



**Universidad**  
Zaragoza

# Trabajo Fin de Grado

Aplicación móvil para gestionar y vender entradas  
para eventos

Mobile application to manage and sell tickets for  
events

Autor

José Daniel Barceló Labuena

Director

Francisco Javier Zarazaga Soria

Escuela de Ingeniería y Arquitectura

2020/2021

# APLICACIÓN MÓVIL PARA GESTIONAR Y VENDER

## ENTRADAS PARA EVENTOS

### RESUMEN

---

El ser humano es sociable por naturaleza y le gusta relacionarse con personas con gustos similares a los suyos. Una vía para esta relación es acudir a acontecimientos que sean de su agrado. Estas actividades son esenciales para permitir la desconexión de la rutina diaria, influyendo así en su estado de ánimo y mejorando la actitud de quienes acuden a estos eventos. Tiene sentido, por tanto, estudiar y mejorar las posibilidades existentes para permitir una mayor flexibilidad y visualización de la información de quienes estén interesados en realizar estas actividades.

La idea de este trabajo de fin de grado surge a través de una experiencia de erasmus en Nápoles (Italia). Cualquier evento en el que se quisiera participar tenía una organización muy rudimentaria y mal organizada, perdiendo así el interés de las personas que pudieran tener en mente acudir al mismo. Concretando más, tanto la información del evento como la información de quienes van a acudir y el método de pago no eran los adecuados dadas las circunstancias de inexperiencia en el idioma y la ausencia de personas conocidas en las que confiar para realizar actividades.

El problema principal que se aborda en este TFG es proveer de una herramienta que combata los problemas de desinformación, organización y pago de eventos tanto a las personas interesadas en acudir a ellos, como a los encargados de organizarlos. La solución consiste en dos aplicaciones móviles, una para cada tipo de usuario, ya que es un método muy sencillo por la rapidez de acceso y la comodidad que ofrece un dispositivo móvil. Esto elimina la necesidad de desplazamiento a cualquier ubicación o la utilización de otras aplicaciones de terceros para informarse acerca de un evento o realizar el pago del mismo. El sistema está orientado, fundamentalmente, a organizadores eventuales de eventos como pueden ser las fiestas de bienvenida de Erasmus, conciertos de fiestas locales, o las exposiciones temporales de arte en sitios no habituales.

## DECLARACIÓN DE AUTORÍA



Escuela de  
Ingeniería y Arquitectura  
Universidad Zaragoza

### DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe remitirse a seceina@unizar.es dentro del plazo de depósito)

D./D<sup>a</sup>. José Daniel Barceló Labuena

en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de Estudios de la titulación de Grado en Ingeniería Informática  (Título del Trabajo) Aplicación móvil para gestionar y vender entradas para eventos

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 22 de Septiembre de 2021

Fdo: José Daniel Barceló Labuena

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

# ÍNDICE

---

Aplicación móvil para gestionar y vender entradas para eventos .....	0
Resumen .....	0
Declaración de autoría .....	1
Agradecimientos.....	4
1 Introducción .....	5
1.1 Contexto del Trabajo .....	5
1.2 Contexto Tecnológico .....	7
1.3 Motivación y problema que se aborda .....	8
1.4 Alcance, objetivos y limitaciones .....	10
1.5 Herramientas de trabajo.....	10
1.6 Esquema general de la memoria del proyecto .....	12
2 Trabajo desarrollado .....	13
2.1 Requisitos del sistema .....	13
2.1.1 Requisitos funcionales aplicación de usuarios .....	13
2.1.2 Requisitos funcionales aplicación de organizadores .....	14
2.1.3 Requisitos no funcionales.....	14
2.1.4 Diccionario de datos .....	15
2.2 Arquitectura software del sistema .....	15
2.2.1 Componente y conector .....	15
2.2.2 Vista de distribución .....	16
2.3 Diseño del sistema .....	17
2.3.1 Navegación de la interfaz de usuario .....	17
2.3.2 Navegación de la interfaz de organizador .....	18
2.3.3 Diagrama de módulos.....	18
2.3.1 Modelo de datos.....	21
2.4 Pasarela de pagos .....	23
2.5 Sistema de garantía de protección de datos .....	24
2.6 Dimensión del trabajo realizado .....	25

2.7	Problemas encontrados .....	26
3	Lecciones aprendidas y conclusiones.....	28
3.1	Conocimientos adquiridos .....	28
3.2	Ideas Futuras.....	28
3.3	Conclusiones .....	29
4	Bibliografía .....	30
5	Anexo I. Manual de usuario aplicación de usuarios.....	31
6	Anexo II. Manual de usuario aplicación de organizadores .....	48

## AGRADECIMIENTOS

---

En primer lugar, me gustaría agradecer a mis padres la oportunidad que me han regalado de poder realizar mis estudios siempre que he querido y me han apoyado tanto económicamente como emocionalmente.

A los profesores de la titulación y en especial, a Javier Zarazaga por su gran ayuda y consejos en el proceso de realización de este TFG.

Y por último, y no por ello menos importante, a mis compañeros de clase que me han acompañado en este camino ayudándome en los momentos más difíciles y compartiendo los mejores momentos.

# 1 INTRODUCCIÓN

## 1.1 Contexto del Trabajo

El trabajo se sitúa en una sociedad de consumo en la que las personas desean en su mayoría pasar tiempo con aquellos a quienes quieren y aprecian, según un estudio realizado en 2020 por el Ministerio de Sanidad, Política Social e Igualdad “Injuve” [1].

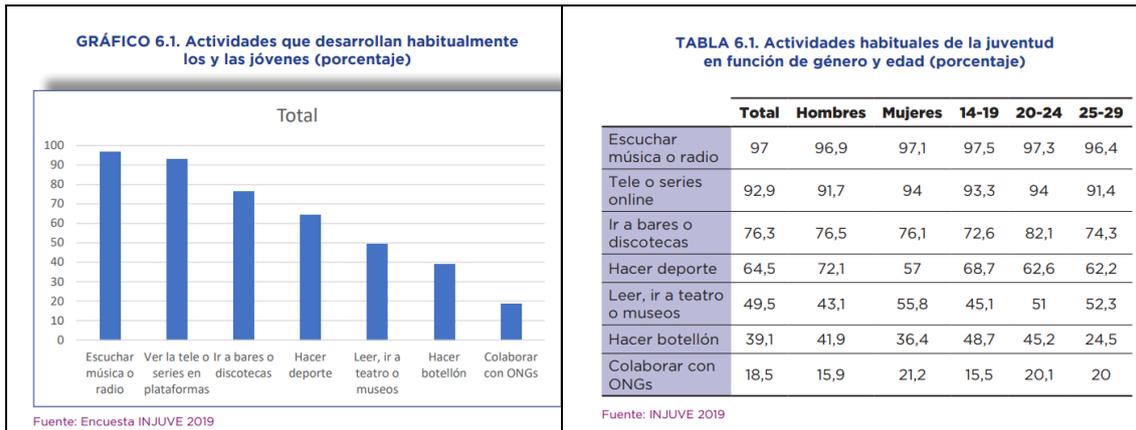


Figura 1: Tabla de actividades que desarrollan habitualmente las personas jóvenes

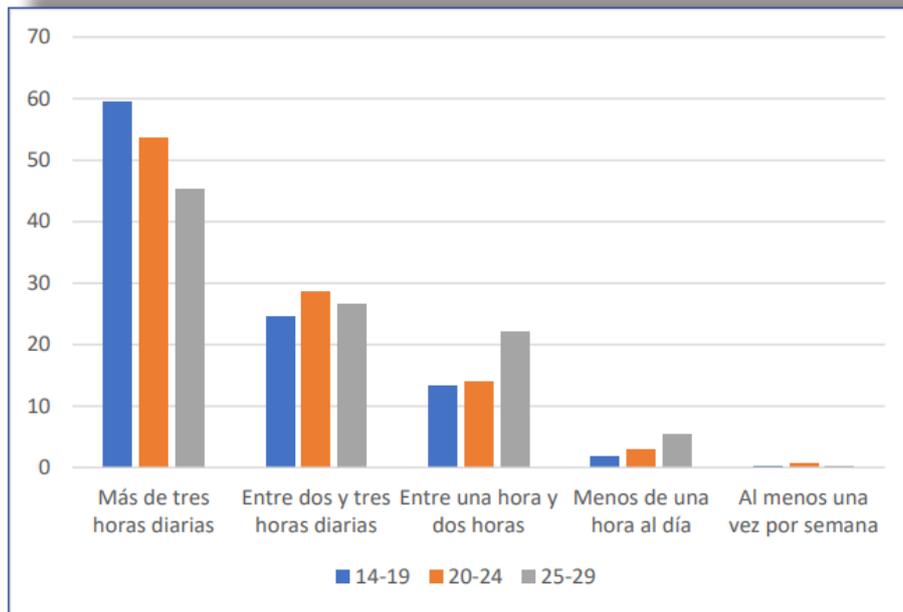
El 76,3% de las personas utiliza su tiempo libre para salir a bares o discotecas y el 49,5% lo utiliza para leer, ir a teatros o museos. Esto constituye aproximadamente un 25% del tiempo de cada día en realizar actividades de ocio, según el instituto Nacional de Estadística en un informe realizado en el 2008 [2]



Figura 2: Distribución del tiempo libre diario

Además de todo esto, el informe de “Injuve” desarrolla una gráfica en la que se observa que más del 50% de las personas entre 14 y 29 años dedica más de tres horas diarias a utilizar internet.

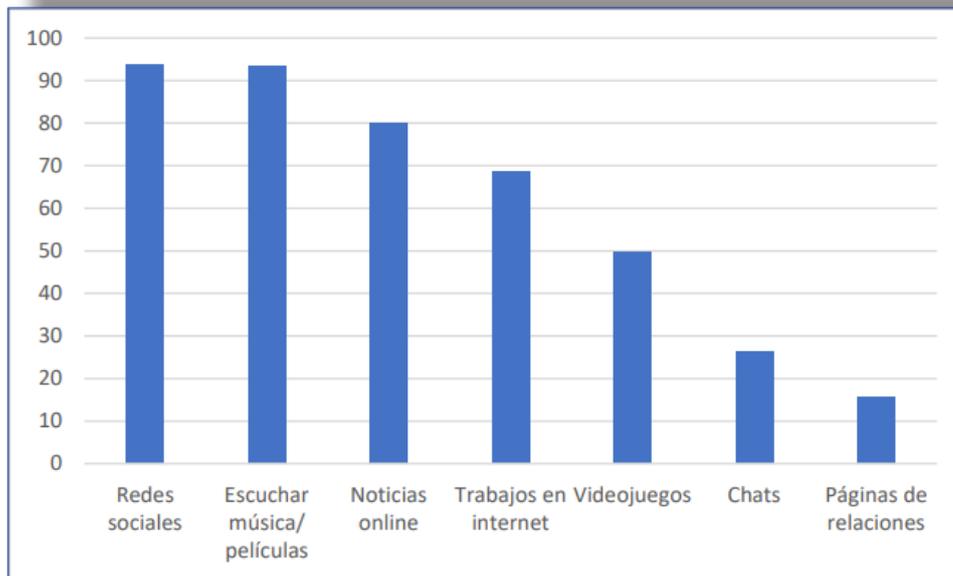
**GRÁFICO 6.5. Porcentaje de consumo de internet según el tramo de edad**



Fuente: Encuesta INJUVE 2019

*Figura 3: Gráfico del uso de internet de la gente joven sesgado por edad*

Más del 90% de este colectivo utiliza el tiempo de consumo de internet en redes sociales, que son la principal herramienta que existe para enterarse de nuevos eventos. Una aplicación complementaria que recopilara todas estas actividades, conociendo que la mayoría de la población joven utiliza gran parte de su tiempo en acudir a sitios de este estilo, haría que los lugares que alojan estas actividades vieran su crecimiento incrementado.

**GRÁFICO 6.6. Porcentaje de actividades desarrolladas en internet**

Fuente: Encuesta INJUVE 2019

Figura 4: Gráfico de actividades desarrolladas en internet

## 1.2 Contexto Tecnológico

Debido a la complejidad asociada al trabajo de desarrollar una aplicación móvil que funcione tanto en iOS<sup>1</sup> como en Android<sup>2</sup>, se ha optado por elegir un lenguaje de programación que permita el desarrollo de ambas aplicaciones de forma simultánea. El lenguaje escogido es React Native<sup>3</sup>, ya que combina los beneficios de JavaScript<sup>4</sup> junto con la tecnología de ReactJS<sup>5</sup>. Esto permite realizar unas transiciones y animaciones muy fluidas en todos los dispositivos en los que se ejecute la aplicación, debido a que las vistas son convertidas en vistas nativas antes de ser renderizadas.

Para el servidor del backend<sup>6</sup> se ha optado por crear un servidor basado en Python<sup>7</sup> que utiliza la librería flask<sup>8</sup> para recibir y enviar peticiones a la aplicación. Se ha elegido esta

<sup>1</sup> <https://es.wikipedia.org/wiki/iOS>

<sup>2</sup> <https://es.wikipedia.org/wiki/Android>

<sup>3</sup> <https://reactnative.dev/>

<sup>4</sup> <https://developer.mozilla.org/es/docs/Web/JavaScript>

<sup>5</sup> <https://es.reactjs.org/>

<sup>6</sup> [https://en.wikipedia.org/wiki/Back\\_end](https://en.wikipedia.org/wiki/Back_end)

<sup>7</sup> <https://es.wikipedia.org/wiki/Python>

<sup>8</sup> <https://es.wikipedia.org/wiki/Flask>

tecnología debido a su gran versatilidad de poder funcionar en cualquier sistema que integre su interpretador. Además, Python es de código abierto y pone a la disposición del usuario una gran variedad de frameworks<sup>9</sup> de gran calibre que permiten realizar cualquier tarea que se desee. Por último, la legibilidad del código es más clara y comprensible, dado que reduce drásticamente los caracteres necesarios para realizar una operación.

### 1.3 Motivación y problema que se aborda

Tal y como se ha visto anteriormente, las personas dedican la mayor parte de su tiempo libre a socializar. Sin embargo, todas estas interacciones llevan haciéndose de la misma manera durante muchos años. Incluso en la época actual de la proliferación de soluciones de todo tipo para todo tipo de problema, tras una búsqueda exhaustiva por la Web, no se ha logrado encontrar ninguna solución que dé respuesta a los retos que se plantean en este TFG.

Los problemas abordados se pueden clasificar desde dos puntos de vista: el de los organizadores y el de los usuarios. Desde el punto de vista de los organizadores de eventos, se tienen que afrontar los siguientes retos/problemas:

- Que los asistentes consigan entrar sin pagar (se “cuelen”) a un evento. Esto puede suceder por falta de organización a la hora de verificar quién ha pagado la entrada. Salvo que se disponga de una infraestructura muy profesional de control (similar a la de los grandes escenarios de actuaciones o los campos de fútbol), lo habitual es que se haga un control manual de la entrada en papel (fácilmente duplicable si se cuenta con una impresora mínimamente buena) o sobre una lista de inscritos (sujeta a falsificaciones en el nombre que se da o en el uso duplicado de un nombre si no se lleva un adecuado seguimiento). Que la gente “se cuele” puede llevar acarreado dos grandes problemas para la organización del evento:
  - La pérdida económica derivada de las personas que acceden al evento pero no pagan la entrada;
  - Los problemas de seguridad vinculados a la superación del aforo permitido/soportado por la instalación del evento.
- Evitar los errores humanos a la hora de cobrar una entrada o de apuntar el nombre de un asistente. La emisión más o menos manual de una entrada suele ir vinculada a la recolección del dinero de la misma. No es extraño que en ocasiones las cuentas entre el número de entradas vendido y el dinero recaudado no cuadre. Por otro lado, si el registro de asistencia o el control de la

---

<sup>9</sup> <https://es.wikipedia.org/wiki/Framework>

misma se lleva a cabo de manera manual sobre listado de nombres, puede darse con mucha facilidad que los datos no se tomen correctamente.

- Facilitar la devolución de entradas. Los problemas de control de asistencia que se describen previamente hacen que resulte complicado el proceso de devolución de entradas ante algún problema que pueda dar lugar a la suspensión del evento.

Como usuarios que utilizan la aplicación para comprar las entradas, se proponen soluciones para los problemas siguientes:

- Facilitar la compra de entradas a través de una aplicación móvil sin necesidad de desplazarse físicamente
- Unificar en un mismo sitio/aplicación todos los eventos de una zona para poder compararlos. En este punto, disponer de una visión geográfica de la ubicación de los eventos puede ser muy útil para poder llegar a ellos.
- Evitar la pérdida de una entrada en formato físico y la posibilidad de entrar al evento.

Algunas de las alternativas existentes en el mercado son Nyxell<sup>10</sup> o Eventbrite<sup>11</sup>, que también son aplicaciones móviles que permiten la compra y fácil gestión de eventos. En el caso de Eventbrite también cuentan con un servicio a través de web. Las dos utilizan el sistema de QR<sup>12</sup> para identificar las entradas y controlar el acceso a los diferentes eventos. Sin embargo, estas alternativas cuentan con el fuerte hándicap de las comisiones<sup>13</sup>. En el caso de Nyxell las comisiones son siempre de un 10%, mientras que en Eventbrite son mayores puesto que el mínimo de comisión que cobran es de un 10%.

En este TFG se plantea desarrollar un producto que, desde un punto de vista de funcionalidad, pretende ser competencia de estas soluciones con añadidos que mejoran las prestaciones como pueden ser la visualización de eventos sobre los mapas o la existencia de una lista de asistentes al evento. La manera de monetizar la solución desarrollada no ha sido un objetivo de este trabajo, pero una posibilidad de competir con estas aplicaciones sería al igual que ellos, trabajar con comisiones, pero más ajustadas. El margen que ofrecen las pasarelas de pago es muy amplio hasta llegar a ese 10%, por lo que un trabajo de calidad, que ofrezca más posibilidades y con unas comisiones menores sería un éxito en el mercado de las aplicaciones de venta y gestión de entradas para eventos.

---

<sup>10</sup> <https://nyxell.com/>

<sup>11</sup> <https://www.eventbrite.es/>

<sup>12</sup> [https://es.wikipedia.org/wiki/C%C3%B3digo\\_QR](https://es.wikipedia.org/wiki/C%C3%B3digo_QR)

<sup>13</sup> <https://es.wikipedia.org/wiki/Comisi%C3%B3n>

## 1.4 Alcance, objetivos y limitaciones

El alcance establecido para este trabajo ha sido el desarrollo de un producto mínimo viable que permitiera acudir ante posibles inversores interesados en hacerlo evolucionar hacia un sistema comercial. Dentro de las limitaciones de este TFG se incluye el no desarrollo del modelo de negocio que debería complementar a este producto mínimo viable.

El objetivo principal del TFG es proveer de una herramienta que combata los problemas de desinformación, organización y pago de eventos tanto a las personas interesadas en acudir a ellos, como a los encargados de organizarlos. Dentro de este objetivo siempre se busca realizar la experiencia de usuario lo más satisfactoria posible a través de la implementación de funcionalidades que faciliten el uso. Este servicio contempla todos los aspectos necesarios para una correcta administración de eventos, desde la venta de estos hasta la comprobación de compra de las entradas para el acceso. Todo este trabajo se va a ver reflejado en dos aplicaciones móviles en las que se podrán realizar las acciones mencionadas anteriormente junto con muchas otras que se desarrollarán más en detalle a lo largo de esta memoria. Este sistema está pensado para ser utilizado por organizadores eventuales de eventos como pueden ser las fiestas de bienvenida de Erasmus, conciertos de fiestas locales, o las exposiciones temporales de arte en sitios no habituales.

Al tratarse de un proyecto personal, la única limitación aparente para mejorar el trabajo o ampliar funcionalidades es el tiempo límite de entrega del TFG, por lo que el margen de mejora es muy amplio. En lo que respecta a otras limitaciones, aparece el hecho de que estas aplicaciones son un prototipo, por lo que éstas no se suben a los marketplaces de sus correspondientes sistemas operativos.

## 1.5 Herramientas de trabajo

En este proyecto se han utilizado tanto herramientas necesarias para el desarrollo del software como herramientas necesarias para la gestión de documentación y otros aspectos.

Herramientas de uso no técnico:

- Microsoft Word<sup>14</sup> para realizar la memoria del proyecto
- Google Meet<sup>15</sup> para reuniones con el tutor del TFG

---

<sup>14</sup> [https://es.wikipedia.org/wiki/Microsoft\\_Word](https://es.wikipedia.org/wiki/Microsoft_Word)

<sup>15</sup> [https://es.wikipedia.org/wiki/Google\\_Meet](https://es.wikipedia.org/wiki/Google_Meet)

- Google Chrome<sup>16</sup> para buscar información y dudas
- Pizarra física de rotuladores para apuntar el seguimiento de tareas pendientes

Herramientas de uso técnico:

- Visual Studio Code<sup>17</sup> como entorno de desarrollo
- Github<sup>18</sup> para almacenamiento del trabajo y gestión de versiones
- Github desktop<sup>19</sup> para gestión de los repositorios
- Xampp<sup>20</sup> para instalar una base de datos local MySQL
- PhpMyAdmin<sup>21</sup> para gestionar directamente la base de datos desplegada
- AWS RDS<sup>22</sup> para desplegar la base de datos
- AWS EC2<sup>23</sup> para desplegar el servidor del backend
- WinSCP<sup>24</sup> para conectar de manera remota vía SSH<sup>25</sup> con el servidor desplegado en AWS EC2
- Expo<sup>26</sup> para emular la aplicación y poder probar el código dentro de un entorno adaptado para ello
- Android Studio<sup>27</sup> para correr la aplicación en un Emulador<sup>28</sup> android y comprobar el correcto funcionamiento
- Postman<sup>29</sup> para simular las peticiones que realizan los clientes a través de la aplicación

Además, se han utilizado máquinas físicas y un entorno de internet propios con las siguientes características

- Equipo de desarrollo: Asus ROG Strix GL553V Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.81 GHz. 128GB SSD & 1TB HDD. Nvidia Geforce GTX 1050Ti. Windows 10.

---

<sup>16</sup> [https://es.wikipedia.org/wiki/Google\\_Chrome](https://es.wikipedia.org/wiki/Google_Chrome)

<sup>17</sup> [https://es.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://es.wikipedia.org/wiki/Visual_Studio_Code)

<sup>18</sup> <https://es.wikipedia.org/wiki/GitHub>

<sup>19</sup> <https://desktop.github.com/>

<sup>20</sup> <https://es.wikipedia.org/wiki/XAMPP>

<sup>21</sup> <https://es.wikipedia.org/wiki/PhpMyAdmin>

<sup>22</sup> [https://en.wikipedia.org/wiki/Amazon\\_Relational\\_Database\\_Service](https://en.wikipedia.org/wiki/Amazon_Relational_Database_Service)

<sup>23</sup> [https://es.wikipedia.org/wiki/Amazon\\_EC2](https://es.wikipedia.org/wiki/Amazon_EC2)

<sup>24</sup> <https://es.wikipedia.org/wiki/WinSCP>

<sup>25</sup> [https://es.wikipedia.org/wiki/Secure\\_Shell](https://es.wikipedia.org/wiki/Secure_Shell)

<sup>26</sup> <https://expo.dev/>

<sup>27</sup> [https://es.wikipedia.org/wiki/Android\\_Studio](https://es.wikipedia.org/wiki/Android_Studio)

<sup>28</sup> <https://es.wikipedia.org/wiki/Emulador>

<sup>29</sup> <https://www.postman.com/>

- Teléfono iOS: iPhone 11 con iOS hasta 14.7.1 con 4GB de memoria RAM
- Teléfono Android: Xiaomi Redmi Note 8T con Android 11 con 3GB de RAM
- Conexión a internet con cable de aproximadamente 100 MB/s de bajada y 100MB/s de subida

## 1.6 Esquema general de la memoria del proyecto

La memoria de este TFG empieza con un resumen sobre el trabajo realizado y su justificación. La primera sección corresponde a la introducción de esta memoria en la que se explica el contexto del proyecto y la motivación para el desarrollo del mismo. A continuación en el apartado 2 “Trabajo desarrollado”, se explica de forma detallada los requisitos del sistema y se expone la estructura y el trabajo mediante diagramas explicados para ayudar a entender todo el proyecto en su conjunto. También se incluyen en otros dos apartados cómo se ha llevado a cabo la protección de datos y la integración de los pagos.

El cuerpo principal de la memoria concluye con un apartado en el que se describen las lecciones aprendidas y conclusiones, así como una sección en la que se habla sobre ideas futuras para seguir desarrollando el proyecto. Al final del documento aparecen dos anexos en los que se desarrollan dos manuales de usuario, uno para cada una de las aplicaciones desarrolladas. Junto a estos manuales aparecen capturas de pantalla de todas las pantallas de las dos aplicaciones.

## 2 TRABAJO DESARROLLADO

### 2.1 Requisitos del sistema

En esta sección se van a describir los requisitos que se han considerado fundamentales para la realización de este TFG.

#### 2.1.1 REQUISITOS FUNCIONALES APLICACIÓN DE USUARIOS

RFU1	El sistema debe permitir registrar a un usuario
RFU2	El sistema debe permitir iniciar y cerrar sesión a un usuario
RFU3	El sistema debe permitir reestablecer la contraseña de un usuario
RFU4	El sistema debe permitir establecer una ubicación geográfica a cada usuario
RFU5	El sistema debe permitir cambiar la ubicación geográfica a cada usuario
RFU6	El sistema debe permitir la visualización de la lista eventos según la ubicación geográfica del usuario
RFU7	El sistema debe permitir la visualización de los eventos según la ubicación geográfica del usuario sobre un mapa interactivo
RFU8	El sistema debe permitir visualizar la información de un evento
RFU9	El sistema debe permitir a un usuario apuntarse a la lista de asistentes de un evento
RFU10	El sistema debe permitir a un usuario visualizar la lista de asistentes de un evento
RFU11	El sistema debe permitir a un usuario obtener la ruta con el uso de una aplicación de mapas hasta la localización de un evento
RFU12	El sistema debe permitir a un usuario pagar la entrada de un evento
RFU13	El sistema debe permitir a un usuario pagar la entrada de un evento para otras personas (Hasta un máximo de 10 por transacción)
RFU14	El sistema debe permitir a un usuario visualizar la lista de entradas de eventos que ha comprado
RFU15	El sistema debe permitir a un usuario visualizar un código QR con la entrada correspondiente a la compra de una entrada para un evento
RFU16	El sistema debe permitir a un usuario eliminar su cuenta de forma permanente

RFU17	El sistema debe permitir a un usuario enviar un formulario de contacto para ayudas o sugerencias
-------	--------------------------------------------------------------------------------------------------

### **2.1.2 REQUISITOS FUNCIONALES APLICACIÓN DE ORGANIZADORES**

RFO1	El sistema debe permitir iniciar y cerrar sesión a un organizador
RFO2	El sistema debe permitir a un organizador crear un evento en la ubicación geográfica del organizador
RFO3	El sistema debe permitir a un organizador visualizar todos los eventos creados por sí mismo
RFO4	El sistema debe permitir visualizar a un organizador la lista de personas que han comprado una entrada a un evento
RFO5	El sistema debe permitir a un organizador añadir manualmente a una persona a la lista de personas que han comprado una entrada a un evento
RFO6	El sistema debe permitir a un organizador eliminar a una persona de la lista de personas que han comprado una entrada a un evento
RFO7	El sistema debe permitir a un organizador modificar la información de un evento creado
RFO8	El sistema debe permitir a un organizador cancelar un evento creado
RFO9	El sistema debe permitir a un organizador escanear un código QR de una entrada comprada de un evento

### **2.1.3 REQUISITOS NO FUNCIONALES**

RNF1	El idioma de la aplicación será inglés
RNF2	El funcionamiento de la aplicación será fluido, con unos tiempos de respuesta cortos siempre que sea posible (alrededor de un segundo)
RNF3	La aplicación contará con sistemas de cifrado de claves
RNF4	La aplicación necesitará de conexión a internet para poder realizar cualquier operación
RNF5	La aplicación debe de estar disponible para teléfonos con sistema operativo iOS y Android
RNF6	El inicio de sesión en la aplicación de usuarios se recuerda aunque la aplicación sea cerrada, manteniendo la sesión abierta(*)

(\*)Este requisito no se ha aplicado a la aplicación destinada a los organizadores para evitar problemas de seguridad en caso de robo del dispositivo móvil.

### **2.1.4 DICIONARIO DE DATOS**

- Usuario: email, alias, teléfono, ubicación geográfica y contraseña (cifrada con el algoritmo sha256).
- Ubicación geográfica: tupla de datos que representan una ubicación, conformado por país y ciudad.
- Evento: título, descripción, fecha, foto, imagen de fondo, precio, comisión, ubicación del evento, máximo de asistentes, asistentes actuales, fecha de deshabilitación de la compra, fecha de deshabilitación del evento en el feed de eventos, URL de la web del evento, usuario organizador, lista de usuarios asistentes y lista de usuarios que han comprado una entrada.
- Feed de eventos: pantalla de la aplicación en la que aparecen los eventos disponibles dentro de la ubicación geográfica configurada por el usuario
- Entrada: objeto digital que se compra a través de la aplicación que permite el acceso al evento asociado a la misma. Tiene asociada un código QR con su información.

## **2.2 Arquitectura software del sistema**

### **2.2.1 COMPONENTE Y CONECTOR**

La arquitectura de la aplicación se basa en el modelo cliente-servidor, siendo los clientes las dos aplicaciones móviles y el servidor el encargado de atender sus peticiones. Las dos aplicaciones móviles han sido desarrolladas en React Native, que como se ha comentado anteriormente, es una librería de JavaScript combinada con ReactJS que nos permite desarrollar una app para iOS y Android con un mismo código.

El servidor escogido ha sido un servidor de Python utilizando la tecnología Flask, que es un framework minimalista que permite crear de forma sencilla, rápida y con muy pocas líneas de código un servidor web. El servidor de Flask utiliza una base de datos RDS de Amazon Web Services para almacenar toda la información. Esta base de datos es una base de datos relacional que se ha configurado para utilizar MySQL<sup>30</sup>. Esto compone una base de datos muy rápida en lectura que se adapta a la perfección a la aplicación desarrollada, ya que la mayoría de consultas van a ser lecturas de datos; y las escrituras serán, en su mayoría, transacciones monetarias que de cualquier manera necesitarían un tiempo de procesamiento mayor.

---

<sup>30</sup> <https://es.wikipedia.org/wiki/MySQL>

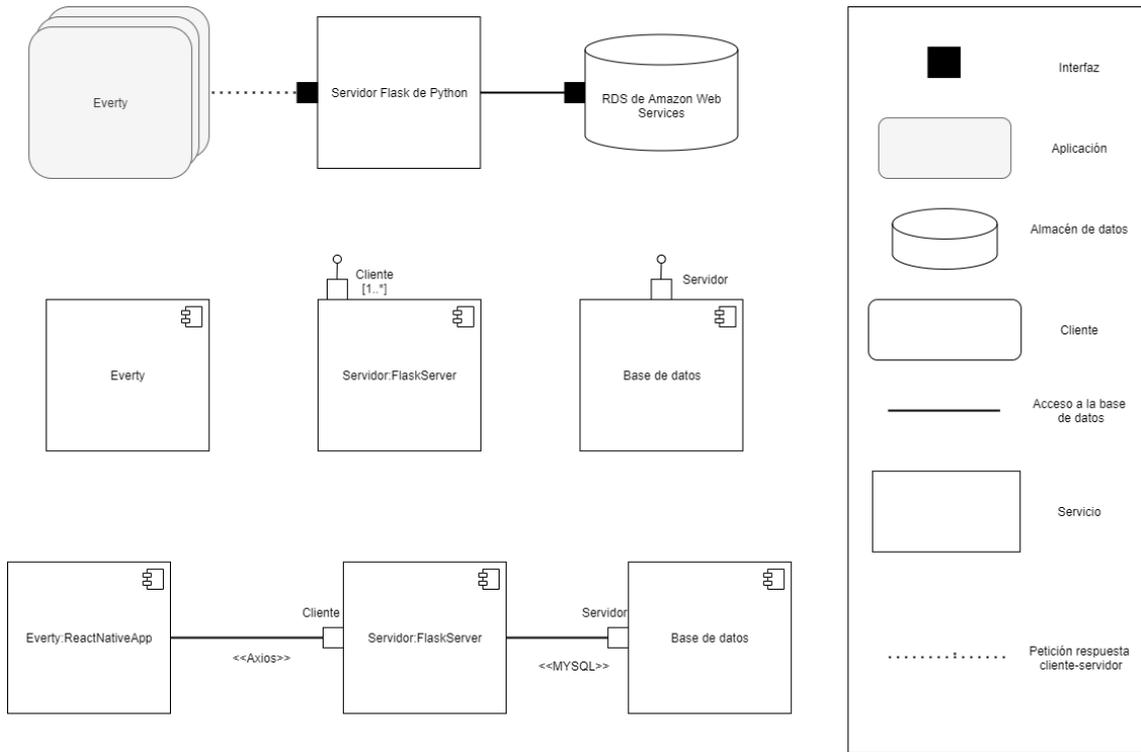


Figura 5: Diagrama componente y conector

### 2.2.2 VISTA DE DISTRIBUCIÓN

La distribución de la aplicación está compuesta de tres capas propias y una externa: la capa de la aplicación, que se ejecuta en los dispositivos móviles de los usuarios, la capa de servidor que se ejecuta en una máquina virtual dentro de Amazon Web Services, la capa de datos que se ejecuta sobre otra máquina virtual especial para bases de datos RDS también en Amazon Web Services y la capa del servicio de la plataforma de pagos.

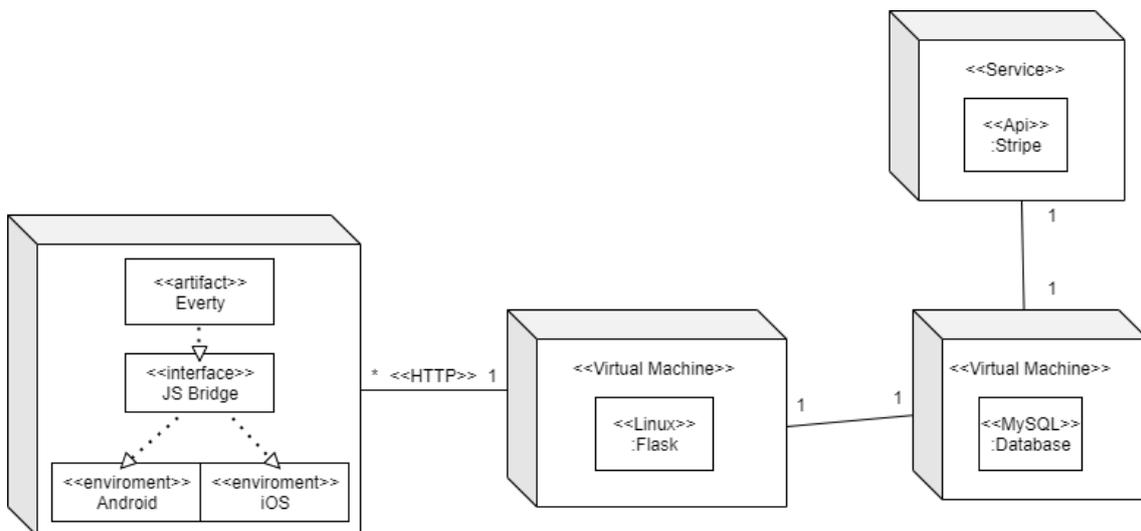


Figura 6: Vista de distribución

## 2.3 Diseño del sistema

### 2.3.1 NAVEGACIÓN DE LA INTERFAZ DE USUARIO

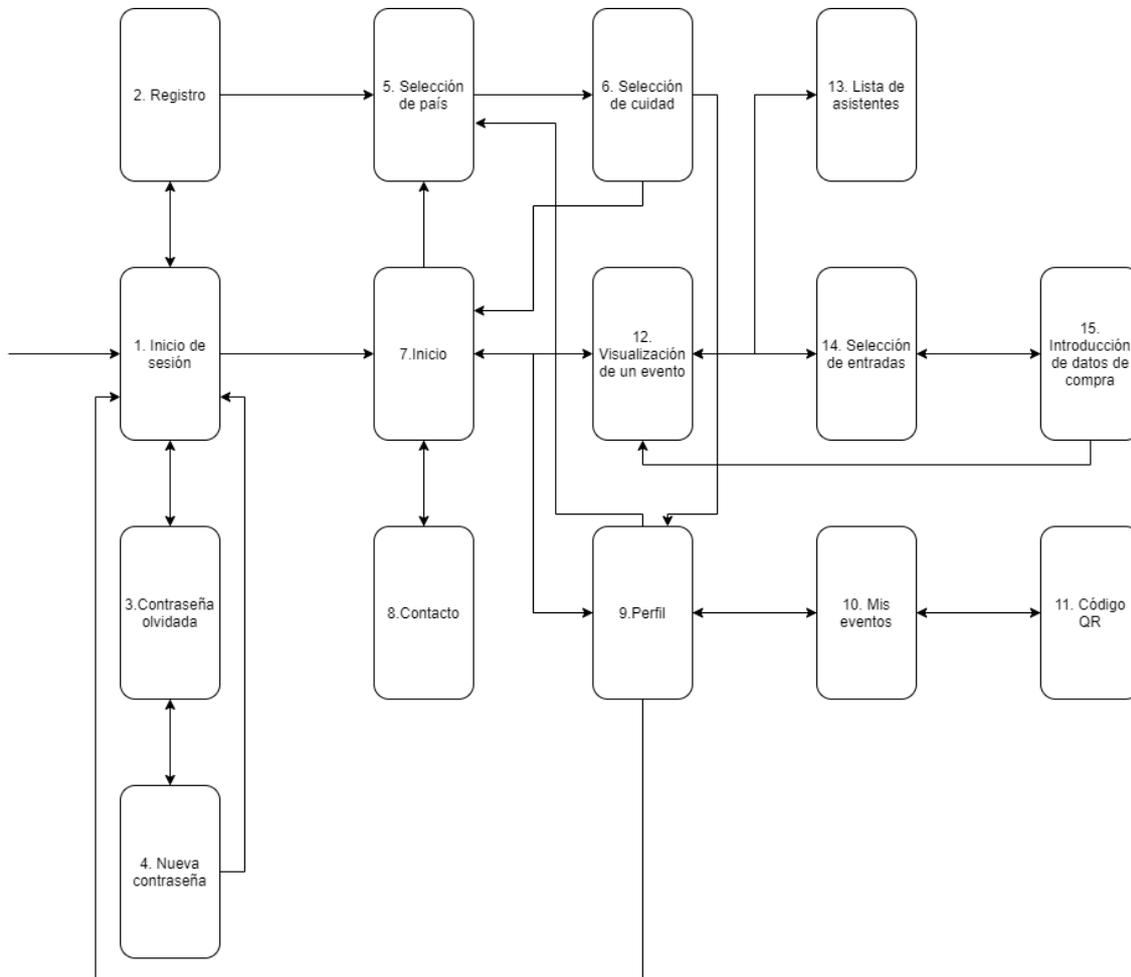


Figura 7: Mapa de navegación de la aplicación de usuarios

En la figura 7 se puede observar la forma de navegar entre las pantallas de la aplicación de usuarios. El sentido de las flechas es importante puesto que, si una flecha es unidireccional, significa que el recorrido debe continuar hasta el final o hasta una pantalla en la que se pueda navegar a otro lugar de la aplicación, como sucede cuando se accede a la pantalla “5. Selección de país”. También cabe destacar que desde la pantalla “6. Selección de ciudad” existen 2 flechas salientes de la pantalla, pero esta navegación es desde un mismo botón; que dependiendo de la pantalla desde la que se haya navegado hasta la pantalla “5. Selección de país”, volverá a la pantalla “9. Perfil” o a la pantalla “7. Inicio”.

### 2.3.2 NAVEGACIÓN DE LA INTERFAZ DE ORGANIZADOR

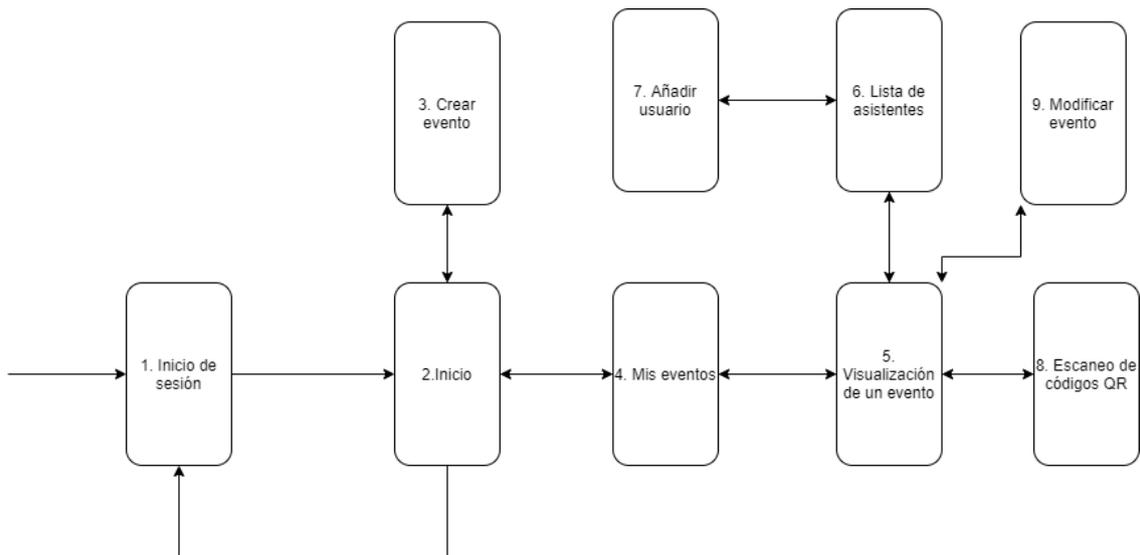


Figura 8: Mapa de navegación de la aplicación de organizadores

De la misma manera que en el mapa de navegación de la aplicación de usuarios, en la figura 8 se puede ver con la misma estructura el mapa de la aplicación de organizadores. Un punto a destacar es que para ir a la pantalla “1.Inicio de sesión”, solo es posible hacerlo desde la pantalla “2.Inicio”, pero está representado con 2 flechas unidireccionales diferentes ya que la acción de “Volver” no está disponible en la pantalla 2, sino que hay que cerrar sesión para poder volver a la pantalla 1.

### 2.3.3 DIAGRAMA DE MÓDULOS

Tanto el servidor como la parte lógica del frontend<sup>31</sup> están formados por dos módulos principales:

- Módulo de eventos: encargado de las acciones relacionadas con los eventos: creación, modificación y borrado de eventos, pago y apuntado de personas a eventos y obtención de eventos.
- Módulo de usuarios: encargado de administrar y gestionar las operaciones relacionadas con los usuarios: registro e inicio de sesión, modificación de datos de usuarios, recuperación de contraseñas, obtención de datos de perfiles y borrado de usuarios.

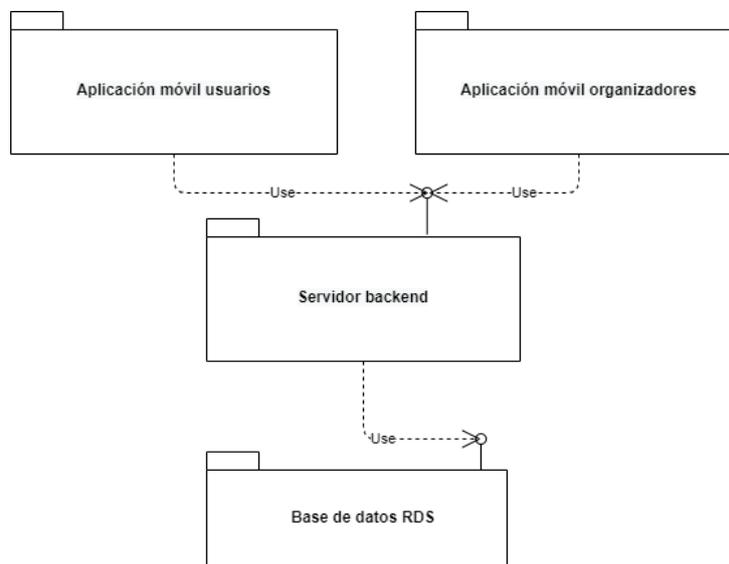
Además, en las aplicaciones en la parte del frontend, se ha seguido una estructura de organización de código que permite separar fácilmente las aplicaciones en cuatro grandes partes:

<sup>31</sup> [https://es.wikipedia.org/wiki/Front\\_end\\_y\\_back\\_end](https://es.wikipedia.org/wiki/Front_end_y_back_end)

- Controladores: encargados de la comunicación con el servidor.
- Pantallas: ficheros utilizados para mostrar la información al usuario.
- Navigation: responsables de la navegación de las aplicaciones.
- Assets: conjunto de imágenes y fuentes de texto utilizadas.

Con lo que respecta al backend, el servidor de Flask se encarga de atender las peticiones provenientes de las aplicaciones y dirigir las a su módulo correspondiente. Estos son los encargados de realizar la comunicación con la base de datos para gestionar la petición y la lógica de aplicación solicitada. Por último, la base de datos RDS utiliza MySQL, que es la encargada de definir el modelo de la base de datos.

En la figura 9a se puede ver una vista general del diagrama de módulos. Estos módulos están desarrollados con mayor profundidad a continuación para favorecer la claridad.



*Figura 9a: Diagrama de módulos (Vista general)*

En esta figura 9a queda claro la estructura del sistema, en la que las dos aplicaciones hacen uso del servidor backend para suplirse de información, que a su vez este servidor obtiene a partir del uso de la base de datos.

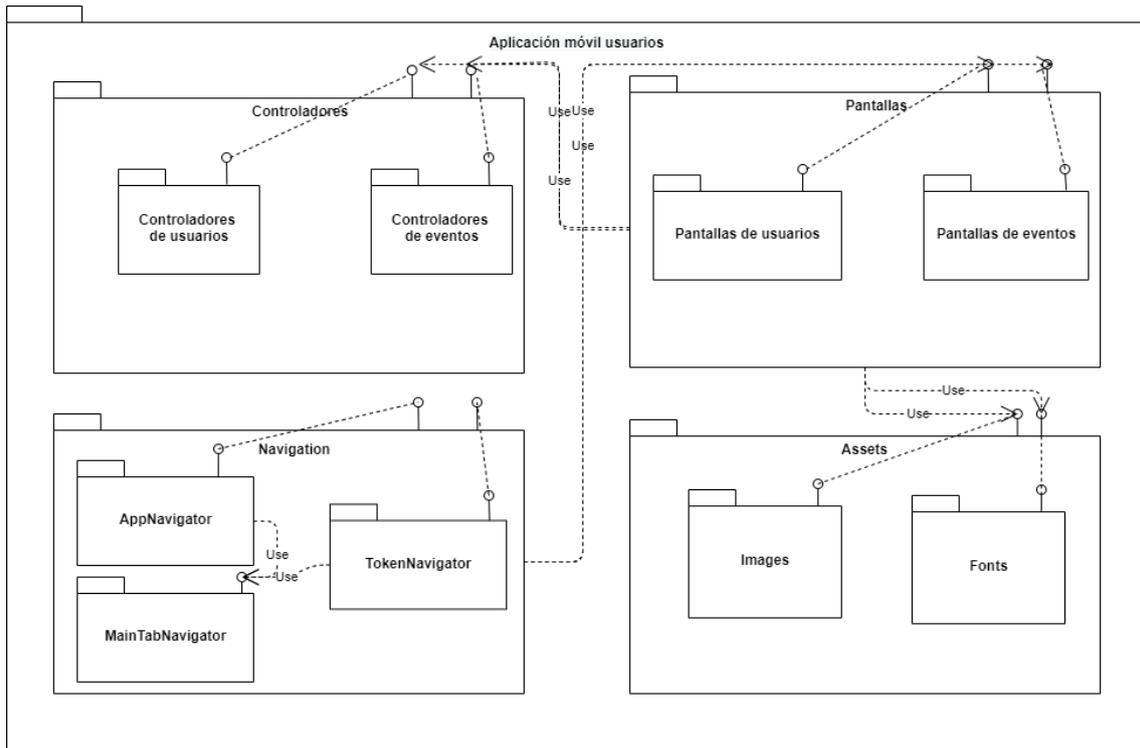


Figura 9b: Diagrama de módulos (Módulo aplicación de usuarios)

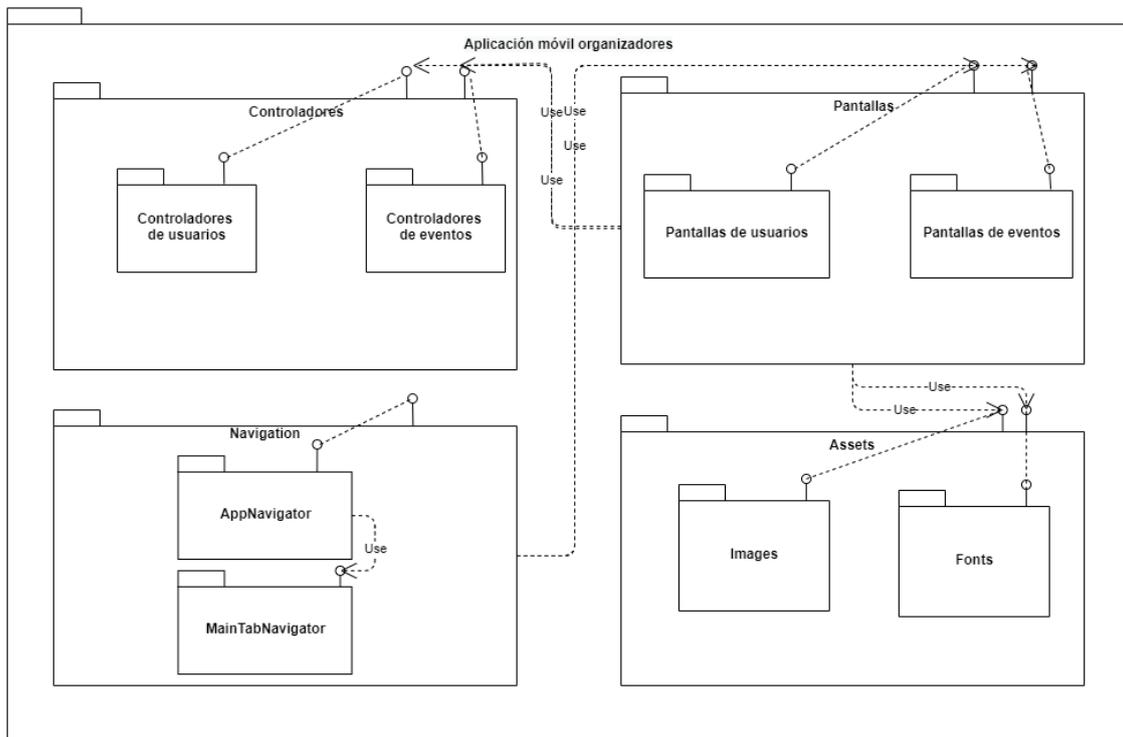


Figura 9c: Diagrama de módulos (Módulo aplicación de organizadores)

En las dos figuras anteriores 9b y 9c se puede distinguir cuatro grandes módulos en los que se dividen las aplicaciones: el módulo de pantallas que es dónde se renderiza el contenido que ve el usuario, el módulo de navegación que se encarga de navegar entre

estas pantallas, el módulo de controladores que se encarga de la conexión con el servidor y el módulo de “assets” que almacena las imágenes y las fuentes utilizados en las aplicaciones.

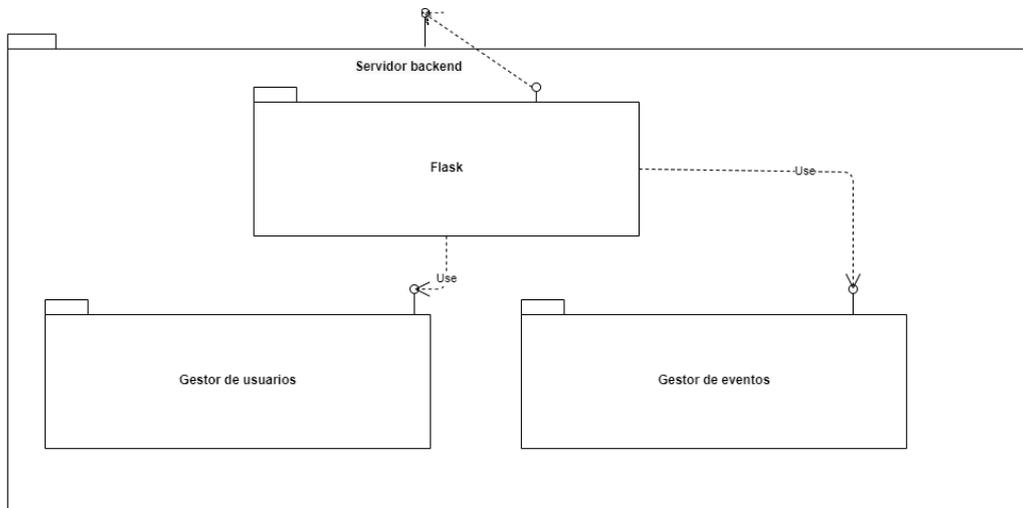


Figura 9d: Diagrama de módulos (Módulo servidor backend)

En este diagrama se observa que el servidor de backend tiene dos grandes módulos separados: el de usuarios y el de eventos. Cada uno de estos tiene la lógica necesaria para gestionar todo lo relacionado con ellos mismos. El módulo de flask es el encargado de agrupar toda esta información para que sea accesible desde las aplicaciones.

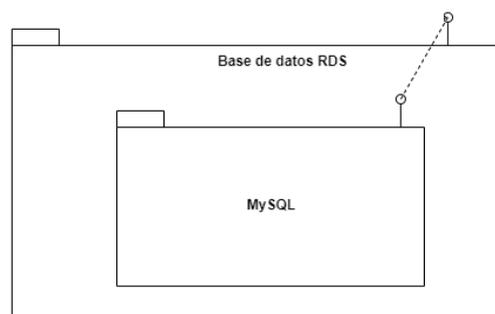


Figura 9e: Diagrama de módulos (Módulo base de datos)

Por último en el diagrama de módulos de la base de datos, hay un único componente que es la base de datos MySQL, utilizada por el servidor backend para obtener los datos con los que trabajar.

### 2.3.1 MODELO DE DATOS

En la base de datos RDS se almacenan todos los datos necesarios para el correcto funcionamiento de las aplicaciones.

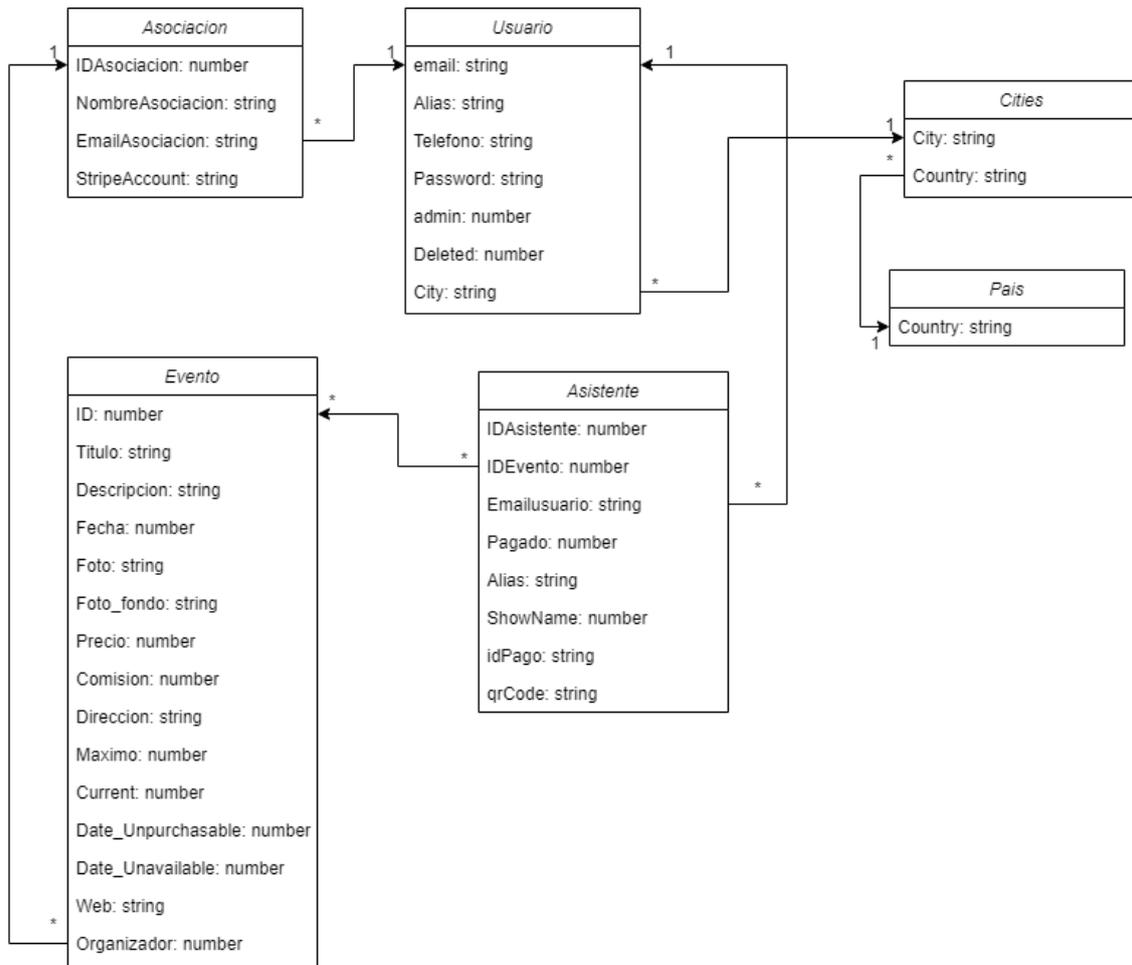


Figura 10: Modelo de datos

Los objetos principales de las aplicaciones son los eventos. Estos están compuestos de los datos necesarios para poder ser utilizados, y están relacionados directamente con un conjunto de asistentes.

Cada asistente puede ser una persona que se haya apuntado a la lista de asistentes a un evento o que haya pagado el precio de asistencia. Esto se distingue en el campo "Pagado" que estará a 1 si lo ha pagado y a 0 si no lo está. Además, cada asistente pagado tendrá asociado un identificador único del pago generado por la plataforma de pagos y un código QR obtenido a partir del identificado para permitirle la entrada al evento. Cada asistente está relacionado a un usuario, pero aún así se almacena el alias del asistente en el objeto ya que un solo usuario puede comprar entradas para otras personas y estas entradas quedan asociadas a su cuenta.

Los eventos tienen un solo organizador con el que están relacionados a través de un identificador único. Estos organizadores son asociaciones que previamente han contactado para poder tener una cuenta de organizadores en la aplicación correspondiente. Para crear esta cuenta, basta con asociar un objeto de "Asociación" a un usuario normal y corriente y cambiar el campo "admin" del usuario a 1. Cabe destacar

que, al hacer esta modificación sobre el usuario, este dejará de funcionar en la aplicación de usuarios y pasará a funcionar únicamente en la aplicación de organizadores.

Además, cada usuario debe configurar una zona geográfica para que le aparezcan los eventos de su área. Esto también sirve para localizar los eventos a través de la cuenta de usuario asociada a una organización: la organización creadora del evento.

## 2.4 Pasarela de pagos

Para poder implementar la funcionalidad de venta de entradas para los eventos, es necesario el uso de una pasarela de pagos<sup>32</sup>. Este es un servicio de un proveedor externo que sirve para autorizar pagos en negocios electrónicos. También se conoce como la versión digital de un TPV (Terminal punto de venta<sup>33</sup>)

Teniendo esto en cuenta, la primera idea que aparece es utilizar las pasarelas de pagos que ofrecen los bancos, ya que éstas son las más extendidas y tienen, por norma general, unas buenas condiciones y bajas comisiones de uso. Una de ellas es la TPV Virtual de Ibercaja, que ofrece unas comisiones de entre un 0,4% y un 0,8% por cada transacción más una cuota fija mensual de 3 euros.

Al intentar contratar este servicio, aparecen los problemas burocráticos provenientes del hecho de ser un estudiante y no un autónomo o empresa consolidada. Tras acudir más de cinco veces a diferentes oficinas, enviar más de diez emails y solicitar hablar con el equipo técnico de Ibercaja, no se obtuvo respuesta alguna más allá de un "Ya te llamaremos". También se intentó contratar el mismo servicio en el banco Santander, en el que las comisiones eran más elevadas y una cuota fija parecida; pero seguía siendo una opción rentable. Para poder optar a esta opción, era necesario ser cliente del banco Santander, algo que no parece un gran dilema, pero obligaban a abrir una cuenta con un mínimo de 10.000€ en ella y con unas comisiones demasiado altas, por lo que también se descartó.

Tras no tener éxito con los bancos, la siguiente posibilidad eran pasarelas de pagos de otros proveedores como pueden ser Paypal<sup>34</sup> o Stripe<sup>35</sup>. El sentido común dice que Paypal es una entidad muy conocida y en la que la gente confía, por lo que parecía una buena opción para implementar los pagos. Sin embargo, aparece de nuevo el problema de las comisiones: 0,35€ fijos más 2,9% del precio a pagar. El porcentaje no parece

---

<sup>32</sup> [https://es.wikipedia.org/wiki/Pasarela\\_de\\_pago](https://es.wikipedia.org/wiki/Pasarela_de_pago)

<sup>33</sup> [https://es.wikipedia.org/wiki/Terminal\\_punto\\_de\\_venta](https://es.wikipedia.org/wiki/Terminal_punto_de_venta)

<sup>34</sup> <https://es.wikipedia.org/wiki/PayPal>

<sup>35</sup> <https://es.wikipedia.org/wiki/Stripe>

demasiado grande, pero teniendo en cuenta que la mayoría de los pagos van a ser pequeños (en torno a 5€ - 10€), la comisión fija lo arruina por completo.

Como última opción aparece Stripe, plataforma que en primera instancia suena desconocida, pero tras investigar se descubre que muchos negocios mundialmente conocidos como Twitter<sup>36</sup>, Facebook<sup>37</sup> o Deliveroo<sup>38</sup> usan Stripe como su pasarela de pagos. Las comisiones que ofrece Stripe son bastante mejores que las de Paypal, ya que para tarjetas europeas la comisión es de 0,25€ fijos más 1,4% del precio que se va a pagar. Esta comisión fija también es algo indeseable, pero al ser más baja junto con el porcentaje en comparación a Paypal, convierte a Stripe en la opción elegida para implementar la pasarela.

Para poder utilizar Stripe basta con confirmar la identidad a través de la vinculación de una cuenta bancaria a la cuenta de usuario de Stripe. Además, a la hora de implementar hay a disposición del usuario una amplia documentación con ejemplos de uso de la API para diferentes lenguajes de programación, lo que facilita bastante la implementación del servicio. Incluso cuentan con un servicio de chat para atención al cliente que está disponible las 24 horas del día en caso de necesitar ayuda o alguna información adicional.

## 2.5 Sistema de garantía de protección de datos

La protección de datos es un tema muy delicado cuando se tratan temas de ámbito personal o pagos por internet. Por este motivo, en las aplicaciones desarrolladas en este proyecto se ha intentado evitar recopilar información sensible de los usuarios y así cumplir la normativa vigente de la Unión Europea para la protección de datos<sup>39</sup>. Esta normativa dice, en resumen, que para que no sea necesaria esta protección de datos no se deben relacionar formas de contacto con datos que representen una persona física. Traducido al ámbito del proyecto esto quiere decir que la normativa no se cumpliría si se almacenase el nombre de los usuarios registrados o de quienes compran una entrada. La forma de evitar esta relación es la utilización de un alias o nombre de usuario. El alias sirve para identificar a una persona de tal manera que los datos no queden comprometidos por estar relacionados con ellos directamente. Estos posibles datos serían el número de teléfono o el correo electrónico.

---

<sup>36</sup> <https://es.wikipedia.org/wiki/Twitter>

<sup>37</sup> <https://es.wikipedia.org/wiki/Facebook>

<sup>38</sup> <https://es.wikipedia.org/wiki/Deliveroo>

<sup>39</sup> [https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu\\_es](https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_es)

En lo que respecta a la seguridad de los pagos, el nombre o la identidad de una persona es un carácter indispensable para verificar que un pago es verídico y no un intento de fraude. La forma de solventar esto es a través de la delegación de esta responsabilidad sobre el proveedor de la pasarela de pagos, en este caso, Stripe. Pese a que sea una entidad externa la encargada de la verificación de los pagos, también sería necesario almacenar un registro de estos pagos en caso de que sucediera algún problema con alguna entrada, y almacenar los datos de estos pagos implicaría tener que guardar el nombre. La solución consiste en guardar un identificador único que Stripe genera cada vez que se realiza un pago. Este identificador permite obtener toda la información necesaria requerida para un pago a través de la aplicación. Para obtenerlo basta con hacer uso de la API de Stripe y obtener de forma sencilla en formato JSON los datos de la transacción para su posterior uso en caso de que fuera necesario.

## 2.6 Dimensión del trabajo realizado

La planificación inicial concuerda con la planificación final, ya que esta apenas se ha modificado en lo que respecta a la distribución de las horas de trabajo.



*Figura 11: Distribución del trabajo realizado*

En total se han invertido 690 horas aproximadamente, distribuidas de la siguiente manera:

- 240 horas en desarrollar la aplicación de usuarios
- 160 horas en desarrollar la aplicación de organizadores
- 170 horas en desarrollar el servidor y la base de datos
- 30 horas en diseñar la estructura de las aplicaciones y servidor
- 60 horas de formación en las tecnologías
- 30 horas de documentación del proyecto

El código queda finalmente distribuido de la siguiente manera:

```

37 text files.
37 unique files.
193 files ignored.

http://cloc.sourceforge.net v 1.60 T=5.07 s (5.9 files/s, 832.2 lines/s)
-----
Language          files          blank          comment          code
-----
Javascript         30             314             34             3873
SUM:               30             314             34             3873
-----

```

Figura 12: Líneas de código Aplicación de usuarios

```

30 text files.
30 unique files.
48 files ignored.

http://cloc.sourceforge.net v 1.60 T=3.68 s (5.7 files/s, 675.5 lines/s)
-----
Language          files          blank          comment          code
-----
Javascript         21             188             35             2262
SUM:               21             188             35             2262
-----

```

Figura 13: Líneas de código Aplicación de organizadores

```

8 text files.
8 unique files.
63 files ignored.

http://cloc.sourceforge.net v 1.60 T=0.92 s (4.3 files/s, 714.8 lines/s)
-----
Language          files          blank          comment          code
-----
Python            4              112             16             533
SUM:              4              112             16             533
-----

```

Figura 14: Líneas de código servidor backend

## 2.7 Problemas encontrados

Un problema encontrado fue la implementación de mapas en la aplicación. Al tratar de desarrollar la aplicación tanto para iOS como para Android, el proveedor de mapas tiene que ser escogido con cuidado, ya que estos sistemas operativos no tienen las mismas compatibilidades. En primera instancia, como el dispositivo principal de desarrollo era un iPhone, se implementaron los mapas con Apple maps, pero pronto apareció el problema de funcionamiento en Android.

Tras esto, se valoró utilizar la API de google maps para permitir la visualización de mapas y el renderizado de los marcadores encima de este para poder interactuar con ellos. Esta solución parece la más adecuada puesto que parece cumplir con los requisitos que se necesitan, pero hay un requisito que no se consigue implementar con esta API: la posibilidad de desplazar un marcador para establecer la ubicación de un evento a la hora de crear o modificar uno de estos.

Con una librería desarrollada para react native se consiguen implementar todos los requisitos relacionados con los mapas por lo que el problema parece resuelto, pero tras esto aparecen los permisos; que son gestionados de manera diferente dependiendo del sistema operativo y de la versión, pero otra librería externa unifica estos permisos.

La integración con Stripe también ha supuesto un problema, ya que la complejidad que tiene una transacción monetaria es bastante elevada y se ha tenido que aprender cómo funciona.

El siguiente problema fue el despliegue del servidor y de la base de datos. En la etapa de desarrollo se había utilizado tanto el servidor como la base de datos de forma local en el dispositivo en el que se estaba desarrollando. Debido a la inexperiencia al desplegar en la nube, el tiempo de realizar esta actividad se ha incrementado notablemente. La decisión para elegir el sitio de despliegue fue fácil, AWS. Se ha elegido AWS porque todo el mundo tiene un periodo de prueba de 1 año para utilizar los servicios dentro del "Free tier" que son más que suficientes para realizar este proyecto.

La mayor complicación consistía en cómo había que instalar todo de forma remota en los servidores contratados en AWS, ya que todos ellos cuentan con un firewall que bloquea cualquier tipo de entrada o salida a los servidores a no ser que se permita específicamente. Además, aparte de modificar el firewall, para acceder al almacenamiento de la máquina a través de SSH es necesario también el uso de una clave privada para una mayor seguridad. Esta restricción costó bastante tiempo solucionarla ya que el programa utilizado para esto, WinSCP, es difícil de configurar con estas características.

## 3 LECCIONES APRENDIDAS Y CONCLUSIONES

---

### 3.1 Conocimientos adquiridos

Lo principal ha sido el conocimiento en las tecnologías de todos los componentes que constituyen este proyecto, ya que antes de empezar todas ellas eran desconocidas y han requerido de una formación previa para poder trabajar.

También se ha aprendido a trabajar con APIs<sup>40</sup> externas con Stripe que, a pesar de tener una documentación excelente, han supuesto un desafío porque a excepción de una asignatura realizada al mismo tiempo que el desarrollo de este trabajo, nunca se había realizado previamente.

El proceso de diseñar una arquitectura correcta y funcional para una aplicación es un conocimiento muy valioso aprendido a raíz de realizar el proyecto. La estructura de cómo se realiza cualquier trabajo y su planificación son los pilares fundamentales para que todo salga perfecto.

Por último, también se ha aprendido a hacer una gestión de tipos de usuarios correcta dentro de las aplicaciones, que es un tema muy delicado y que casi todos los proyectos de hoy en día poseen esta característica.

### 3.2 Ideas Futuras

Debido a limitaciones temporales y personales, no se han podido realizar todas las ideas que se tenían para poder mejorar el trabajo, pero se tienen pensadas bastantes ideas para realizar en un futuro y proveer de más funcionalidades a la aplicación.

La primera idea, sería adaptar el código de react native para que funcione también en web y desplegarlo. De esta manera, no sería necesario instalar explícitamente la aplicación en un dispositivo móvil, por lo que se podría utilizar en cualquier dispositivo con un navegador. Esto permitiría un alcance a un público mucho mayor.

En lo que respecta a la aplicación se tienen varias ideas entre las que destaca el añadir una sección de favoritos en las que los usuarios pueden marcar eventos como favoritos para poder acceder de una forma más rápida a ellos sin necesidad de tener que buscarlos. Relacionada con esta idea, aparece la siguiente: implementar filtros y una barra de búsqueda para los eventos, que ayudaría a buscar eventos según los gustos o disponibilidad de los usuarios.

---

<sup>40</sup> [https://es.wikipedia.org/wiki/Interfaz\\_de\\_programaci%C3%B3n\\_de\\_aplicaciones](https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones)

Otras ideas son: integrar el uso de mapas para mostrar la dirección de un evento, añadir una sección de FAQ, almacenar lugares físicos en los que se realicen eventos para poder valorarlos, añadir la funcionalidad de enviar notificaciones push o crear una URL personalizada para cada evento para poder compartirlo con mayor facilidad.

Además, una mejora que incrementaría muchísimo la utilidad de la aplicación sería fusionarla con funcionalidades propias de una red social, en la que cada persona podría subir publicaciones o posts y relacionarlos con los eventos a los que se ha asistido. Junto con estas mejoras, lo que mejor funcionaría es una sección de comentarios en los posts o publicaciones y un chat y así poder hablar con otros usuarios a través de la aplicación.

### 3.3 Conclusiones

El objetivo que tenía realizar este TFG era completar un proyecto personal nacido de una experiencia de erasmus en Italia, en la que se echaba mucho en falta un servicio como el que ofrecen estas aplicaciones. De esa manera se podría realizar una gestión mucho menos caótica y de forma estructurada y segura, evitando impagos de entradas o fallos al realizar los cobros.

A partir de esa idea se desarrolló lo que ahora mismo son las dos aplicaciones: la de usuarios y la de organizadores. Este proyecto no está terminado ni mucho menos. Todavía faltan muchas ideas por implementar que harán del trabajo una experiencia mucho mejor.

En lo que respecta personalmente, este proyecto me ha ayudado a darme cuenta de que puedo conseguir cualquier meta que me proponga sin importar las limitaciones. Lo más importante es la motivación. Cuando empecé con el proyecto, no sabía cómo hacer la gran mayoría de tareas necesarias para poder llevarlo a cabo, pero poco a poco he ido aprendiendo por mi cuenta algo que en un principio parecía impensable.

He aprendido tecnologías muy importantes en el mundo laboral y que, gracias a ello, a día de hoy estoy trabajando en una empresa que utiliza exactamente las mismas tecnologías que yo he utilizado para este TFG.

## 4 BIBLIOGRAFÍA

---

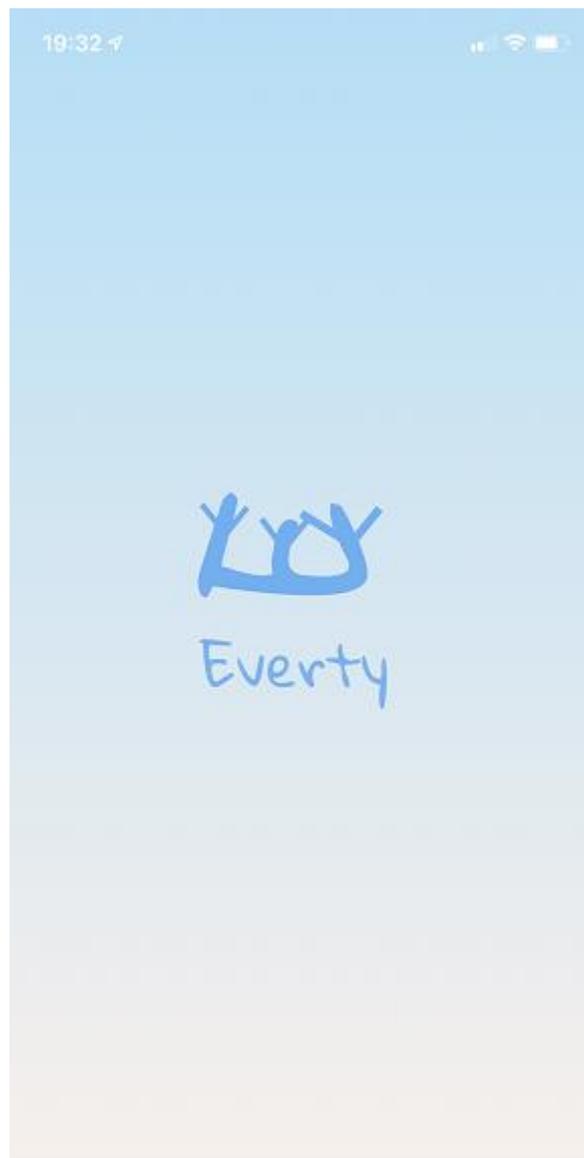
- [1] [http://www.injuve.es/sites/default/files/adjuntos/2021/03/informe\\_juventud\\_es\\_pana\\_2020.pdf](http://www.injuve.es/sites/default/files/adjuntos/2021/03/informe_juventud_es_pana_2020.pdf)
- [2] [https://www.ine.es/ss/Satellite?L=es\\_ES&c=INECifrasINE\\_C&cid=1259925107326&p=1254735116567&pagename=ProductosYServicios%2FPYSLayout](https://www.ine.es/ss/Satellite?L=es_ES&c=INECifrasINE_C&cid=1259925107326&p=1254735116567&pagename=ProductosYServicios%2FPYSLayout)

## 5 ANEXO I. MANUAL DE USUARIO APLICACIÓN DE USUARIOS

---

A continuación, se va a detallar el funcionamiento de la aplicación de usuarios y qué acciones se pueden realizar sobre las diferentes pantallas.

Nada más abrir la aplicación aparece una pantalla de carga que se muestra mientras está cargando la aplicación



*Figura 15: Pantalla de carga de la aplicación*

Tras terminar la carga, la aplicación navega a la pantalla de inicio de sesión en la que aparecen dos campos "Email" y "Password" que se han de rellenar para acceder a la aplicación. También está disponible una opción de recuperación de contraseña en caso

de que esta haya sido olvidada. En la situación en la que el usuario entra por primera vez a la aplicación deberá acceder a la sección de “Register” para crear una cuenta nueva.

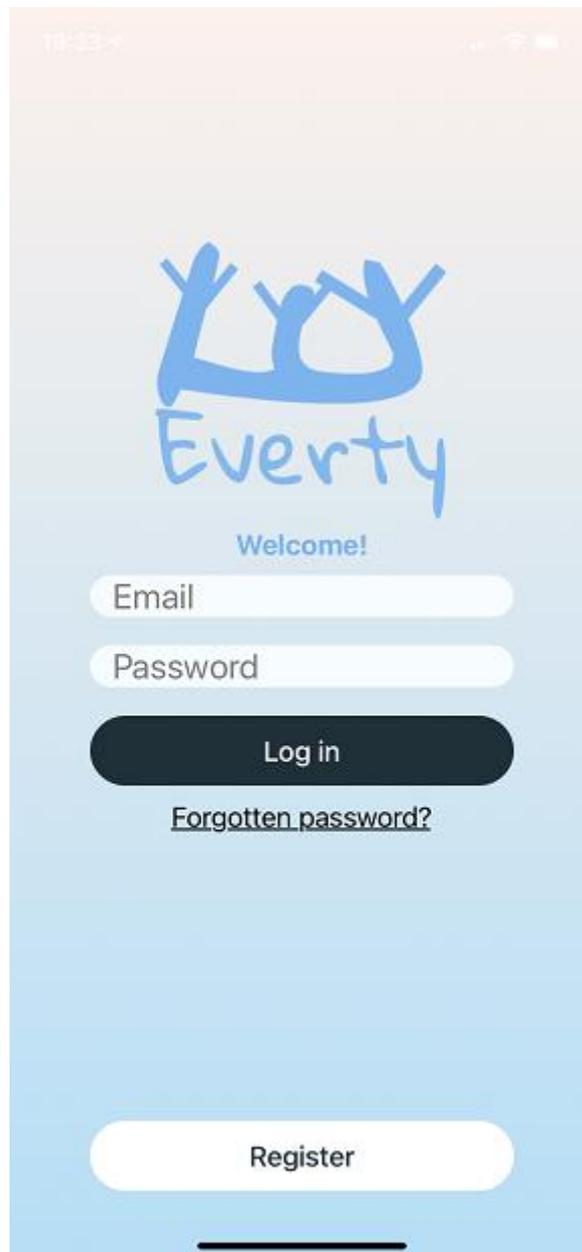


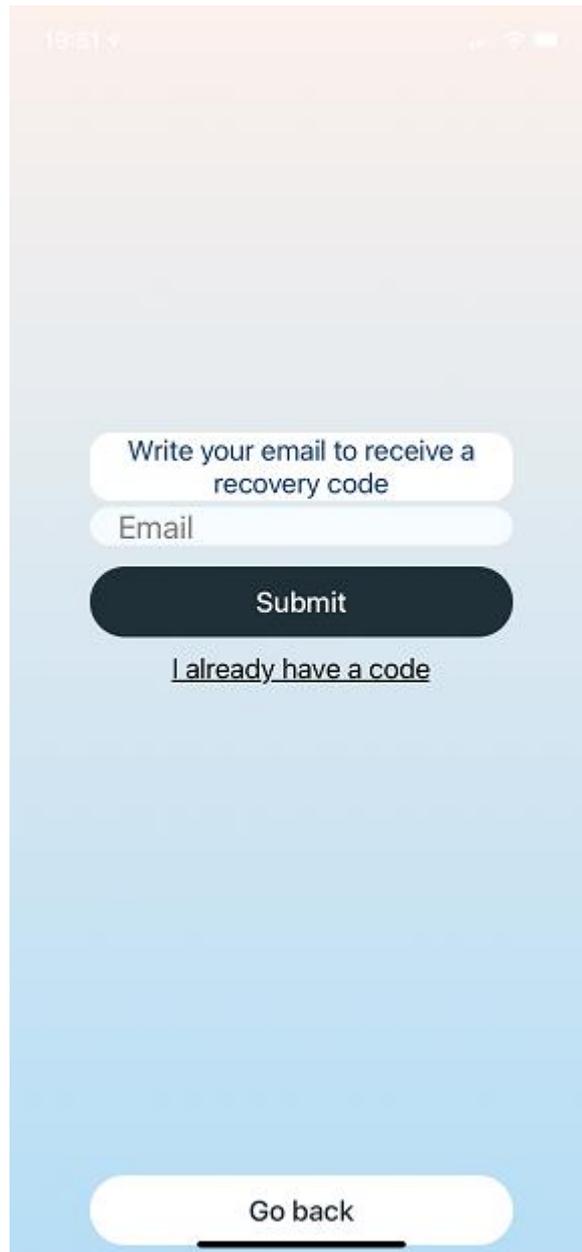
Figura 16: Pantalla de inicio de sesión

En la pantalla de registro, se han de rellenar algunos campos para crear una nueva cuenta: “email”, “password”, “alias” y “phone number”. Se pide repetir la contraseña para evitar problemas de escritura ya que esta aparece con un formato seguro en el que no se puede leer lo que se ha escrito. Además, es necesario aceptar los términos y condiciones del servicio y la política de privacidad para poder registrarse. En caso de que alguno de los campos no se complete o de que no se acepten los términos y condiciones, la aplicación mostrará una alerta por pantalla para informar al usuario. Para completar

el registro se deberá pulsar en el botón de “Register”. Para cancelar el registro y volver a la pantalla de inicio de sesión, se deberá pulsar el botón de “Login”

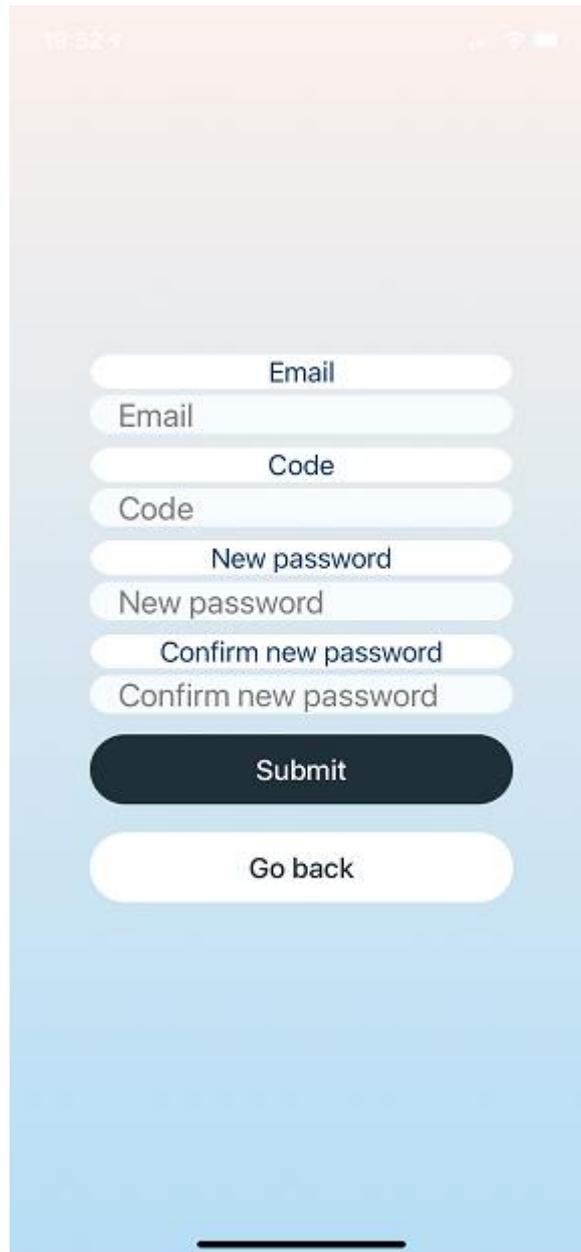
*Figura 17: Pantalla de registro*

Si en la pantalla de inicio de sesión se pulsa sobre “Forgotten password?” la aplicación redirigirá a una pantalla en la que se solicitará al usuario rellenar el campo de “Email” para proceder a enviarle un email con un código que le servirá para reestablecer la contraseña. Si se desea cancelar la acción se deberá pulsar el botón de “Go back”. Si se desea enviar la solicitud, se deberá pulsar en el botón de “Submit”. Al pulsar sobre este botón, el sistema mostrará una alerta en la que informará al usuario de que, en caso de existir una cuenta asociada al email escrito, se habrá enviado un correo electrónico con un código de recuperación.



*Figura 18: Pantalla de contraseña olvidada*

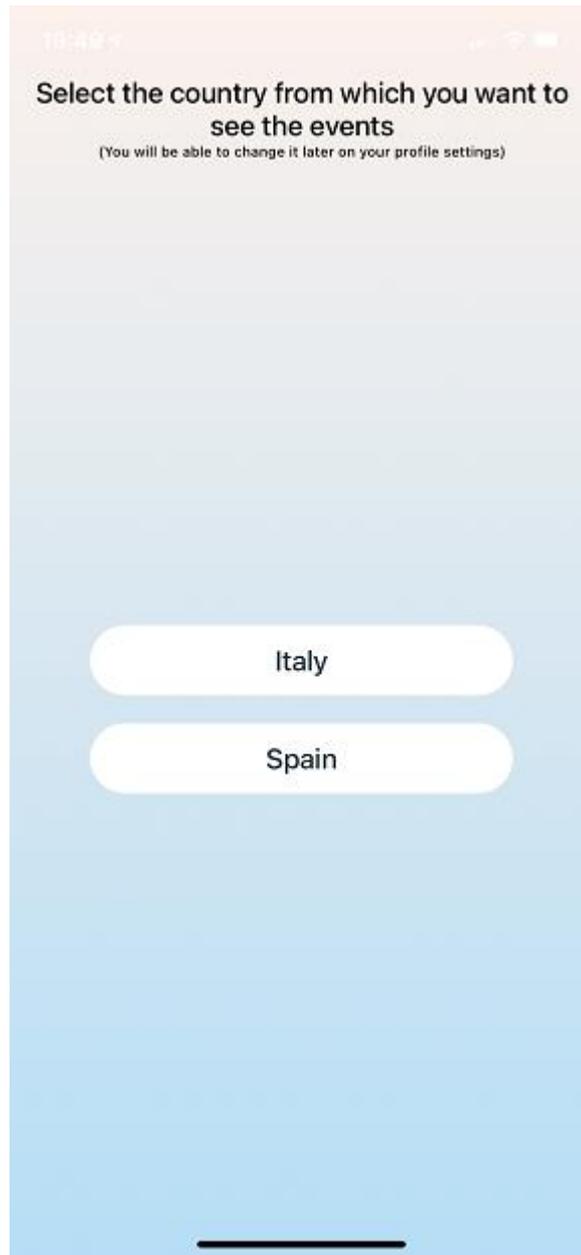
Al pulsar sobre el botón de "I already have a code" la aplicación navegará hasta la pantalla de restablecimiento de contraseña. En esta se le pedirá al usuario completar los campos de "email", "code" y "new password". Al igual que en el registro, la contraseña se deberá de escribir dos veces para evitar errores tipográficos.



*Figura 19: Pantalla de nueva contraseña*

Para guardar los cambios el usuario deberá pulsar en el botón “Submit”, mientras que para cancelarlos deberá pulsar en el botón “Go back”. Después de guardar los cambios, el usuario deberá iniciar sesión de forma normal en la pantalla de inicio de sesión utilizando su nueva contraseña.

Desde la pantalla de registro, al pulsar “Register” con unos datos correctos, la aplicación navegará hasta la sección de elección de localización, que consta de dos pantallas.



*Figura 20: Pantalla de elección de país*

En la primera se solicitará al usuario elegir el país del que quiere que se le muestren los eventos disponibles. También se le avisará de que podrá cambiar estos datos más adelante. Al pulsar sobre uno de los países disponibles se navegará hasta una segunda pantalla en la que le aparecerán las localidades disponibles dentro del país seleccionado.



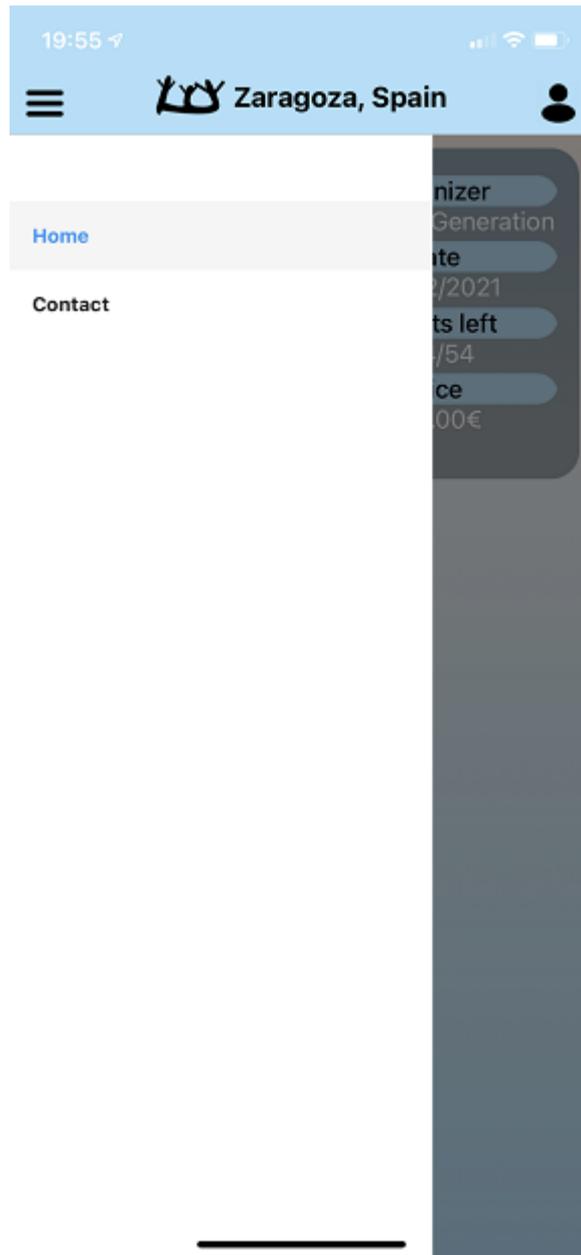
*Figura 21: Pantalla de elección de ciudad*

Al pulsar sobre la localización deseada, la aplicación navegará hasta la pantalla de inicio de la aplicación. Este comportamiento será el mismo que tendrá cuando un usuario pulse en el botón de "Log in" en la pantalla de inicio de sesión.



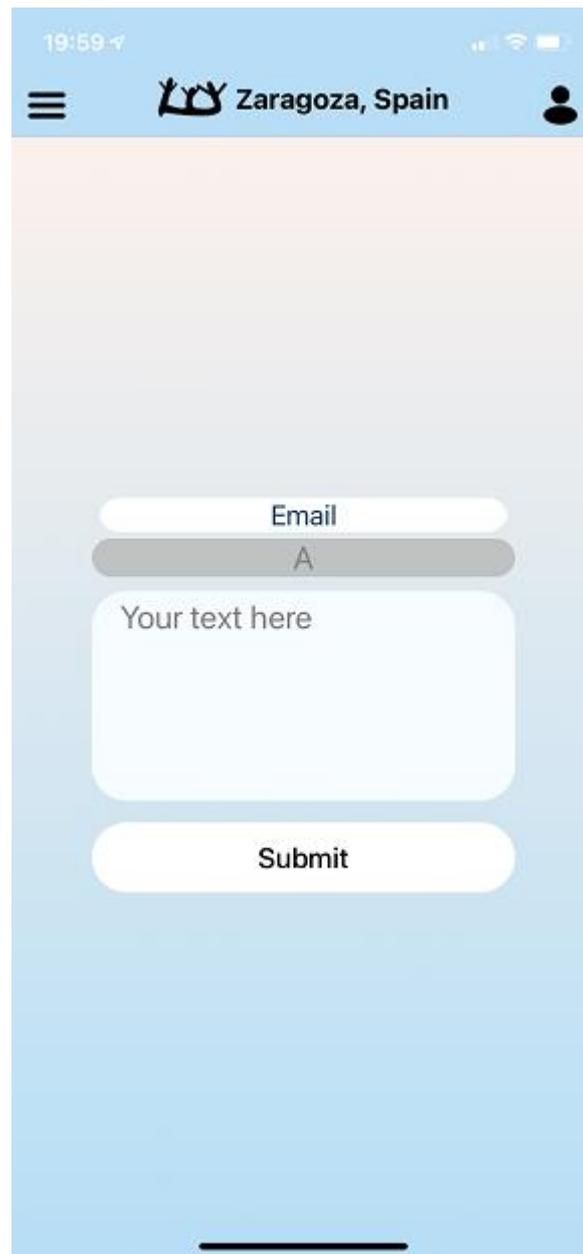
Figura 22: Pantalla de inicio

En esta pantalla se podrán visualizar en forma de lista los diferentes eventos disponibles en la localización seleccionada. Además, en la barra de navegación superior hay tres botones que tienen diferentes funcionalidades. Al pulsar sobre el botón superior izquierdo, se abrirá un menú lateral con dos opciones: “Home” y “Contact”.



*Figura 23: Pantalla de barra lateral*

Al pulsar sobre “Home” la aplicación navegará a la pantalla de inicio independientemente de la pantalla sobre la que se encuentre el usuario. Si se pulsa sobre “Contact”, la aplicación redirigirá a una pantalla de ayuda en la que se podrá enviar un mensaje a los propietarios de la aplicación para cualquier motivo que concierna al usuario.



*Figura 24: Pantalla de contacto*

Para enviar el formulario de contacto, será necesario rellenar el campo del mensaje. Si esto no se cumple, el pulsar sobre el botón “Submit” la aplicación mostrará una alerta avisando de que se ha de rellenar el campo para poder continuar. Si se ha escrito el mensaje deseado, este botón avisará con una alerta del estado de la petición, tanto si el mensaje se ha enviado correctamente como si ha sucedido una excepción y no se ha podido enviar.

Volviendo a la pantalla de inicio, encontramos en la barra de navegación superior una sección en la que aparece el logo de la aplicación seguido de la localización configurada por el usuario. Al pulsar sobre esto, la aplicación redirigirá al usuario a las pantallas de

selección de localización para poder cambiar la localización deseada. Tras terminar, devolverá al usuario a la pantalla de inicio de la aplicación.

Por último, en la parte superior derecha, al pulsar sobre el botón del perfil la aplicación navegará hasta el perfil de un usuario en que se podrán observar sus datos y varias acciones a realizar.

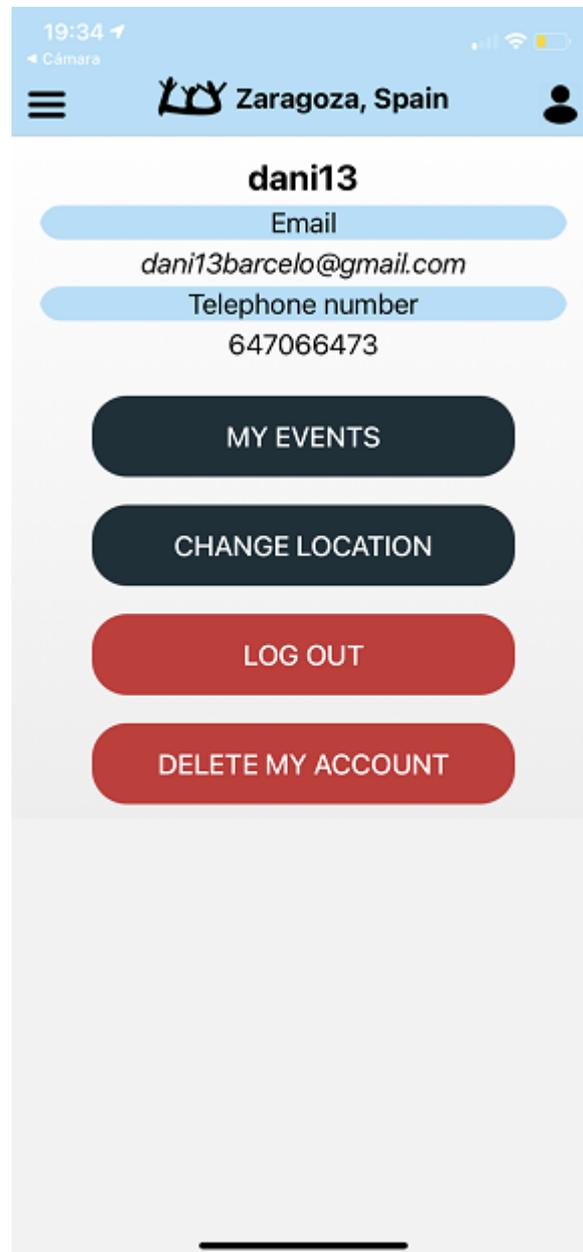


Figura 25: Pantalla de perfil

En esta pantalla se podrá observar el alias del usuario, su email y su número de teléfono. Además será posible acceder a la sección de mis eventos a través del botón "My events". El usuario también podrá cambiar la localización deseada de la aplicación a través del botón "Change location", que tiene un funcionamiento igual que el botón central superior de la barra de navegación que se ha mencionado anteriormente.

También podrá cerrar sesión a través del botón de “Log out” y podrá borrar su cuenta con el botón “Delete my account”. Al pulsar sobre este último botón, la aplicación mostrará una alerta al usuario de que el proceso de borrado de la cuenta es irreversible, en caso de que alguien lo pulse por error. En caso de confirmar el borrado de la cuenta, se comprobará que el usuario no tenga eventos pagados pendientes de ocurrir. En ese caso, la aplicación mostrará una alerta por pantalla avisando de que la cuenta no puede ser borrada por el motivo mencionado. En caso contrario de que no se tenga ningún evento pendiente, la cuenta se borrará y se cerrará la sesión, redirigiendo a la pantalla de inicio de sesión. Al pulsar sobre el botón “My events” el usuario será redirigido a la pantalla de sus eventos comprados.



Figura 26: Pantalla de mis eventos

En esta pantalla se podrán ver en forma de lista los eventos que han sido pagados. Para volver a la pantalla de perfil hay que pulsar en el botón "Go back". Al pulsar sobre alguno de los eventos, aparecerá una alerta sobre la pantalla informando del número de entradas compradas para ese evento. La alerta tendrá dos opciones: "Ok" y "View QR". Si se pulsa sobre "Ok", la alerta se cerrará y no ocurrirá ninguna acción. Si se pulsa sobre "View QR", la aplicación navegará a otra pantalla en la que se mostrará el código QR asociado a esa compra.



*Figura 27: Pantalla de código QR*

En esta pantalla se podrá visualizar el código QR para su posterior escaneo en la aplicación de los organizadores para permitir el acceso al evento. También aparece un campo en el que se informa del número de entradas restantes que quedan asociadas a

ese código QR. Al pulsar sobre el botón “Go back” la aplicación navegará hasta la pantalla de “My events” para poder seguir visualizándolos.

Volviendo a la pantalla de inicio, aparecen en la pantalla una lista de eventos de la localización seleccionada junto a un breve resumen de lo que consiste el evento. Concretamente aparece la información del organizador, la fecha, el número de entradas restantes y el precio del evento. También aparece el título del evento y una imagen asociada. De manera opcional, puede aparecer una imagen de fondo semi-opaca, que en caso de no ser especificada aparecerá una predeterminada con el logo de la aplicación. Al pulsar sobre uno de los eventos disponibles, la aplicación redirigirá a la pantalla de visualización de un evento.

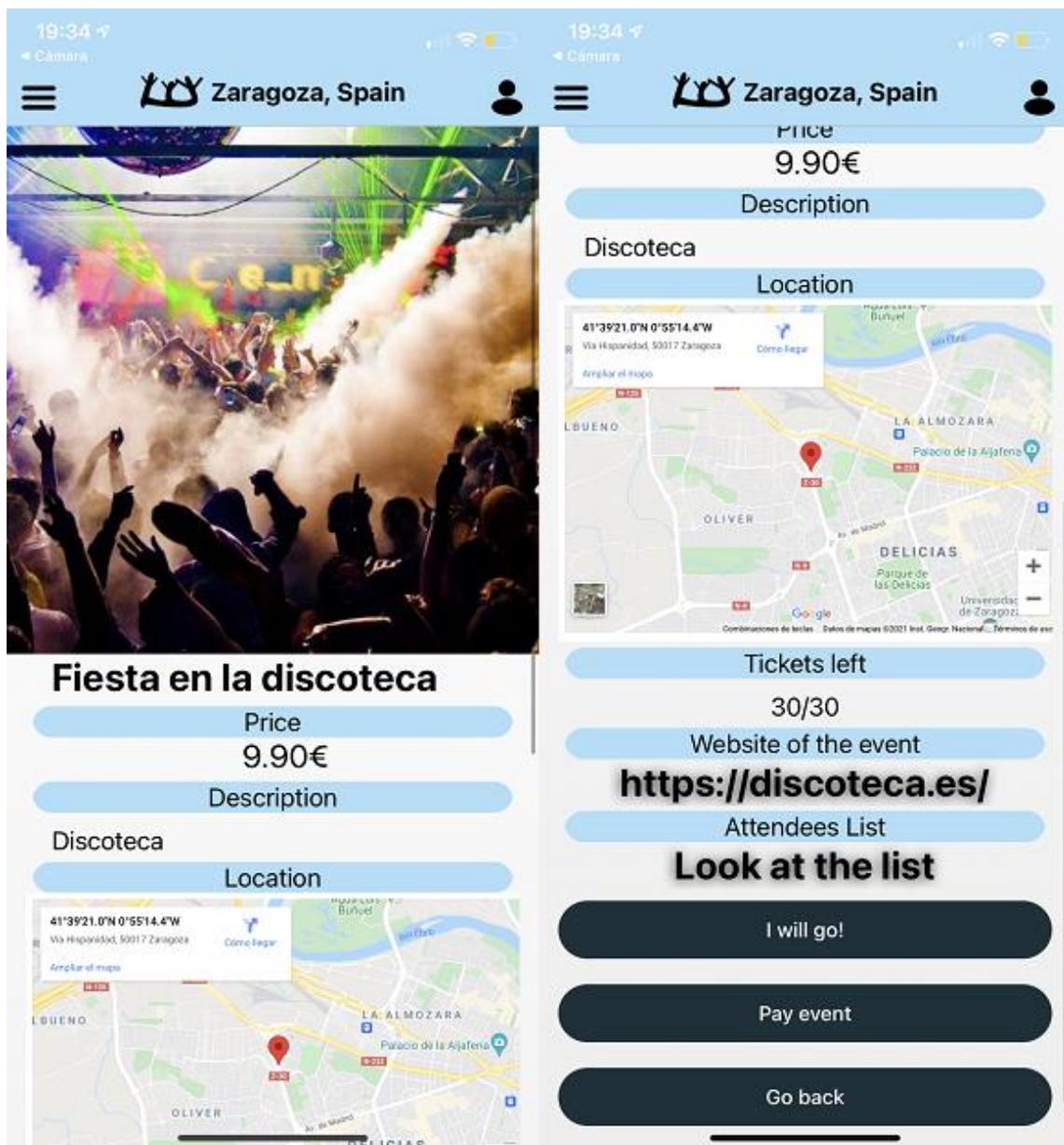
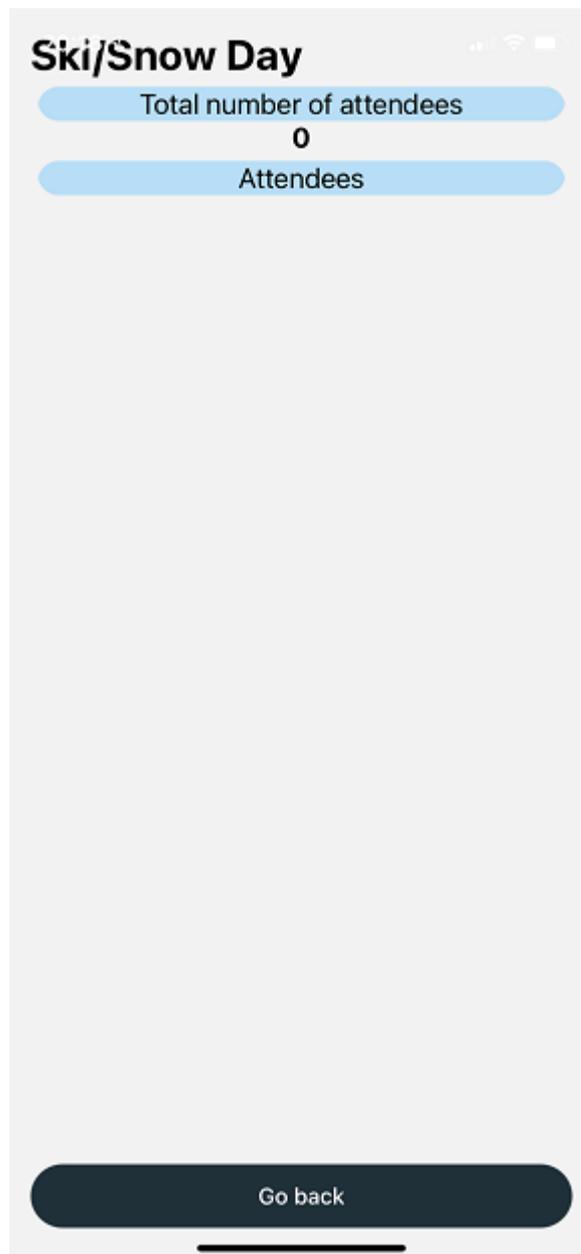


Figura 28: Pantalla de visualización de un evento

En esta pantalla se podrá ver de forma más detallada la información del evento. En la parte superior de la pantalla aparecerá la imagen del evento. Seguido de esto aparecerán en orden el título del evento, el precio, la descripción, la localización, el número de entradas restantes y un link a la página web del evento. Este link se podrá pulsar para copiarlo al portapapeles para navegar a la web a través de un navegador web.

A continuación aparecerá la sección de asistentes a un evento. Al pulsar sobre el botón "Look at the list" se redirigirá al usuario a una pantalla en la que podrá ver la lista de personas que van a asistir al evento.



*Figura 29: Pantalla de asistentes*

En esta pantalla no se puede realizar ninguna acción excepto refrescar la búsqueda arrastrando desde la parte superior hacia abajo. Para volver se ha de pulsar sobre el botón de “Go back”, que devolverá a la pantalla de visualización del evento.

Cuando se pulse sobre el botón “I will go!” aparecerá en la pantalla una alerta en la que informa al usuario de lo que va a suceder. El alias del usuario aparecerá en la lista de asistentes al evento, pero se le avisa de que eso no es una entrada. Para poder acceder al evento será necesario que pague la entrada.

Al pulsar sobre el botón “Pay event” la aplicación navegará hasta la pantalla de selección de entradas para el evento.



Figura 30: Pantalla de selección de entradas

En esta pantalla, se podrán seleccionar hasta un máximo de diez entradas por cada compra para el evento seleccionado. También aparecerá el precio total de las entradas. En caso de que el número de entradas disponibles sea menor a diez, se informará al usuario de la situación. Al pulsar sobre el botón "Next" se navegará hasta la pantalla de introducción de datos de la compra.



Figura 31: Pantalla de introducción de datos de la compra

Antes de permitir al usuario acceder a esta pantalla, se mostrará una alerta por pantalla informando de que los datos que se van a introducir en esta pantalla deben de ser correctos, puesto que serán los que se utilicen para verificar la entrada al evento.

Por cada entrada seleccionada, aparecerá en la pantalla un nuevo campo que será necesario rellenar con el alias de las personas que vayan a asistir al evento.

También se tendrá que rellenar el campo de la tarjeta de crédito con unos valores correctos para una tarjeta de crédito o débito válida. Al pulsar sobre el botón “Pay” se comenzará a realizar la transacción. Tanto si la transacción es correcta como si es errónea, se avisará al usuario a través de una alerta sobre la pantalla.

## 6 ANEXO II. MANUAL DE USUARIO APLICACIÓN DE ORGANIZADORES

---

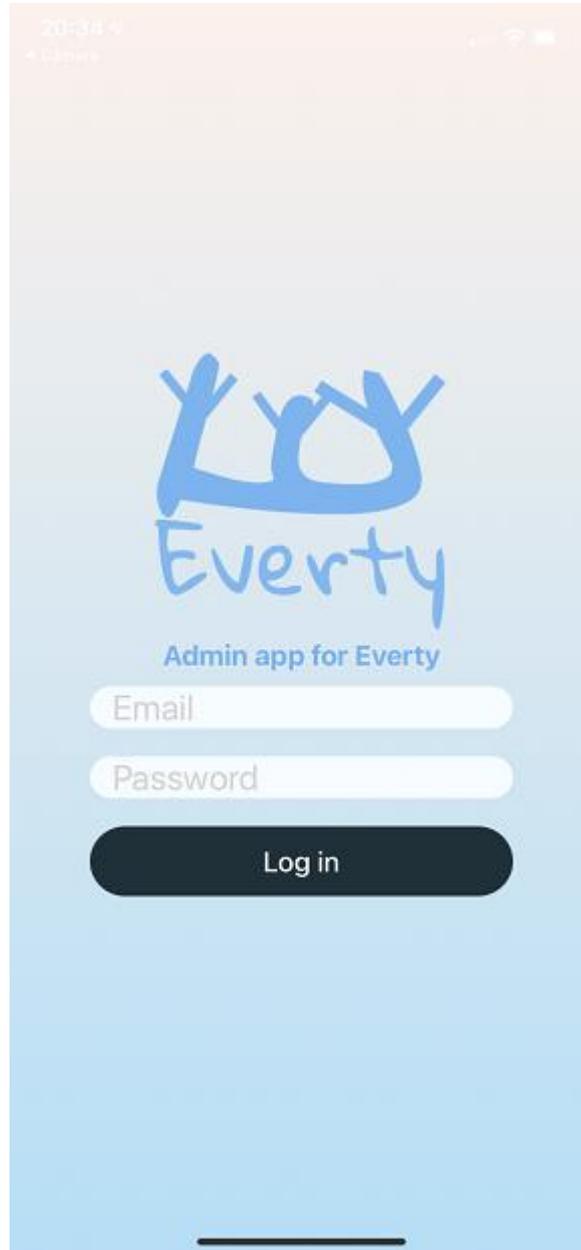
A continuación, se va a detallar el funcionamiento de la aplicación de usuarios y qué acciones se pueden realizar sobre las diferentes pantallas.

Nada más abrir la aplicación aparece una pantalla de carga que se muestra mientras está cargando la aplicación.



*Figura 32: Pantalla de carga de la aplicación*

Tras terminar la carga, la aplicación navega a la pantalla de inicio de sesión en la que aparecen dos campos “Email” y “Password” que se han de rellenar para acceder a la aplicación.

*Figura 33: Pantalla de inicio de sesión*

Al pulsar sobre el botón “Log in” la aplicación navegará hasta la pantalla de inicio de la aplicación.

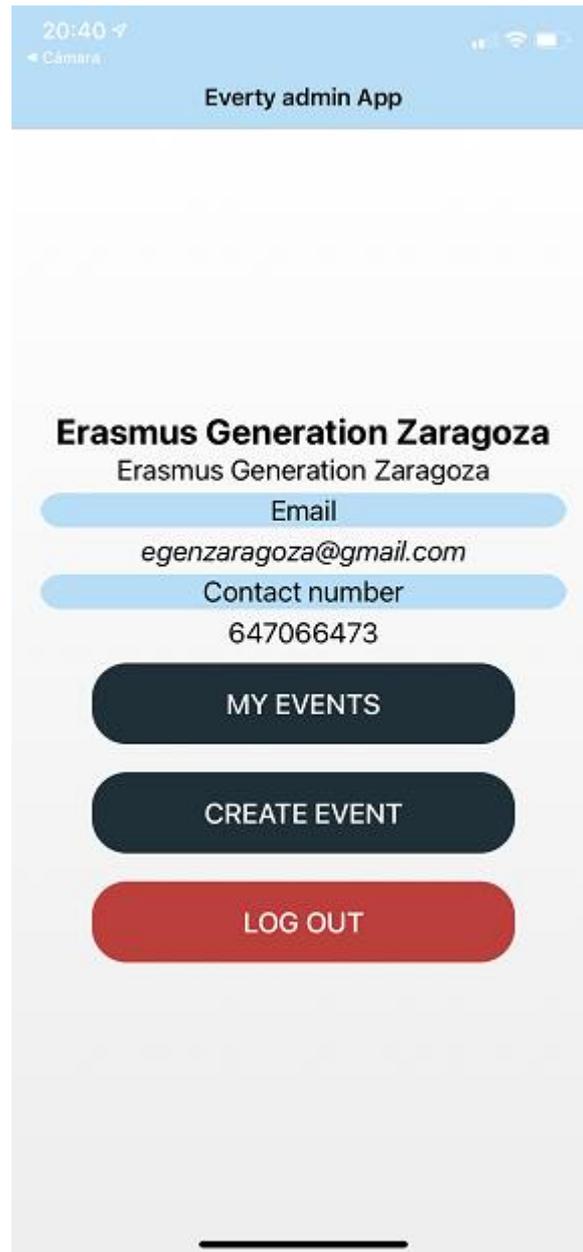


Figura 34: Pantalla de inicio

En esta pantalla se podrá visualizar el nombre de la asociación, el email asociado a la cuenta y el número de contacto. Desde esta pantalla, se podrán ver los eventos creados pulsando el botón de “My events”; se podrá crear un evento a través del botón de “Create event” y se podrá cerrar sesión a través del botón “Log out”.

Al pulsar sobre el botón de “Create event” la aplicación navegará hasta la pantalla correspondiente en la que se podrá crear un nuevo evento.

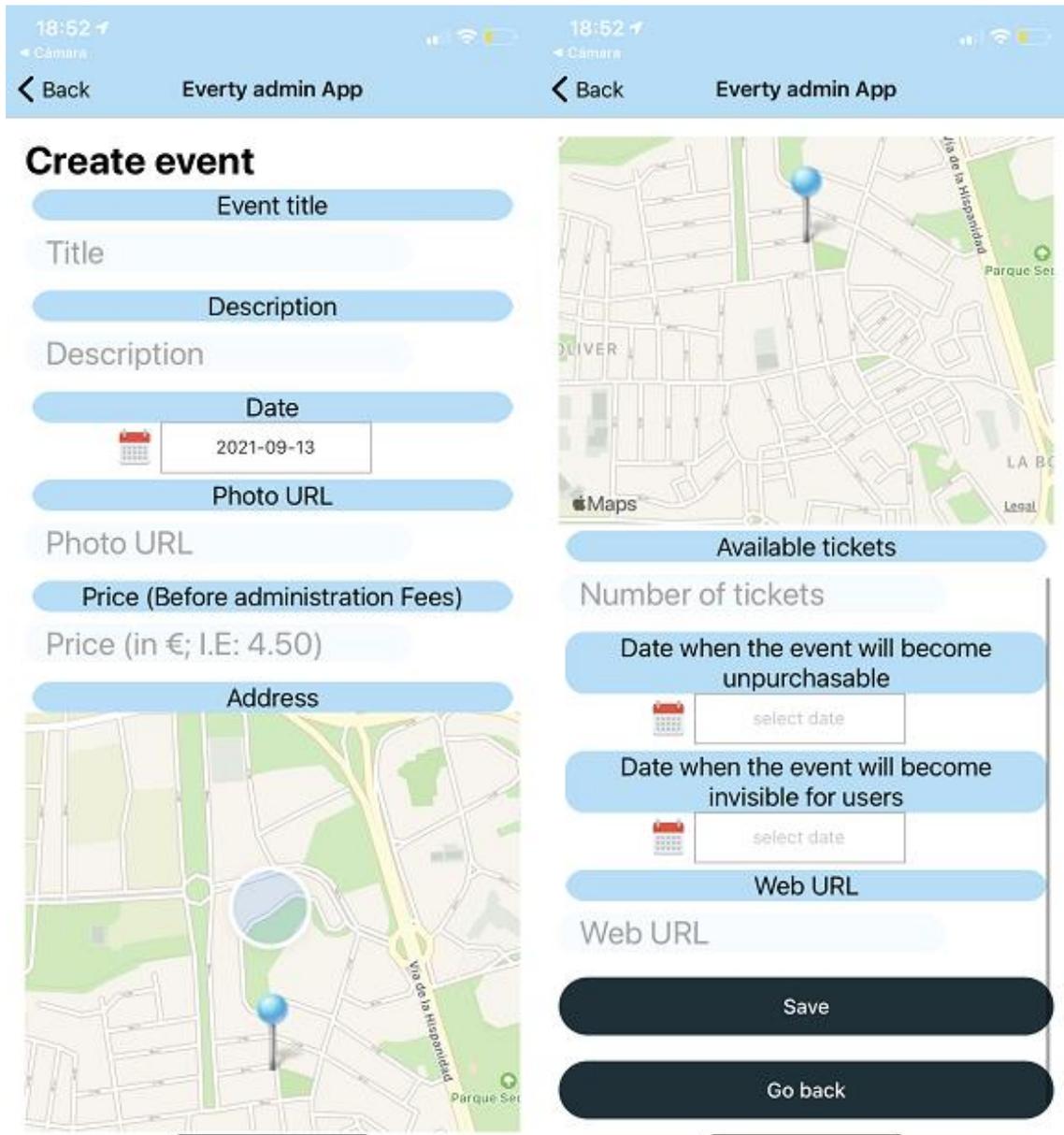


Figura 35: Pantalla creación de un nuevo evento

En esta pantalla se pedirán rellenar los siguientes campos para poder crear un evento: título del evento, descripción, fecha del evento, foto, precio, dirección, entradas máximas disponibles, fecha de imposibilitación de compra de entradas para el evento, fecha de ocultación del evento para los usuarios en el feed de eventos y la url de la web del evento. Si todos estos campos se han rellenado correctamente, al pulsar sobre el botón “Save” se informará al usuario de que el evento se ha creado con éxito a través de una alerta y se volverá a la pantalla de inicio de la aplicación. Si por el contrario falta por rellenar algún campo, o alguno de ellos contiene información errónea; o si la creación falla por cualquier motivo, se avisará al usuario de esto a través de una alerta sobre la pantalla.

Para cancelar la creación del evento se tendrá que pulsar el botón de “Go back” que devolverá al usuario a la pantalla de inicio.

Desde la pantalla de inicio, si se pulsa sobre “My events” la aplicación navegará hasta la pantalla de visualización de los eventos creados por el organizador.



Figura 36: Pantalla de eventos

Para volver a la pantalla de inicio bastará con pulsar sobre el botón de “Go back”.

Los eventos en esta pantalla aparecen en forma de lista. En cada uno de ellos aparece un breve resumen de lo que consiste el evento. Concretamente aparece la información del organizador, la fecha, el número de entradas restantes y el precio del evento. También aparece el título del evento y una imagen asociada. De manera opcional, puede

aparecer una imagen de fondo semi-opaca, que en caso de no ser especificada aparecerá una predeterminada con el logo de la aplicación.

Al pulsar sobre uno de estos eventos, se navegará hasta la pantalla de visualización del evento pulsado.

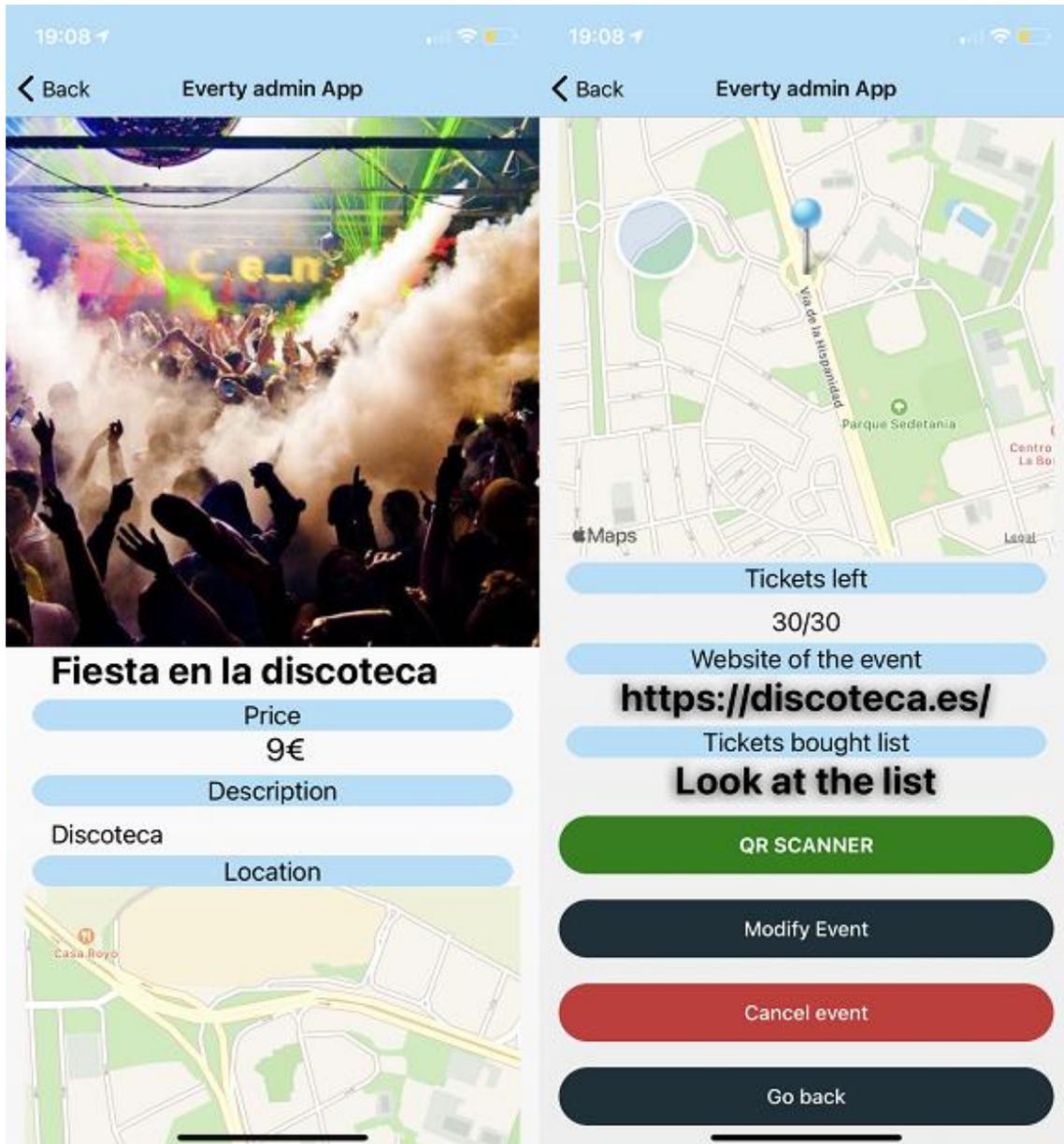
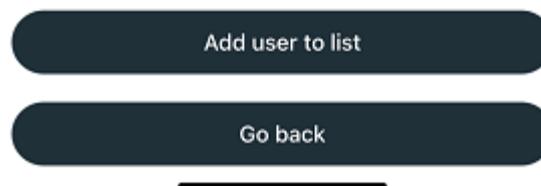


Figura 37: Pantalla de visualización de un evento

En esta pantalla se podrá ver de forma más detallada la información del evento. En la parte superior de la pantalla aparecerá la imagen del evento. Seguido de esto aparecerán en orden el título del evento, el precio, la descripción, la localización, el número de entradas restantes y un link a la página web del evento. Este link se podrá pulsar para copiarlo al portapapeles para navegar a la web a través de un navegador web.

A continuación aparecerá la sección de asistentes a un evento. Al pulsar sobre el botón “Look at the list” se redirigirá al usuario a una pantalla en la que podrá ver la lista de personas que han pagado una entrada del evento.



*Figura 38: Pantalla de asistentes*

En esta pantalla se podrán visualizar la lista de personas que han pagado una entrada. Pulsando sobre el alias de una de ellas se permite la eliminación de la persona pulsada de la lista a través de una alerta. En caso de confirmar la eliminación, la persona dejará de aparecer en la lista.

También se podrán añadir personas a esta lista de manera manual pulsando sobre el botón de “Add user to list”. Al pulsar sobre este botón, se navegará a otra pantalla para realizar la acción seleccionada.



---

*Figura 39: Pantalla de añadir asistente*

En esta pantalla, se pedirá rellenar un formulario con el alias del asistente que se desea añadir a la lista. Para guardar será necesario pulsar sobre el botón "Save". Si se desea cancelar la acción, hay que pulsar el botón de "Go back" que devolverá al usuario a la pantalla de la lista de asistentes pagados.

Si se vuelve a la pantalla de visualización del evento, la siguiente opción que se puede pulsar es el botón de "QR SCANNER" que navegará hasta una nueva pantalla.



*Figura 40: Pantalla de escaneo de códigos QR*

En esta pantalla, la primera vez que se acceda en el dispositivo móvil, solicitará al usuario permisos para utilizar la cámara del móvil. Tras aceptarlos, aparecerá en la pantalla el contenido enfocado por la cámara. Al colocar un código QR asociado a una entrada a un evento en la cámara, este se reconocerá automáticamente y se mostrará una alerta por pantalla avisando de que el código se ha escaneado correctamente. Para volver a escanear se deberá pulsar un nuevo botón que aparecerá para poder escanear de nuevo. Si el código QR escaneado no tiene nada que ver con una entrada de un evento, o si el código QR pertenece a un evento el sistema mostrará una alerta informando del error.

Para salir de la pantalla será necesario pulsar en el botón superior izquierdo “Back” de la barra de navegación de la aplicación.

Si desde la pantalla de visualización del evento se pulsa el botón de “Modify Event”, la aplicación navegará hasta una pantalla similar a la pantalla de creación de un evento, pero con los formularios completos con la información actual del evento seleccionado.

The screenshot displays the 'Modify event' interface. At the top, there's a navigation bar with a back arrow and the text 'Everyt admin App'. The main content area is divided into two columns. The left column contains form fields for:
 

- Event title:** Fiesta en la discoteca
- Description:** Discoteca
- Date:** 2021-10-31
- Photo URL:** https://fotos02.noticias...
- Price (Before administration Fees):** 9
- Address:** A map view showing a location in a residential area.

 The right column contains fields for:
 

- Maximum available tickets:** 30
- Current available tickets:** 30
- Date when the event will become unpurchasable:** 2021-10-30
- Date when the event will become invisible for users:** 2021-11-01
- Web URL:** https://discoteca.es/

 At the bottom, there are two dark buttons: 'Save' and 'Go back'.

Figura 41: Pantalla de modificación de evento

Para guardar los cambios habrá que pulsar en el botón “Save”. Para cancelar la acción se tendrá que pulsar el botón “Go back”.

Por último, si se selecciona en la pantalla de visualización de un evento la opción de “Cancel event” aparecerá una alerta sobre la pantalla avisando de lo que sucederá: los usuarios que hayan pagado una entrada a un evento recibirán su dinero de manera

íntegra, por lo que a los organizadores se les cargará el precio total de la entrada y no solo la parte del dinero que hayan recibido después de las comisiones.