado sesión y se encuentra en la pantalla del perfil, **CUANDO** pulsa sobre la imagen del perfil, **ENTONCES** se abre la galería del dispositivo, si elige una imagen su foto de perfil se actualiza.

- **DADO** que un usuario ha iniciado sesión y no ha rechazado/aceptado a todos los demás usuarios, **CUANDO** abre la sección de “Networking”, **ENTONCES** se muestran usuarios aleatorios a los que agregar.
- **DADO** que un usuario ha iniciado sesión y ha rechazado/aceptado a todos los demás usuarios, **CUANDO** abre la sección de “Networking”, **ENTONCES** se muestra un mensaje diciendo que no hay más usuarios con los que relacionarse y se le redirige a su perfil.
- **DADO** que un usuario ha iniciado sesión, se encuentra en el apartado de “Networking” y hay usuarios a los que aceptar/rechazar, **CUANDO** rechaza a un usuario, **ENTONCES** se guardan las preferencias en la base de datos, no vuelve a aparecer en su lista, el propio usuario no le aparecerá como usuario aleatorio al usuario rechazado y se muestra un mensaje con esta información.
- **DADO** que un usuario ha iniciado sesión, se encuentra en el apartado de “Networking” y hay usuarios a los que aceptar/rechazar, **CUANDO** acepta a un usuario que no le ha aceptado, **ENTONCES** se guardan las preferencias en la base de datos y se muestra un mensaje de que está a la espera de que el otro usuario acepte.
- **DADO** que un usuario ha iniciado sesión, se encuentra en el apartado de “Networking” y hay usuarios a los que aceptar/rechazar, **CUANDO** acepta a un usuario que ya le ha aceptado, **ENTONCES** se guardan las preferencias en la base de datos y se muestra un mensaje de que son contactos interesados, ahora el usuario se ha añadido a su lista de contactos.
- **DADO** que un usuario ha iniciado sesión y no tiene contactos, **CUANDO** accede a la sección “Contactos”, **ENTONCES** se muestra un mensaje de que no tiene contactos.
- **DADO** que un usuario ha iniciado sesión y tiene al menos un contacto, **CUANDO** accede a la sección “Contactos”, **ENTONCES** se muestra una lista con todos sus contactos, para cada uno se muestra: imagen de perfil, nombre, apellidos, comunidad, ciudad, un botón para llamarle y otro para enviar un email.
- **DADO** que un usuario ha iniciado sesión, tiene al menos un contacto y se encuentra en la sección “Contactos”, **CUANDO** pulsa el botón de enviar un email, **ENTONCES** se abre un diálogo para que el usuario elija la

aplicación con la que enviarlo y se abre un correo con el email del receptor y como asunto “Hackathon”.

- **DADO** que un usuario ha iniciado sesión, tiene al menos un contacto y se encuentra en la sección “Contactos”, **CUANDO** pulsa el botón de llamar y no da permisos, **ENTONCES** se muestra un mensaje indicando que si no da permisos no puede llamar.
- **DADO** que un usuario ha iniciado sesión, tiene al menos un contacto y se encuentra en la sección “Contactos”, **CUANDO** pulsa el botón de llamar y da permisos, **ENTONCES** se abre el teléfono del dispositivo con el número del contacto.
- **DADO** que un usuario ha iniciado sesión, tiene al menos un contacto y se encuentra en la sección “Contactos”, **CUANDO** pulsa el icono de la lupa se muestra un campo de texto para buscar contactos, **ENTONCES** se muestran los contactos que cumplen el filtro (o todos si no se ha escrito nada en el campo de búsqueda).
- **DADO** que un usuario ha iniciado sesión, **CUANDO** accede a la sección “Acreditación”, **ENTONCES** se muestra una pantalla con sus datos, un ID único y un código QR también único.
- **DADO** que un usuario ha abierto la aplicación al menos una vez en su dispositivo, **CUANDO** hay algún cambio en el calendario, **ENTONCES** se envía una notificación al dispositivo con la información del cambio aunque la aplicación esté cerrada.
- **DADO** que un usuario usa la aplicación y ha habido algún cambio en el calendario, **CUANDO** abre la sección “Notificaciones”, **ENTONCES** se añade a la lista de notificaciones.

### 6.5.3 Resultados

En la ilustración 15 se pueden ver las pantallas que se han completado hasta el momento rodeadas por un rectángulo verde y las que se han empezado pero no están acabadas por uno amarillo.

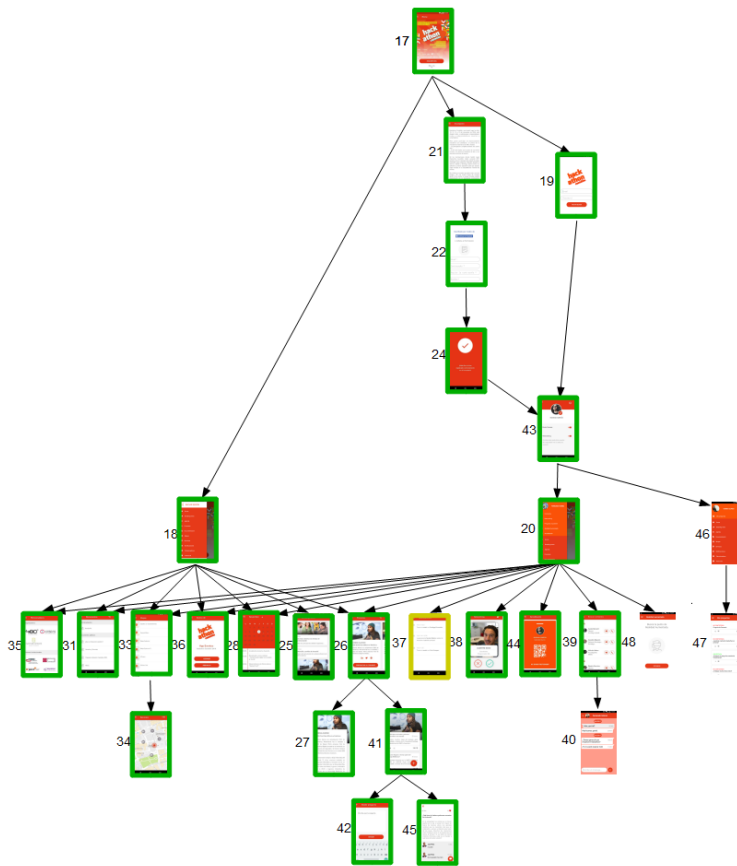


Ilustración 15: Pantallas hechas en verde y empezadas en amarillo hasta el Sprint 5

## 6.6. Sprint 6 (08/05/2017 a 21/05/2017)

### 6.6.1 Resumen del sprint

En la tabla 7 se muestra la actualización del estado de la pila del producto al inicio de este sexto y último sprint.

ID	FUNCIONALIDAD	ESTADO
1	Acceder a la información sobre la inscripción en el evento.	✓
2	Inscribirse en el evento mediante un formulario.	✓
3	Poder navegar por la aplicación gracias a un menú.	✓
4	Poder ver las últimas noticias actualizadas sobre el evento.	✓


















5	Consultar la agenda con la planificación temporal del evento.	
6	Ver información sobre los ponentes del evento y hacerles preguntas.	
7	Descargar documentos en PDF con información más detallada del evento.	
8	Poder acceder a un mapa en el que estén indicados todos los servicios que ofrece la ciudad (hoteles, restaurantes, ...).	
9	Poder acceder a un mapa en el que estén indicados todos los sitios importantes del evento.	
10	Recibir notificaciones sobre algún cambio en el calendario o alguna interacción con otro usuario.	
11	Poder consultar los patrocinadores del evento.	
12	Poder consultar la información legal del evento.	
13	Una vez creada la cuenta, acceder al perfil.	
14	Una vez creada la cuenta, modificar al perfil.	
15	Obtener una acreditación que permita participar en el evento.	
16	Interactuar con otros usuarios y tenerlos en tu lista de contactos.	
17	Una vez el usuario se ha registrado, puede iniciar sesión.	
18	Dar feedback a los administradores con quejas o sugerencias.	
19	Chat privado que permite comunicarse a un usuario con sus contactos.	
20	Implementar la parte del ponente para contestar preguntas.	
21	Realidad aumentada.	

Tabla 7: Pila del producto al inicio del Sprint 6

A partir del estado de la pila del producto se planificaron las siguientes tareas para ser implementadas durante este sexto sprint y así dar por acabado el proyecto. Hay que destacar que de la parte de realidad aumentada simplemente se implementará la vista, no

habrá funcionalidad. Esto lo decidió el supervisor, Sergio Aguado, porque no lo consideraba una parte fundamental de la aplicación.

**Recibir notificaciones sobre algún cambio en el calendario o alguna interacción con otro usuario.**

- Crear un backend programado en JavaScript usando Node que detecte cuándo dos usuarios se añaden a contactos para enviar la notificación a ambos (5 horas).
- Implementar las funciones correspondientes en la aplicación para que sean llamadas por Firebase cuando haya una notificación y la lancen (1 hora).

Tiempo total: 6 horas → 2 puntos de historia (10 horas).

**Una vez el usuario se ha registrado, puede iniciar sesión.**

- Modificar la interfaz de login para que haya un enlace que ponga “¿Has olvidado tu contraseña?” (10 minutos).
- Funcionamiento que mande un correo para cambiar la contraseña mediante Firebase (50 minutos).

Tiempo total: 1 horas → 1/2 punto de historia (2 horas y 30 minutos).

**Chat privado que permite comunicarse a un usuario con sus contactos.**

- Crear la interfaz del chat, muy similar a WhatsApp: imagen del contacto, nombre, apellidos y evidentemente la lista de mensajes y un campo para enviar uno (3 horas).
- Administrar el envío de mensajes aprovechando la base de datos en tiempo real que ofrece Firebase (6 horas).

Tiempo total: 9 horas → 2 puntos de historia (10 horas).

### **Implementar la parte del ponente para contestar preguntas.**

- Controlar que el menú solo muestra a lo que un ponente puede acceder cuando inicia sesión (30 minutos).
- Crear una lista que muestre las preguntas que tiene contestadas y sin contestar (1 hora).
- Crear la funcionalidad de contestar una pregunta (30 minutos).

Tiempo total: 1 horas → 1/2 punto de historia (2 horas).

### **Realidad aumentada.**

- Crear la interfaz siguiendo el diseño establecido (10 minutos).

Tiempo total: 10 minutos → 0 puntos de historia (0 minutos).

En total se han asignado 5 puntos de historia, que equivalen a 5 días o a 25 horas. El resto del tiempo se empleará a alguna de estas tres tareas si no se han completado a tiempo y a optimizar y refactorizar código que pueda estar repetido o ser ineficiente.

### **6.6.2 Pruebas de aceptación**

A continuación se muestran las pruebas de aceptación que se han pasado satisfactoriamente al final del sexto sprint.

- **DADO** que un usuario ha iniciado sesión y está en la sección de “Networking”, **CUANDO** acepta a un usuario que ya le ha aceptado, **ENTONCES** se añade a la lista de contactos, se envía una notificación al dispositivo y se añade dicha notificación a la lista de notificaciones.
- **DADO** que un usuario ha iniciado sesión y ha aceptado a algún usuario en la sección “Networking”, **CUANDO** dicho usuario le acepta, **ENTONCES** se añade a la lista de contactos, se envía una notificación al dispositivo aunque no tenga la aplicación abierta y se añade dicha notificación a la lista de notificaciones.



- **DADO** que un usuario está en la pantalla de login, quiere iniciar sesión pero se ha olvidado de su contraseña y no ha escrito nada en el campo email, **CUANDO** pulsa el enlace “¿Has olvidado tu contraseña?”, **ENTONCES** se muestra un mensaje de error diciendo que debe introducir su email.
- **DADO** que un usuario está en la pantalla de login, quiere iniciar sesión pero se ha olvidado de su contraseña y ha escrito en el campo email, **CUANDO** pulsa el enlace “¿Has olvidado tu contraseña?”, **ENTONCES** se envía un correo a la dirección especificada para cambiar la contraseña.
- **DADO** que un usuario recibe el correo de cambio de contraseña y lo ignora, **CUANDO** intenta iniciar sesión con sus credenciales antiguas, **ENTONCES** accede correctamente ya que no hay nada que haya cambiado.
- **DADO** que un usuario recibe el correo de cambio de contraseña y la cambia, **CUANDO** intenta iniciar sesión con sus credenciales antiguas, **ENTONCES** se muestra un mensaje de error.
- **DADO** que un usuario recibe el correo de cambio de contraseña y la cambia, **CUANDO** intenta iniciar sesión con sus nuevas credenciales, **ENTONCES** inicia sesión correctamente.
- **DADO** que un usuario ha iniciado sesión, tiene al menos un contacto y está en la sección de “Contactos”, **CUANDO** pulsa sobre un contacto, **ENTONCES** se abre un chat privado que contiene: un botón para volver atrás, una foto del contacto, el nombre y apellidos de éste, un campo de texto para escribir mensajes, un botón para enviar un mensaje y una lista con todos los mensajes (vacía si no hay ninguno).
- **DADO** que un usuario ha iniciado sesión y está la pantalla de un chat privado, **CUANDO** envía un mensaje, **ENTONCES** se añade a la lista de mensajes como enviado por él o ella, con la fecha y hora y le llega al otro usuario.
- **DADO** que un usuario ha iniciado sesión y está la pantalla de un chat privado, **CUANDO** envía un mensaje en un día diferente al del último mensaje, **ENTONCES** se añade a la lista de mensajes como enviado por él o ella, con la fecha y hora, se añade un separador que indica el día, mes y año y le llega al otro usuario.

- **DADO** que un usuario ha iniciado sesión con una cuenta de ponente, **CUANDO** abre el menú, **ENTONCES** solo se muestran las secciones a las que tiene acceso.
- **DADO** que un usuario ha iniciado sesión con una cuenta de ponente, **CUANDO** abre la sección “Mis preguntas”, **ENTONCES** se muestran las preguntas que se le han hecho indicando si las ha contestado o no.
- **DADO** que un usuario ha iniciado sesión con una cuenta de ponente y se encuentra en la sección “Mis preguntas”, **CUANDO** toca una de las preguntas, **ENTONCES** se muestra toda la información de la pregunta: comentarios, likes y un botón para responder.
- **DADO** que un usuario ha iniciado sesión con una cuenta de ponente y se encuentra en la pantalla de información de una de sus preguntas, **CUANDO** toca el botón de contestar, **ENTONCES** se muestra una pantalla en la que escribir su contestación.
- **DADO** que un usuario ha iniciado sesión con una cuenta de ponente y se encuentra en la pantalla de contestar una pregunta, **CUANDO** no escribe nada y toca el botón de contestar, **ENTONCES** se muestra un mensaje de error.
- **DADO** que un usuario ha iniciado sesión con una cuenta de ponente y se encuentra en la pantalla de contestar una pregunta, **CUANDO** escribe algo y toca el botón de contestar, **ENTONCES** se guarda la respuesta y se vuelve a la pantalla de información de la pregunta.
- **DADO** que un usuario ha iniciado sesión, **CUANDO** accede a la sección “Realidad aumentada”, **ENTONCES** se muestra una pantalla en la que hay un botón para escanear que no hace nada.

### 6.6.3 Resultados

En la ilustración 16 se pueden ver las pantallas que se han completado hasta el momento rodeadas por un rectángulo verde. Es el final del proyecto y están todas completadas al 100%.

## 6.7 Pila final del producto (21/05/2017)

En la tabla 8 se muestra la pila del producto al finalizar el sexto y último sprint. Como se puede apreciar, se han completado todas las tareas previstas al iniciar el proyecto más 3 tareas extra, además de las ya existentes que se han modificado durante el desarrollo de la aplicación.

<b>ID</b>	<b>FUNCIONALIDAD</b>	<b>ESTADO</b>
1	Acceder a la información sobre la inscripción en el evento.	✓
2	Inscribirse en el evento mediante un formulario.	✓
3	Poder navegar por la aplicación gracias a un menú.	✓
4	Poder ver las últimas noticias actualizadas sobre el evento.	✓
5	Consultar la agenda con la planificación temporal del evento.	✓
6	Ver información sobre los ponentes del evento y hacerles preguntas.	✓
7	Descargar documentos en PDF con información más detallada del evento.	✓
8	Poder acceder a un mapa en el que estén indicados todos los servicios que ofrece la ciudad (hoteles, restaurantes, ...).	✓
9	Poder acceder a un mapa en el que estén indicados todos los sitios importantes del evento.	✓
10	Recibir notificaciones sobre algún cambio en el calendario o alguna interacción con otro usuario.	✓
11	Poder consultar los patrocinadores del evento.	✓
12	Poder consultar la información legal del evento.	✓
13	Una vez creada la cuenta, acceder al perfil.	✓
14	Una vez creada la cuenta, modificar al perfil.	✓

15	Obtener una acreditación que permita participar en el evento.	✓
16	Interactuar con otros usuarios y tenerlos en tu lista de contactos.	✓
17	Una vez el usuario se ha registrado, puede iniciar sesión.	✓
18	Dar feedback a los administradores con quejas o sugerencias.	✓
19	Chat privado que permite comunicarse a un usuario con sus contactos.	✓
20	Implementar la parte del ponente para contestar preguntas.	✓
21	Realidad aumentada.	✓

Tabla 8: Pila del producto al finalizar el proyecto

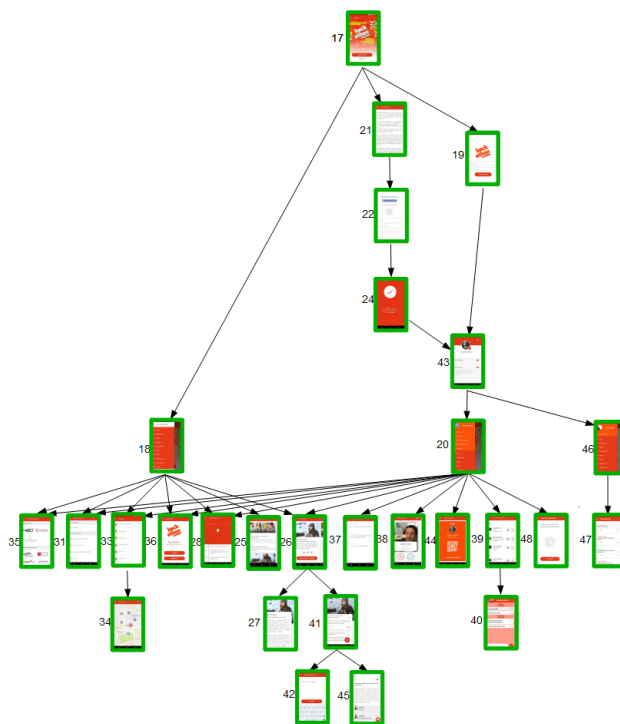


Ilustración 16: Pantallas hechas en verde en el Sprint 6

## Capítulo 7

# Futuras extensiones

### Índice

---

7.1. Futuras extensiones de la aplicación .....	77
7.2. Otras aplicaciones complementarias .....	77

---

### 7.1 Futuras extensiones de la aplicación

Para empezar, al final del proyecto ha sobrado tiempo para implementar una funcionalidad extra que no se había previsto al inicio del proyecto, el chat. Permite enviar mensajes privados entre usuarios, muy similar a WhatsApp.

Además de lo propuesto al inicio del proyecto y de las extensiones ya implementadas, hay una serie de mejoras y extensiones que se le pueden hacer a la aplicación:

- Puntuar al ponente.
- Bloquear usuario.
- Consultar información de los usuarios.
- Ver fotos de las instalaciones y los sitios de interés.
- Buscar usuarios.
- Realidad aumentada.

### 7.2 Otras aplicaciones complementarias

Para que la aplicación tenga sentido y pueda comercializarse, sería necesario implementar una aplicación más: la del administrador.

La primera serviría para cambiar la fecha y la hora de los eventos o añadir nuevos. También permitiría añadir nuevos ponentes, nuevas instalaciones, nuevos sitios de

interés, nuevos patrocinadores y noticias sobre el evento. Además, se podría consultar el feedback proporcionado por los usuarios.

De todas formas, la aplicación sirve para dar información al usuario y podría implementarse en cualquier plataforma. No es estrictamente necesario que sea una aplicación móvil, podría ser una web o incluso una aplicación de escritorio.

## Capítulo 8

# Conclusiones

### Índice

---

8.1. Resumen .....	79
8.2. Objetivos logrados .....	79
8.3. Valoración personal .....	80

---

### 8.1 Resumen

En este documento se detalla el proyecto realizado en las 300 horas de la estancia en prácticas en la empresa Cuatroochenta S.L, del 27 de febrero de 2017 al 21 de mayo de 2017. El proyecto consiste en la implementación de una aplicación Android para la gestión de eventos y congresos. A lo largo del documento se explica el ámbito de la empresa, la aplicación a implementar, las tecnologías usadas, el diseño de la aplicación, cómo se ha desarrollado y las posibles mejoras del sistema.

### 8.2 Objetivos logrados

Al acabar la estancia se han conseguido todos los objetivos propuestos antes de empezar las prácticas. El objetivo principal era programar una aplicación Android para la gestión de eventos y congresos.

Además, se han conseguido otros objetivos: aprender a programar aplicaciones Android y dominar el uso de bases de datos NoSQL. La aplicación desarrollada está al nivel exigido por la empresa para ser publicada. Esto indica que el nivel adquirido en programación Android es bueno y suficiente para crear una aplicación competente desde cero. Por otro lado, también se ha adquirido un buen nivel en bases de datos NoSQL y, más concretamente, en Firebase.

## 8.3 Valoración personal

Estoy muy satisfecho tanto con los conocimientos adquiridos como con el trato de la empresa. Se me ha tenido en cuenta desde el primer día y los trabajadores de Cuatroochenta se han mostrado dispuestos a integrarme en el grupo. Además, cualquier duda que me ha surgido ha sido resuelta de inmediato por Sergio Aguado o por cualquier otro compañero de la sección de Android.

En cuanto a los conocimientos adquiridos, estoy muy contento con las tecnologías aprendidas. Me han gustado e interesado desde el primer momento, además son modernas y están muy demandadas hoy en día. La programación para Android se ha hecho más llevadera por el hecho de saber Java. Avancé bastante rápido desde el principio, esto no habría sido posible si hubiera sido otro lenguaje que no conozco. Por otro lado, las bases de datos NoSQL no se estudian en la carrera pero cada vez se usan más y más, sobre todo para bases de datos con grandes cantidades de información que necesitan un tiempo de consulta bajo. Es muy útil dominar diversos tipos de bases de datos. También he de decir que al haber aprendido SQL se hace extraño que no haya relaciones y que se repitan datos, me costó un poco adaptarme al principio.

Finalmente, decir que tengo intención de seguir usando estas tecnologías para ampliar mis conocimientos, ya sea en mi tiempo libre o trabajando para una empresa. He adquirido un buen nivel en estos tres meses, pero es evidente que una tecnología no se domina completamente en tan poco tiempo y que aún me quedan muchísimas cosas por aprender.



# Bibliografía

[1] “Soluciones Cuatroochenta S.L.”, última vez visitado el 12 de mayo del 2017, <http://www.cuatroochenta.com>

[2] “Digital in 2016”, última vez visitado el 8 de junio del 2017, la imagen se encuentra en este enlace <http://www.bialarblog.com/espana-digital-2016/>, pero el artículo original es este <https://wearesocial.com/special-reports/digital-in-2016>

[3] “Android”, última vez visitado el 12 de mayo del 2017, <https://es.wikipedia.org/wiki/Android>

[4] “Arquitectura del software”, última vez visitado el 2 de mayo del 2017, [https://es.wikipedia.org/wiki/Arquitectura\\_de\\_software](https://es.wikipedia.org/wiki/Arquitectura_de_software)

[5] “Arquitectura Android”, última vez visitado el 26 de abril del 2017, <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

[6] “Aplicaciones en Android”, última vez visitado el 12 de mayo de 2017, <https://sites.google.com/site/swcuc3m/home/android/generalidades/aplicacionespag3>

[7] “Golden Rules of User Interface Design”, última vez visitado el 13 de mayo de 2017, <http://theomandel.com/resources/golden-rules-of-user-interface-design/>

[8] “Singleton”, última vez visitado el 17 de mayo de 2017, <https://es.wikipedia.org/wiki/Singleton>



## Anexo A

# Modelo de datos

Como ya se ha comentado anteriormente, las bases de datos NoSQL no tienen relaciones entre las tablas y la información puede repetirse. A continuación se muestra una lista con las tablas y los atributos de la base de datos creada para este proyecto.

- Chats (contiene la información de los chats)
  - chatId (id único de un chat)
    - messageId (id único de un mensaje de un chat)
      - message (texto enviado)
      - senderId (id del usuario que lo envía)
      - receiverId (id del usuario al que va dirigido)
      - time (día y hora en el que se envía)
- Documents (contiene los documentos informativos del evento)
  - title (título del documento)
  - href (referencia a la ubicación del documento)
- Events (contiene los eventos que se muestran en el calendario)
  - title (título del evento)
    - address (dirección y sala donde se realiza)
    - dayOfWeek (día de la semana en el que se realiza)
    - endTime (hora a la que acaba)
    - image (referencia a la ubicación de la imagen, acepta nulos)
    - info (descripción del evento)
    - startTime (hora a la que empieza)
    - time (día y hora a la que tiene lugar)
- Feedback (contiene las sugerencias y quejas de los usuarios)
  - feedback (mensaje enviado por el usuario)
  - type (tipo de feedback: queja o sugerencia)
- Information (contiene los textos largos con información del evento)
  - title (título de la información)

- text (texto de la información)
- Maps (contiene los lugares del evento, ya sean instalaciones o servicios)
  - name (nombre del lugar)
    - address (dirección del lugar)
    - description (descripción del lugar)
    - latitude (latitud en la que se encuentra)
    - longitude (longitud en la que se encuentra)
    - phone (teléfono del lugar)
    - type (tipo de lugar: hotel, restaurante, tienda, o instalación)
- Matches (contiene la información de la interacción entre usuarios)
  - matchId (id único de la interacción entre dos usuarios)
    - answer1 (booleano que indica si el usuario 1 ha contestado)
    - answer2 (booleano que indica si el usuario 2 ha contestado)
    - match1 (booleano que indica si el usuario 1 ha aceptado)
    - match2 (booleano que indica si el usuario 2 ha aceptado)
    - uid1 (id del usuario 1)
    - uid2 (id del usuario 2)
- News (contiene la información de las noticias del evento)
  - title (título de la noticia)
    - image (referencia a la ubicación de la imagen, acepta nulos)
    - info (texto con la información de la noticia)
    - time (día y hora en la que se publica la noticia)
- Notifications (contiene la información de las notificaciones públicas)
  - title (título de la notificación)
  - time (día y hora en la que se genera la notificación)
- NotificationsUser (contiene las notificaciones de los usuarios)
  - userId (id del usuario)
    - title (título de la notificación)
    - time (día y hora en la que se genera la notificación)
- Speakers (contiene la información de los ponentes del evento)
  - name (nombre del ponente)
  - description (descripción del ponente)
  - image (referencia a la ubicación de la imagen del ponente)
  - job (puesto de trabajo del ponente)

- linkedin (enlace al perfil de LinkedIn del ponente)
- questions (contiene las preguntas que se le han realizado al ponente)
  - answer (contestación del ponente a la pregunta)
  - comments (contiene los comentarios de los usuarios sobre la respuesta dada por el ponente)
    - commentId (id único de un comentario)
    - comment (texto del comentario)
    - name (nombre y apellidos del usuario que comenta)
    - time (día y hora en la que hace el comentario)
    - userImage (referencia a la ubicación de la imagen del usuario)
  - likes (contiene todos los usuarios que le han dado me gusta)
    - userId (id del usuario que ha dado me gusta)
  - time (día y hora en la que se hace la pregunta)
  - title (la pregunta)
- speakerId (id único del ponente)
- twitter (enlace al perfil de Twitter del ponente)
- website (enlace al sitio web del ponente)
- Tokens (contiene la información sobre los tokens registrados)
  - token (token único de un dispositivo concreto)
  - userId (id del usuario que ha iniciado sesión en dicho dispositivo)
- Users (contiene la información de los usuarios)
  - userId (id único de un usuario)
  - city (ciudad en la que vive el usuario)
  - companyName (empresa en la que trabaja)
  - country (país en el que vive)
  - department (departamento de la empresa en el que trabaja)
  - email (correo electrónico)
  - fact (booleano para saber si desea facturar o no)
  - image (referencia a la ubicación de la imagen de perfil)
  - name (nombre)
  - networking (booleano para saber si desea hacer contactos o no)
  - nif (DNI)

- `phoneNumber` (número de teléfono)
- `position` (puesto que ocupa en la empresa)
- `postalCode` (código postal de donde vive)
- `sector` (sector de la empresa en el que trabaja)
- `snooze` (booleano para saber si desea recibir notificaciones o no)
- `speaker` (booleano para saber si es un ponente o no)
- `state` (comunidad en la que vive)
- `surname` (apellidos)
- `website` (sitio web)

## Anexo B

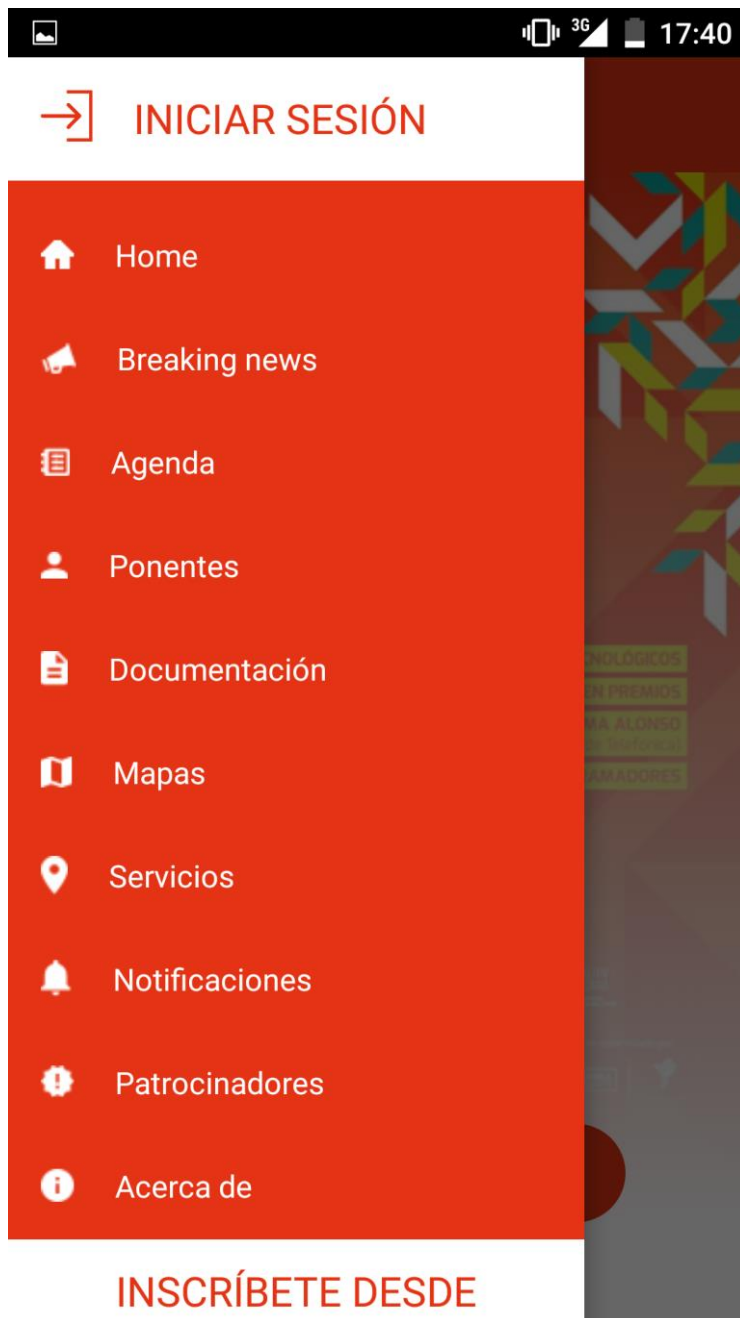
# Capturas de pantalla

En este anexo se muestran todas las pantallas de la aplicación para mostrar la interfaz gráfica creada.



### Hackathon Castellón 2016

*Ilustración 17: Página principal*



*Ilustración 18: Menú sin iniciar sesión*



# hack athon castellón

INICIAR SESIÓN

*Ilustración 19: Formulario para iniciar sesión*

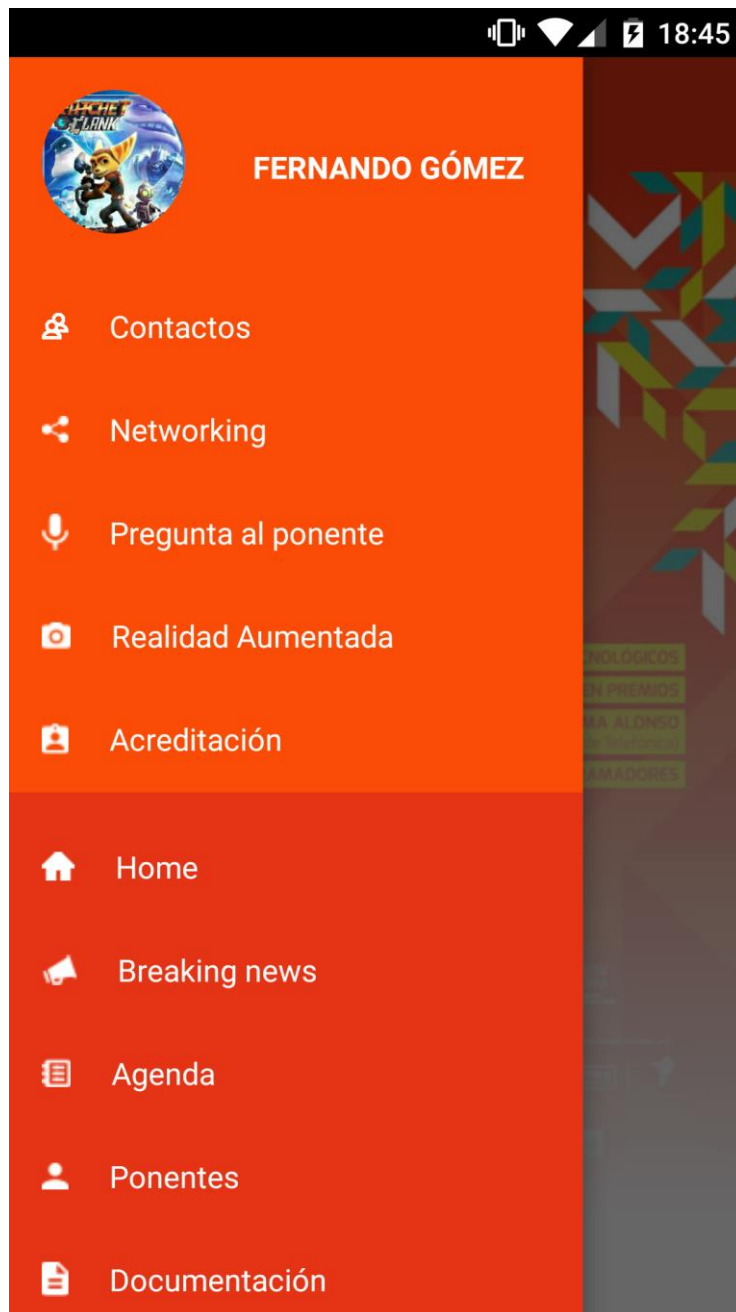


Ilustración 20: Menú con sesión iniciada



Hackathon Castellón, que tendrá lugar los días 10, 11, 12 y 13 de noviembre de 2016, está dirigido tanto a profesionales emprendedores como a emprendedores noveles y universitarios.

Para poder participar es imprescindible inscribirse con antelación. El precio de la inscripción al evento es de 10€ e incluye:

- La participación completa durante los cuatro días.
- El Kit del Hacker, que consta de una bolsa mochila, una pulsera acreditativa de tela y una camiseta exclusiva del evento.

En las instalaciones donde tendrá lugar Hackathon Castellón habrá conexión WiFi gratuita, pero es necesario que traigas tu propio ordenador para programar. Además, durante el viernes, sábado y domingo, habrá servicio de cafetería a precios populares para que puedas coger fuerzas sin la necesidad de moverte del recinto.

No olvides que puedes apuntarte solo o con tus amigos. ¡Es una buena forma de conocer gente! Si ya tenéis un grupo formado, podéis especificarlo en el formulario escribiendo, individualmente, el nombre del grupo. Si no lo

*Ilustración 21: Información de la inscripción*

Inscríbete por medio de

 Continuar con Facebook

o rellena el formulario



Ilustración 22: Formulario de la inscripción, parte 1



Nombre

Apellidos

País



Estado



Ciudad



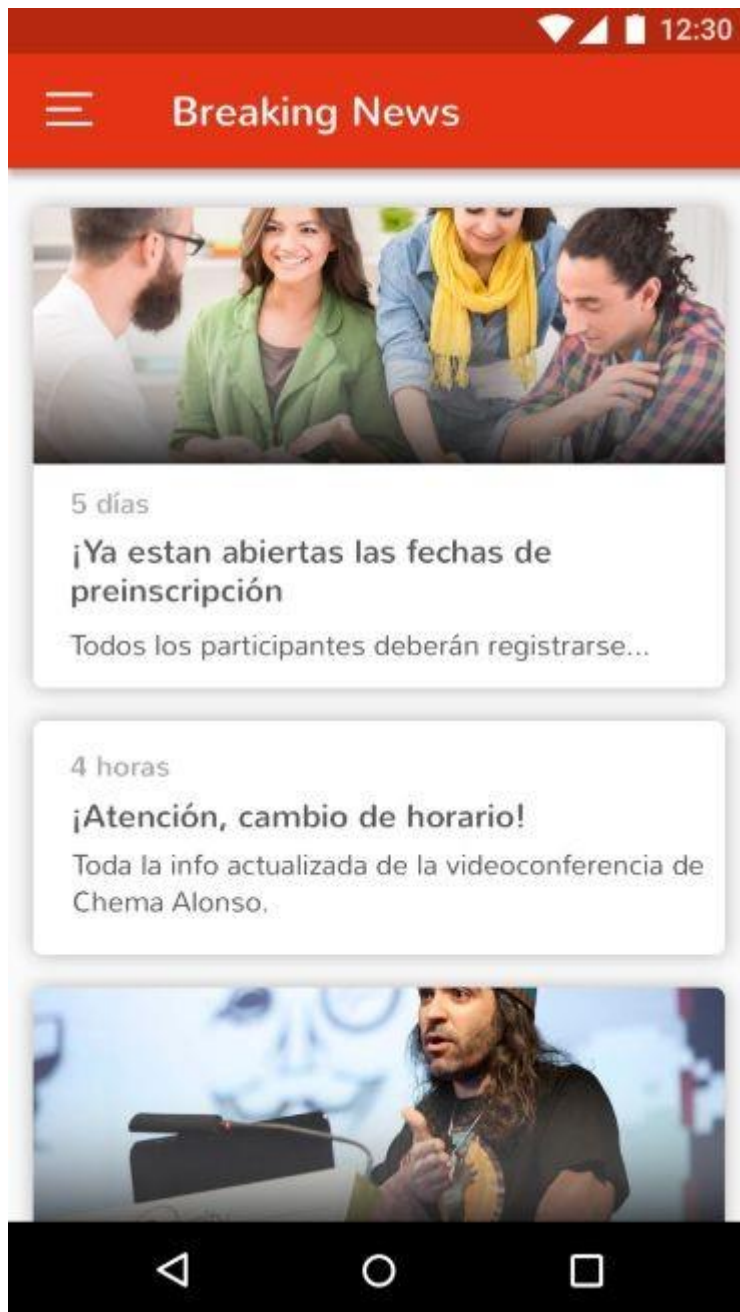
Codigo Postal

Teléfono Móvil

*Ilustración 23: Formulario de la inscripción, parte 2*



*Ilustración 24: Confirmación de la inscripción*



*Ilustración 25: Sección de noticias*



*Ilustración 26: Sección de ponentes*



A screenshot of a mobile application interface. At the top, there is a status bar with a close button (X), signal strength, battery, and the time 12:30. Below the status bar is a photograph of a man with long hair and a beard, wearing a blue and brown striped beanie, speaking at a podium. In the background of the photo is a large, stylized graphic of a face with wide eyes. A white text box is overlaid on the bottom half of the image, containing the following text:

**CHEMA ALONSO**  
**CDO (Chief Data Officer) en Telefónica**

Chema Alonso es actualmente CDO de Telefónica. Previamente fue el fundador y CEO de Eleven Paths empresa filial de Telefónica Digital centrada en la innovación en productos de seguridad y el Director General de Global Security Business en la unidad B2B de Telefónica Business Solutions.

Anteriormente trabajó y dirigió Informática 64 durante 14 años, empresa centrada en Seguridad Informática y formación. Es Dr. en Seguridad Informática por la Universidad Rey Juan Carlos de Madrid, Ingeniero Informático por la URJC e Ingeniero Informático de Sistemas por la Universidad Politécnica de Madrid, que además le nombró Embajador

*Ilustración 27: Información del ponente*



*Ilustración 28: Agenda con calendario*

The image shows a mobile application interface for an agenda. At the top, there is a red header bar with a hamburger menu icon on the left, the word 'Noviembre' in the center, and a dropdown arrow on the right. To the right of the header, there are icons for signal strength, Wi-Fi, and battery, along with the time '12:30'. Below the header, the date '10 Jueves' is displayed in a light gray bar. The main content is a list of events for the day, each with a time slot on the left and a description on the right, separated by a vertical red line. The events are: 10:30h Acreditación Asistentes Hackatón; 11:00h Presentación y mesa redonda: "Transformación digital: de Castellón al mundo"; 12:00h Videoconferencia de Chema Alonso: "Hackers & IoT"; 12:30h Comida; 16:00h Taller de creatividad e innovación con Kike Algora; 18:00h Networking entre partners, patrocinadores e instituciones; 20:00h Fiesta de presentación; 22:00h Finalización de primer día. At the bottom, the date '11 Viernes' is displayed in a light gray bar.

10 Jueves	
10:30h	Acreditación Asistentes Hackatón
11:00h	Presentación y mesa redonda: "Transformación digital: de Castellón al mundo"
12:00h	Videoconferencia de Chema Alonso: "Hackers & IoT"
12:30h	Comida
16:00h	Taller de creatividad e innovación con Kike Algora
18:00h	Networking entre partners, patrocinadores e instituciones
20:00h	Fiesta de presentación
22:00h	Finalización de primer día
11 Viernes	

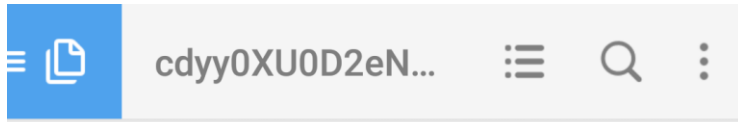
*Ilustración 29: Agenda sin calendario*



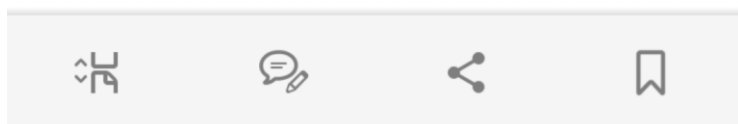
*Ilustración 30: Información de un acto*



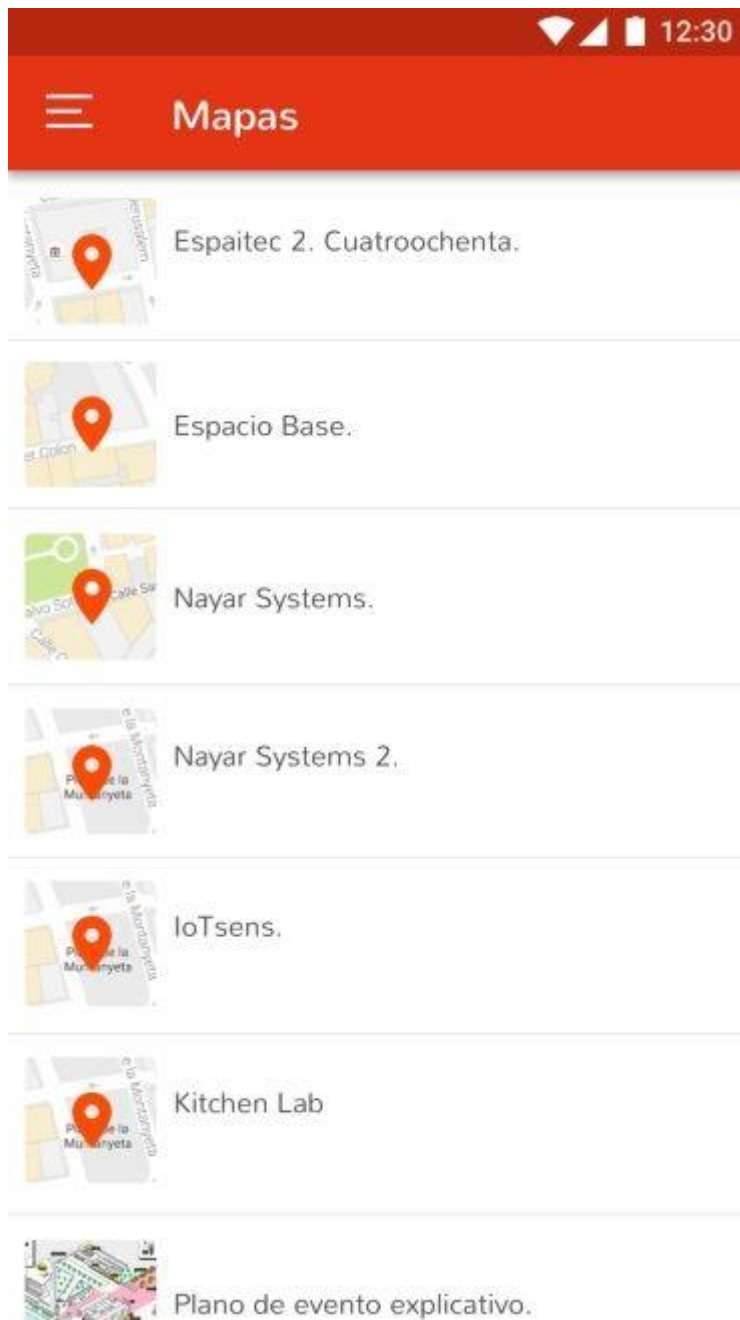
*Ilustración 31: Lista de documentos con información*



Inscripción de Fernando GA para el Hackathon 2016.



*Ilustración 32: Ejemplo PDF personalizado*



*Ilustración 33: Lista de los lugares importantes del evento*



*Ilustración 34: Mapa con los servicios de la ciudad*





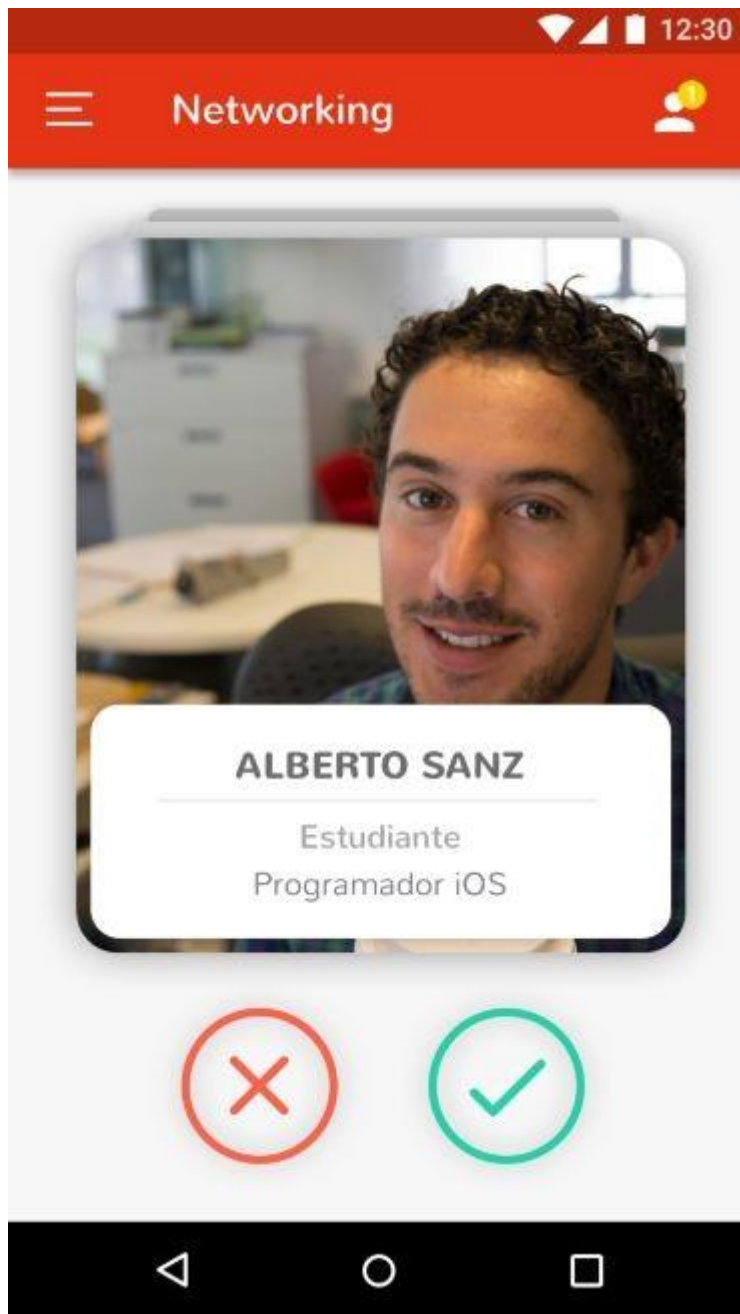
Il·lustració 35: Patrocinadores



*Ilustración 36: Información de la empresa*



*Ilustración 37: Lista de notificaciones*



*Ilustración 38: Interacción con otro usuario*

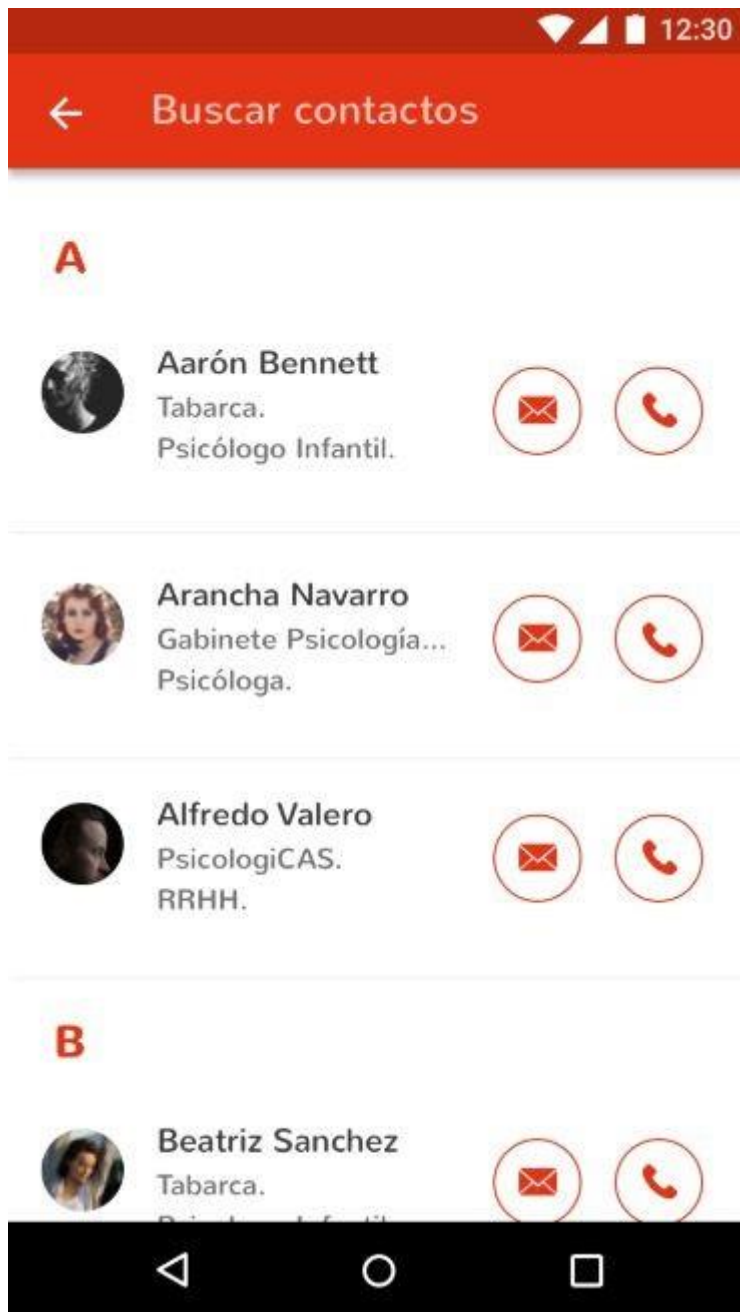


Ilustración 39: Lista de contactos

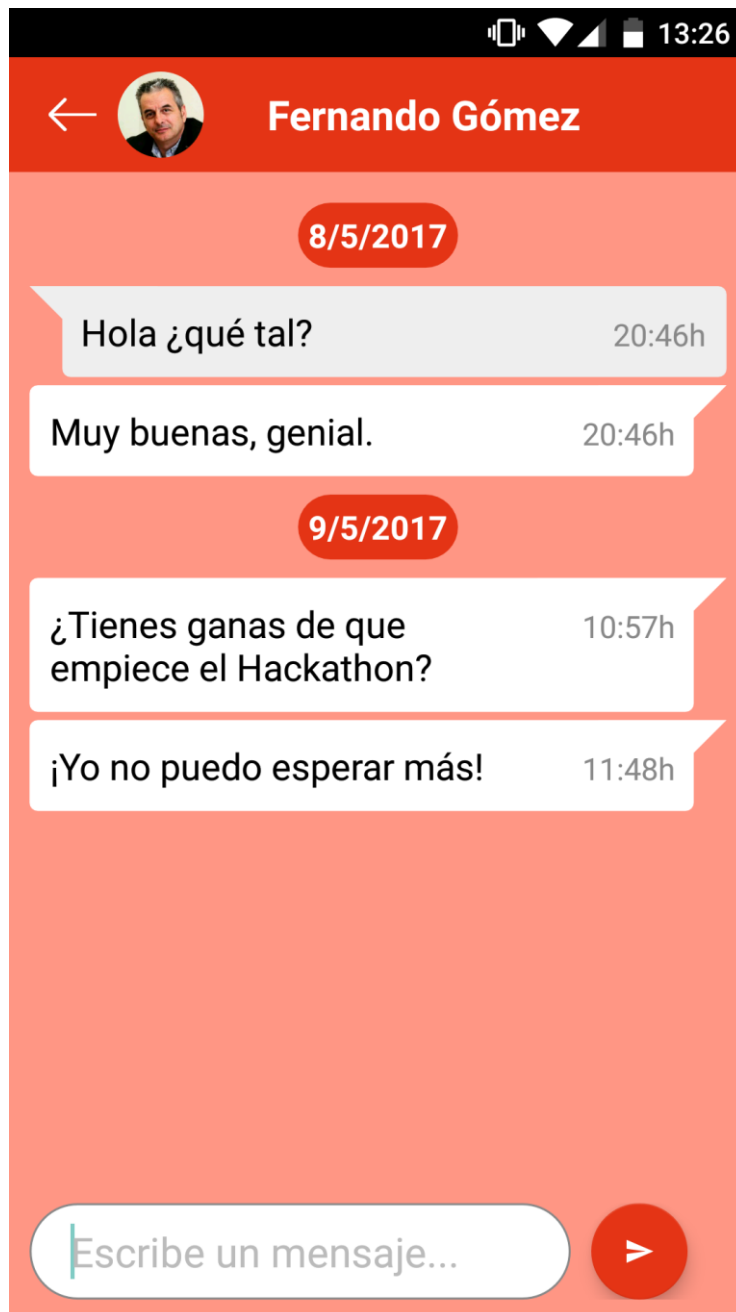


Ilustración 40: Chat privado

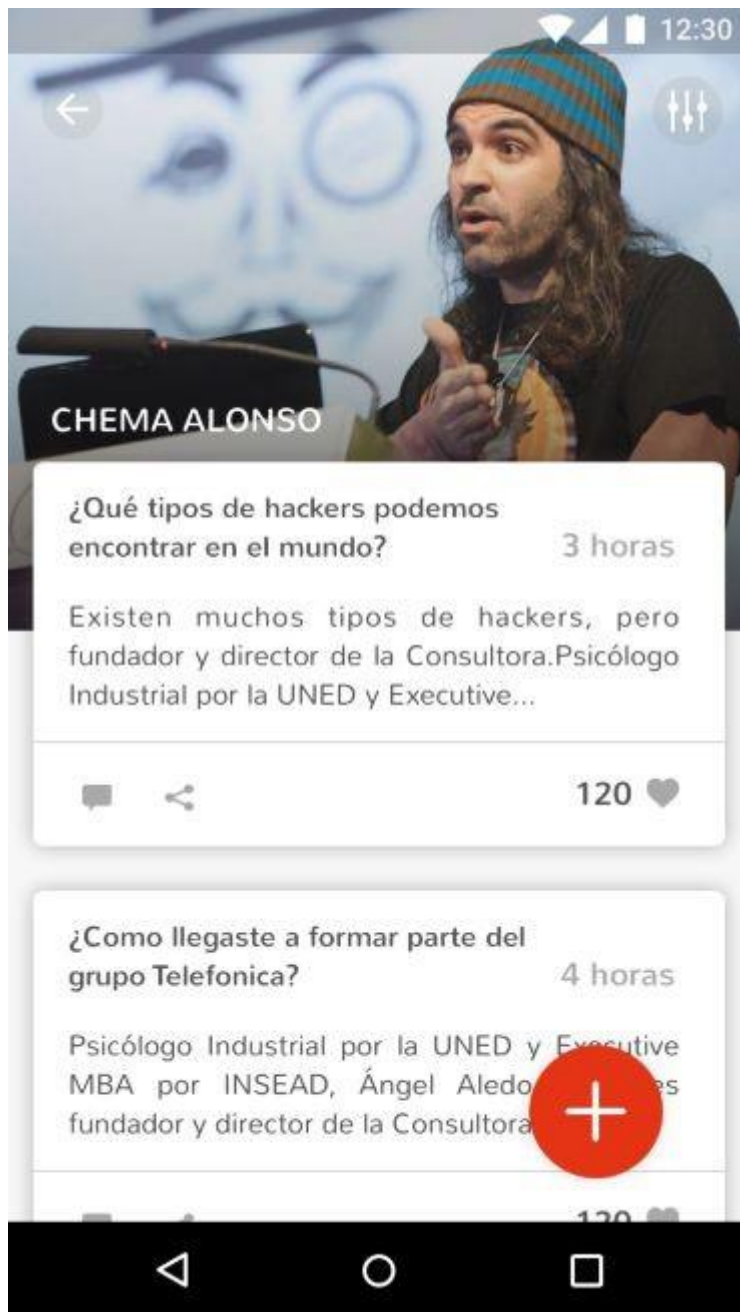
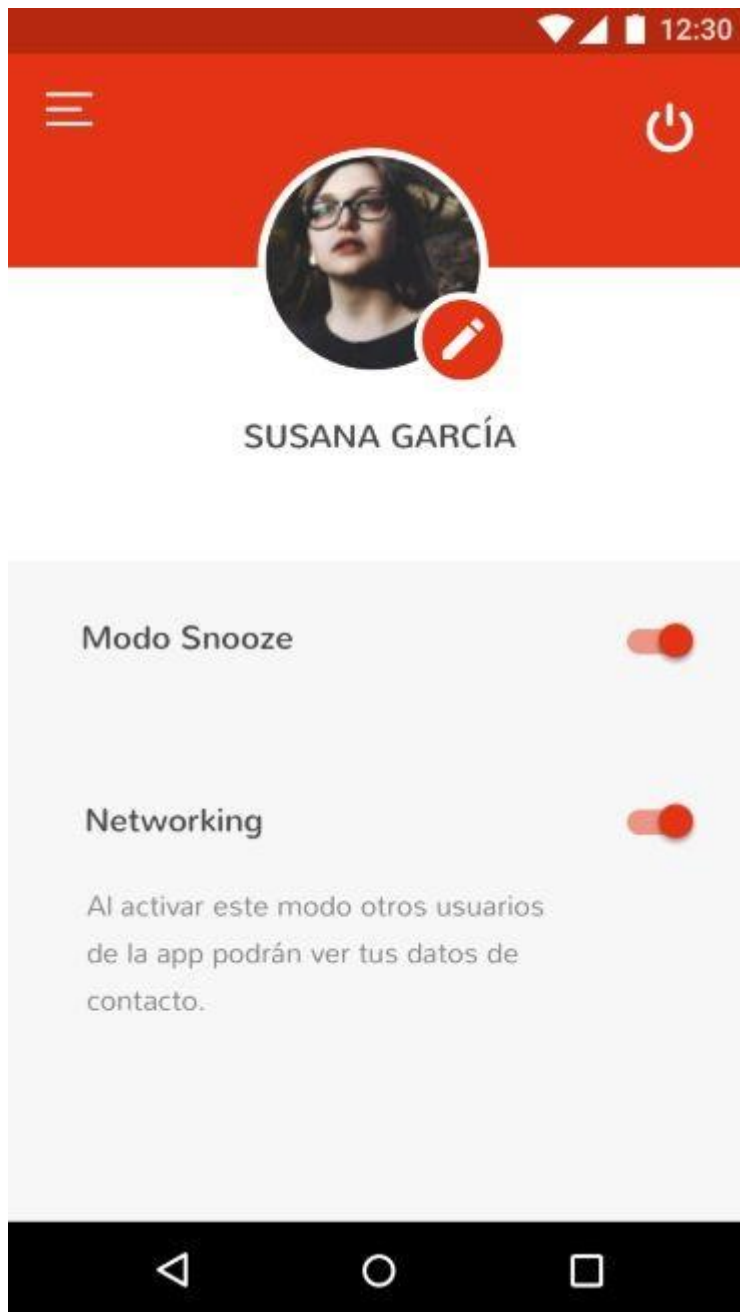


Ilustración 41: Lista de preguntas al ponente

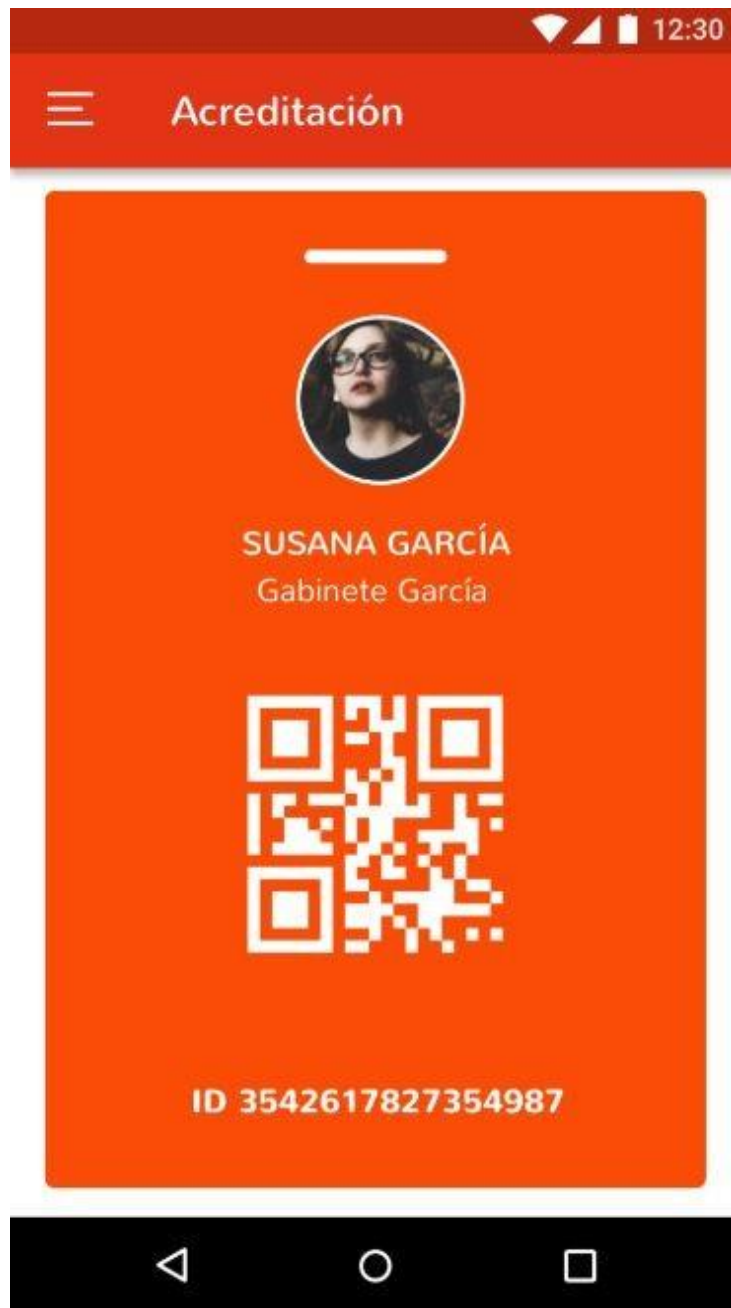


*Ilustración 42: Preguntar al ponente*





*Ilustración 43: Perfil*



*Ilustración 44: Acreditación*



34 días.

3 

¿Qué tipos de hackers podemos encontrar en el mundo?

En la actualidad, los piratas ya no llevan parche en el ojo y surcan los mares en busca de tesoros, ahora son grandes cerebros que no necesitan más que un ordenador y acceso a Internet para hacer de las suyas. Nosotros los conocemos como "hackers" pero, en la mayoría de los casos, el término no se utiliza de forma correcta. Entonces, ¿Qué es un hacker y qué tipos de hacker existen?




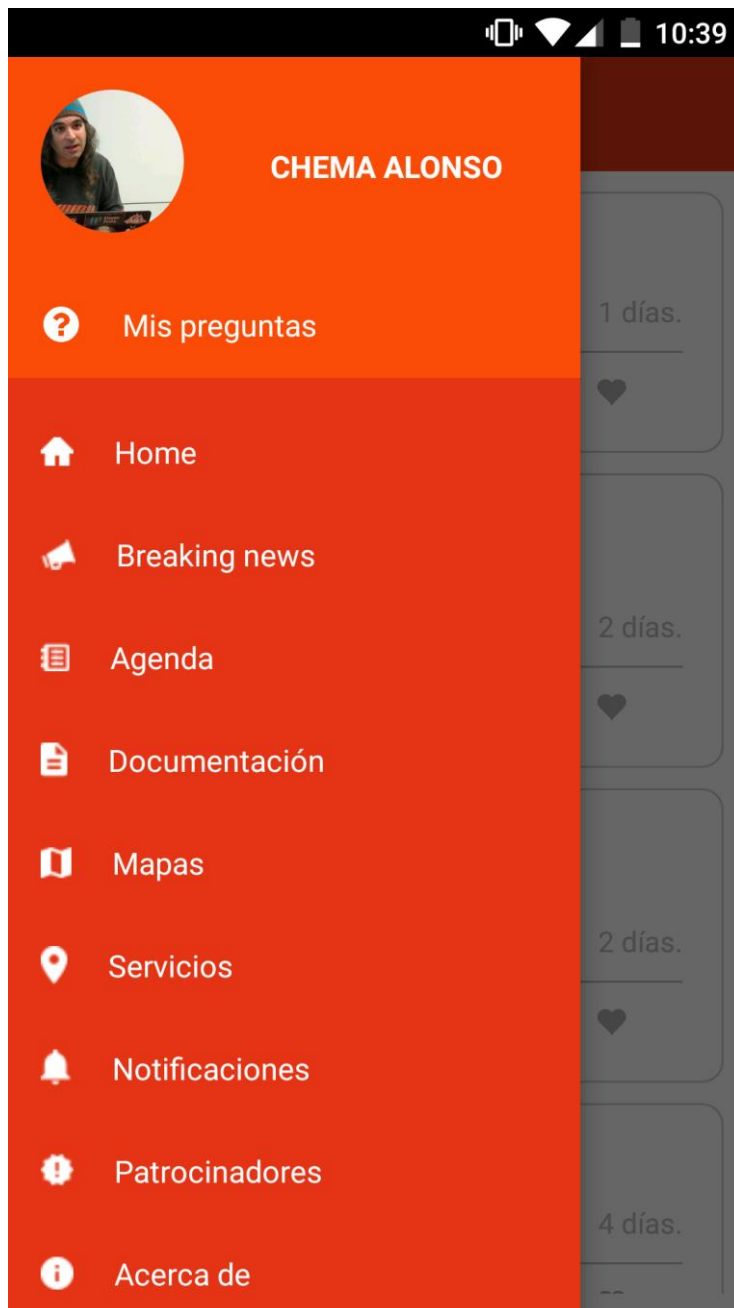
	<b>Juan Pérez</b> muy bien	1 días.
	<b>Juan Pérez</b> Me ha quedado muy claro.	10 días. 

Ilustración 45: Vista de una pregunta con comentarios



*Ilustración 46: Menú de un ponente*

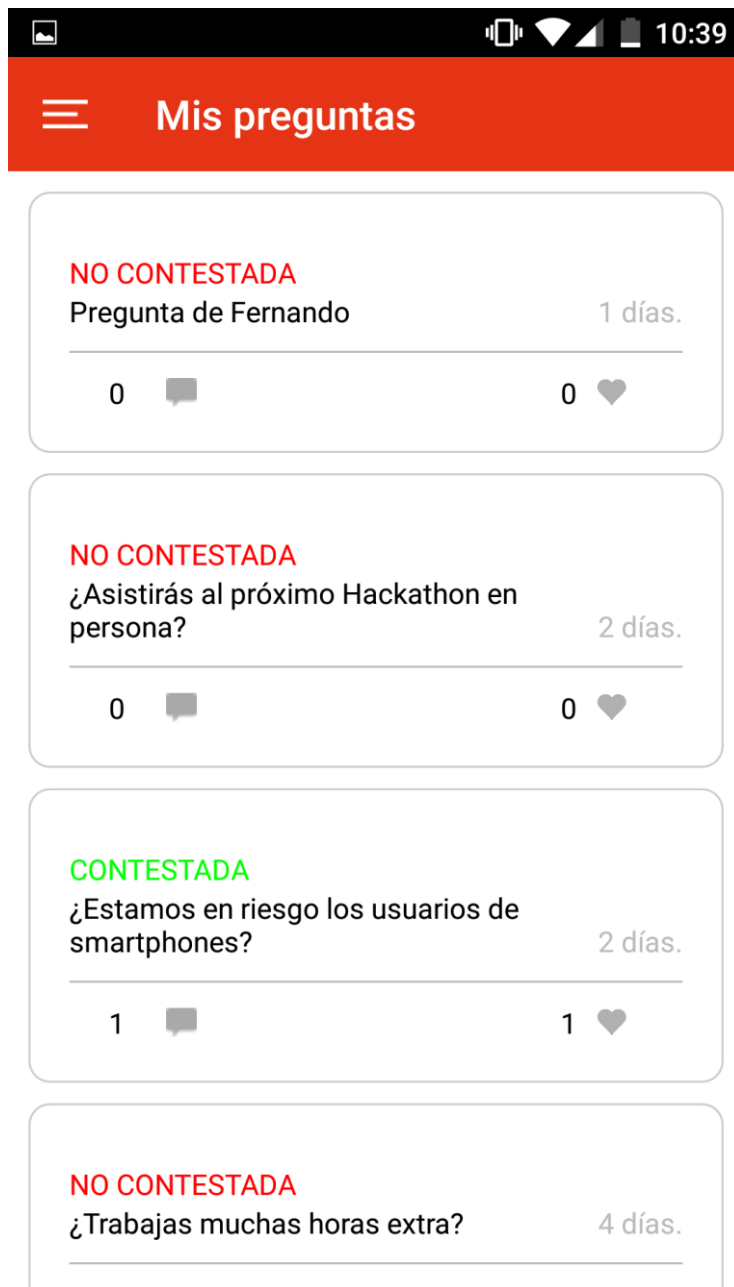
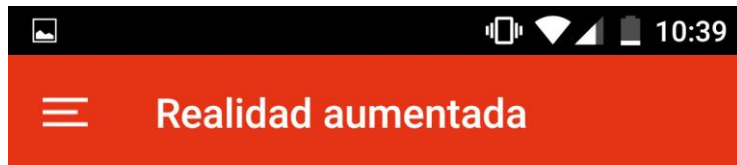


Ilustración 47: Lista de preguntas de un ponente



Busca tu punto de  
Realidad Aumentada.



*Ilustración 48: Pantalla de realidad aumentada*