

# Desarrollo de una aplicación Android para festivales

**Janire Fernández Iglesias**

Máster Universitario en Ingeniería Informática

Desarrollo de Aplicaciones sobre Dispositivo móviles

**Nombre Consultor/a** Jordi Ceballos Villach

**Nombre Profesor/a responsable de la asignatura** Jordi Ceballos Villach

06/2020



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

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Desarrollo de una aplicación Android para festivales</i>
<b>Nombre del autor:</b>	<i>Janire Fernández Iglesias</i>
<b>Nombre del consultor/a:</b>	<i>Jordi Ceballos Villach</i>
<b>Nombre del PRA:</b>	<i>Jordi Ceballos Villach</i>
<b>Fecha de entrega (mm/aaaa):</b>	06/2020
<b>Titulación::</b>	<i>Máster Universitario en Ingeniería Informática</i>
<b>Área del Trabajo Final:</b>	<i>Desarrollo de Aplicaciones sobre Dispositivos Móviles</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>DCU, prototipado, festival</i>
<b>Resumen del Trabajo</b>	
<p>Este trabajo muestra el desarrollo de una aplicación Android para festivales basándose en el uso de la filosofía del Diseño Centrado en el Usuario (DCU), la cual puede considerarse como la aplicación práctica de la disciplina de Interacción Persona Ordenador (IPO), la usabilidad y la experiencia de usuario (UX).</p> <p>Se tienen tres fases en la aplicación de DCU: la fase de investigación la cual se basa en la recopilación de información para definir el producto. Para ello se analizan aplicaciones similares para conocer sus puntos fuertes, características, y descubrir las tendencias de diseño que existen. En la fase de diseño se definen los escenarios de uso y los flujos de interacción para describir cómo va a ser la interacción de un usuario con la aplicación y se realizarán los prototipos. Y por último, en la fase de evaluación se aplica el método de test de usuarios con el objetivo de encontrar posibles errores y mejoras en el diseño realizado.</p> <p>A continuación se pasa a la fase de diseño técnico donde se definen los casos de uso y el diseño de la arquitectura. Una vez esto, se pasa a la implementación donde se transforma el diseño realizado en un producto en el cual se implementan las diferentes funcionalidades para conseguir la aplicación Android. En esta fase se comentan todas las herramientas y decisiones tomadas.</p> <p>Se puede concluir que gracias a la utilización de la filosofía DCU se ha podido obtener un producto de alta calidad que satisface las necesidades del usuario.</p>	

## **Abstract**

This document presents the development of an Android application for festivals based on the use of the philosophy of User Centered Design (UCD), which can be considered as the practical application of the discipline of Human-Computer Interaction (HCI), usability and user experience (UX).

There are three phases in the application of UCD: the research phase which is based on the collection of information to define the product. For this, similar applications are analyzed to know their strengths, characteristics, and discover the design trends that exist. In the design phase, use scenarios and flowcharts are defined to describe how a user will interact with the application and prototypes will be made. And finally, in the evaluation phase, the user test method is applied in order to find possible errors and improvements in the design made.

Then the next step is the technical design phase where the use cases and architecture design are defined. Once this is done, the implementation takes place where the design is transformed into a product which the different functionalities are implemented to get the Android application. In this phase all the tools and decisions made are discussed.

It can be concluded that thanks to the use of the UCD philosophy, it has been possible to obtain a high quality product that meets the user's needs.

## Índice

1. Introducción .....	1
1.1 Contexto y justificación del Trabajo .....	1
1.2 Objetivos del Trabajo.....	2
1.4 Planificación del Trabajo.....	4
1.5 Breve resumen de productos obtenidos .....	7
1.6 Breve descripción de los otros capítulos de la memoria .....	7
2. Análisis.....	8
2.1 Introducción.....	8
2.2 Análisis competitivo o benchmarking.....	8
2.3 Encuesta.....	12
2.5 Observación con usuarios.....	15
2.6 Conclusiones del análisis .....	18
3. Diseño.....	20
3.1 Introducción.....	20
3.2 Escenarios de uso .....	20
3.3 Flujos de interacción.....	21
3.4 Prototipo de baja fidelidad .....	24
3.5 Prototipo de alta fidelidad.....	27
4. Evaluación .....	31
4.1 Introducción.....	31
4.2 Test con usuarios .....	31
4.3 Revisión del prototipo .....	34
5. Diseño técnico .....	36
5.1 Definición de los casos de uso .....	36
5.2 Diseño arquitectura .....	39
6. Implementación .....	43
6.1 Introducción.....	43
6.2 Desarrollo .....	43
6.3 Análisis del estado del proyecto.....	52
6.4 Pruebas .....	52
7. Conclusiones.....	54
7.1 Introducción.....	54
7.2 Lecciones aprendidas.....	54
7.3 Logro de los objetivos.....	54
7.4 Seguimiento de la planificación y metodologías .....	54
7.5 Líneas de trabajo futuro .....	55
8. Glosario .....	56
9. Bibliografía .....	57
10. Anexos.....	60
ANEXO I. Manual de Instrucciones de compilación.....	60

## Lista de ilustraciones

Ilustración 1. Diagrama de Gantt .....	6
Ilustración 2. Imágenes aplicación Festival International .....	8
Ilustración 3. Imágenes aplicación Sziget Festival.....	9
Ilustración 4. Imágenes aplicación Festival de Cannes - Official .....	10
Ilustración 5. Imágenes aplicación FestivAll, Music Festival Guide .....	10
Ilustración 6. Imágenes aplicación Festival One .....	11
Ilustración 7. Prototipo de baja fidelidad del menú principal.....	24
Ilustración 8. Prototipo de baja fidelidad para la consulta de artistas y horarios .....	24
Ilustración 9. Prototipo de baja fidelidad para la consulta de restaurantes o bares.....	25
Ilustración 10. Prototipo de baja fidelidad para el mapa.....	25
Ilustración 11. Prototipo de baja fidelidad para el método de pago para evitar colas de espera .....	26
Ilustración 12. Prototipo de baja fidelidad para las notificaciones.....	26
Ilustración 13. Prototipo de baja fidelidad.....	27
Ilustración 14. Prototipo de alta fidelidad del menú principal .....	27
Ilustración 15. Prototipo de alta fidelidad para la consulta de artistas y horarios .....	28
Ilustración 16. Prototipo de alta fidelidad para la consulta de restaurantes o bares.....	29
Ilustración 17. Prototipo de alta fidelidad para el mapa .....	29
Ilustración 18. Prototipo de alta fidelidad para el método de pago .....	30
Ilustración 19. Modificación prototipo de alta fidelidad para artistas .....	34
Ilustración 20. Modificación prototipo de alta fidelidad para restaurantes o bares .....	34
Ilustración 21. Modificación prototipo de alta fidelidad para el mapa .....	35
Ilustración 22. Modificación prototipo de alta fidelidad para el método de pago.....	35
Ilustración 23. Captura de pantalla de las reglas definidas en Firebase database y Firebase Cloud Storage.....	43
Ilustración 24. Captura de pantalla de menú principal en aplicación.....	44
Ilustración 25. Algunos de los índices definidos en Firebase database para artistas. ....	44
Ilustración 26. Carpetas que se tienen en Firebase Cloud Storage para artistas.....	45
Ilustración 27. Captura de pantalla de consulta de artistas y horario en aplicación. ....	45
Ilustración 28. Índices definidos en Firebase database para restaurantes.....	46
Ilustración 29 . Carpetas que se tienen en Firebase Cloud Storage para restaurantes. ....	46
Ilustración 30. Captura de pantalla de consulta de restaurantes en aplicación. ....	47
Ilustración 31. Carpetas que se tienen en Firebase Cloud Storage para mapa. ....	48
Ilustración 32. Captura de pantalla de consulta de mapa en aplicación .....	48
Ilustración 33. Captura de pantalla de realización de pago en aplicación.....	50
Ilustración 34. Listado notificaciones enviadas con <i>Firebase Cloud Messaging</i> .....	51
Ilustración 35. Captura de pantalla de recibimiento de notificación en aplicación.....	51
Ilustración 36. Captura de pantalla de pruebas realizadas con Firebase Test Lab .....	53
Ilustración 37. Captura de pantalla de cómo ejecutar una aplicación en Android Studio.....	60
Ilustración 38. Captura de pantalla para visualizar donde se debe introducir el código del servidor PHP para el método de pago .....	61
Ilustración 39. Panel de control de XAMPP.....	61
Ilustración 40. Captura de cómo se ejecuta apache y MySQL en XAMPP .....	61
Ilustración 41. Captura de pantalla de ngrok en funcionamiento. ....	62
Ilustración 42. Captura de pantalla de dónde se debe introducir en el código la dirección del servidor PHP.....	62

## Lista de tablas

Tabla 1. Etapas de realización del proyecto junto con su duración de desarrollo.....	4
Tabla 2. Plan de trabajo de la primera etapa del proyecto.....	5
Tabla 3. Plan de trabajo de la segunda etapa del proyecto.....	5
Tabla 4. Hitos del proyecto. ....	6
Tabla 5. Desviación de plan de trabajo para añadir mejoras a la aplicación. ....	52

## Lista de figuras

Figura 1. Resultado encuesta desarrollo de aplicación móvil (pregunta 1.1).....	12
Figura 2. Resultado encuesta desarrollo de aplicación móvil (pregunta 1.2).....	13
Figura 3. Resultado encuesta desarrollo de aplicación móvil (pregunta 2.1).....	13
Figura 4. Flujo de interacción para consultar los horarios de un artista.....	21
Figura 5. Flujo de interacción para consultar los restaurantes o bares.....	22
Figura 6. Flujo de interacción para consultar mapa.....	22
Figura 7. Flujo de interacción para método de pago evitando colas de espera.....	23
Figura 8. Flujo de interacción de la aplicación.....	23
Figura 9. Diagrama UML correspondiente al diseño de las entidades y clases.....	40
Figura 10. Diagrama explicativo de la arquitectura del sistema.....	41

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

Los festivales se han convertido en una de las actividades centro del verano en los cuáles se agrupan personas para poder ver a sus artistas favoritos o para poder disfrutar de la música del momento.

A la hora de organizar un festival se tienen que tener en cuenta diferentes aspectos relacionados con los permisos, licencias, seguros, derechos laborales de los músicos, entre otros. Es decir, se tiene que realizar el cumplimiento de la normativa de los festivales. Sin embargo, aunque se tenga un cumplimiento de la normativa, muchas veces se olvida que también se quiere conseguir el mayor grado de satisfacción posible de los asistentes.

Debido a la cantidad de personas que se juntan en estos eventos, se tiene que tener una buena organización en cuanto a la información que se comparte a los presentes y se debe proporcionar el acceso a dicha información de manera sencilla en cualquier momento.

Se presentan a continuación algunos de los problemas de organización que se pueden encontrar en la mayoría de los festivales, y por tanto, se quieren mejorar.

- Hoy en día cuando se llega a un festival es necesaria la compra de cupones para poder consumir en éste. Dicha compra supone estar en la cola durante 30 minutos (incluso más) al comienzo del festival, y unos 10 minutos cada vez que se necesiten más.

Se requiere proporcionar un método con el cual se pueda pagar cualquier producto evitando las colas de espera.

- Se debería proporcionar una forma sencilla para poder consultar los horarios de cada escenario pudiendo ver los diferentes artistas que actuarán. Actualmente sólo se puede consultar dicha información a través de fotos que se cuelgan en las redes sociales como Facebook o Instagram, e ir buscando en cuál de los escenarios está el artista que le interesa al usuario en concreto. Cabe destacar que muy pocas veces se encuentra dicha información en la página web oficial.

Se debería tener el acceso a dicha información de manera sencilla, de tal manera que se puedan realizar búsquedas con los filtros necesarios para encontrar la información deseada.

- Además de la necesidad de poder consultar los diferentes horarios, es necesaria una forma sencilla de consulta de los diferentes restaurantes o bares que haya en el festival. Muchas veces los usuarios van recorriendo el festival para poder ver la comida que se ofrece en éstos y poder elegir el que gusté más.

Se considera que se debería proporcionar una forma sencilla de consulta para poder ver los diferentes restaurantes o bares que se tengan.

- En algunos casos es necesario informar a los usuarios de cancelaciones, retrasos u otro tipo de notificaciones. Normalmente dicha información es compartida a través de las redes sociales como Facebook o Instagram, lo que puede dar lugar a que haya personas



que no reciban dicho aviso porque no siguen a las cuentas concretas del festival, o simplemente porque no están atentos a éstas.

Es por ello que se considera que debería haber una mejor gestión a la hora de enviar notificaciones a los usuarios, de tal manera que no se requiera tener que buscar en las redes sociales posibles publicaciones con avisos.

- Normalmente en los festivales se hace la entrega de un folleto el cual contiene el mapa para saber la ubicación de los diferentes puntos principales de éste.

Debería de proporcionarse una manera sencilla de acceso al mapa de tal forma que no se tenga que estar pendiente de no perder el folleto.

En definitiva, se quiere mejorar la organización de los festivales, permitiendo la consulta de la información más relevante consiguiendo que se tenga una mayor satisfacción por parte de los usuarios, ya que se tiene acceso a ésta de forma sencilla y rápida.

## **1.2 Objetivos del Trabajo**

El objetivo de la aplicación es facilitar la organización en un festival para proporcionar la mayor satisfacción a los usuarios presentes.

Se considera que la mejor opción para conseguir dicho objetivo es mediante la realización de una aplicación para dispositivos móviles.

En dicha aplicación se tendrán las siguientes funcionales.

- Consulta de los diferentes horarios de cada escenario pudiendo filtrar en función de los campos definidos.
- Consulta de los diferentes restaurantes o bares pudiendo filtrar en función de los campos definidos.
- Capacidad para recibir notificaciones en caso de cancelación, retraso u otro tipo de aviso.
- Consulta del mapa del festival concreto donde se encuentran todos los puntos principales de éste.
- Método de pago para poder consumir en el festival concreto evitando las colas de espera para poder comprar cupones.

Por otro lado, la aplicación deberá cumplir con una serie de requisitos no funcionales, los cuales se presentan a continuación.

- El sistema funcionará en dispositivos móviles con sistema operativo Android.
- La aplicación debe presentar facilidad de uso, teniendo una interfaz sencilla e intuitiva.
- Los datos de la aplicación deberán ser almacenados en un sistema gestor de bases de datos.
- El sistema debe presentar escalabilidad y rendimiento acorde al número de usuarios que estén utilizando la aplicación.

### 1.3 Enfoque y método seguido

Inicialmente se consideraron diferentes estrategias a llevar a cabo para poder proporcionar una mejor organización en los festivales.

#### 1. Funcionalidades en la página web oficial

Se consideró poder añadir las funcionalidades en la página web oficial donde se podría incluir toda la información que el usuario necesitaría consultar. Para ello, la página web debería ser capaz de adaptarse en cualquier dispositivo, es decir, que tuviese un diseño “*responsive*” y se pudiesen acceder a todas las funcionalidades a través de ésta.

Los inconvenientes que presentaba dicha propuesta es que se tenía un problema a la hora de enviar notificaciones a los usuarios sobre diferentes sucesos que se produjesen en el evento, ya que requeriría que el usuario estuviese en la página web en ese momento. Para solucionar dicho problema se pensó en enviar notificaciones por email, pero se considera que es más cómodo para los usuarios recibir las notificaciones en un mismo lugar en vez de tener que consultar el correo electrónico.

Por otro lado, realizar un método de pago a través de la página web podría presentar problemas, ya que si el usuario sale de dicha página podría perder la información. Además, puede resultar incomodo tener que acceder a la página web e ir navegando hasta encontrar el apartado de compra de cupones cada vez que se quisiese realizar una compra.

#### 2. Creación aplicación de dispositivos móviles

Se consideró la realización de una aplicación móvil, ya que sería más cómodo para los usuarios poder consultar la información deseada y además, se podrían recibir las notificaciones a través de dicha aplicación y el método de pago resultaría más sencillo debido a que no se perdería la información si se saliese.

Además de la comodidad, se vio que el desarrollo de una aplicación proporciona mayores ventajas a la hora de poder hacerla configurable para que pueda ser usada por diferentes festivales. Es decir, cada festival podría utilizar dicha aplicación para su organización. Para ello, la introducción de datos como los horarios, restaurantes, bares o mapa no debe suponer un cambio en el código, sino que solamente se cambie la información de la base de datos.

Por tanto, se considera que con el desarrollo de una aplicación móvil se podrían añadir todos los requerimientos funcionales y no funcionales comentados en el apartado anterior. También se conseguiría satisfacción por parte del usuario ya que es la propuesta que proporciona un fácil acceso, y además resulta más sencilla a la hora de hacerla configurable.

## 1.4 Planificación del Trabajo

A la hora de la realización de la planificación se van a definir los diferentes recursos necesarios para poder realizar el trabajo, se va a realizar una planificación temporal de las diferentes tareas a llevar a cabo y se van a definir los hitos parciales de cada una de las PEC.

### 1.4.1. Recursos necesarios para realizar el trabajo

Para poder desarrollar una aplicación de dispositivos móviles se necesitan los siguientes recursos.

- Software Android Studio: Software donde se desarrollará el código de la aplicación.
- Ordenador: Necesario para poder utilizar el software anteriormente comentado.
- Dispositivo móvil: Necesario para poder probar la aplicación desarrollada.
- Material de oficina.

### 1.4.2. Planificación temporal de las tareas a llevar a cabo

A la hora de realizar la planificación del proyecto se tiene que tener en cuenta que la duración de las tareas se ha definido en horas, ya que dependiendo de si son días laborables o no, el número de horas de dedicación varía.

- Días laborables: 3h al día.
- Días no laborables o festivos: 4h al día.

La realización del proyecto se organiza en etapas. Cada etapa tendrá una serie de tareas a realizar.

La primera etapa corresponderá al diseño y la arquitectura de la aplicación, en el cual se elegirán las librerías o tecnologías que se van a utilizar para poder almacenar la información en un sistema gestor de bases de datos.

Más adelante se pasará a la etapa de desarrollo, la cual presenta la mayor duración. En ésta se comenzarán a implementar los diferentes requerimientos para poder conseguir la aplicación Android.

Cabe destacar que a lo largo del proyecto se tendrá que ir redactando la documentación, es por ello que éste tendrá una duración desde el comienzo hasta la finalización del proyecto.

Tabla 1. Etapas de realización del proyecto junto con su duración de desarrollo.

Nombre de tarea	Duración	Comienzo	Fin
Documentación	370 horas	mié 19/02/20	mié 10/06/20
▶ Análisis y diseño	91 horas	jue 05/03/20	mié 01/04/20
▶ Desarrollo y pruebas	229 horas	jue 02/04/20	mié 10/06/20

En la primera etapa se realizará el análisis de la aplicación, en el cual se tienen que encontrar las tecnologías o librerías para poder cumplir los diferentes requisitos de la aplicación. Por tanto, se deberá estudiar cómo debe ser la arquitectura de ésta y cómo se podrán implementar las diferentes funcionalidades.

Además, en esta etapa se deberá empezar a pensar cómo debe presentarse al usuario la aplicación, es por ello que se llevará a cabo el diseño de las interfaces gráficas gracias a la realización de un prototipo de alta fidelidad.

**Tabla 2. Plan de trabajo de la primera etapa del proyecto.**

Nombre de tarea	Duración	Comienzo	Fin
<b>↙ Análisis y diseño</b>	<b>91 horas</b>	<b>jue 05/03/20</b>	<b>mié 01/04/20</b>
Análisis de requisitos	25 horas	jue 05/03/20	jue 12/03/20
Análisis tecnologías	48 horas	jue 05/03/20	jue 19/03/20
Arquitectura de la aplicación	43 horas	jue 19/03/20	mié 01/04/20
Diseño interfaces gráficas	43 horas	jue 19/03/20	mié 01/04/20

En la siguiente etapa es cuando se comienza con el desarrollo de la aplicación. En ésta se empezarán a incluir las funcionalidades definidas. En cada una de las funcionalidades se tendrá un proceso de codificación y uno de pruebas. En este último es donde se realizarán comprobaciones para verificar el buen funcionamiento de cada componente añadido. Una vez esto se realizarán las pruebas para poder verificar que la aplicación completa funciona correctamente.

**Tabla 3. Plan de trabajo de la segunda etapa del proyecto.**

Nombre de tarea	Duración	Comienzo	Fin
<b>↙ Desarrollo y pruebas</b>	<b>229 horas</b>	<b>jue 02/04/20</b>	<b>mié 10/06/20</b>
<b>↙ Implementación consulta de horarios</b>	<b>71 horas</b>	<b>jue 02/04/20</b>	<b>jue 23/04/20</b>
Codificación	62 horas	jue 02/04/20	lun 20/04/20
Pruebas y resolución incidencias	9 horas	lun 20/04/20	jue 23/04/20
<b>↙ Implementación consulta de restaurantes/bares</b>	<b>22 horas</b>	<b>vie 24/04/20</b>	<b>jue 30/04/20</b>
Codificación	18 horas	jue 23/04/20	dom 26/04/20
Pruebas y resolución incidencias	11 horas	lun 27/04/20	jue 30/04/20
<b>↙ Implementación envío notificaciones</b>	<b>25 horas</b>	<b>vie 01/05/20</b>	<b>vie 08/05/20</b>
Codificación	16 horas	vie 01/05/20	mar 05/05/20
Pruebas y resolución incidencias	9 horas	mar 05/05/20	vie 08/05/20
<b>↙ Implementación método de pago</b>	<b>56 horas</b>	<b>sáb 09/05/20</b>	<b>sáb 23/05/20</b>
Codificación	35 horas	sáb 09/05/20	mar 19/05/20
Pruebas y resolución incidencias	20 horas	mié 20/05/20	sáb 23/05/20
<b>↙ Implementación visualización de mapa</b>	<b>33 horas</b>	<b>dom 24/05/20</b>	<b>dom 31/05/20</b>
Codificación	11 horas	dom 24/05/20	mié 27/05/20
Pruebas y resolución incidencias	21 horas	jue 28/05/20	dom 31/05/20
Pruebas de la aplicación completa	31 horas	lun 01/06/20	mié 10/06/20

Se presenta a continuación el diagrama de Gantt resultante.

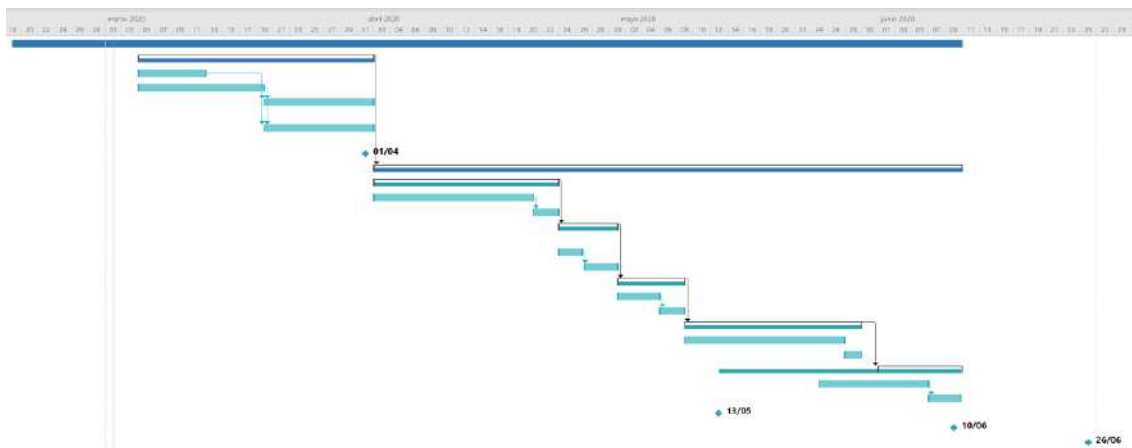


Ilustración 1. Diagrama de Gantt

### 1.4.3. Hitos parciales de cada una de las PECs

Se presentan los hitos que se pueden encontrar en el proyecto. Se ha realizado una descomposición de éstos para poder ver qué tareas requiere cada uno.

Tabla 4. Hitos del proyecto.

Nombre	Duración	Inicio	Final
<b>PEC2. Diseño y arquitectura</b>	91 horas	05/03/2020	01/04/2020
Análisis: Usuarios y contexto de uso	19 horas	05/03/2020	10/03/2020
Diseño: diseño conceptual	25 horas	11/03/2020	18/03/2020
Diseño: Prototipado	28 horas	19/03/2020	27/03/2020
Evaluación: Evaluación	13 horas	27/03/2020	30/03/2020
<b>PEC3. Implementación</b>	137 horas	02/04/2020	13/05/2020
Desarrollo: desarrollo de cada componente junto con sus pruebas unitarias.	205 horas	02/04/2020	31/05/2020
Pruebas: pruebas de la aplicación completa.	31 horas	01/06/2020	10/06/2020
<b>Entrega final</b>	91 horas	14/05/2020	10/06/2020
Producto: manual de usuario, instrucciones, entre otros.	25 horas	14/05/2020	21/05/2020
Memoria del trabajo: repaso y corrección.	25 horas	22/05/2020	29/05/2020
Presentación	38 horas	30/05/2020	10/06/2020
<b>Defensa Virtual</b>	11 horas	22/06/2020	26/06/2020

## **1.5 Breve resumen de productos obtenidos**

En la entrega final es donde se deberá adjuntar la aplicación desarrollada. Además de la entrega del código, será necesario incluir un manual de usuario explicando cómo se utiliza la aplicación y un manual de instrucciones sobre cómo compilar y ejecutar la aplicación en un dispositivo o simulador.

Por otro lado, se deberá entregar la memoria del trabajo por lo que se realizará un repaso de éste junto con sus correcciones.

Y por último se preparará una presentación sobre la aplicación realizada explicando los objetivos del trabajo, los aspectos clave del desarrollo, las decisiones tomadas y las conclusiones y resultados obtenidos.

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

Inicialmente se comenzará con el estudio y análisis del diseño y la arquitectura que tendrá la aplicación. Se aplicarán los correspondientes métodos para conocer las características de los usuarios, sus necesidades y objetivos, así como el contexto de uso. Esto va a permitir que se puedan identificar las funcionalidades que deberá tener la aplicación para satisfacer a los usuarios. Para ello, se tendrá que hacer uso de estudios o investigaciones que permitirán definir y conocer usuarios, y se incluirá la justificación de elección de los métodos de indagación, así como su proceso de planteamiento, desarrollo, resultados y conclusiones obtenidas.

A partir de la información recopilada se realizarán los escenarios de uso describiendo desde el punto de vista del usuario cómo utilizará un producto en un contexto concreto. Tomando los flujos de interacción definidos se realizará un prototipo de alta fidelidad del cuál posteriormente se llevará a cabo su evaluación mediante un proceso de DCU. Gracias al desarrollo del prototipo se obtendrá una representación de la aplicación que permitirá comunicar decisiones de diseño y evaluarlo antes de desarrollar el producto final, de tal forma que la aplicación resultante transmita una buena experiencia de usuario.

A continuación se pasará a la implementación, que es cuando se llevará a cabo el desarrollo del código de la aplicación junto con sus correspondientes pruebas para la validación del funcionamiento. Se deberá incluir una breve descripción de las herramientas, editores, APIs, entre otros, utilizados y una justificación de su elección. Además, se realizará un breve análisis en relación a la planificación planteada, identificando posibles desviaciones y las medidas correctivas a tomar.

Por último, en la entrega final es donde se tiene que entregar la aplicación desarrollada junto con la memoria del trabajo. Como se ha comentado anteriormente será necesario incluir un manual de usuario, un manual de instrucciones y se deberá preparar una presentación sobre la aplicación realizada.

## 2. Análisis

### 2.1 Introducción

Este apartado se basa en involucrar a los usuarios en las diferentes actividades que se llevan a cabo y están orientados a la obtención de información para definir el producto. El conocimiento sobre los usuarios, sus contextos de uso, sus necesidades, objetivos y actitudes son imprescindibles para un diseño centrado en el usuario y para desarrollar la aplicación.

### 2.2 Análisis competitivo o benchmarking

En el análisis competitivo o benchmarking se van a analizar productos similares y/o que compiten con el sistema interactivo que se está diseñando. De esta manera se van a conocer las expectativas de los usuarios (dado que puede que ya conozcan estos productos), entender las tendencias del mercado, aprender de los errores pero también de lo que funciona, estudiar las interfaces, entre otros.

#### Festival International



Ilustración 2. Imágenes aplicación Festival International

**Ofrecida por:** Google commerce Ltd

**Versión actual:** 4.0.0

**Actualizada:** 24 de enero de 2020

**Precio:** Gratuita

**Descripción:** Se trata de la aplicación oficial para el Festival Internacional de Louisiane. Permite ver bandas favoritas, crear su propio horario del Festival y navegar por el centro de Lafayette.

**Funcionalidades:**

- Crear horario personalizado organizado por día, con posibilidad de añadir o eliminar artistas con un solo toque.
- Ver todos los artistas que tocan en las próximas dos horas.
- Ver información del artista. Se puede ver su biografía, sus videos, la página web del artista y sus últimos albums.
- Ver una lista completa de todos los vendedores de cocina, artes y manualidades en el Festival.
- Ver información del festival como el estacionamiento, transporte, objetos perdidos, entre otros.
- Evitar cola y comprar souvenirs oficiales del festival desde el dispositivo móvil u obtén indicaciones para llegar a la tienda más cercana.

Se puede decir que dicha aplicación es bastante completa y ofrece muchas funcionalidades. Pero uno de los inconvenientes que se puede apreciar nada más descargar la aplicación es que su diseño no es muy amigable para el usuario, es decir, no es simple e intuitivo. No cumple el principio de información perceptible ya que no comunica de manera eficaz la información necesaria para el usuario, no proporciona un contraste adecuado entre la información y sus alrededores, no maximiza la legibilidad de información esencial, entre otros. Además, se ha podido comprobar que no tiene una buena tolerancia a errores, hay veces que cuando se presiona una opción la aplicación deja de responder.

Por otro lado en cuanto a las funcionalidades, se considera que debería tenerse en cuenta para nuestra aplicación la opción de añadir más información relevante cuando se consultan los artistas, ya que es información que le puede interesar al usuario. En cambio, cabe destacar que en nuestro diseño se tendrá la opción de consultar todos los artistas en todo momento, no solo cuando vayan a tocar en las próximas dos horas.

### Sziget Festival



Ilustración 3. Imágenes aplicación Sziget Festival

**Ofrecida por:** Google commerce Ltd

**Versión actual:** 7.2.0

**Actualizada:** 16 de Noviembre de 2019

**Precio:** Gratuita

**Descripción:** Se trata de la aplicación oficial para el Festival Sziget, el cual se celebra en Budapest, Hungría.

**Funcionalidades:**

- Navegar por escenario y sedes.
- Búsqueda de programas.
- Añadir programas a favoritos.
- Usa los mapas par navegar fácilmente por el festival.
- Información actualizada sobre el festival.
- Llamada de emergencia.
- Opción de comprar tickets.

Se trata de una aplicación bastante útil para la gestión del festival Sziget. Además, su diseño es bastante intuitivo y sencillo, el único inconveniente que se podría decir es que debería ser un poco más minimalista ya que los colores son un poco fuertes.

En cuanto a las funcionalidades se puede ver que es bastante completa, aunque no se solventa el problema que se tienen en los festivales en cuanto a las colas de espera para comprar cupones, pero en cambio, se tiene la opción de comprar tickets del festival. Esto debería tenerse en cuenta para poder añadirla a nuestra aplicación.



## Festival de Cannes - Official

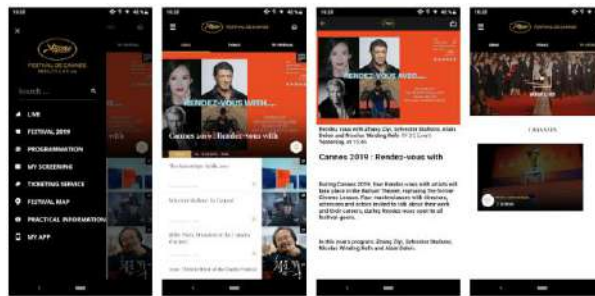


Ilustración 4. Imágenes aplicación Festival de Cannes - Official

**Ofrecida por:** Google commerce Ltd

**Versión actual:** 8.0.4

**Actualizada:** 24 de Mayo de 2019

**Precio:** Gratuita

**Descripción:** Se trata de la aplicación oficial para el de Cannes.

**Funcionalidades:**

- Servicios dedicados a profesionales acreditados: programación, marcaje de sesiones, acceso a la venta de entradas online, geolocalización e información práctica.
- Anuncios exclusivos y noticias sobre la selección de jurados, eventos y los ganadores.
- Videos de la alfombra roja, fotos, conferencias de prensa, entrevistas, etc.
- Trailers y extractos de películas.
- Galería de fotos diarias.

Es una aplicación con un buen diseño y una amplia variedad de funcionalidades. Aunque, cabe destacar que este tipo de festivales no son los que se quieren cubrir, ya que nuestra aplicación estaría más enfocada a festivales de música al aire libre donde la gente se reúne para ver a los artistas. De todas maneras, se deberían tener en cuenta algunas funcionalidades que presenta como el acceso a la venta de entradas online o que se puedan ver videos, ya que podrían resultar de interés.

## FestivAll, Music Festival Guide

Ilustración 5. Imágenes aplicación FestivAll, Music Festival Guide



**Ofrecida por:** Google commerce Ltd

**Versión actual:** 4.2.5

**Actualizada:** 23 de enero de 2020

**Precio:** Gratuita

**Descripción:** Se trata de la aplicación para poder ver los diferentes festivales de música que se van a celebrar y poder ver la información de éstos.

**Funcionalidades:**

- Ver los diferentes festivales que tienen lugar filtrados por fecha o popularidad.
- Obtener una lista completa de las bandas que se presentarán en el festival elegido.
- Comprar tickets para los festivales.

Es una aplicación bastante útil y con un buen diseño para la búsqueda de festivales y además permite ver los diferentes artistas que se encuentran junto con información relevante de éstos, como un link de su spotify para escuchar sus canciones, últimos videoclips que han lanzado, las diferentes actuaciones que tienen en distintos festivales, entre otros. Aunque nuestra aplicación vaya a ser para un festival concreto, se ha podido comprobar que cuando se consulte información de un artista debería presentar información relevante sobre éste como se hace en esta aplicación.

### Festival One

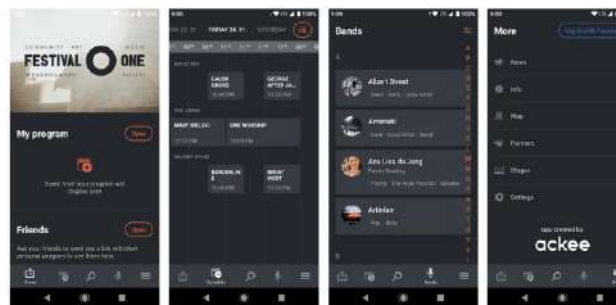


Ilustración 6. Imágenes aplicación Festival One

**Ofrecida por:** Google commerce Ltd

**Versión actual:** 4.0.3

**Actualizada:** 23 de enero de 2020

**Precio:** Gratuita

**Descripción:** Se trata de la aplicación para el festival One para poder organizarse con los diferentes eventos y poder ver los diferentes artistas en éste.

**Funcionalidades:**

- Ver la lista de artistas y eventos.
- Agregar actos favoritos para crear un propio horario personalizado.
- Conectarse con amigos y compartir horarios para el fin de semana.
- Mantenerse actualizado con información de última hora a través de notificaciones push.
- Preguntas frecuentes del festival.

A simple vista es una aplicación con un diseño muy intuitivo y muy simple. Como inconveniente se podría decir que se deberían incluir funcionalidades más útiles ya que no se puede ver ninguna sobre la compra de cupones o tickets y además se debería de mostrar un poco más de información sobre los diferentes artistas.

### Conclusiones

Este análisis ha ayudado a recopilar información de las aplicaciones similares que existen actualmente en el mercado. Se van a coger ideas para tenerlas como referencia y también se evitarán errores y problemas que se han detectado que hacen que la experiencia del usuario no sea la esperada.

Algunas de las funcionalidades recopiladas que deberían tenerse en cuenta son:

- Añadir información relevante de los artistas como la biografía, videos, página web, link de su spotify para escuchar sus canciones, últimos lanzamientos o diferentes actuaciones.
- Posibilidad de crear un horario personalizado con las actuaciones de interés.
- Comprar tickets del festival
- Poder ver videos de las actuaciones que se hayan realizado o en streaming.
- Tener un buen diseño de aplicación que resulte simple e intuitivo para el usuario y que tenga colores suaves presentando una vista minimalista.

## 2.3 Encuesta

Las encuestas permiten obtener información de manera remota, sin que los usuarios tengan que estar presentes. Además, presenta la ventaja de que la información que se obtiene tiene significación estadística.

En este caso la encuesta se ha realizado a través de google forms. En ésta se tienen diferentes preguntas para poder saber cuáles son las funcionalidades que debería tener la aplicación y cuales considera cada usuario que debe tener mayor prioridad. El link al formulario es el siguiente: <https://forms.gle/MtKcQWqbdqhsn7A8>.

En el primer bloque se recopila información básica de los usuarios para ver el rango de edad de los usuarios que están interesados en el desarrollo de una aplicación móvil para festivales.

### 1. Edad

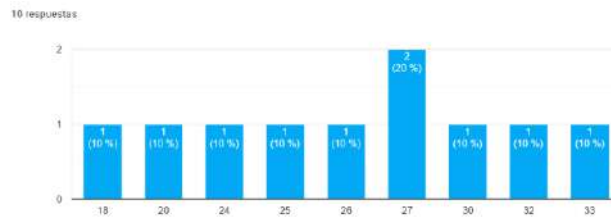


Figura 1. Resultado encuesta desarrollo de aplicación móvil (pregunta 1.1)

Se puede comprobar que el rango de edades va entre 18-33 años, por lo que se puede ver que en un principio que los usuarios a los que les ha causado interesante dicha aplicación son jóvenes. Aún así, a la hora de realizar el diseño debería hacerse lo más intuitivo y simple posible, para que a las personas que no estén acostumbradas al uso de aplicaciones móviles les resulte fácil su utilización.

**2. ¿Utilizas habitualmente internet? (redes sociales, búsqueda de ofertas, compras online, etc...)**

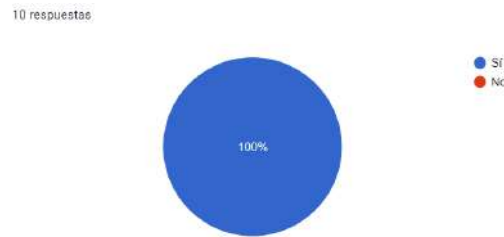


Figura 2. Resultado encuesta desarrollo de aplicación móvil (pregunta 1.2)

Todos los usuarios encuestados hacen uso de redes de forma habitual.

**3. ¿Qué sistema operativo utiliza su smartphone?**

De las 10 personas que han realizado la encuesta solamente una utiliza el sistema operativo IOS, los demás tienen Android.

En el segundo bloque se tienen preguntas de interés sobre qué debería tener la aplicación. Cabe destacar que estas preguntas han sido realizadas en base a las conclusiones sacadas en el análisis competitivo para poder ver las prioridades de éstas.

**1. ¿Le parece útil tener una aplicación en un festival?**

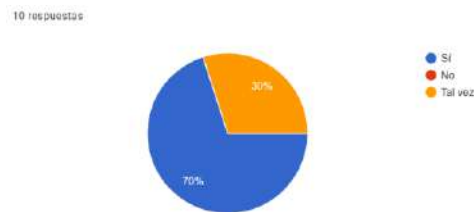


Figura 3. Resultado encuesta desarrollo de aplicación móvil (pregunta 2.1)

Se puede observar que ninguno de los encuestados considera que no es útil utilizar una aplicación Android para un festival.

**2. ¿Qué funcionalidades cree que serían necesarias?**

Una de las respuestas más comunes encontradas es que deberían verse los artistas que participan en el festival con sus horarios.

Por otro lado, en una de las respuestas se ha comentado que se debería proporcionar información de las salidas de emergencias e información del estado de seguridad. Debería tenerse en cuenta este detalle a la hora de realizar nuestra aplicación porque es información bastante relevante.

**3. Si tuviese las siguientes funcionalidades. ¿Cómo las ordenarías de mayor a menor prioridad? "Prioridad 1" corresponde a la máxima prioridad y "Prioridad 7" a la mínima.**

Se tienen las siguientes prioridades a las siguientes funcionalidades en la encuesta.

1. **Poder consultar los diferentes artistas**  
El 80% de los usuarios considera que debería ser de prioridad 1.  
El 20% de los usuarios considera que debería ser de prioridad 2.
2. **Poder crear horario personalizado con las actuaciones de los artistas**  
El 20% de los usuarios considera que debería ser de prioridad 1.  
El 30% de los usuarios considera que debería ser prioridad 3.  
El 50% de los usuarios considera que debería ser prioridad 4.
3. **Poder consultar los diferentes restaurantes o bares en el festival**  
El 10% de los usuarios considera que debería ser de prioridad 1.  
El 40% de los usuarios considera que debería ser de prioridad 2.  
El 50% de los usuarios considera que debería ser de prioridad 3.
4. **Capacidad de recibir notificaciones con alertas**  
El 50% de los usuarios considera que debería ser de prioridad 1.  
El 50% de los usuarios considera que debería ser de prioridad 2.
5. **Método de pago para poder comprar consumiciones**  
El 30% de los usuarios considera que debería ser de prioridad 1.  
El 10% de los usuarios considera que debería ser de prioridad 2.  
El 30% de los usuarios considera que debería ser de prioridad 3.  
El 30% de los usuarios considera que debería ser de prioridad 5.
6. **Posibilidad de comprar tickets para el festival**  
El 20% de los usuarios considera que debería ser de prioridad 2.  
El 40% de los usuarios considera que debería ser de prioridad 4.  
El 40% de los usuarios considera que debería ser de prioridad 6.
7. **Poder consultar el mapa del festival**  
El 30% de los usuarios considera que debería ser de prioridad 1.  
El 30% de los usuarios considera que debería ser de prioridad 2.  
El 40% de los usuarios considera que debería ser de prioridad 3.

Por lo tanto se puede ver que las funcionalidades más prioritarias según la encuesta son las de poder consultar los diferentes artistas, la de recibir notificaciones con alertas y la posibilidad de consultar el mapa del festival.

#### 4. ¿Le gustaría poder compartir fotos a través de la aplicación? ¿Por qué?

La respuesta a esta pregunta por parte de todos los usuarios ha sido negativa porque ya se tienen redes sociales para poder compartir fotos del festival, por lo que no debería ser necesario incluir esta funcionalidad.

#### 5. ¿Le gustaría que se incluyesen en la aplicación vídeos sobre las actuaciones? ¿Por qué?

Aunque la mayoría de las respuestas ha sido que no, se han encontrado dos usuarios que les parecería buena idea poder ver los festivales en directo. Por tanto, en caso de añadir dicha funcionalidad debería plantearse realizar la transmisión de los vídeos en directo.

### Conclusiones

Se ha podido ver que de las funcionalidades que deberían tenerse en cuenta recopiladas gracias al análisis competitivo, algunas no resultan prioritarias para los usuarios.

- Se considera que la opción de ver vídeos en streaming no debería ser algo prioritario, ya que la mayoría de usuarios considera que dicha opción no es necesaria.
- La posibilidad de crear horarios personalizados no destaca entre las de mayor prioridad, es por ello que en caso de querer implementar dicha funcionalidad se realizará en etapas posteriores al proyecto, y no estará en la planificación de este semestre.
- La posibilidad de ver los puntos de salida de emergencia se hará a través del mapa.

## 2.5 Observación con usuarios

Con el objetivo de poder evaluar alguna aplicación con funcionalidades similares a nuestra aplicación, se ha realizado una observación con usuarios con la aplicación "Sziget Festival", ya que se considera que es la más completa respecto a funcionalidades y tiene un diseño bastante intuitivo. Esta observación se hace para comprobar cómo se maneja a nivel usuario. Se quiere saber la facilidad de uso, las mejoras que se podrían realizar, los diferentes elementos que se deberían destacar más, entre otros.

Para realizar las sesiones se ha utilizado google forms como formulario donde se tienen las diferentes preguntas para saber el perfil de los usuarios y las tareas a realizar para poder ver su opinión respecto a las dificultades que se encuentran al realizar determinadas acciones en la plataforma. Aunque cabe destacar que el formulario sirve de guía y para apuntar los resultados, pero las sesiones han sido a través de Skype. El formulario está disponible en el siguiente enlace: <https://forms.gle/PHzxpsjCzuX3mea78>.

### Perfiles

Se presenta la información obtenida de los diferentes perfiles de usuario.

#### **Personas jóvenes**

Usuario 1:

- Edad: 26 años.
- Ciudad: Gante.
- Ocupación: con empleo jornada completa.
- Estado: con pareja.
- Hijos: no.
- Utiliza tanto en su trabajo como en su vida social internet a través del ordenador y smartphones: redes sociales, búsqueda de ofertas, compras online, entre otros.
- Vida activa.

Usuario 2:

- Edad: 16 años.
- Ciudad: Bilbao
- Ocupación: estudiante.
- Estado: soltero.
- Hijos: no.
- Utiliza tanto en su trabajo como en su vida social internet a través del ordenador y smartphones: redes sociales, búsqueda de ofertas, compras online, entre otros.
- Vida activa.

## Personas de mediana edad

Usuario 1:

- Edad: 40 años.
- Ciudad: Bilbao.
- Ocupación: con empleo jornada completa.
- Estado: con pareja.
- Hijos: no.
- Utiliza tanto en su trabajo como en su vida social internet a través del ordenador y smartphones: redes sociales, búsqueda de ofertas, compras online, entre otros.
- Vida activa.

Usuario 2:

- Edad: 43 años.
- Ciudad: Bilbao
- Ocupación: estudiante.
- Estado: con pareja.
- Hijos: sí.
- Utiliza tanto en su trabajo como en su vida social internet a través del ordenador y smartphones: redes sociales, búsqueda de ofertas, compras online, entre otros.
- Vida activa.

## Formulario de consentimiento

Debido a que la información recopilada del usuario a partir del cuestionario demográfico no da pie a que pueda ser identificado y las sesiones realizadas no son grabadas, no se ha realizado un formulario de consentimiento.

## Contexto de uso

Antes de realizar la sesión se informa al usuario que para realizarla es necesario un dispositivo móvil con sistema operativo Android, y la aplicación que se quiere evaluar se utilizaría cuando se estuviese en un festival o se quisiese consultar información de éste.

## Tareas

### Tarea 1

<b>Escenario</b>	Imagina que quieres ir al festival de Sziget y quieres consultar los artistas que acuden. Descárgate la aplicación "Sziget Festival". Una vez realizado esto, intenta buscar a la artista "Dua Lipa" para ver información relevante sobre ella.
<b>Tarea</b>	Búsqueda de artistas.
<b>Objetivos</b>	<ul style="list-style-type: none"><li>• Comprobar la facilidad con la que se puede buscar un artista en el sistema.</li><li>• Comprobar que la información que se presenta del artista es útil.</li></ul>
<b>Preguntas</b>	<ul style="list-style-type: none"><li>• ¿Le ha resultado difícil encontrar la lista de artistas?</li><li>• ¿Qué información echas de menos al ver los datos de Dua Lipa? ¿Qué quitarías?</li></ul>

Todos los usuarios han conseguido realizar la búsqueda sin ningún problema. Pero se han tenido algunos comentarios respecto al diseño y a la información que se presenta del artista.

- Todos los usuarios han comentado que la aplicación es demasiado colorida.
- Aparece un calendario junto con el texto "miércoles" y no especifica qué día exactamente corresponde a la actuación, lo que ha ocasionado frustración a algunos usuarios.
- Aparece un icono de reproducir que cuando se pulsa empieza a sonar una canción del artista, pero se considera que debería ser mejor poder ver todas sus canciones, en vez de escuchar una al azar.

## Tarea 2

<b>Escenario</b>	Imagina que quieres organizarte para poder ver ciertas actuaciones. Busca los diferentes horarios de las actuaciones y selecciona varios como favoritos. A continuación, visualiza solamente los favoritos seleccionados.
<b>Tarea</b>	Búsqueda de horarios y selección de favoritos.
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Comprobar la facilidad con la que se puede encontrar los horarios en el sistema.</li> <li>• Comprobar la facilidad con la que se pueden seleccionar las actuaciones favoritas.</li> </ul>
<b>Preguntas</b>	<ul style="list-style-type: none"> <li>• ¿Le ha resultado difícil encontrar los horarios?</li> <li>• ¿Has podido seleccionar fácilmente los horarios preferidos y ver una lista sólo de éstos?</li> </ul>

A dos de los usuarios, en este caso los perfiles de mediana edad, les ha costado bastante encontrar la opción de ver solamente los seleccionados. Se considera que se debería poner en el menú la opción de ver solamente los favoritos en vez de que cuando se presenten todos los horarios tener que pulsar a un icono concreto.

## Tarea 3

<b>Escenario</b>	Imagina que te vienen bien las fechas para poder ir al festival y quieres comprar los tickets. Busca dónde se encuentra la venta de tickets y selecciona uno para proceder a comprarlo. (Nota: No es necesario que lo compres, sino que encuentres dónde se realiza dicha acción).
<b>Tarea</b>	Intentar comprar un ticket.
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Comprobar la facilidad con la que se puede acceder a la venta de tickets.</li> <li>• Comprobar facilidad para proceder al pago.</li> </ul>
<b>Preguntas</b>	<ul style="list-style-type: none"> <li>• ¿Has podido encontrar dónde se compran entradas?</li> <li>• ¿Has tenido alguna dificultad intentando comprar las entradas?</li> </ul>



Gracias al menú es bastante fácil acceder a la opción de compra de tickets, es por ello que ningún usuario ha tenido problema en encontrar la opción. En cambio, cuando se va a proceder a realizar el pago ha habido 3 usuarios que les ha costado encontrar la opción de pagar. Esto es debido a que aparece una opción de añadir al carro, pero para proceder al pago hay que ir hasta el final de la lista donde se tiene un botón que dice "coger estos artículos". Por tanto, se considera que debería proporcionarse una manera más intuitiva de compra.

Por otro lado, ha habido un usuario que le ha llamado la atención que en la pantalla de compra de billetes se tiene la opción de cambiar el idioma, pero en las otras pantallas no había un botón para ello.

#### Tarea 4

<b>Escenario</b>	Ahora imagina que ya estás en el festival y que quieres ir a un escenario concreto para poder ver a Dua Lipa. Para ello, intenta acceder al mapa y ver dónde se encuentran los escenarios.
<b>Tarea</b>	Búsqueda los diferentes espacios en el mapa.
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Comprobar la facilidad con la que se puede acceder al mapa.</li> <li>• Comprobar si la visualización del mapa es correcta.</li> </ul>
<b>Preguntas</b>	<ul style="list-style-type: none"> <li>• ¿Has tenido alguna dificultad al visualizar el mapa?</li> <li>• ¿Te gusta la visualización del mapa?</li> </ul>

A ninguno de los usuarios le ha costado encontrar esta opción, pero todos han comentado que el diseño podría mejorarse.

#### Conclusiones

Analizando toda la información se pueden sacar las siguientes conclusiones:

- El diseño de la interfaz no debe ser muy colorido, ya que puede causar molestia a los usuarios. Por el contrario, sería mejor un diseño minimalista.
- La funcionalidad de selección de actuaciones favoritas debería incluirse en el menú principal, es decir, la opción de ver los seleccionados, en vez de tener que buscar en la pantalla, donde se ve la lista completa, qué botón permite hacer esto.
- El método de pago a través de la aplicación debe ser lo más intuitivo posible.
- Debe poder cambiarse el idioma en cualquier lugar de la aplicación, no solamente en una pantalla concreta.

#### 2.6 Conclusiones del análisis

Gracias a la realización del análisis se ha podido comprobar cómo debería estar realizado el diseño de la interfaz para proporcionar facilidad y comodidad a los usuarios. Además, se han vistos diferentes fallos en distintas aplicaciones similares y por tanto en el diseño de nuestra aplicación deberían tenerse en cuenta. Por otro lado, también se han visto las necesidades reales de los usuarios en cuanto a la prioridad de las funcionalidades.

Por tanto, se obtienen los siguientes resultados en el análisis:

- Se debe realizar un diseño minimalista que sea intuitivo y simple para el usuario.
- No se van a añadir las funcionalidades adicionales obtenidas en el análisis competitivo ya que en la encuesta se ha comprobado que a los usuarios no les parecen prioritarias. En caso de ser añadidas serán en versiones posteriores de la aplicación.
- La consulta de los horarios se realizará a través de los artistas, ya que en la encuesta se ha visto un interés mayor por consultar artistas que escenarios.
- La funcionalidad para poder pagar en el festival a través de la aplicación deberá ser lo más intuitiva posible para no ocasionar confusión al usuario.

## 3. Diseño

### 3.1 Introducción

En esta fase se comenzará a realizar el diseño de la aplicación a partir de los datos recopilados en el análisis. Inicialmente se definirán escenarios de uso para ver cómo sería la interacción del usuario. A continuación se harán los diferentes flujos de interacción y con esta información se pasará a realizar los prototipos de baja y alta fiabilidad.

### 3.2 Escenarios de uso

Los escenarios de uso describen en detalle la interacción de un usuario con la aplicación con una cierta motivación y un objetivo específico. En éstos se cuenta una historia corta de una persona que hace uso de la aplicación en la cual se incluye toda la información relevante que el usuario experimenta para alcanzar su objetivo.

Cabe destacar que los perfiles que utilizarán dicha aplicación, como se ha podido comprobar en el análisis serán perfiles jóvenes y de mediana edad. Es por ello, que los escenarios de uso se realizarán con este tipo de perfiles.

#### **Escenario 1. Consultar los horarios de un artista**

"Enrique es un hombre de 40 años que se ha enterado de que su grupo favorito va a ir a su ciudad porque se ha organizado un festival con varios artistas. Debido a motivos de trabajo a Enrique solo le gustaría ir a ver a su grupo favorito y por tanto necesita saber exactamente el día y hora en la que actuarán. Su amigo le ha dicho que normalmente se cuelgan fotos en Instagram cuando comience el festival donde aparecerá el horario, pero Enrique no quiere comprar el ticket porque igual el horario no le permite acudir. Este problema se lo comentó a su mujer y ella le dijo que había leído en las noticias que se había creado una aplicación donde podía consultar esto, por lo que él se dispuso a descargarla y probarla. Al ver que podía ver todos los artistas que acudirían al festival con su horario específico de su actuación se puso muy contento porque sí podía acudir. Además, pudo ver que otro artista que le gustaba iba a actuar después de su grupo favorito."

#### **Escenario 2. Consultar de los diferentes restaurantes o bares**

"María es una estudiante de psicología en la universidad de la UOC de 23 años. Últimamente ha estado estudiando mucho así que le gustaría tomarse un respiro. Por ello, ha decidido con sus amigas acudir a un festival que se celebra en su misma ciudad y dura dos días, por lo que tendrán que quedarse a dormir allí. Aunque está muy emocionada se encuentra un poco preocupada porque no puede comer gluten y no sabe qué tipo de comida se va a ofrecer e igual necesita llevar su comida en la mochila. Debido a su preocupación ha llamado a atención al cliente y le han informado de que había una aplicación donde podría ver esto. Al saber esto, María se la ha descargado y se ha puesto muy contenta al ver que podía ver todos los restaurantes y qué tipo de comida iban a ofrecer."

#### **Escenario 3. Consultar mapa del festival**

"Luis es un hombre de 37 años que tiene 3 hijas. La hija mayor le dijo que se iba a organizar al lado de su casa un evento de música clásica y que le gustaría ir. Él no quería ir porque todavía recuerda que cuando fueron a Disney se perdieron porque no tenían el folleto con el mapa y no quería vivir otra vez esa experiencia. Su mujer también quería ir pero Luis seguía muy cabezota, así que se informó sobre el evento y vio que existía una aplicación donde podía ver el mapa y todos los escenarios. Por ello, al final acudieron."

#### Escenario 4. Pagar a través del móvil evitando colas de espera.

"Gaizka es un hombre de 27 años jugador de baloncesto. Hace poco se lesionó y le tuvieron que operar de la rodilla y tuvo que estar un mes en casa. Ahora ya se encuentra mejor pero no puede estar mucho tiempo de pies. Su amigo Matías, también jugador de baloncesto organizó con el equipo una sorpresa para llevarlo a un festival muy conocido de Barcelona. Pero cuando le informaron a Gaizka, él se negó ya que además de que odiaba tener que esperar las colas para la compra de cupones su rodilla no se lo iba a permitir. Unos días más tarde, se empezó a arrepentir y se intentó informar sobre cuántas personas iban a acudir, y para su sorpresa vio que existía una aplicación del festival. Cuando se la descargó vio que en este festival no se requería esperar colas de cupones ya que se podía consumir pagando a través de la aplicación, por lo que al final decidió ir. "

#### Escenario 5. Recibir notificaciones en caso de avisos

"Elena es una estudiante en prácticas en la empresa Idom de Bilbao. Todos los becarios son muy amigos y decidieron ir juntos a un festival pero Elena no estaba convencida porque la última vez que fue a uno no acudió el artista que quería ver, y estuvo toda la noche esperando para nada porque nadie le informó de que se había cancelado su actuación. Aún así, uno de sus amigos, Pedro, le consiguió convencer, pero con la condición de que se tenía que encargar de cotillear en Instagram a su artista favorito en todo momento para ver si iba a ir o no. Por suerte, Pedro sabía que la aplicación del festival informaba de todo tipo de cambios, retrasos u cancelaciones así que pudo disfrutar del festival y no tuvo que estar pendiente de las redes sociales. "

### 3.3 Flujos de interacción

Para poder tener una visión esquematizada de los escenarios de uso, se van a realizar los flujos de interacción, los cuales son una representación gráfica del proceso paso a paso del escenario en concreto.

#### Flujo interacción. Consultar los horarios de un artista

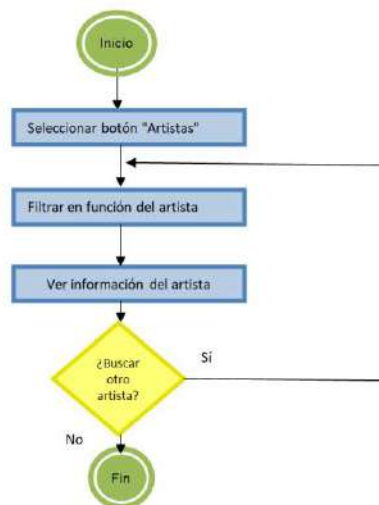


Figura 4. Flujo de interacción para consultar los horarios de un artista

Cómo se ha dicho anteriormente, gracias al proceso de análisis se sabe que la consulta de horarios es más interesante realizarla a través de los artistas en vez de los escenarios. Los usuarios prefieren consultar los artistas que acudirán y los diferentes horarios que tendrán.

Por ello, en el flujo de interacción cuando se filtra un artista se podrá ver el horario que tendrá.

#### Flujo interacción. Consultar de los diferentes restaurantes o bares

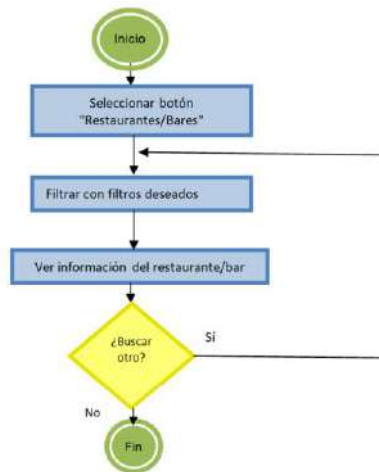


Figura 5. Flujo de interacción para consultar los restaurantes o bares

El flujo de interacción para consultar los diferentes restaurantes o bares será parecido a la consulta de horarios, pero en esta caso el restaurante mostrará información sobre los diferentes comidas que ofrecerá y las opiniones de éstos.

#### Flujo interacción. Consultar mapa del festival



Figura 6. Flujo de interacción para consultar mapa

El flujo de interacción de la consulta del mapa es muy simple, ya que sólo requerirá pulsar desde la pantalla inicial el botón "mapa" y se podrán ver dónde se encuentran todos los elementos del festival. En caso de querer encontrar una ubicación se realizará una búsqueda.

## Flujo interacción. Método de pago evitando colas de espera

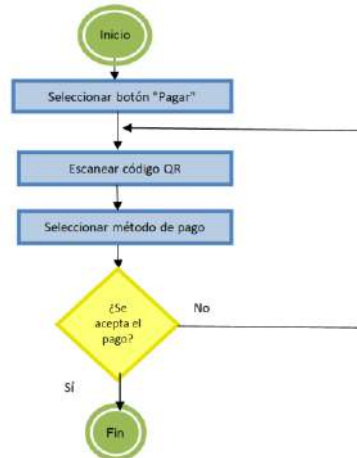


Figura 7. Flujo de interacción para método de pago evitando colas de espera

Cómo se ha podido ver en el análisis la forma de pagar debe ser lo más simple e intuitiva posible, ya que si no puede causar frustración al usuario y esto puede ocasionar que no se realice el pago. Es por ello que el flujo de interacción en este caso no es muy complicado, simplemente se tiene que pulsar desde la pantalla inicial el botón "pagar/consumir" y a continuación se procede escanear un código QR. A continuación se preguntará por el método de pago y si se acepta el pago o no.

Por otro lado, cabe destacar que no se ha realizado un flujo de interacción de la funcionalidad relacionada con el recibimiento de notificaciones porque el usuario no debe realizar ninguna interacción para poder recibirlas. En caso de que se produzca un cambio, retraso u cancelación, el sistema automáticamente enviará una notificación a los usuarios.

Por lo que el flujo de interacción resultante para todas las funcionalidades de la aplicación sería de la siguiente manera.

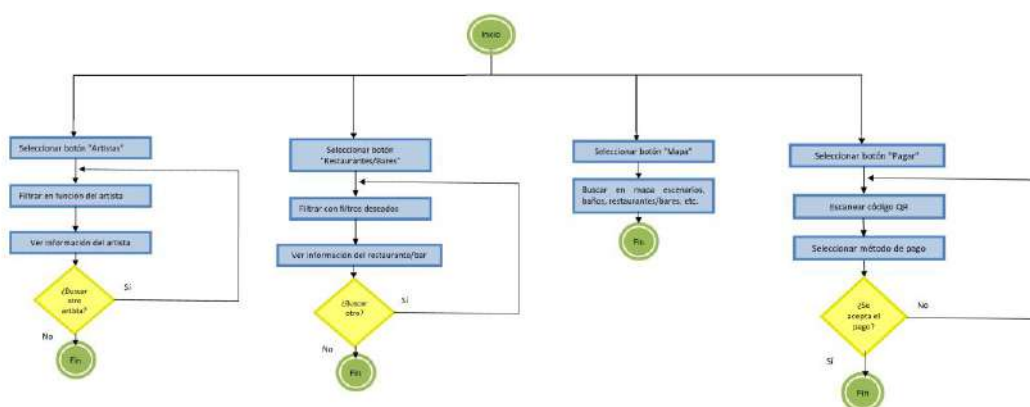


Figura 8. Flujo de interacción de la aplicación

### 3.4 Prototipo de baja fidelidad

En este apartado se comienza a plasmar toda la información recogida en las fases anteriores en un modelo visual del sistema a diseñar. Inicialmente se realiza un prototipo de baja fidelidad debido a su facilidad de creación y la de realizar cambios ya que no se necesita conocimiento de ninguna herramienta para su diseño. Se deberán incluir en éste todas las funcionalidades de la aplicación pero sin entrar mucho en detalle de tal forma que se pueda tener una primera visualización de la aplicación.

#### Menú principal

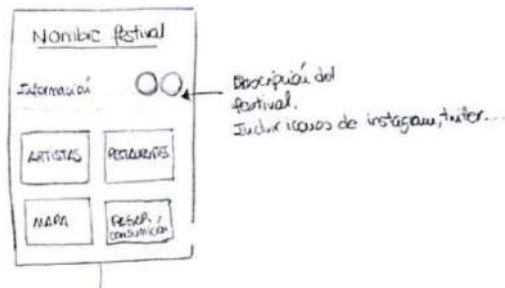


Ilustración 7. Prototipo de baja fidelidad del menú principal

La primera vez que se acceda a la aplicación se tendrá una pantalla como la de la ilustración 7. En ésta se tendrán los botones principales para poder acceder a las diferentes funcionalidades. Además, se tendrá una pequeña descripción o eslogan del festival concreto y se incluirán iconos para poder acceder a las cuentas oficiales de instagram, twitter, entre otros.

#### Artistas



Ilustración 8. Prototipo de baja fidelidad para la consulta de artistas y horarios

Para poder consultar los diferentes artistas y sus horarios, se debe pulsar el botón de "artistas" desde el menú principal. En la lista se deberá ver una pequeña imagen junto con el nombre e información relevante en cuanto a su estilo de música y el escenario en el que actúa. Además, se tiene la opción de poder filtrar en la lista en función de los campos deseados.

En caso de presionar un artista concreto, se podrá ver la misma imagen que aparecía anteriormente pero con un tamaño mayor y se incluirá más información sobre éste.

### Restaurantes o bares



Ilustración 9. Prototipo de baja fidelidad para la consulta de restaurantes o bares

Para poder consultar los diferentes restaurantes o bares, se debe pulsar el botón de "restaurantes/bares" desde el menú principal. En la lista se deberá ver una pequeña imagen junto con el nombre y se añadirá una puntuación a éstos en base a las opiniones de los usuarios.

En caso de presionar un restaurante o bar concreto, se podrá ver la misma imagen que aparecía anteriormente pero con un tamaño mayor, imágenes del menú que se ofrece y las diferentes opiniones de los usuarios. Además, se tendrá un botón para poder añadir opiniones adicionales.

### Mapa

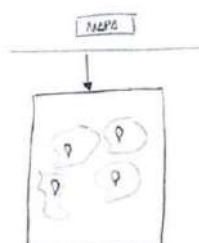


Ilustración 10. Prototipo de baja fidelidad para el mapa

Para poder consultar el mapa, se debe pulsar el botón de "mapa" desde el menú principal. Una vez se accede, se podrá ver directamente el mapa con los puntos principales del festival.



## Método de pago evitando colas de espera

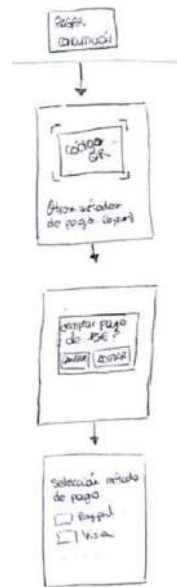


Ilustración 11. Prototipo de baja fidelidad para el método de pago para evitar colas de espera

Para poder realizar el pago a través de la aplicación se tendrá que ir a los restaurantes y el camarero entregará un código QR que deberá escanearse. A continuación aparecerá una alerta en la cual se preguntará si se está de acuerdo con el pago o no y se seleccionará la aplicación con la que se desea pagar (paypal, tarjeta crédito, entre otros).

## Notificaciones



Ilustración 12. Prototipo de baja fidelidad para las notificaciones

Las notificaciones se recibirán en cualquier pantalla de la aplicación o incluso cuando no se esté dentro de ésta. Aparecerá un pop-up en la parte superior con el aviso en concreto.

Una representación completa del prototipo de baja fidelidad es de la siguiente manera:

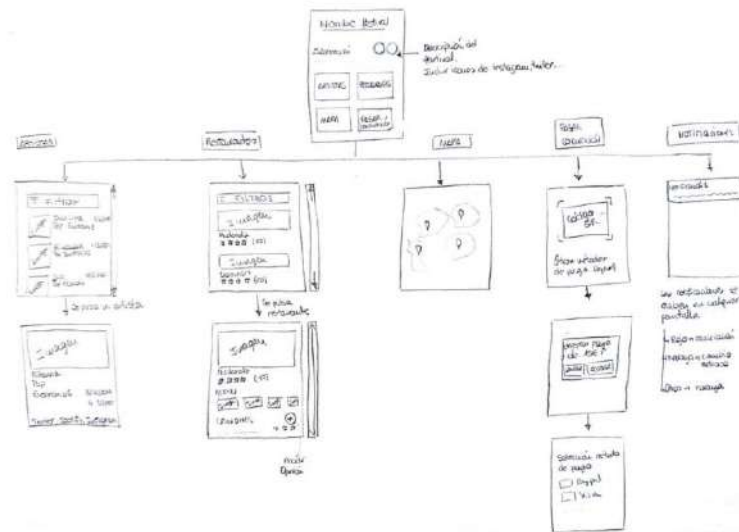


Ilustración 13. Prototipo de baja fidelidad

### 3.5 Prototipo de alta fidelidad

Tras haber realizado el prototipo de baja fidelidad, en el cual se han podido realizar de forma rápida los cambios para que la aplicación contenga las funcionalidades principales con una primera idea del diseño que se tendrá, se pasa a realizar el prototipo de alta fidelidad.

#### Menú principal

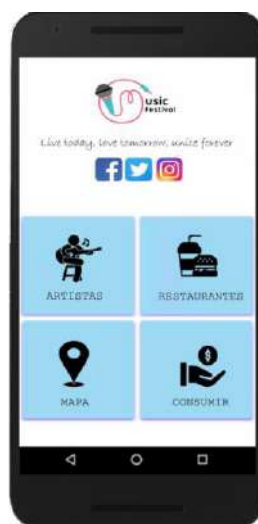


Ilustración 14. Prototipo de alta fidelidad del menú principal

Se puede observar que el menú principal presenta un diseño minimalista bastante sencillo e intuitivo para que sea fácil su comprensión. Inicialmente se tiene un pequeño eslogan del festival y se incluyen sus redes sociales relevantes. A continuación, se han incluido botones del mismo estilo de tal manera que se sigue la ley de similitud, la cual indica que los elementos

que comparten características visuales son interpretados de forma conjunta, en este caso corresponden a las funcionalidades de la aplicación.

## Artistas

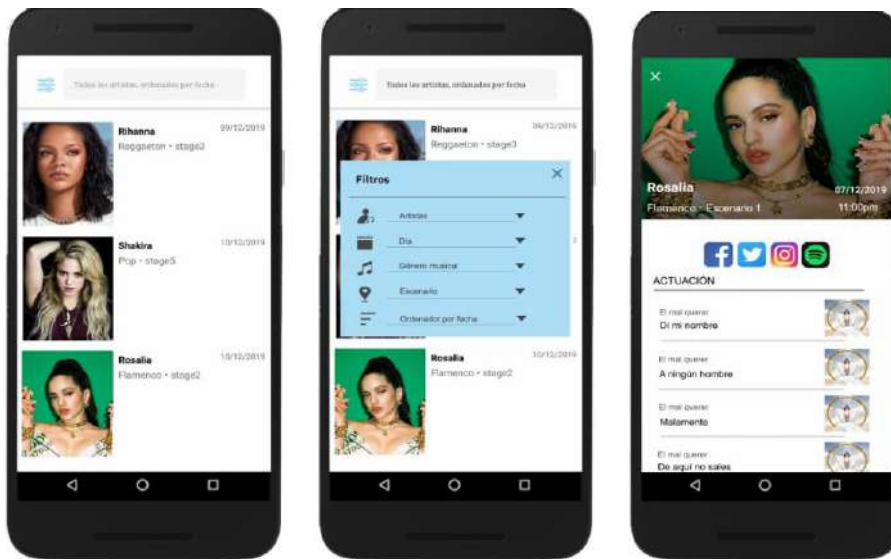


Ilustración 15. Prototipo de alta fidelidad para la consulta de artistas y horarios

Cuando se pulsa el botón "artistas" del menú principal se podrá ver una lista con todos ellos junto con información relevante. Además, cuando se quiera filtrar aparecerá una ventana pequeña con todas las opciones con las que se puede buscar.

En caso de seleccionar un artista se verá información sobre el lugar, fecha y hora de actuación y sus redes sociales. También se ha considerado relevante incluir una lista de las diferentes canciones que cantará en el festival.

Se puede observar que el estilo sigue siendo muy simple, ya que se quiere mostrar de manera eficaz la información. Además, cabe destacar que se intenta seguir con el color principal de la aplicación, el azul, es por ello que la ventana de filtros tiene el mismo color que los botones del menú principal.

## Restaurantes o bares

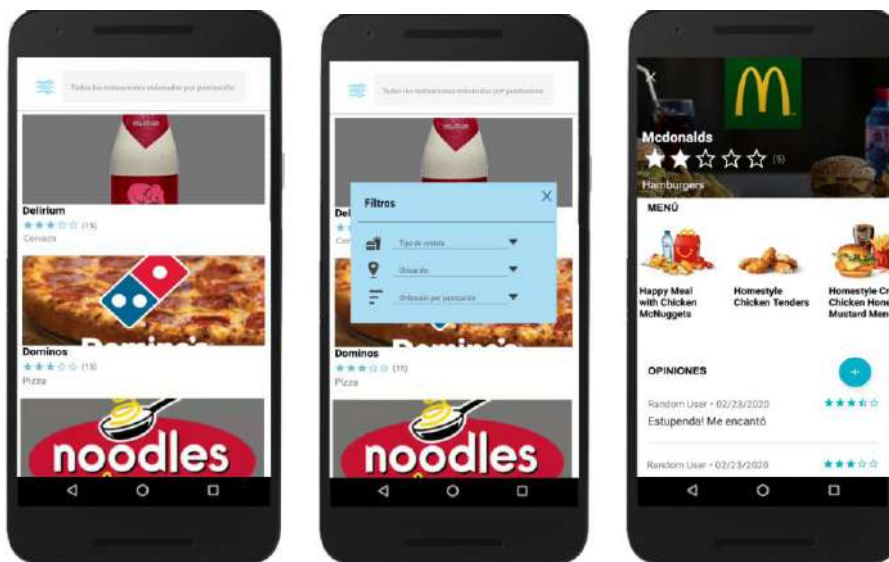


Ilustración 16. Prototipo de alta fidelidad para la consulta de restaurantes o bares

En caso de presionar el botón restaurantes desde el menú principal se verá una lista de éstos. Se puede ver que el diseño es muy parecido al de la lista de artistas, pero en este caso las imágenes ocupan todo el ancho de la aplicación. Esto es debido a que se quiere focalizar la atención en las imágenes ya que mostrarán el tipo de comida que se ofrecerá, en cambio en los artistas la información importante son el lugar y fecha de actuación.

Por otro lado, cuando se selecciona un restaurante aparecerá el menú con imágenes y el nombre del plato, y a continuación aparecerán las diferentes opiniones de los usuarios con la posibilidad de añadir más.

## Mapa

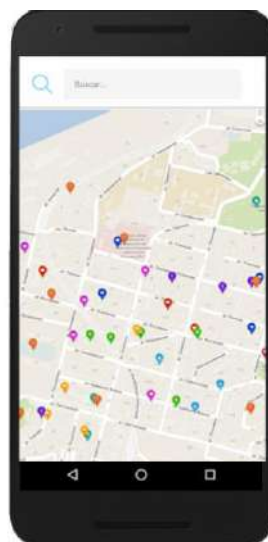


Ilustración 17. Prototipo de alta fidelidad para el mapa

En el mapa se verán todos los puntos de interés y se tendrá la posibilidad de buscar un punto concreto, como podría ser la ubicación de un escenario.

## Método de pago evitando colas de espera



Ilustración 18. Prototipo de alta fidelidad para el método de pago

Se ha intentado realizar un método muy fácil de interpretar para que no dé lugar a errores. Nada más acceder a esta funcionalidad se deberá escanear un código QR. Una vez escaneado se seleccionará el método de pago, el cual redireccionará a la página oficial para introducir los datos necesarios. A continuación el camarero, el cual tiene el código QR, recibirá la notificación de pago realizado.

A la hora de realizar el prototipo de alta fidelidad se han tenido presentes los principios básicos de diseño. Por tanto en el diseño se ha buscado que:

- Se garantice un uso equiparable, es decir, que proporcione las mismas formas de uso para todos los usuarios, teniendo las mismas características y teniendo un diseño atractivo.
- Se garantice un uso flexible para que pueda usarse con la mano izquierda y derecha, y se adapta al paso o ritmo del usuario.
- Presente un diseño simple e intuitivo para su fácil entendimiento.
- La información sea perceptible comunicando de manera eficaz la información necesaria al usuario.
- Se minimicen los riesgos de realizar acciones involuntarias o accidentales.
- Exija poco esfuerzo físico si puede ser usado eficaz y confortablemente.
- Proporcione un tamaño y espacio apropiados para el acceso, el alcance, la manipulación y el uso.

## 4. Evaluación

### 4.1 Introducción

En esta fase se busca realizar una evaluación del diseño del prototipo realizado, con el objetivo de encontrar posibles errores y mejorarlo. Las técnicas que pueden llevarse a cabo para esta evaluación pueden ser métodos con usuarios y sin usuarios, como podría ser una evaluación con test con usuarios y una evaluación heurística, respectivamente.

### 4.2 Test con usuarios

Cómo ya se podido ver en la fase de análisis, este método permite saber la facilidad de uso, las mejoras que se podrían realizar, los diferentes elementos que se deberían destacar más, entre otros.

Las sesiones se van a realizar de forma remota siguiendo un guión, el cual podrá compartirse al usuario a través de google forms: <https://forms.gle/ZBZQypYGXUMpnrPW6>. En éste se recopilará características del perfil de usuario y se indicarán las diferentes tareas a realizar. En el caso de que la sesión sea grabada deberá presentarse un formulario de consentimiento.

Una vez ejecutadas todas las tareas y respondidas las preguntas, se realizará un breve cuestionario sobre qué le ha parecido la sesión al usuario.

#### Perfiles

Los perfiles que deberán ser entrevistados serán del mismo perfil que en la fase de análisis ya que se ha visto que los usuarios que utilizarán la aplicación serán perfiles jóvenes y de mediana edad.

La información que se recopilará para saber el perfil del usuario serán las siguientes:

- Edad.
- Ciudad.
- Ocupación:
  - En paro.
  - Estudiante.
  - Con empleo a jornada completa.
  - Con empleo a media jornada.
  - Otro.
- Estado:
  - Soltero.
  - Con pareja.
  - Otro.
- Hijos:
  - Sí.
  - No.
- ¿Utilizas habitualmente internet? (redes sociales, búsqueda de ofertas, compras online,etc...).
  - Sí.
  - No.
- ¿Le gusta realizar actividades al aire libre? ¿Ir a eventos , fiestas, festivales u otros?.

- Nunca.
- Casi nunca.
- A veces.
- Casi siempre.
- Siempre.

### **Tareas del test**

#### **Tarea 1**

<b>Escenario</b>	Imagina que quieres ir un festival muy conocido y acaban de lanzar su aplicación. En ésta quieres consultar los artistas que acuden para poder ver si está tu favorita y poder ver su horario de actuación.
<b>Tarea</b>	Búsqueda de artistas junto con su horario.
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Comprobar la facilidad con la que se puede buscar un artista en la aplicación.</li> <li>• Comprobar que la información que se presenta del artista es útil.</li> </ul>
<b>Preguntas</b>	<ul style="list-style-type: none"> <li>• ¿Le ha resultado difícil encontrar la lista de artistas?</li> <li>• ¿Has podido ver el horario de actuación?</li> <li>• ¿Qué información echas de menos al seleccionar el artista?</li> </ul>

En general las respuestas obtenidas en este apartado han sido muy positivas. A los usuarios les ha sorprendido toda la información que se puede obtener de un artista. Además, el hecho de que se presenten las diferentes redes sociales de éstos supone comodidad a los usuarios ya que tienen un acceso directo a toda su información y trabajo.

Por otro lado, se ha visto a partir de los comentarios que el punto de mejora más evidente es el de la paleta de colores usada especialmente en el azul cielo de las emergentes.

#### **Tarea 2**

<b>Escenario</b>	Imagina que quieres consultar la comida que se ofrece en el festival ya que eres vegana y no sabes si habrá restaurantes de este tipo.
<b>Tarea</b>	Búsqueda de restaurantes.
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Comprobar la facilidad con la que se puede buscar un restaurante en la aplicación</li> <li>• Comprobar que la información que se presenta del restaurante es útil.</li> </ul>
<b>Preguntas</b>	<ul style="list-style-type: none"> <li>• ¿Le ha resultado difícil encontrar la lista de restaurantes?</li> <li>• ¿Qué información echas de menos al seleccionar un restaurante?</li> </ul>

En este apartado también se ha comentado que debería modificarse la paleta de colores utilizada para la ventana emergente. El resto de comentarios obtenidos han sido positivos, por lo que la información que se presenta para los restaurantes resulta correcta.

### Tarea 3

<b>Escenario</b>	Imagina que ya te has organizado y sabes en qué escenarios actuarán los artistas que quieres ver pero te gustaría saber si los servicios están cerca de éstos.
<b>Tarea</b>	Consulta de mapa.
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Comprobar la facilidad con la que se accede al mapa.</li> <li>• Comprobar que la información que se presenta del mapa es la apropiada</li> </ul>
<b>Preguntas</b>	<ul style="list-style-type: none"> <li>• ¿Le ha resultado difícil acceder al mapa?</li> <li>• ¿La información presentada es suficiente? ¿Qué añadirías?</li> </ul>

En esta tarea se ha obtenido como sugerencia que se deberían tener diferentes iconos para cada ubicación, es decir, que se tenga un icono para los escenarios, para los restaurantes, los baños, entre otros.

### Tarea 4

<b>Escenario</b>	Imagina que ya estás en el festival y te gustaría comprar algo para comer pero no quieres esperar en la cola de cupones. Utiliza la aplicación para poder pagar la comida.
<b>Tarea</b>	Utilizar método de pago de la aplicación.
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Comprobar la facilidad con la que se accede al pago a través de la aplicación.</li> <li>• Comprobar que el proceso de pago es sencillo.</li> </ul>
<b>Preguntas</b>	<ul style="list-style-type: none"> <li>• ¿Le ha resultado difícil encontrar cómo se puede pagar a través de la aplicación?</li> <li>• ¿Le ha parecido sencillo el método de pago? ¿Qué mejorarías?</li> </ul>

En esta tarea se han tenido varios comentarios indicando que en vez de tener que seleccionar el tipo de tarjeta que se tiene, (maestro, visa, entre otros) que simplemente se pueda introducir la información necesaria sin necesidad de seleccionar el tipo.

Por otro lado, también se considera que debe modificarse el color de la ventana emergente.

### Conclusiones

Por tanto, analizando toda la información recopilada a través de la evaluación se pueden sacar las siguientes conclusiones:

- Se debe modificar la paleta de colores utilizada especialmente el azul cielo de las ventanas emergentes.



- Se deben introducir diferentes iconos para el tipo de localización que se está buscando: baños, escenarios, restaurantes, entre otros.
- Se debería poder introducir una tarjeta sin necesidad de seleccionar el tipo de tarjeta que es.

### 4.3 Revisión del prototipo

A partir de las conclusiones obtenidas en la evaluación con el test con usuarios se van a realizar las modificaciones necesarias sobre el prototipo de alta fidelidad inicialmente realizado.

A continuación se presentan las modificaciones realizadas:

#### Artistas

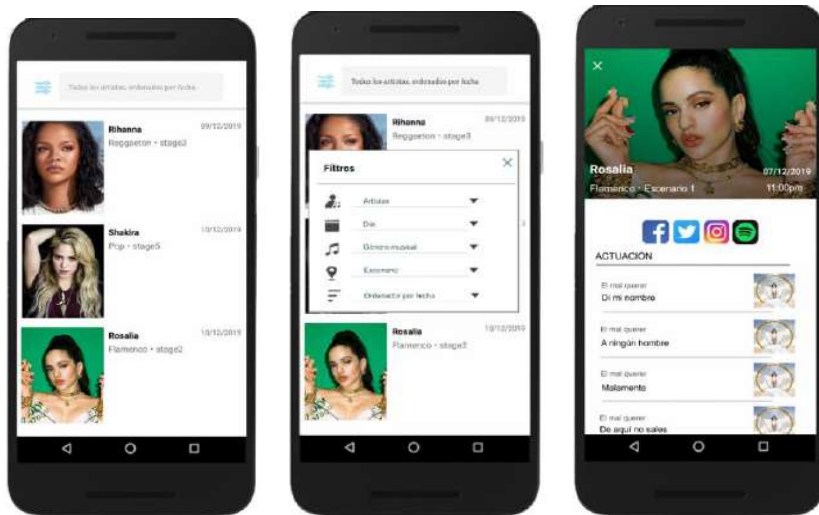


Ilustración 19. Modificación prototipo de alta fidelidad para artistas

Se ha modificado el color de las ventana emergente para realizar búsqueda de artistas.

#### Restaurantes o bares



Ilustración 20. Modificación prototipo de alta fidelidad para restaurantes o bares

Se ha modificado el color de las ventana emergente para realizar búsqueda de restaurantes o bares.

### Mapa

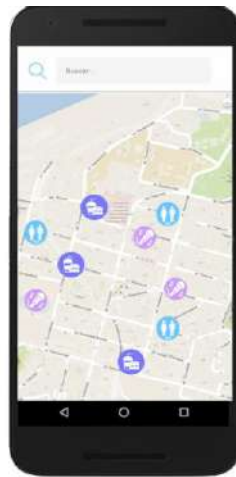


Ilustración 21. Modificación prototipo de alta fidelidad para el mapa

Se ha realizado la modificación para que los diferentes puntos de interés se plasmen en el mapa con iconos representativos.

### Método de pago evitando colas de espera



Ilustración 22. Modificación prototipo de alta fidelidad para el método de pago

Se ha modificado el color de la ventana emergente y además con este diseño ya no es necesario seleccionar el tipo de tarjeta que se tiene.

## 5. Diseño técnico

### 5.1 Definición de los casos de uso

En esta fase se realizan los diferentes casos de uso para las diferentes funcionalidades que se van a incluir en la aplicación. Un caso de uso es una descripción de las actividades que se deberán realizar para llevar a cabo el proceso.

A continuación se presenta el listado de casos de uso:

#### Artistas

<b>Identificador</b>	CU-001
<b>Nombre</b>	Consultar artistas
<b>Prioridad</b>	Alta
<b>Actores</b>	Usuario
<b>Precondiciones</b>	Ninguna
<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe seleccionar el botón "artistas" del menú principal.
<b>Postcondiciones</b>	Ninguna.

<b>Identificador</b>	CU-002
<b>Nombre</b>	Filtrar artistas
<b>Prioridad</b>	Media
<b>Actores</b>	Usuario
<b>Precondiciones</b>	El usuario debe estar inicialmente en la pantalla donde se ve la lista de artistas.
<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe indicar los filtros que quiere aplicar.
<b>Postcondiciones</b>	Ninguna.

<b>Identificador</b>	CU-003
<b>Nombre</b>	Ver información artista concreto
<b>Prioridad</b>	Alta
<b>Actores</b>	Usuario
<b>Precondiciones</b>	El usuario debe estar inicialmente en la pantalla donde se ve la lista de artistas.
<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe pulsar el artista deseado cuando se esté en la pantalla donde se ven todos los artistas.
<b>Postcondiciones</b>	Ninguna.

#### Restaurantes

<b>Identificador</b>	CU-004
<b>Nombre</b>	Consultar restaurantes
<b>Prioridad</b>	Alta
<b>Actores</b>	Usuario
<b>Precondiciones</b>	Ninguna

<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe seleccionar el botón "restaurantes" del menú principal.
<b>Postcondiciones</b>	Ninguna.

<b>Identificador</b>	CU-005
<b>Nombre</b>	Filtrar restaurantes
<b>Prioridad</b>	Media
<b>Actores</b>	Usuario
<b>Precondiciones</b>	El usuario debe estar inicialmente en la pantalla donde se ve la lista de restaurantes.
<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe indicar los filtros que quiere aplicar.
<b>Postcondiciones</b>	Ninguna.
<b>Identificador</b>	CU-006
<b>Nombre</b>	Ver información restaurante concreto
<b>Prioridad</b>	Alta
<b>Actores</b>	Usuario
<b>Precondiciones</b>	El usuario debe estar inicialmente en la pantalla donde se ve la lista de restaurantes.
<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe pulsar el artista deseado cuando se esté en la pantalla donde se ven todos los restaurantes.
<b>Postcondiciones</b>	Ninguna.

<b>Identificador</b>	CU-006
<b>Nombre</b>	Incluir opinión en restaurante
<b>Prioridad</b>	Baja
<b>Actores</b>	Usuario
<b>Precondiciones</b>	El usuario debe estar inicialmente en la pantalla donde se ve la información de un restaurante concreto.
<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe pulsar el botón para añadir una opinión y rellenar los datos correspondientes.
<b>Postcondiciones</b>	Ninguna.

## Mapa

<b>Identificador</b>	CU-007
<b>Nombre</b>	Consultar mapa
<b>Prioridad</b>	Media
<b>Actores</b>	Usuario
<b>Precondiciones</b>	Ninguna.
<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe pulsar desde el menú principal el botón "mapa".
<b>Postcondiciones</b>	Ninguna.

<b>Identificador</b>	CU-008
<b>Nombre</b>	Buscar ubicación en mapa
<b>Prioridad</b>	Baja
<b>Actores</b>	Usuario

<b>Precondiciones</b>	El usuario debe estar inicialmente en la pantalla donde se ve el mapa.
<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe introducir la ubicación que se desea mirar en el mapa.
<b>Postcondiciones</b>	Ninguna.

#### Método de pago

<b>Identificador</b>	CU-009
<b>Nombre</b>	Escanear código
<b>Prioridad</b>	Baja
<b>Actores</b>	Usuario
<b>Precondiciones</b>	El usuario debe estar inicialmente en la pantalla de "consumir".
<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe introducir escanear el código QR.
<b>Postcondiciones</b>	Ninguna.

<b>Identificador</b>	CU-010
<b>Nombre</b>	Seleccionar método de pago
<b>Prioridad</b>	Baja
<b>Actores</b>	Usuario
<b>Precondiciones</b>	El usuario debe haber escaneado el código QR. Este paso se debería haber realizado en la pantalla de "consumir".
<b>Iniciado por</b>	Usuario
<b>Flujo</b>	El usuario debe seleccionar el método de pago entre las opciones y se redireccionará a la página oficial del seleccionado.
<b>Postcondiciones</b>	Ninguna.

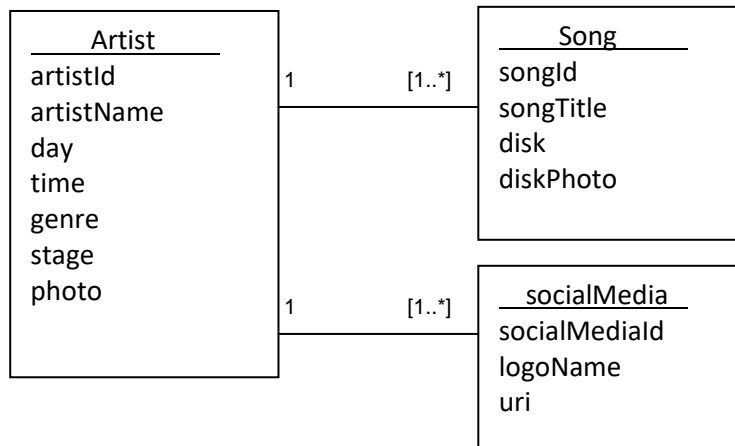
#### Notificaciones

<b>Identificador</b>	CU-011
<b>Nombre</b>	Enviar notificación
<b>Prioridad</b>	Baja
<b>Actores</b>	Usuario administrador/Sistema
<b>Precondiciones</b>	Ninguna.
<b>Iniciado por</b>	Usuario administrador/Sistema
<b>Flujo</b>	El usuario administrador indicará al sistema que se deben enviar notificaciones y éste realizará dicha acción.
<b>Postcondiciones</b>	Ninguna.

## 5.2 Diseño arquitectura

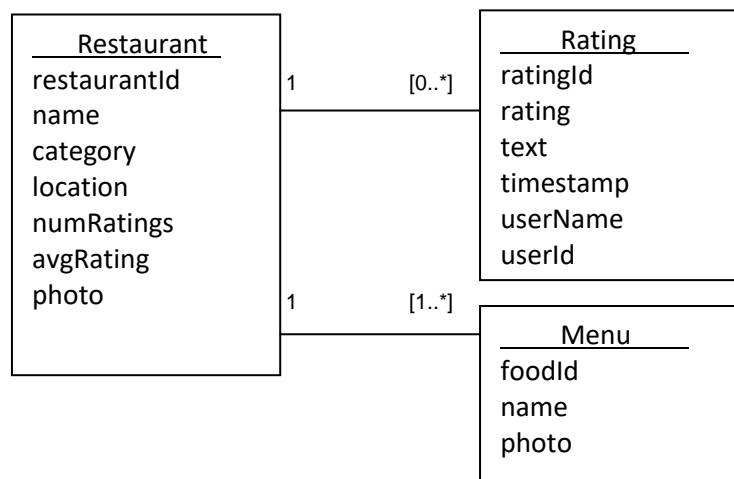
### 5.2.1 Diagrama UML correspondiente al diseño de la base de datos

#### Artistas



Los artistas tendrán como atributos todos los campos que deberán ser visualizados en la lista y cuando se selecciona un artista concreto. Cabe destacar que un artista tendrá varias canciones para realizar la actuación y varias redes sociales que serán añadidas al ver su información detallada.

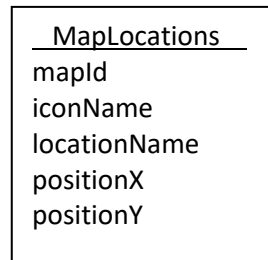
#### Restaurantes



Los restaurantes tendrán diferentes valoraciones de los usuarios. Estas serán mostradas cuando se selecciona un restaurante concreto, pero cabe destacar que se necesitará en la entidad restaurante una media y el número total de todas las valoraciones, ya que estas serán mostradas en la lista.

Se considera que se debería realizar de esta manera para aportar mayor rapidez a la hora de coger todos los elementos de la lista y presentarlos por pantalla, en vez de tener que calcular la media y el número total de todos los restaurantes cada vez que se quiera la lista. Por otro lado, un restaurante tendrá muchas comidas para poder formar su menú.

## Mapa



Un mapa tendrá diferentes ubicaciones para poder indicar donde están los puntos relevantes del festival. Además, cada ubicación tendrá un icono concreto.

## Método de pago para consumir

No será almacenado ningún dato para realizar el pago. Se considera que deberá usarse una librería externa para poder garantizar la seguridad necesaria para realizar dicho pago.

## Notificaciones

Se considera que las notificaciones serán enviadas por parte de los administradores pero no serán almacenadas.

### 5.2.2 Diagrama UML correspondiente al diseño de las entidades y clases

Se presenta un diagrama UML donde se pueden ver las diferentes clases que se tendrán en la aplicación para poder proporcionar las funcionalidades.

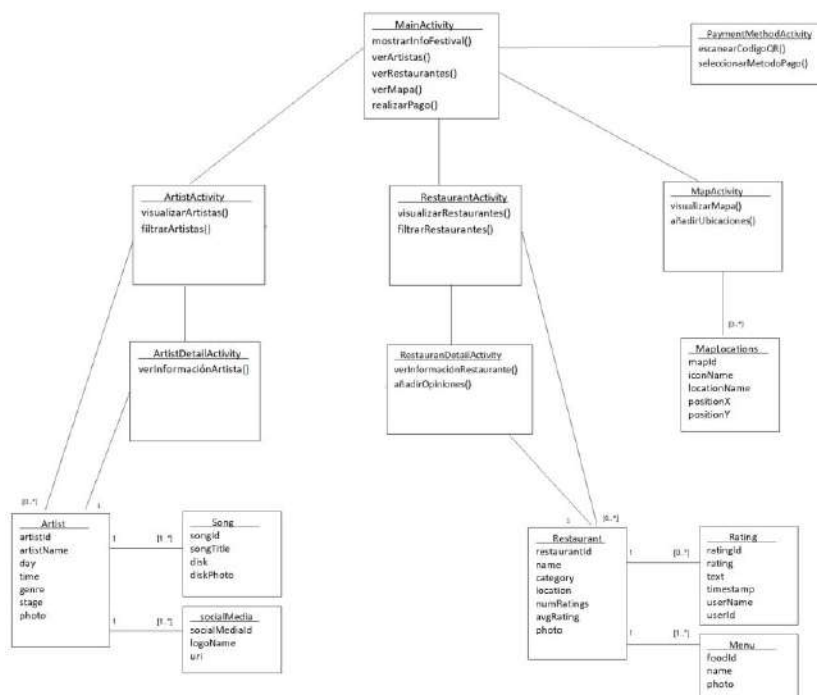


Figura 9. Diagrama UML correspondiente al diseño de las entidades y clases

### 5.2.3 Diagrama explicativo de la arquitectura del sistema

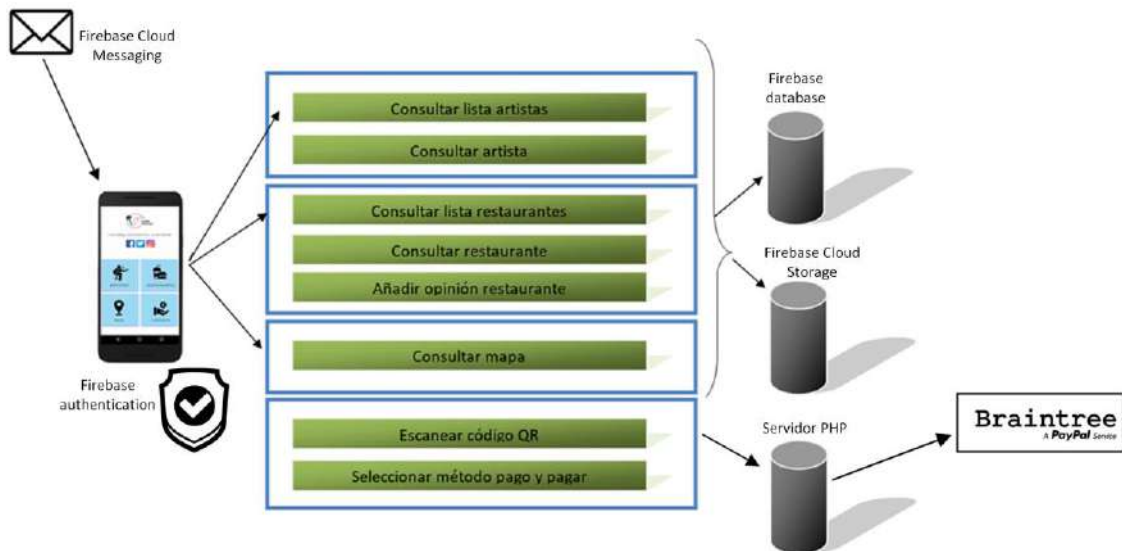


Figura 10. Diagrama explicativo de la arquitectura del sistema

La aplicación utilizará Firebase para almacenar los datos de ésta. Firebase es una plataforma creada por Google cuya principal función es desarrollar y facilitar la creación de aplicaciones de forma rápida. Proporciona diferentes herramientas como pueden ser base de datos, autenticación, almacenamiento, hosting, entre otros. Es por ello que se considera mejor opción utilizar dicha plataforma que construir una base de datos local.

En el desarrollo de la aplicación de festivales se va a hacer uso de las siguientes herramientas que proporciona Firebase:

- **Firestore authentication:** proporciona herramientas para poder autenticar a los usuarios en la aplicación. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Twitter, entre otros.
- **Firestore database- Cloud Firestore:** se trata de una base de datos flexible y escalable para la programación en servidores, dispositivos móviles y web. Mantiene los datos sincronizados entre aplicaciones cliente a través de agentes de escucha en tiempo real.
- **Firestore Cloud Storage:** se trata de un servicio de almacenamiento de objetos potente, simple y rentable construido para la escala de Google. Los SDK pueden ser usados para almacenar imágenes, audio, video y otros tipo de contenido generados por el usuario.
- **Firestore Cloud Messaging:** es una solución de mensajería multiplataforma que permite enviar mensajes de forma segura y gratuita. Es decir, se puede gestionar el envío de notificaciones a los usuarios.

Por otro lado, se utiliza una librería externa para la funcionalidad de realizar pagos en la aplicación, "brain tree payments". Ésta ofrece la capacidad de aceptar pagos en línea o dentro de la aplicación. Permite una variedad de pagos como puede ser tarjetas visa, maestro, mastercard, Paypal, Apple Pay, entre otros.



Para su utilización es necesario configurar un servidor con el cual se realizará la comunicación entre la aplicación y el servidor "brain tree". Más adelante se explica mejor su funcionamiento.

Además, también se utiliza la librería "mobile visión api" para poder escanear códigos qr, ya que el diseño planteado para el método de pago requiere escanear un código qr. Por tanto, esta librería va a permitir detectar códigos de barras en tiempo real en cualquier orientación.

## 6. Implementación

### 6.1 Introducción

En esa fase se transforma el diseño realizado en un producto, es decir, se implementan las diferentes funcionalidades para poder conseguir la aplicación Android. Se van a comentar todas las herramientas que son utilizadas para su implementación y las diferentes decisiones tomadas.

### 6.2 Desarrollo

La implementación de la aplicación se realiza con Android Studio y como se ha podido ver en el diagrama explicativo de la arquitectura, se hace uso de *Firebase* y un servidor PHP para conectar la aplicación con el servidor "brain tree".

Se comenta a continuación cómo son implementados cada uno de los requerimientos y las herramientas utilizadas para cada uno de ellos.

#### Menú principal

A la hora de acceder al menú principal de la aplicación no se requiere registro de usuarios, puesto que se requiere aportar comodidad en su acceso, pero será necesaria una autenticación para poder acceder a los diferentes datos que se tengan almacenados en *Firebase database* y *Firebase Cloud Storage*. Esto es debido a que se tienen reglas de seguridad en las cuales se especifica que los usuarios requieren de autenticación para realizar cualquier operación de tipo lectura o escritura que afecte a los datos y archivos almacenados.

```
1 rules_version = '2';
2 service firebase.storage {
3   match /b/{bucket}/o {
4     match /{allPaths=**} {
5       allow read, write: if request.auth != null;
6     }
7   }
8 }
```

Ilustración 23. Captura de pantalla de las reglas definidas en *Firebase database* y *Firebase Cloud Storage*.

Por ello, cuando se accede a la aplicación se realiza automáticamente una autenticación anónima con *Firebase authentication*, de tal forma que se permite a los usuarios no registrados trabajar con datos protegidos.

A continuación se muestra una captura de pantalla del funcionamiento de menú principal.



Ilustración 24. Captura de pantalla de menú principal en aplicación.

## Artistas

Para poder realizar la consulta de los diferentes artistas se va a hacer uso de *Firebase database* para almacenar los metadatos de los artistas y *Firebase Cloud Storage* para poder almacenar las correspondientes imágenes.

En *Firebase database* será necesario configurar determinados índices para permitir la búsqueda de artistas haciendo uso de determinados filtros. En el caso de búsqueda de artistas se han añadido todas las combinaciones posibles con sus posibles filtros (nombre del artista, día, género musical, escenario y orden).

ID de la colección	Campos indexados	Alcance de la consulta	Estado
artistas	day Ascendente, genre Ascendente, artistName Ascendente	Colección	Habilitado
artistas	day Ascendente, stage Ascendente	Colección	Habilitado
artistas	artistName Ascendente, day Ascendente	Colección	Habilitado
artistas	genre Ascendente, day Ascendente	Colección	Habilitado
artistas	day Ascendente, genre Ascendente	Colección	Habilitado
artistas	day Ascendente, genre Ascendente, stage Ascendente, artistName	Colección	Habilitado
artistas	genre Ascendente, stage Ascendente	Colección	Habilitado
artistas	day Ascendente, artistName Ascendente	Colección	Habilitado
artistas	artistName Ascendente, genre Ascendente, stage Ascendente	Colección	Habilitado
artistas	day Ascendente, genre Ascendente, stage Ascendente	Colección	Habilitado

Ilustración 25. Algunos de los índices definidos en *Firebase database* para artistas.

Por otro lado, en *Firebase Cloud Storage* se va a tener una estructura a la hora de guardar imágenes de un determinado artista, de tal forma que la obtención de éstas sea de forma sencilla y en caso de añadir nuevos artistas no se requieran cambios en el código.

- Carpeta "Artists": se guardan todas las fotos de perfil de los artistas, siendo el nombre de la foto el del artista concreto.

- Carpeta "Albums": se tienen las imágenes de los diferentes álbumes de las canciones que se cantarán en la actuación. El nombre de éstas imágenes serán las del nombre del álbum concreto.



Ilustración 26. Carpetas que se tienen en Firebase Cloud Storage para artistas.

Cabe destacar que el nombre de las diferentes imágenes para cada artista se obtiene a partir de *Firestore database*, ya que cada artista y canción tienen asociado una imagen. Por tanto, cada vez que se quiera mostrar un artista se buscará en *Firestore Cloud Storage* las imágenes que sean necesarias. En el caso de no encontrar una imagen se mostrará una imagen por defecto.

Se muestra a continuación algunas capturas de pantalla del funcionamiento de la aplicación cuando se hace consulta de artistas.

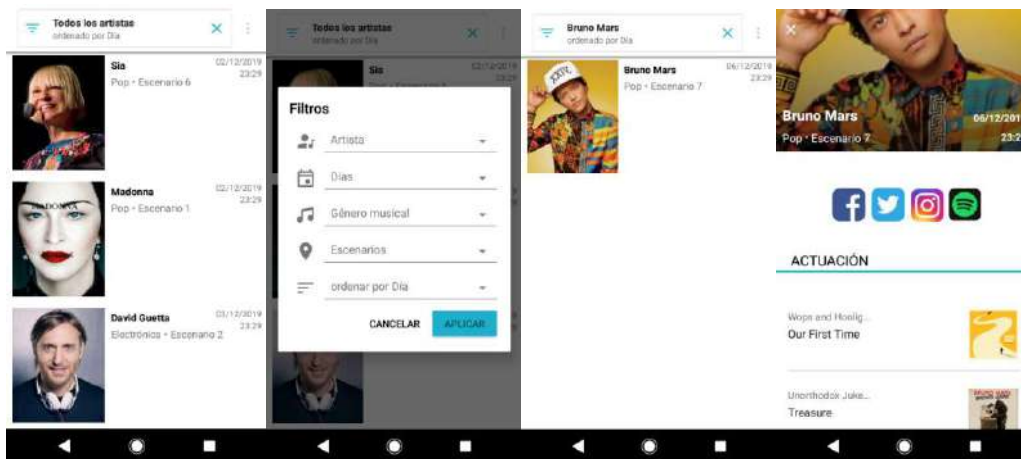


Ilustración 27. Captura de pantalla de consulta de artistas y horario en aplicación.

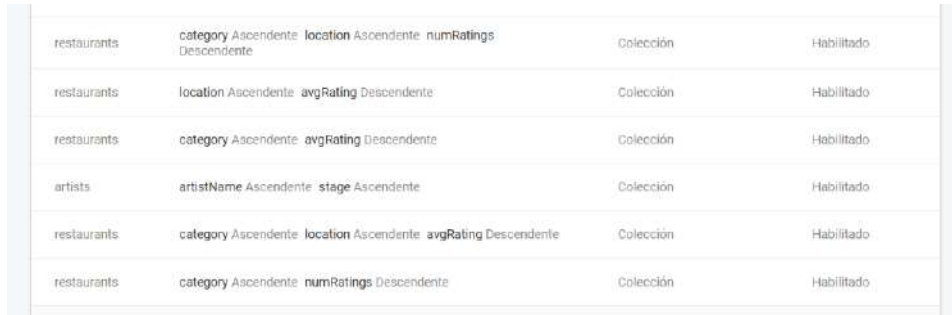
Se puede ver en la captura como se realiza una consulta de todos los artistas y a continuación se aplica un filtro para buscar un artista concreto. En caso de querer eliminar el valor del filtro se puede pulsar el botón "x" de la esquina superior derecha.

Cabe mencionar que se tiene un botón menú (icono con tres puntos) el cual es utilizado para añadir nuevos artistas a *Firestore*, es decir, se trata de un botón utilizado para generar datos de prueba. Cuando la aplicación se lleve a producción dicho botón será eliminado.

## Restaurantes

En la consulta de los diferentes restaurantes del festival también se hará uso de *Firebase database* y *Firebase Cloud Storage*.

En *Firebase database* será necesario configurar determinados índices necesarios para permitir la búsqueda de restaurantes con filtros (tipo de comida, lugar, orden).



restaurants	category Ascendente Descendente	location Ascendente	numRatings	Colección	Habilitado
restaurants	location Ascendente	avgRating Descendente		Colección	Habilitado
restaurants	category Ascendente	avgRating Descendente		Colección	Habilitado
artists	artistName Ascendente	stage Ascendente		Colección	Habilitado
restaurants	category Ascendente	location Ascendente	avgRating Descendente	Colección	Habilitado
restaurants	category Ascendente	numRatings Descendente		Colección	Habilitado

Ilustración 28. Índices definidos en *Firebase database* para restaurantes.

Por otro lado, en *Firebase Cloud Storage* se deben almacenar imágenes para el perfil principal del restaurante y de los diferentes menús que ofrece. Para ello se tienen las siguientes carpetas:

- Carpeta "Restaurants": carpeta principal para almacenar las diferentes carpetas de los diferentes restaurantes que se tengan.
- Lista de carpetas dentro de "Restaurants": se tendrán todas las imágenes correspondientes al menú que ofrece cada restaurante y su imagen de perfil.

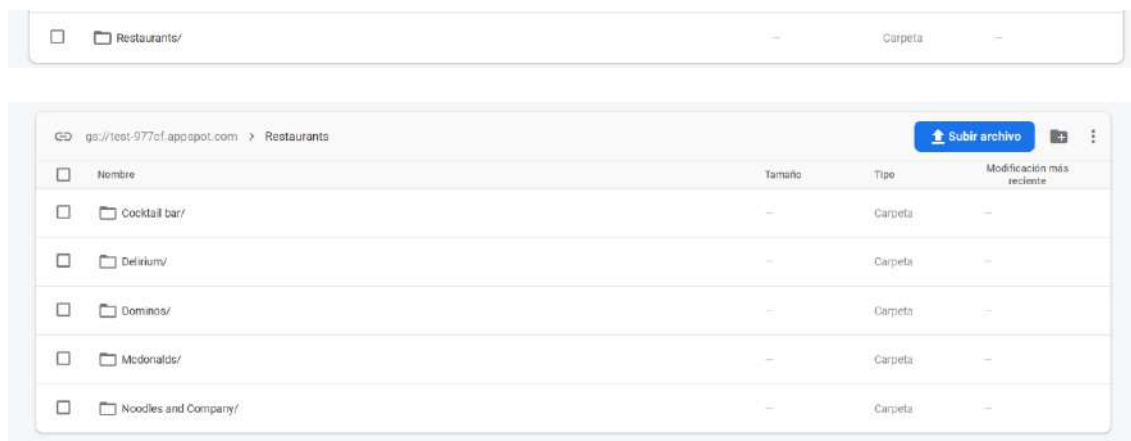


Ilustración 29 . Carpetas que se tienen en *Firebase Cloud Storage* para restaurantes.

Cabe destacar que al igual que en la funcionalidad artistas, el nombre de las diferentes imágenes se obtiene a partir de *Firebase database*. En este caso cada restaurante y cada comida que se ofrece tienen una imagen asociada.

Por otro lado, además de la consulta de los diferentes restaurantes se debe permitir incluir opiniones sobre éstos. Para ello, se ha añadido de forma opcional la opción de añadir el nombre del usuario que está realizando la valoración, ya que cuando se accede a la aplicación

se realiza una autenticación anónima. En caso de que el usuario no incluya el nombre, se mostrará por defecto "Random User".

A continuación se muestran algunas capturas de pantalla del funcionamiento de la aplicación cuando se hace consulta de restaurantes.

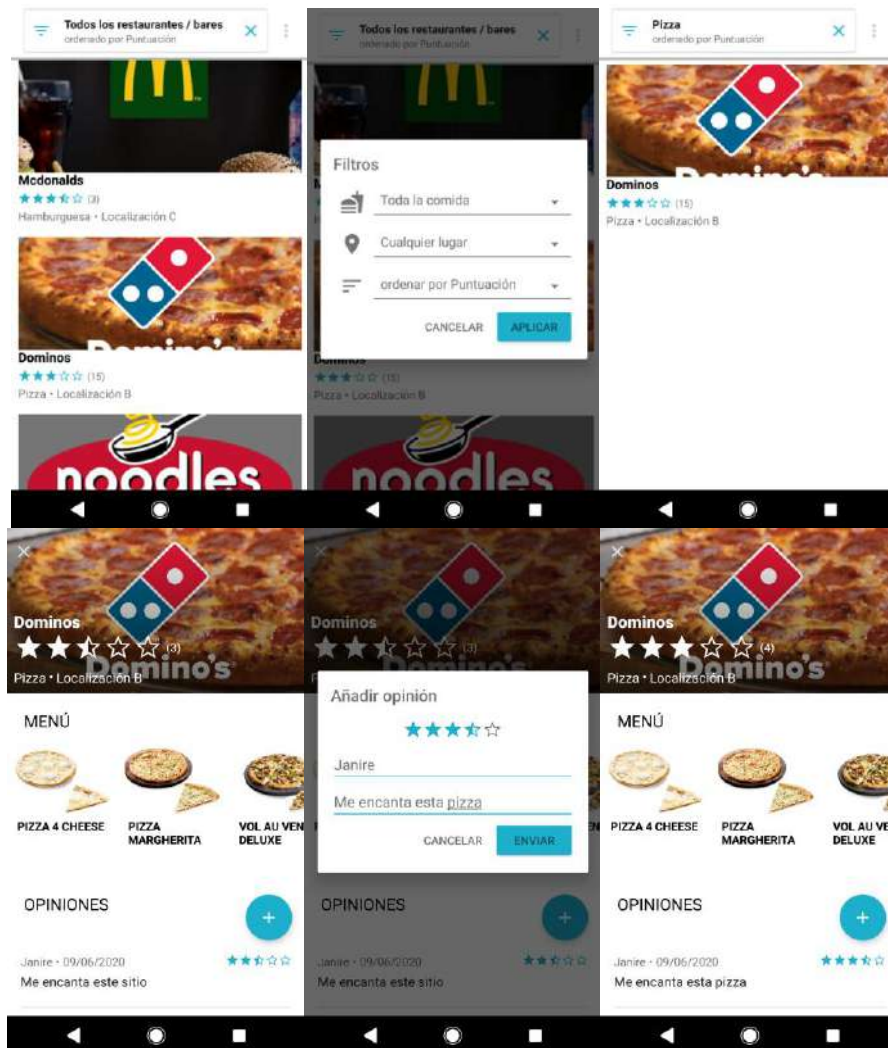


Ilustración 30. Captura de pantalla de consulta de restaurantes en aplicación.

Se puede ver en la captura como se realiza una consulta de todos los restaurantes y a continuación se aplica un filtro para buscar un tipo de comida concreto. En caso de querer eliminar el valor del filtro se puede pulsar el botón "x" de la esquina superior derecha.

Una vez se ha seleccionado el restaurante deseado se puede ver el menú que ofrece y las diferentes opiniones que se tienen. A continuación se añade una valoración a dicho restaurante.

Cabe mencionar que al igual que en la funcionalidad de consulta de artistas, se tiene un botón menú (icono con tres puntos), el cual es utilizado para añadir datos de prueba a *Firestore*. Cuando la aplicación se lleve a producción dicho botón será eliminado.

## Mapa

Para la consulta del mapa y los diferentes puntos de interés se ha considerado que la imagen del mapa del festival concreto deberá ser almacenado en *Firestore Cloud Storage* y los diferentes puntos de interés de dicho mapa serán almacenados en *Firestore database*.

En *Firestore Cloud Storage* se tendrá la siguiente carpeta que contendrá la imagen del mapa.



Ilustración 31. Carpetas que se tienen en *Firestore Cloud Storage* para mapa.

Una vez se obtiene la imagen del mapa en la aplicación se incluyen los diferentes puntos de interés, los cuales se obtendrán a partir de los datos almacenados en *Firestore database*. Dependiendo del tipo de localización que sea, se utilizará un icono distinto para representar el punto de interés.

Cabe destacar que para poder ampliar e ir moviéndose sobre el mapa, se ha utilizado la librería "subsampling-scale-image-view" creada por David Morrissey.

Dicha librería permite tener una vista de imagen personalizada para Android, diseñada para galerías de fotos y que muestra imágenes enormes (por ejemplo, mapas y planos de construcción) sin tener errores de memoria. Incluye soporte para hacer zoom, paneo, rotación y animación, y permite una extensión fácil para que se pueda agregar propias superposiciones.

En este caso se ha creado una clase View que extiende de la librería, la cual ha permitido poder incluir puntos de interés a la imagen del mapa. Además, cuando se hace zoom a la imagen, la librería permite que el icono del punto de interés se escale en función de la ampliación.

A continuación se muestran algunas capturas de pantalla del funcionamiento de la consulta del mapa.

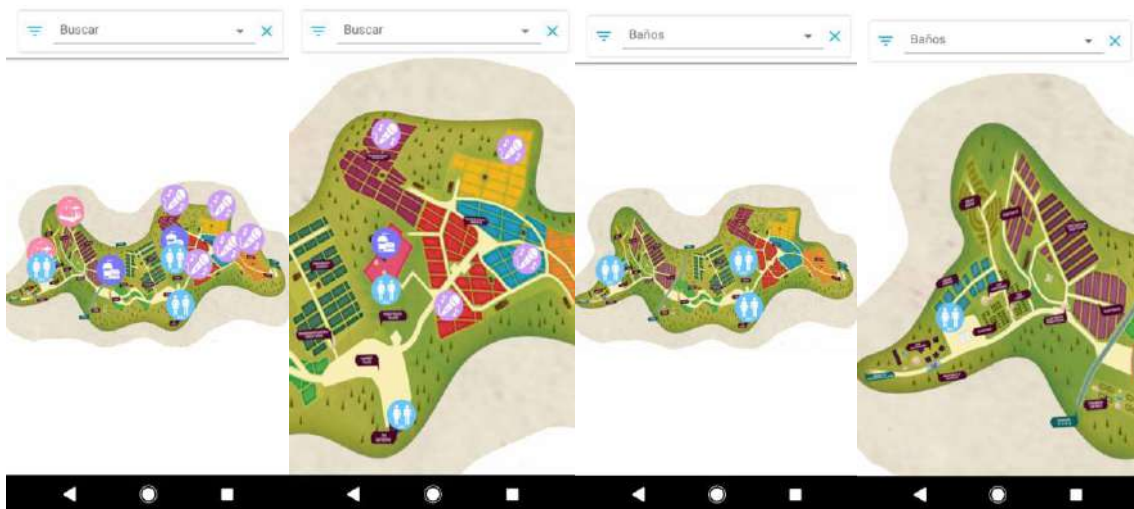


Ilustración 32. Captura de pantalla de consulta de mapa en aplicación

## Método de pago para consumir

Para poder realizar el pago en la aplicación se debe escanear un código qr y a continuación realizar el pago a través del método seleccionado. Para poder implementar dicha funcionalidad se hace uso de diferentes librerías: "mobile vision api" y "brain tree payments".

"Mobile vision api" permite diferentes funcionalidades:

- Detectar caras: encuentra rostros humanos en fotos, videos o transmisiones en vivo. También rastrea posiciones de puntos de referencia faciales como los ojos, nariz y la boca.
- Reconocer texto: Reconoce texto en cualquier idioma latino. También presenta la estructura del texto reconocido, incluido los párrafos y las líneas.
- Escanear código de barras: Detecta códigos de barras en tiempo real en cualquier orientación. También puede detectar y analizar varios códigos de barras en diferentes formatos al mismo tiempo.

La funcionalidad de interés para esta aplicación es la de escanear códigos de barras, ya que permite escanear el código qr para poder obtener el precio del producto a pagar. Además, una de las ventajas que se ha visto es que no hace falta colocar el código en una posición concreta, sino que da igual la orientación y su posición en la cámara para realizar el escaneo, lo que aportará comodidad al usuario.

Por otro lado, se utiliza la librería "brain tree payments" para poder proporcionar diferentes métodos de pagos al usuario. Esta librería ha sido creada por una división de PayPal que se especializa en sistemas de pagos móviles y web para empresas de comercio electrónico.

Para poder añadirla a la aplicación se tienen que implementar dos partes: el lado del cliente (aplicación Android) y el servidor PHP. Es necesaria la realización de un servidor para poder comunicar el cliente con el servidor "brain tree".

En el lado del cliente se añade el código necesario para que pueda comunicarse con el servidor PHP. Y por otro lado, el servidor PHP se tiene que comunicar con "brain tree". En este caso, el servidor PHP se ejecutará con XAMPP y para que pueda ser accedido desde el exterior se utilizará la herramienta ngrok.

El funcionamiento de "brain tree" con la parte del cliente y el servidor PHP es la siguiente:

- La aplicación Android solicita el token a nuestro servidor.
- Nuestro servidor genera un token usando el SDK de "brain tree" y lo envía de vuelta a la aplicación Android.
- La aplicación Android obtiene la información de la tarjeta de crédito del usuario y contacta con nuestro servidor. A continuación, nuestro servidor contacta con el servidor "brain tree" para realizar el pago.
- Nuestro servidor recibe notificación del pago por parte del servidor "brain tree" y se lo comunica a la aplicación Android.

A continuación se muestran algunas capturas de pantalla del funcionamiento de método de pago para consumir.



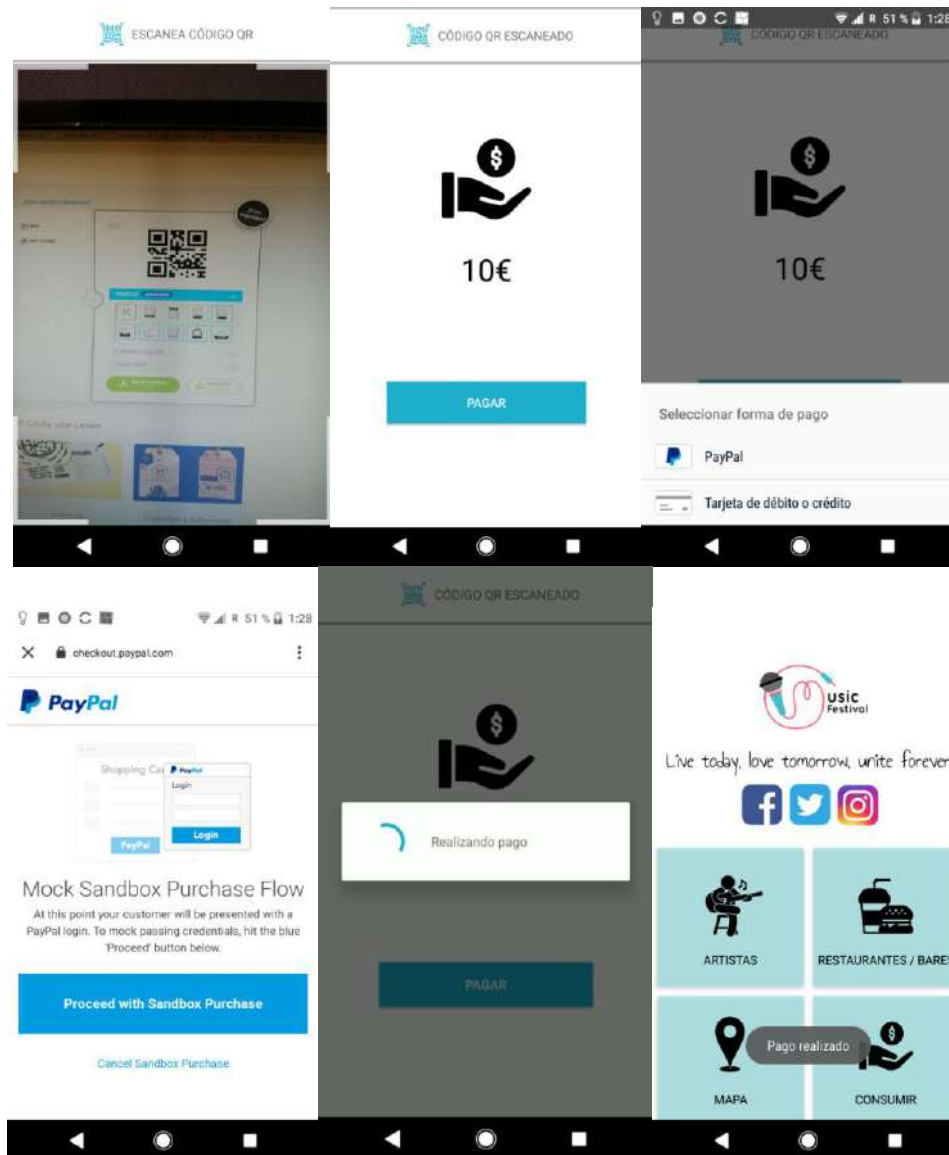


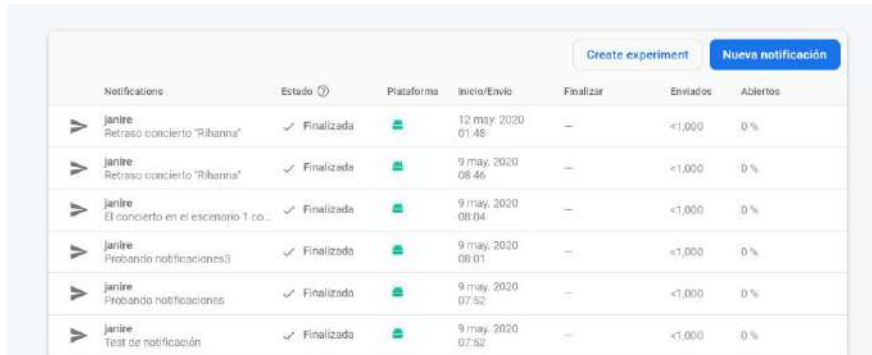
Ilustración 33. Captura de pantalla de realización de pago en aplicación

Se puede ver en la captura que se realiza un escaneo de un código qr y se obtiene el precio que se debe pagar. Una vez esto, cuando se pulsa el botón de pagar aparecen los diferentes métodos de pago implementados en "brain tree" y se procede a realizar el pago con el método seleccionado.

## Notificaciones

Para implementar las notificaciones se hace uso de *Firebase Cloud Messaging* la cual permite enviar mensajes de forma segura y gratuita .

Para poder enviar notificaciones a nuestra aplicación se tiene que acceder a *Firebase Cloud Messaging* y escribir el mensaje que se quiere enviar. Cabe destacar que se tendrá un listado de todas las notificaciones enviadas.



Notificaciones	Estado	Plataforma	Inicio/Envío	Finalizar	Enviados	Abiertos
janire Retraso concierto "Rihanna"	Finalizada		12 may. 2020 01:46	—	<1,000	0%
janire Retraso concierto "Rihanna"	Finalizada		9 may. 2020 08:46	—	<1,000	0%
janire El concierto en el escenario 1 co...	Finalizada		9 may. 2020 08:04	—	<1,000	0%
janire Probando notificaciones3	Finalizado		9 may. 2020 08:01	—	<1,000	0%
janire Probando notificaciones	Finalizada		9 may. 2020 07:52	—	<1,000	0%
janire Test de notificación	Finalizada		9 may. 2020 07:52	—	<1,000	0%

Ilustración 34. Listado notificaciones enviadas con *Firebase Cloud Messaging*

A continuación se muestra una captura de pantalla del funcionamiento de notificaciones.

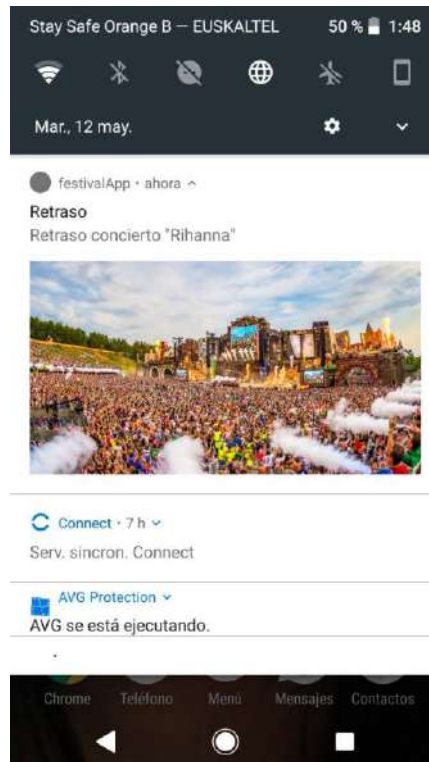


Ilustración 35. Captura de pantalla de recibimiento de notificación en aplicación.

### 6.3 Análisis del estado del proyecto

Según la planificación realizada al inicio del proyecto se puede observar que se ha llegado a los objetivos previstos. Se han implementado los requisitos definidos y se pueden ver los resultados de su ejecución.

Aún así cabe destacar que en la ejecución de las pruebas se han visto mejoras que podrían realizarse. Estas mejoras son tanto relacionadas con el diseño como con algunas implementaciones de las funcionalidades.

A continuación se realiza una propuesta de las desviaciones que se van a tener hasta realizar la fecha de la entrega final (tarea "mejoras").

Tabla 5. Desviación de plan de trabajo para añadir mejoras a la aplicación.

Nombre de tarea	Duración	Comienzo	Fin
Entrega PEC3	0 horas	mié 13/05/20	mié 13/05/20
<b>▲ Mejoras</b>	<b>88 horas</b>	<b>jue 14/05/20</b>	<b>mar 09/06/20</b>
Mejoras de implementación requisitos	88 horas	jue 14/05/20	mar 09/06/20
Mejoras del diseño (UX)	88 horas	jue 14/05/20	mar 09/06/20
<b>Entrega final</b>	<b>0 horas</b>	<b>mié 10/06/20</b>	<b>mié 10/06/20</b>
<b>Defensa Virtual</b>	<b>0 horas</b>	<b>vie 26/06/20</b>	<b>vie 26/06/20</b>

A la hora de realizar la nueva planificación para las mejoras del proyecto se tiene que tener en cuenta que la duración de las tareas se ha definido en horas, y el tiempo de dedicación será el mismo propuesto al inicio del proyecto.

- Días laborables: 3h al día.
- Días no laborables o festivos: 4h al día.

### 6.4 Pruebas

Para probar la aplicación se van a realizar pruebas unitarias, de tal forma que se puede comprobar si un fragmento de código funciona correctamente.

Se realizarán pruebas unitarias en el propio código. Para ello se utilizarán las librerías de JUnit y Mockito.

- Junit es conjunto de librerías que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java. Permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado. Si la clase cumple con la especificación, entonces la prueba ha sido realizada exitosamente, en caso contrario se devolverá un fallo.
- Mockito se trata de una librería de prueba que facilita la creación de "Test Doubles". Es decir, permite crear objetos mockeados sin tener que crear nuevas clases que aumenten la complejidad del código. Se usa para simular interfaces de tal forma que

se pueda agregar una funcionalidad ficticia a una interfaz simulada para realizar pruebas unitarias.

Por otro lado debido a que se está utilizando Firebase, se puede utilizar su herramienta *Firestore Test Lab*, la cual es una infraestructura de prueba de aplicaciones basada en la nube. Con una sola operación se puede probar la aplicación Android en una amplia variedad de dispositivos y configuraciones, y ver los resultados (que incluyen registros, videos y capturas de pantalla) en Firebase console.

*Firestore Test Lab* usa dispositivos reales de producción en un centro de datos de Google para probar la aplicación. Ejecuta pruebas de Espresso y UI Automator 2.0 en aplicaciones para Android.

A continuación se muestra una captura de pantalla de los resultados obtenidos al hacer pruebas en *Firestore Test Lab*.

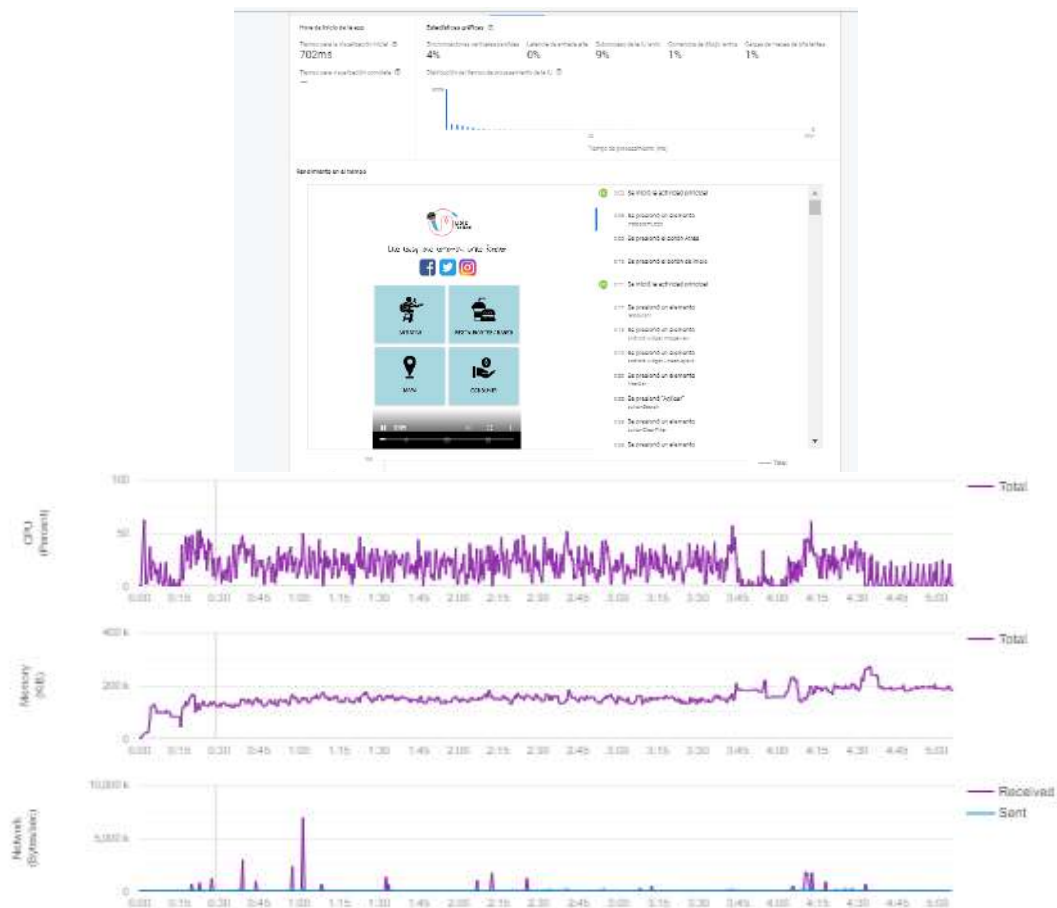


Ilustración 36. Captura de pantalla de pruebas realizadas con *Firestore Test Lab*

## 7. Conclusiones

### 7.1 Introducción

En este apartado se va a realizar un análisis de todo el trabajo realizado para la realización de la aplicación Android para festivales. Se va a comentar si se han conseguido los objetivos, si se han necesitado cambios en la planificación, líneas de trabajo futuras, entre otros.

### 7.2 Lecciones aprendidas

Se ha aprendido que antes de comenzar a codificar una aplicación o producto se debe realizar inicialmente un estudio de lo que existe en el mercado. Es decir, se debe realizar un análisis competitivo para poder analizar productos similares y poder ver qué aspectos se podrían mejorar para poder implementarlas en nuestro producto. De esta manera se va a recopilar la información necesaria para evitar errores y tener una mayor satisfacción de los usuarios.

Por otro lado, también se tiene que tener presente a los usuarios a la hora de diseñar la aplicación. Para ello se puede seguir un proceso DCU en el cual las decisiones de diseño están dirigidas para el usuario. Esto se puede conseguir mediante la realización de encuestas o indagación con los usuarios para obtener información sobre la facilidad de uso de una aplicación o producto y poder ver cuáles son los aspectos más difíciles para que en nuestro diseño se pueda implementar de forma más sencilla.

Además, hay que tener en cuenta que en un proceso DCU la usabilidad del diseño es evaluada de forma iterativa y mejorada incrementalmente, de tal forma que se pueda conseguir un buen producto que satisfaga las necesidades del usuario.

### 7.3 Logro de los objetivos

Se han conseguido implementar todos los requisitos definidos en la planificación inicial del proyecto, tanto los objetivos funcionales como los no funcionales.

Por tanto, se ha conseguido la implementación y el correcto funcionamiento de los objetivos en la aplicación y se han podido ver los diferentes resultados obtenidos.

### 7.4 Seguimiento de la planificación y metodologías

Se ha seguido satisfactoriamente la planificación definida al comienzo del proyecto. Aunque, cabe destacar que algunas de las tareas a realizar han necesitado de más horas de dedicación que las mencionadas en la planificación. Pero en todo momento se ha tenido en cuenta la fecha de finalización de cada tarea, lo que ha supuesto que se hayan completado las implementaciones a tiempo.

También se debe nombrar que debido a los resultados obtenidos en la ejecución de las pruebas, se vieron mejoras que podrían realizarse. Es por ello que se realizó una nueva propuesta de planificación, a partir de la fecha que se hicieron las pruebas hasta la fecha de la entrega final del proyecto, en la cual se debían implementar dichas mejoras.

Por otro lado, se debe señalar que a la hora de la realización de las diferentes tareas se han utilizado herramientas de metodología ágil. En concreto se ha utilizado *Trello* el cual es un

software de administración de proyectos con interfaz web que sirve para gestionar tareas permitiendo organizar el trabajo.

### **7.5 Líneas de trabajo futuro**

Debido a que se ha seguido un proceso DCU se podrían realizar nuevas iteraciones para evaluar la usabilidad de diseño y poder ver si aún se tienen mejoras por realizar. Esto se podría realizar con evaluaciones heurísticas o mediante la realización de test con usuarios, entre otros.

Además, gracias a dichas iteraciones se podría recopilar información sobre nuevas funcionalidades que podrían añadirse en la aplicación para que sea más completa. Éstas podrían ser implementadas de forma incremental mediante actualizaciones.

## 8. Glosario

### **DCU - Diseño Centrado en el usuario.**

Metodología de diseño que toma como centro las personas que utilizarán el producto/servicio.

### **IPO - Interacción Persona Ordenador**

Definida por la Association for Computing Machinery (ACM) como "la disciplina relacionada con el diseño, la evaluación y la implementación de sistemas informáticos interactivos para el uso de seres humanos y con el estudio de los fenómenos más importantes con los que está relacionado".

### **UX - Experiencia de usuario**

Conjunto de sensaciones, percepciones, razones y satisfacción de un usuario que interactúa con un producto o sistema.

### **Benchmarking**

Consiste en evaluar los productos que son competidores del que estamos diseñando, desde el punto de vista del usuario final. El objetivo es conocer sus puntos fuertes y características, y descubrir las tendencias de diseño que existen en el contexto al que nos dirigimos.

### **Prototipo**

Es un modelo del producto o servicio que incluye los aspectos funcionales, estructurales y gráficos del diseño. Se realiza en la fase de generación, después de haber realizado la investigación y de haber definido los objetivos y principales aspectos del diseño.

### **Test de usuario**

Es un método de evaluación de la usabilidad que permite comprobar si las interfaces y los procesos de interacción están bien diseñados y si los usuarios encuentran obstáculos y dificultades para lograr sus objetivos.

## 9. Bibliografía

- [1] Grand Stand Apps (2020) Festival International App. Consultado el 15 de Marzo de 2020.  
<https://play.google.com/store/apps/details?id=com.seedlabs.festintl&hl=en>
- [2] Sziget Festival (2020) Sziget Festival App. Consultado el 15 de Marzo de 2020.  
<https://play.google.com/store/apps/details?id=com.greencopper.android.sziget&hl=en>
- [3] Festival Cannes (2020) Festival de Cannes – Official App. Consultado el 17 de Marzo de 2020.  
<https://play.google.com/store/apps/details?id=fr.festivaldecannes&hl=en>
- [4] FestivAll (2020) Festival FestivAll, Music Festival Guide App. Consultado el 17 de Marzo de 2020.  
[https://play.google.com/store/apps/details?id=com.festivall\\_new&hl=en](https://play.google.com/store/apps/details?id=com.festivall_new&hl=en)
- [5] Festival One (2020) Festival One App. Consultado el 18 de Marzo de 2020.  
<https://play.google.com/store/apps/details?id=cz.ackee.a4f.festivalone&hl=en>
- [6] UOC (2020) Benchmarking. Consultado el 15 de Marzo de 2020.  
<http://design-toolkit.recursos.uoc.edu/es/benchmarking/>
- [7] UOC (2020) Escenarios de uso. Consultado el 24 de Marzo de 2020.  
<http://design-toolkit.recursos.uoc.edu/es/escenarios/>
- [8] UOC (2020) Diagrama de flujo. Consultado el 25 de Marzo de 2020.  
<http://design-toolkit.recursos.uoc.edu/es/guia/diagrama-de-flujo/>
- [9] UOC (2020) Prototipado. Consultado el 25 de Marzo de 2020.  
<http://design-toolkit.recursos.uoc.edu/es/guia/prototipado/>
- [10] UOC (2020) Diseño de interfaces. Consultado el 25 de Marzo de 2020.  
<http://multimedia.uoc.edu/blogs/dii/es/prototipatge/>
- [11] UOC (2020) Evaluación de la usabilidad. Consultado el 28 de Marzo de 2020.  
[http://openaccess.uoc.edu/webapps/o2/bitstream/10609/9861/1/PID\\_00159105.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/9861/1/PID_00159105.pdf)
- [12] UOC (2020) Test con usuarios. Consultado el 28 de Marzo de 2020.  
<http://design-toolkit.recursos.uoc.edu/es/test-con-usuarios/>
- [13] UOC (2020) Introducción al lenguaje de Modelado Unificado. Consultado el 29 de Marzo de 2020.  
[http://openaccess.uoc.edu/webapps/o2/bitstream/10609/9121/1/Intro\\_UML.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/9121/1/Intro_UML.pdf)
- [14] UOC (2020) Introducción a la interacción persona ordenador. Consultado el 3 de Abril de 2020.  
[http://openaccess.uoc.edu/webapps/o2/bitstream/10609/76105/6/Introducci%C3%B3n%20a%20la%20interacci%C3%B3n%20persona%20ordenador\\_M%C3%B3dulo%203\\_Dise%C3%B1o%20centrado%20en%20el%20usuario.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/76105/6/Introducci%C3%B3n%20a%20la%20interacci%C3%B3n%20persona%20ordenador_M%C3%B3dulo%203_Dise%C3%B1o%20centrado%20en%20el%20usuario.pdf)
- [15] Google (2020) Firebase Authentication. Consultado el 5 de Abril de 2020.  
<https://firebase.google.com/docs/auth>



- [16] Google (2020) Cloud Firestore. Consultado el 9 de Abril de 2020.  
<https://firebase.google.com/docs/firestore>
- [17] Google (2020) Primeros pasos con Cloud Firestore. Consultado el 9 de Abril de 2020.  
<https://firebase.google.com/docs/firestore/quickstart>
- [18] Google (2020) Cloud Storage. Consultado el 15 de Abril de 2020.  
<https://firebase.google.com/docs/storage>
- [19] Lars Vogel, Fabian Pfaff (2020) Unit tests with Mockito - Tutorial. Consultado el 18 de Abril de 2020.  
<https://www.vogella.com/tutorials/Mockito/article.html>
- [20] Junit (2020) JUnit. Consultado el 18 de Abril de 2020.  
<https://junit.org/junit4/>
- [21] Lars Vogel (2007) Unit Unit Testing with JUnit - Tutorial. Consultado el 18 de Abril de 2020.  
<https://www.vogella.com/tutorials/JUnit/article.html>
- [22] Cecilio Álvarez Caules (2015) Java Mockito y los Mock Object. Consultado el 18 de Abril de 2020.  
<https://www.arquitecturajava.com/java-mockito-y-los-mock-objects/>
- [23] Google (2020) Firebase Cloud Messaging. Consultado el 2 de Mayo de 2020.  
<https://firebase.google.com/docs/cloud-messaging>
- [24] Braintree (2020) Brain Tree Payment. Consultado el 13 de Mayo de 2020.  
<https://www.braintreepayments.com/be>
- [25] Braintree (2020) Set Up Your Client. Consultado el 14 de Mayo de 2020.  
<https://developers.braintreepayments.com/start/hello-client/android/v2>
- [26] Ngrok (2020) Ngrok. Consultado el 14 de Mayo de 2020.  
<https://ngrok.com/>
- [27] Apache friends (2020) XAMPP. Consultado el 14 de Mayo de 2020.  
<https://www.apachefriends.org/es/index.html>
- [28] The Code City (2017) Complete Braintree Integration in Android Tutorial. Consultado el 14 de Mayo de 2020.  
<https://www.thecodecity.com/2017/04/braintree-android-integration-tutorial.html>
- [29] Dave Morrissey (2020) Subsampling Scale Image View Library. Consultado el 25 de Mayo de 2020.  
<https://github.com/davemorrissey/subsampling-scale-image-view>
- [30] Google (2020) Firebase Test Lab. Consultado el 3 de Junio de 2020.  
[https://firebase.google.com/docs/test-lab/?hl=es-419&gclid=Cj0KCQjw\\_ez2BRCyARIsAJfg-kt3ZUiSodlR0K5m49kJenDHv5drl8bNYGscZVemsHUuY7u92Qnl8l0aAiqLEALw\\_wcB](https://firebase.google.com/docs/test-lab/?hl=es-419&gclid=Cj0KCQjw_ez2BRCyARIsAJfg-kt3ZUiSodlR0K5m49kJenDHv5drl8bNYGscZVemsHUuY7u92Qnl8l0aAiqLEALw_wcB)
- [31] Google (2020) Primeros pasos con Firebase Test Lab desde Firebase console. Consultado el 3 de Junio de 2020.  
<https://firebase.google.com/docs/test-lab/ios/firebase-console?hl=es-419>

[32] Trello (2020) Trello. Consultado el 7 de Marzo.  
<https://trello.com/es>

[33] UOC (2020) Diseño Centrado en el Usuario (DCU). Consultado el 23 de Febrero.  
<http://multimedia.uoc.edu/blogs/dii/es/disseny-centrat-en-lusuari-dcu/>

[34] Android Studio (2020) Cómo compilar y ejecutar una app. Consultado el 12 de Mayo.  
<https://developer.android.com/studio/run?hl=es-419>

## 10. Anexos

### ANEXO I. Manual de Instrucciones de compilación

Este manual de instrucciones tiene como finalidad ver cómo se ejecuta la aplicación a partir del código proporcionado.

#### Cómo ejecutar una aplicación en Android Studio

Para compilar y ejecutar la app, sigue estos pasos:

1. En la barra de herramientas, selecciona la app en el menú desplegable de configuraciones de ejecución.
2. En el menú desplegable del dispositivo de destino, selecciona el dispositivo en el que deseas ejecutar la aplicación.

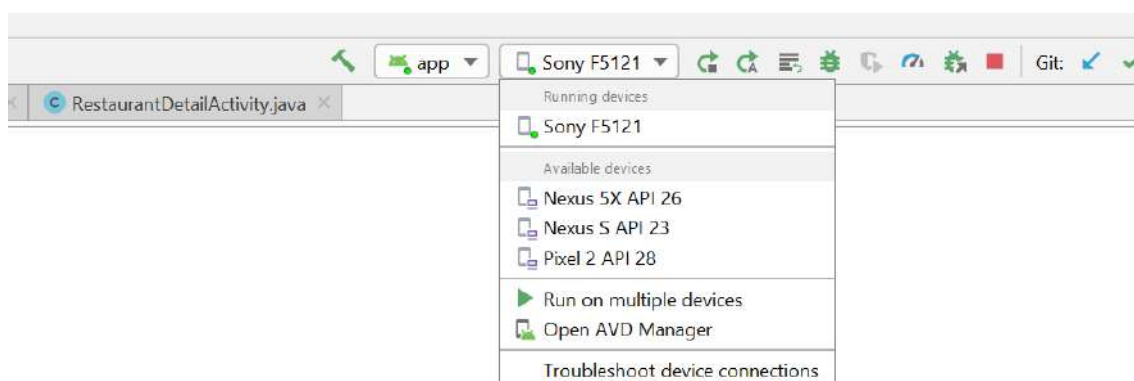





Ilustración 37. Captura de pantalla de cómo ejecutar una aplicación en Android Studio.

\*Nota: Si no tienes ningún dispositivo configurado, debes conectar un dispositivo mediante USB o crear un AVD para usar Android Emulator.

3. Haz clic en **Run** 

\*Nota: También puedes implementar la app en modo de depuración haciendo clic en **Debug** . La ejecución de la app en el modo de depuración te permite configurar puntos de interrupción en el código, examinar variables y evaluar expresiones en el tiempo de ejecución, así como ejecutar herramientas de depuración. Para obtener más información

#### Cómo compilar una aplicación en Android Studio

El botón Run  compila e implementa la app en un dispositivo. Sin embargo, a fin de compilar la app para compartirla o subirla a Google Play, tendrás que usar una de las opciones del menú Build para compilar partes del proyecto o su totalidad.

## Iniciar Servidor PHP para librería "Brain Tree Payments"

Se tiene un servidor para poder utilizar la librería "brain tree payments" en la funcionalidad de pagar a través de la aplicación. Para ejecutar el servidor se usa XAMPP.

1. Se descarga la librería de brain tree para php a través del siguiente enlace <https://developers.braintreepayments.com/start/hello-server/php>, y se debe incluir la librería en la ruta C:\xampp\htdocs\braintree.
2. Se incluye en la siguiente ruta la carpeta del código proporcionado del servidor PHP:

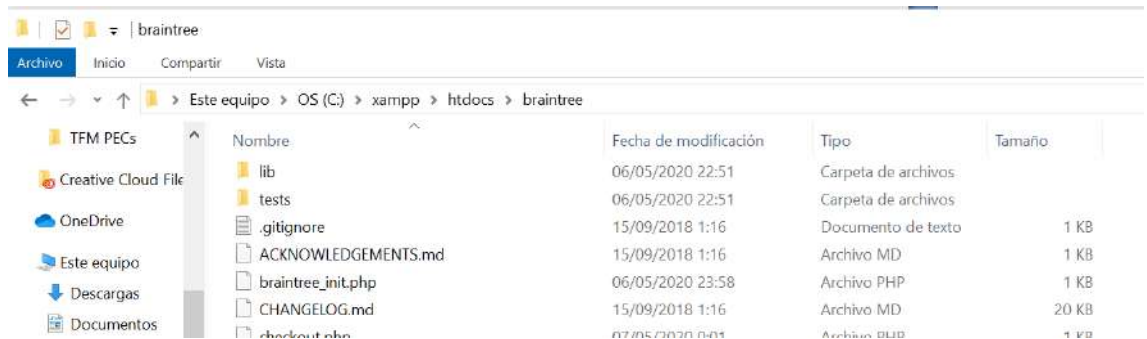


Ilustración 38. Captura de pantalla para visualizar donde se debe introducir el código del servidor PHP para el método de pago

3. Se abre el panel de control XAMPP.

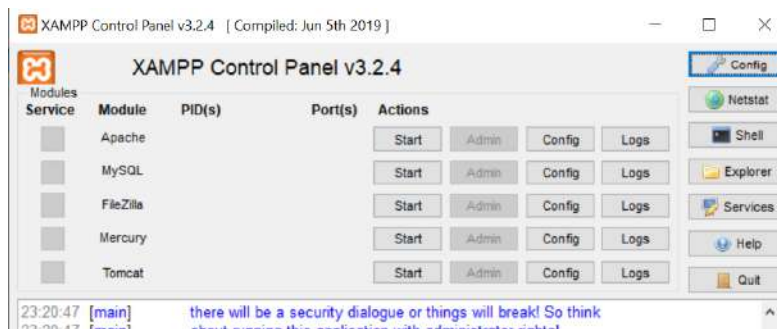


Ilustración 39. Panel de control de XAMPP

4. Se ejecuta Apache y MySQL.

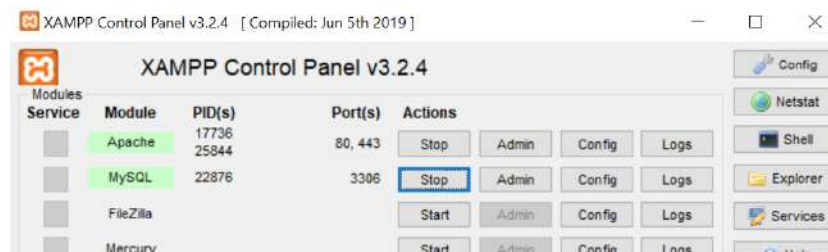



Ilustración 40. Captura de cómo se ejecuta apache y MySQL en XAMPP

5. En el caso de que se quiera tener acceso al servidor desde el exterior se puede utilizar ngrok. Se trata de una herramienta que crea un túnel hacia el puerto en concreto (en este caso Apache está utilizando puerto 80) para que pueda ser accedido desde

cualquier dispositivo a través de una URL aleatoria. Se debe introducir como comando " ngrok.exe http 80".

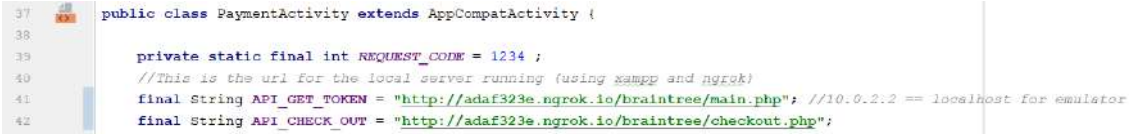
C:\Users\Janire Fernandez\Documents\UOC\ngrok.exe - ngrok.exe http 80



```
ngrok by @inconshreveable
Session Status      online
Session Expires    7 hours, 57 minutes
Version             2.3.35
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding          http://adaf323e.ngrok.io -> http://localhost:80
                   https://adaf323e.ngrok.io -> http://localhost:80
Connections
  ttl   opn   rt1   rt5   p50   p90
   0    0    0.00  0.00  0.00  0.00
```

Ilustración 41. Captura de pantalla de ngrok en funcionamiento.

Además, dicha url deberá incluirse en el código en PaymentActivity.



```
37 public class PaymentActivity extends AppCompatActivity {
38
39     private static final int REQUEST_CODE = 1234 ;
40     //This is the url for the local server running (using xampp and ngrok)
41     final String API_GET_TOKEN = "http://adaf323e.ngrok.io/brainree/main.php"; //10.0.2.2 == localhost for emulator
42     final String API_CHECK_OUT = "http://adaf323e.ngrok.io/brainree/checkout.php";
```

Ilustración 42. Captura de pantalla de dónde se debe introducir en el código la dirección del servidor PHP