

---

# Aplicación móvil para la gestión de carreras deportivas

---

TÍTULACIÓN:

Grado en Ingeniería Informática

AUTOR:

Luz Marina Pulido Santiago

TUTOR:

Francisco Alexis Quesada Arencibia

CO-TUTOR:

Orlando Montesdeoca Santana

Julio 2022

# Agradecimientos

A mi familia, por apoyarme y motivarme a seguir adelante en esos momentos difíciles cuando menos tenía ganas.

A mis amigos, que me acompañaron, ayudaron y motivaron durante estos años.

A mi tutor Alexis Quesada Arencibia,  
por guiarme y ayudarme  
en este proyecto.

# Resumen

La gestión de la organización de las carreras deportivas conlleva un trabajo considerable por parte de las personas encargadas de la organización de las mismas. Por este motivo, además del personal responsable de la organización, estos eventos suelen involucrar a un número importante de voluntarios de apoyo que ayudan en muchas de las tareas organizativas.

Con la idea de organizar a todo el personal involucrado en las diferentes tareas, en este proyecto se propone el desarrollo de una aplicación móvil multiplataforma que ayude a los organizadores a que el evento se pueda desarrollar con éxito. Entre las diferentes tareas que debe soportar la aplicación a desarrollar, deberá permitir la inscripción de voluntarios (individual o grupal), la asignación y seguimiento de tareas, envío de notificaciones, ...

# Abstract

The management of the organization of sports races entails a considerable job for manager people of the same organization. For this reason, besides responsible staff for the organization, these events usually involve an important number of supporting volunteers who help in many of the organizational tasks.

With the idea of organizing all the involved staff in different tasks, in this project it is proposed the development of a multiplatform mobile application that helps the organizers to carry out the event successfully. Among the different tasks that this application must support, it must allow the registration of volunteers (individual or group), the designation and monitoring of tasks, sending notifications, ...

# Índice

<b>Capítulo 1. Introducción.....</b>	<b>3</b>
1.1. Descripción general.....	3
1.2. Estado del arte .....	4
<b>Capítulo 2. Competencias específicas. ....</b>	<b>11</b>
2.1. IS01.....	11
2.2. TI03.....	11
2.3. TI06.....	11
<b>Capítulo 3. Aportaciones. ....</b>	<b>13</b>
<b>Capítulo 4. Normativa y legislación.....</b>	<b>14</b>
4.1. Licencias de software .....	14
4.2. Leyes.....	15
<b>Capítulo 5. Metodología y planificación del proyecto.....</b>	<b>16</b>
5.1. Metodología.....	16
5.2. Planificación del proyecto .....	17
<b>Capítulo 6. Herramientas y tecnologías utilizadas.....</b>	<b>20</b>
6.1. Herramientas de diseño .....	20
6.2. Herramientas para el desarrollo .....	20
6.2.1. Lenguajes y librerías .....	20
6.2.2. IDE y frameworks .....	20
6.3. Herramientas organizativas.....	22
<b>Capítulo 7. Análisis.....</b>	<b>23</b>
7.1. Roles .....	23
7.2. Casos de uso.....	23
7.3. Requisitos funcionales.....	25

7.4. Requisitos no funcionales.....	26
<b>Capítulo 8. Diseño.....</b>	<b>27</b>
8.1. Objetivos del diseño.....	27
8.2. Diseño de mockups.....	27
<b>Capítulo 9. Desarrollo.....</b>	<b>44</b>
9.1. Estructura del proyecto .....	44
9.2. Modelo de datos .....	45
9.2.1. Chats .....	46
9.2.2. Messages.....	46
9.2.3. Users .....	46
9.2.4. Events .....	47
9.2.5. Tasks .....	47
9.3. Firebase.....	48
9.3.1. Estructura de Storage y Realtime Database.....	48
9.4. Servicios .....	49
9.4.1. User service .....	49
9.4.2. Images service.....	50
9.4.3. Chat service.....	50
9.4.4. Event service.....	50
9.4.5. Task service.....	51
9.5. Principales componentes y funcionalidades.....	52
9.5.1. Iniciar sesión y registrar.....	52
9.5.2. Perfil del usuario.....	54
9.5.3. Inicio .....	54
9.5.4. Chats y chat.....	55
9.5.5. Evento .....	58

9.5.6.	Crear evento y tareas.....	59
9.5.7.	Listado de voluntarios .....	60
9.5.8.	Listado de organizadores .....	61
9.5.9.	Buscador de eventos.....	61
<b>Capítulo 10. Pruebas. ....</b>		<b>62</b>
<b>Capítulo 11. Resultados, conclusiones y futuras mejoras. ....</b>		<b>63</b>
11.1.	Resultados y conclusiones .....	63
11.2.	Futuras mejoras.....	64
<b>Capítulo 12. Bibliografía. ....</b>		<b>66</b>
<b>Anexos.....</b>		<b>70</b>
Anexo I: Especificación de casos de uso .....		70
Anexo II: Manual del usuario .....		77

# Índice de ilustraciones

Ilustración 1-1 Comprar una entrada un evento.....	5
Ilustración 1-2 Dashboard de EvenBrite.....	5
Ilustración 1-3 Editar un evento.....	6
Ilustración 1-4 Home de TeamHub.....	7
Ilustración 1-5 Calendario con los eventos próximos. ....	7
Ilustración 1-6 Home de RunnApp.....	8
Ilustración 1-7 Filtro de búsqueda.....	9
Ilustración 1-8 Eventos guardados.....	9
Ilustración 1-9 Información del evento. ....	10
Ilustración 5-1 Metodología SCRUM. ....	16
Ilustración 7-1 Diagrama de casos de uso. ....	24
Ilustración 8-1. Paleta de color de la aplicación.....	27
Ilustración 8-2. Mockups: Iniciar sesión.....	28
Ilustración 8-3. Mockups: Seleccionar quien va a registrarse.....	29
Ilustración 8-4. Mockups: Registrarse. ....	30
Ilustración 8-5. Mockups: Formulario para el responsable.....	31
Ilustración 8-6. Mockups: Inicio.....	32
Ilustración 8-7. Mockups: Evento. ....	33
Ilustración 8-8. Mockups: Confirmar inscripción en una tarea. ....	33
Ilustración 8-9. Mockups: Buscar un evento.....	34
Ilustración 8-10. Mockups: Perfil.....	35
Ilustración 8-11. Mockups: Chats.....	36
Ilustración 8-12. Mockups: Chat con una persona.....	37
Ilustración 8-13. Mockups: Nuevo chat indicando destinatario.....	38
Ilustración 8-14. Mockups: Menú del organizador.....	38
Ilustración 8-15. Mockups: Crear evento. ....	39
Ilustración 8-16. Mockups: Crear tareas. ....	40
Ilustración 8-17. Mockups: Listado de organizadores. ....	41
Ilustración 8-18. Mockups: Eventos del organizador. ....	42
Ilustración 8-19. Mockups: Voluntarios del evento.....	43

Ilustración 8-20 Mockups: Listado de voluntario de una tarea .....	43
Ilustración 9-1 Estructura del proyecto .....	44
Ilustración 9-2 Modelo de datos de la aplicación. ....	45
Ilustración 9-3 Estructura de la base de datos de Realtime Database. ....	48
Ilustración 9-4 Estructura de los componentes.....	52
Ilustración 9-5 Creación de la validación para poder registrar a un voluntario.....	53
Ilustración 9-6 Método que valida las contraseñas al registrarse.....	53
Ilustración 9-7 Mensaje de la validación al intentar registrarse el voluntario .....	54
Ilustración 9-8 Método que comprueba el rol del usuario .....	55
Ilustración 9-9 Método que cuenta el número de mensajes sin leer .....	55
Ilustración 9-10 Vista: Seleccionas usuario/s para crear un nuevo chat. ....	56
Ilustración 9-11 Verificación de chats.....	57
Ilustración 9-12 Cambiar el estado de los mensajes.....	58
Ilustración 9-13 Confirmación de inscripción y eliminación .....	59
Ilustración 9-14 Crear y guardar fichero excel.....	61

# **Estructura del documento.**

Este documento se ha estructurado en 12 capítulos y dos anexos. A continuación, iremos explicando los distintos capítulos que se han desarrollado a lo largo del mismo.

## **Capítulo 1. Introducción.**

En este capítulo describiremos el problema que procuraremos solucionar, los objetivos que pretendemos alcanzar y también analizaremos las herramientas que existen actualmente.

## **Capítulo 2. Competencias específicas.**

En este capítulo enumeraremos las competencias que se han cubierto durante del desarrollo de este proyecto.

## **Capítulo 3. Aportaciones.**

En este capítulo especificaremos las distintas aportaciones en el ámbito socio-económico y personal.

## **Capítulo 4. Normativa y legislación.**

En este capítulo presentaremos la normativa y legislación que tendrá este documento.

## **Capítulo 5. Metodología y planificación del proyecto.**

En este capítulo especificaremos que metodología se ha seguido y cual ha sido la planificación definida para este proyecto.

## **Capítulo 6. Herramientas y tecnologías utilizadas.**

En este capítulo mencionaremos y explicaremos qué herramientas y tecnologías van a utilizarse para el desarrollo del proyecto.

## **Capítulo 7. Análisis.**

En este capítulo analizaremos que requisitos del sistema son los que deberá tener la aplicación, qué actores participarán y describiremos el diagrama de casos de uso.

## **Capítulo 8. Diseño.**

En este capítulo expondremos los objetivos que debe tener el diseño y mostraremos un prototipo que refleje la idea que se ha planteado para esta herramienta.

## **Capítulo 9. Desarrollo.**

En este capítulo explicaremos como está estructurado el proyecto, los servicios que hemos utilizado de Firebase. Además, describiremos el modelo de datos donde podremos apreciar los datos que almacenaremos en la base de datos y, por último, presentaremos que funcionalidades que implementaremos.

## **Capítulo 10. Pruebas.**

En este capítulo señalaremos que pruebas se han realizado para garantizar el correcto funcionamiento del sistema y la interacción de los usuarios con la aplicación.

## **Capítulo 11. Resultado, conclusiones y futuras mejoras.**

En este capítulo comentaremos los resultados y las conclusiones obtenidas tras la fase de implementación. Asimismo, mencionaremos algunas de las mejoras y nuevas funcionalidades que podrían desarrollarse en el futuro.

## **Capítulo 12. Bibliografía.**

En este capítulo especificaremos las fuentes bibliográficas que se han empleado.

## **Anexo I: Especificaciones de casos de usos.**

En este anexo describiremos las especificaciones de casos de uso que muestran el comportamiento de la aplicación.

## **Anexo II: Manual del usuario.**

En este apartado explicaremos de forma sencilla y detallada cómo los usuarios pueden interactuar con la ella, además de mostrar las diferentes vistas y funcionalidades que posee.

# Capítulo 1. Introducción.

## 1.1. Descripción general

Las carreras deportivas son eventos mundialmente conocidos, donde grupos de personas participan con diferentes objetivos, que pueden ser: llegar a la meta en un cierto tiempo, superación, resistencia personal, velocidad, concienciar a la población, fines turísticos, recaudación de fondos para una asociación benéfica, entre otros.

La gestión de la organización de estos tipos de eventos no es nada sencilla. Es más, cada vez resulta más compleja y en ocasiones supone un gran desafío. De forma habitual, las personas que quieren realizar este tipo de eventos contratan a empresas profesionales que se encargan de la gestión de la organización de los mismos.

Dependiendo del tipo de carreras, los responsables de estos eventos se apoyan en los voluntarios que normalmente provienen de clubes deportivos y centro de formación (Institutos). Estas personas se encargan de las tareas organizativas que tienen previsto realizar en sus eventos y, entre ellas, podemos mencionar varias: la entrega de dorsales a corredores, señalización del circuito, la entrega de premios, la asistencia en el puesto de avituallamiento, el puesto de la salida y la llegada a la meta, etc.

Una de las actividades que desempeña el personal que se encarga de organizar estos eventos, es la gestión de voluntarios. Esto consiste en la búsqueda de personas que puedan realizar tareas el día del evento. Esta fase es un aspecto fundamental e imprescindible debido a la gran cantidad de actividades que son desempeñadas por ellos.

Otra de las funciones esenciales que realizan estos organizadores, consiste en la asignación e inscripción de dichas tareas. Esta gestión de inscripción es importante ya que una buena organización permite ofrecer un evento de calidad y sin complicaciones.

Dentro de todas las actividades que este personal realiza, hay que mencionar que muchas de ellas son bastante tediosas y laboriosas. De ahí que, se pretenda proporcionar una solución eficaz que ayude a gestionar esas funcionalidades críticas que realizan.

Finalmente, se ha considerado la idea de desarrollar una herramienta que ayude a mejorar la gestión de estas funciones básicas y necesarias, que consistirán en la gestión de voluntarios, la asignación y el seguimiento de tareas, la coordinación entre los organizadores y todos los voluntarios, entre otros.

## 1.2. Estado del arte

Para dar solución al problema descrito existen diferentes herramientas en el mercado que abarcan algunas de las características tratadas anteriormente. Como tal, no hay una aplicación que se enfoque en los temas que hemos abordado, pero hay varias que se aproximan a lo que queremos desarrollar en este proyecto. Estas son:

1. **EvenBrite** <sup>[1]</sup>. Esta aplicación móvil multiplataforma es una de las más famosas debido a la gestión de todo tipo de eventos. Sus características principales son:
  - Crear eventos.
  - Venta de entradas.
  - Modificar los detalles del evento. (tipo, descripción, cantidad, etc.)
  - Editar la lista de eventos que se organiza.
  - Escanear las entradas con la cámara del móvil.
  - Resolver las solicitudes de los clientes.
  - Los datos en tiempo real.

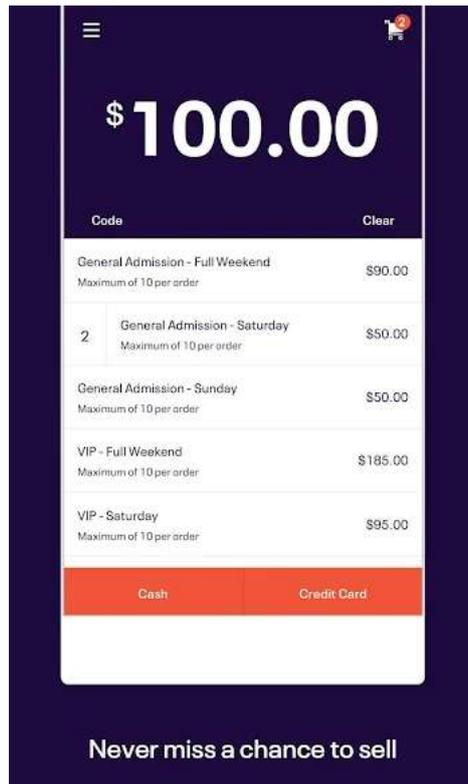


Ilustración 1-1 Comprar una entrada un evento.

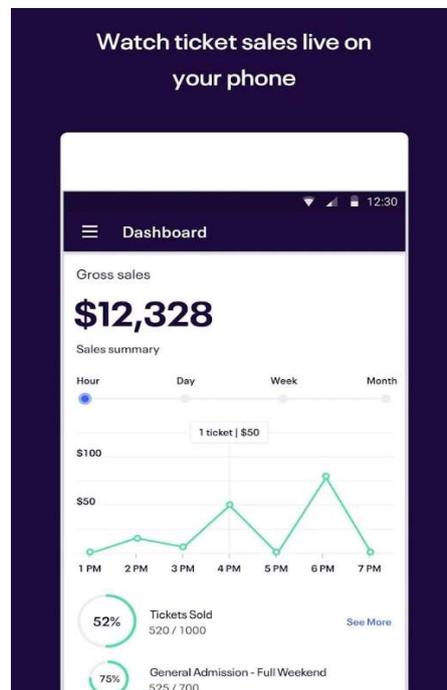


Ilustración 1-2 Dashboard de EvenBrite.

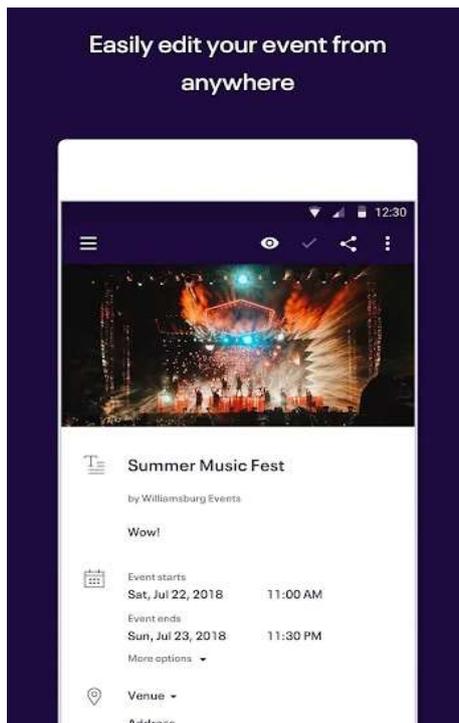


Ilustración 1-3 Editar un evento.

2. **TeamHub** <sup>[2]</sup>. Esta herramienta consiste en la gestión de equipos deportivos, como son beisbol, pádel, futbol, etc. Destaca por varias características:
- Control de asistencia al evento.
  - Comunicación con todo el equipo.
  - Calendario con la fecha del evento (partido).
  - Gestión de miembros.
  - Genera automáticamente estadísticas.

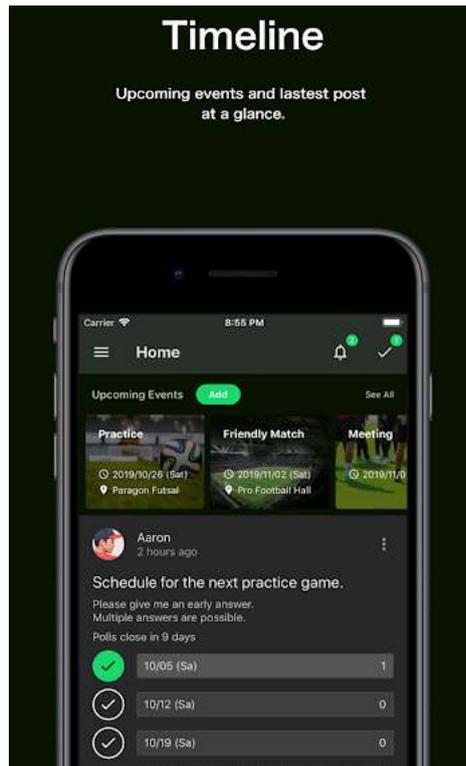


Ilustración 1-4 Home de TeamHub.

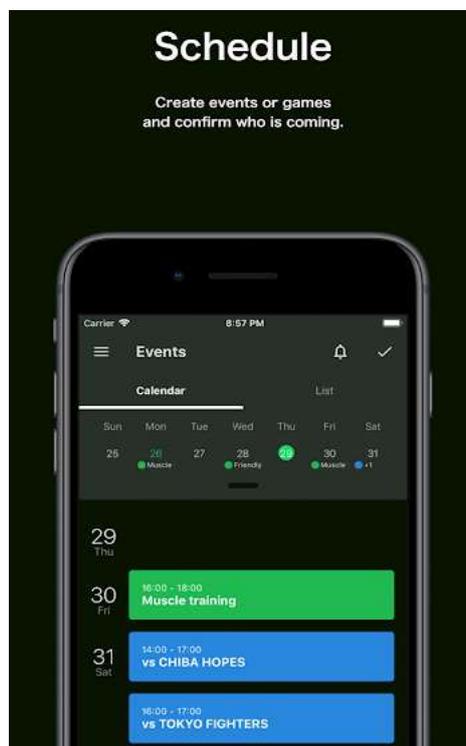


Ilustración 1-5 Calendario con los eventos próximos.

3. **RunnApp** [3]. Se trata de un buscador de carreras deportivas. Sus funciones principales son:

- Permite al usuario buscar carreras deportivas mediante filtros (provincia, modalidad, palabra clave, etc.)
- Un calendario con las fechas próximas de los eventos.
- Información detallada de las carreras deportivas.

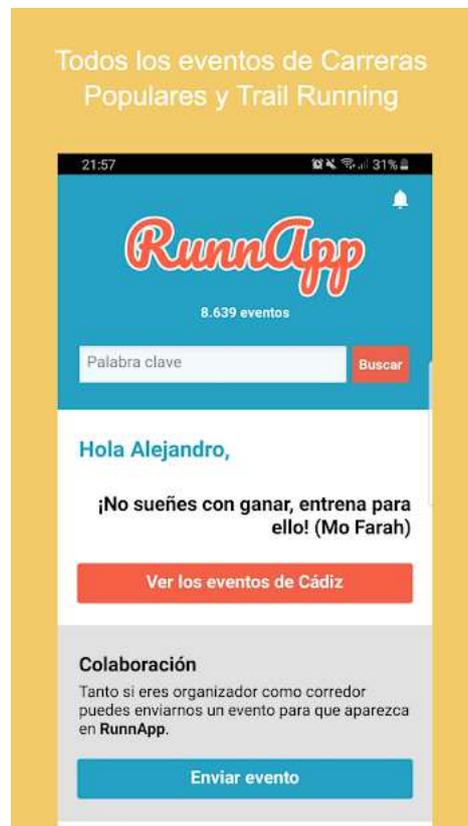


Ilustración 1-6 Home de RunnApp.

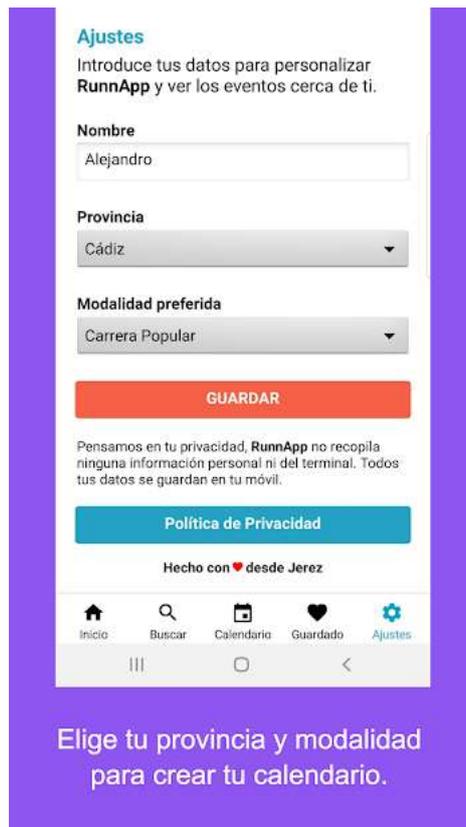


Ilustración 1-7 Filtro de búsqueda.

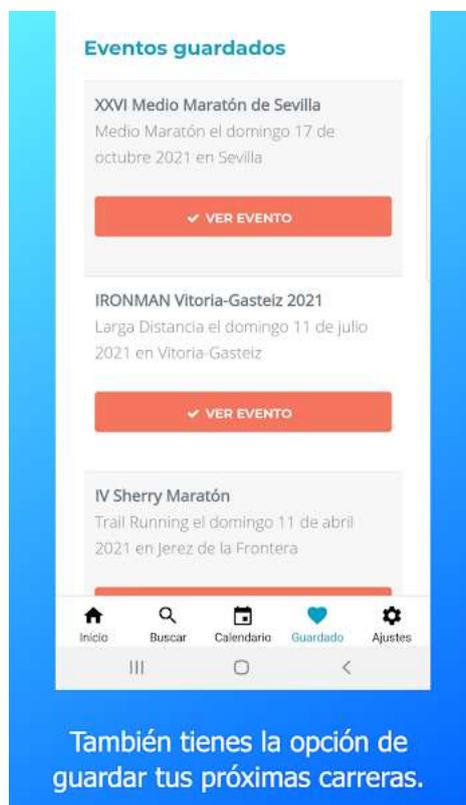


Ilustración 1-8 Eventos guardados.



Ilustración 1-9 Información del evento.

Como podemos observar en las ilustraciones mostradas, no satisfacen en su totalidad el problema que hemos descrito. Estas aplicaciones son muy útiles para muchas personas, pero carecen de algunas características como la de asignar tareas a los voluntarios, el seguimiento de dichas tareas, la comunicación entre los organizadores y voluntarios, entre otras.

## Capítulo 2. Competencias específicas.

En este proyecto las competencias específicas que se han cubierto durante su desarrollo han sido:

### 2.1. IS01

“Capacidad de integrar soluciones de Tecnologías de la información y las comunicaciones y procesos empresariales para satisfacer las necesidades de información de las organizaciones, permitiéndoles alcanzar sus objetivos de forma efectiva y eficiente, dándoles así ventajas competitivas.”

Esta competencia está relacionada con la fase de análisis de este trabajo. En este proceso, se tuvo que realizar un estudio previo de los requisitos funcionales, no funcionales y del sistema. También observamos que actores participarían y se describen las especificaciones de los casos de uso. A la hora de implementar dicha herramienta se ha seguido con los principios fundamentales de la ingeniería del software y la metodología SCRUM.

### 2.2. TI03

“Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas.”

En este proyecto hemos utilizado el test de usabilidad para evaluar el diseño de esta aplicación. Además, para la gestión del desarrollo se han empleado herramientas que permiten gestionar el desarrollo del mismo.

### 2.3. TI06

“Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.”

Para esta aplicación móvil era necesario entender los servicios, sistemas y tecnologías de red que fueron utilizados para la implementación, principalmente para el almacenamiento de datos y archivos en Firebase.

# Capítulo 3. Aportaciones.

## 3.1. Entorno socio-económico

La intención de este proyecto ha sido desde un principio mejorar el trabajo que desempeñan estos organizadores en su día a día. Con esta aplicación conseguimos que se simplifiquen las actividades que son rutinarias en todas las organizaciones de las carreras deportivas. La gestión de los voluntarios y las inscripciones proporcionan una organización más eficiente y automatizada. Además de permitir que los organizadores y todos los voluntarios tengan una vía de comunicación directa para poder coordinar y seguir mejor las tareas organizativas favoreciendo aún más el trabajo que se tiene que desempeñar.

Como hemos destacado en el capítulo de la introducción, actualmente no existe ninguna aplicación que contenga todas estas funcionalidades. Es por ello por lo que se ha desarrollado una aplicación móvil innovadora que nos permitirá mejorar el trabajo de muchas empresas que se encargan de estos eventos.

En definitiva, el principal objetivo de este TFF es satisfacer a este tipo de personal ofreciéndoles una mejor herramienta que les permita realizar y organizar de manera cómoda, sencilla y ordenada su trabajo.

## 3.2. Entorno personal

En cuanto al entorno personal, es importante señalar que la elaboración de este proyecto ha resultado ser muy gratificante porque he podido emplear y poner en práctica los conocimientos que he adquirido durante estos últimos años.

He descubierto nuevas habilidades y capacidades, que hasta el momento desconocía, y que me ayudarán en un futuro no muy lejano. También, he aprendido nuevas herramientas y tecnologías que me permitirán mejorar como programadora. Es más, conocer las diferentes especialidades que tiene la informática es una de las principales razones que me cautivaron para desarrollar esta aplicación. Además, desde un primer momento quise realizar un proyecto en el que tuviera muy pocas similitudes con trabajos que he realizado durante estos últimos años en esta carrera.

# Capítulo 4. Normativa y legislación.

## 4.1. Licencias de software

### 4.1.1. GNU GPL

La licencia GNU GPL <sup>[4]</sup> del inglés General Public License, es una licencia de software libre y de código abierto utilizada en todo el mundo. Con ella, se garantiza a los usuarios finales la libertad de usar, estudiar, copiar y editar el software. Es una licencia copyleft, que significa que cualquier software que se encuentre con esta, tiene que distribuirse con las mismas o equivalentes condiciones.

Esta licencia se encuentra en las siguientes herramientas: Git y StarUML.

### 4.1.2. MIT

La licencia MIT <sup>[5]</sup> es una licencia software libre permisiva que se origina en el famoso Instituto Tecnológico de Massachusetts. Esta aplica muy pocas limitaciones en la reutilización del software que respalda. Además, posee una excelente compatibilidad de licencia permitiendo utilizar otras licencias como son la GNU GPL.

Las herramientas que se encuentran bajo esta licencia son las siguientes: Angular, Ionic, Ionic-icons y Visual Studio Code.

### 4.1.3. Apache License

La licencia Apache <sup>[6]</sup>, o también conocida como Apache Software License, es una licencia de software libre permisiva. Fue creada por la Fundación Apache Software y actualmente cuenta con la versión 2.0. Requiere de la conservación del aviso de derechos de autor y el descargo de responsabilidad. No es una licencia copyleft por lo que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas.

Esta licencia se encuentra en las siguientes herramientas: Android Studio, Typescript, Bootstrap y Firebase.

#### **4.1.4. Microsoft Office: Office 365 Education Plus for students**

Es un conjunto de servicios gratuitos que Microsoft ofrece para los estudiantes y profesores que están inscritos en instituciones educativas, como por ejemplo la Universidad de Las Palmas de Gran Canaria. <sup>[7]</sup>

## **4.2. Leyes**

La seguridad de la información es uno de los aspectos importantes que se tienen que mencionar en este trabajo. Por este motivo, existen diferentes leyes que garantizan los datos personales de las personas.

### **4.2.1. Ley de protección de datos**

Una de estas leyes es la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y que garantiza los Derechos Digitales (LOPDPGDD) <sup>[8]</sup>. Esta ley española tiene por objeto adaptar el derecho interno español al Reglamento general de protección de datos <sup>[9]</sup>.

Los datos personales de esta aplicación son tratados de acuerdo con esta ley. Se limita su acceso empleando las reglas de seguridad que proporciona Firebase para garantizar la seguridad de la base de datos. Además, para la contraseña se almacena el hash de la misma y no la contraseña en texto plano.

### **4.2.2. Reglamento general de protección de datos**

El Reglamento General de Protección de Datos <sup>[10]</sup>, también conocido como RGPD, es un nuevo reglamento que fue aprobado por el Parlamento y el Consejo Europeo. Consiste en la protección de las personas físicas en lo que respecta al tratamiento de los datos personales y a la libre circulación de estos datos.

El tratamiento de datos se restringirá a lo imprescindible, cumpliendo así con la normativa. Esto solo sucede cuando el organizador crea un fichero Excel en el que se muestran los datos de los voluntarios (nombre, apellidos, teléfono, D.N.I., talla de camiseta y email) para que se les pueda dar de alta.

# Capítulo 5. Metodología y planificación del proyecto.

## 5.1. Metodología

La metodología que se ha llevado a cabo para el desarrollo de esta aplicación móvil ha sido SCRUM [11], que es un proceso en el que se utiliza un conjunto de buenas prácticas para trabajar en equipo. Existen tres roles en ella, que son:

- Scrum Master que se encarga de liderar al equipo y solventar los inconvenientes que vayan surgiendo durante el sprint.
- Product Owner es quien se comunica con el cliente y tiene la visión del proyecto.
- Equipo (Team) es un grupo de programadores que se encargan de implementar las historias de usuarios.

En la siguiente ilustración podemos observar cómo sería el proceso de esta metodología. En cada iteración (sprint) se va desarrollando las historias de usuarios que se han planificado al comienzo de la iteración. Estas historias de usuarios están en la pila del producto, que es un listado de todas las funcionalidades que se implementarán en el proyecto.

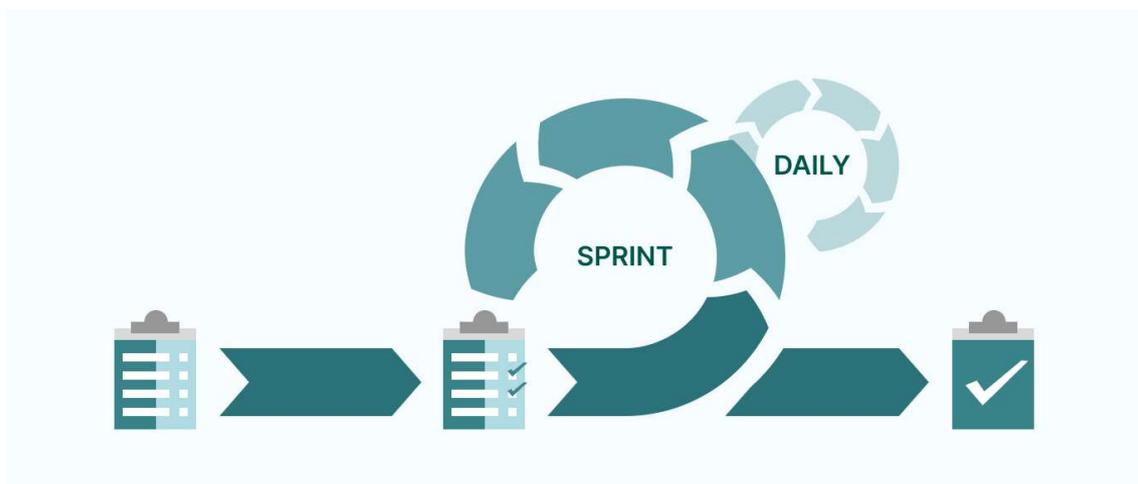


Ilustración 5-1 Metodología SCRUM.

## 5.2. Planificación del proyecto

Al comienzo de este proyecto se diseñó una planificación para su desarrollo. Esta planificación se ha planteado con la intención de realizar las diferentes fases de forma eficiente y estructurada.

En la siguiente tabla 1, podemos distinguir las distintas tareas que se han desempeñado.

Fase	Estimación	Tareas
Estudio previo/ Análisis	30	Tarea 1.1: Estudio del estado del arte.
		Tarea 1.2: Obtención y análisis de los requisitos de la aplicación.
		Tarea 1.3: Aprendizaje sobre las tecnologías a utilizar y estudio de la documentación de estas.
Diseño/ Desarrollo/ Implementación	200	Tarea 2.1: Definición de las tareas a desarrollar y planificación de ellas.
		Tarea 2.2: Diseño del modelo de datos.
		Tarea 2.3: Diseño de la interfaz.
		Tarea 2.4: Implementación de la aplicación.
Evaluación/ Validación/ Prueba	30	Tarea 3.1: Pruebas de funcionamiento.
		Tarea 3.2: Verificación del funcionamiento de la aplicación móvil.
Documentación/ Presentación	40	Tarea 4.1: Realización de la memoria.
		Tarea 4.2: Realización de la presentación

Tabla 1. Planificación del TFT.

En la primera fase se realizaron las siguientes tareas:

- Estudio del estado del arte. Consiste en realizar un estudio de las aplicaciones que se asemejan a la idea propuesta. Posteriormente, con esta investigación se puede realizar una comparativa de las funcionalidades que tienen.

- Obtención y análisis de los requisitos de la aplicación. En estas tareas fue necesario de la intervención del co-tutor, Orlando Montesdeoca Santana, para recopilar la información necesaria. Esta nos permitió poder definir los requisitos que tendría la aplicación móvil. De este modo, pudimos especificar los diferentes actores que participarían, los casos de usos y así poder diseñar el prototipo.
- Aprendizaje sobre las tecnologías a utilizar y estudio de la documentación de estas. Esta investigación requirió de un estudio previo de las tecnologías para seleccionar las más adecuadas para la implementación de la herramienta.

Finalizadas las tareas que se han mencionado, comenzamos a realizar la fase de diseño, desarrollo e implementación, que consistieron en las siguientes tareas:

- Definición de las tareas a desarrollar y planificación de ellas. Se realizó un listado de las funcionalidades que se implementarían y se describió las historias de usuarios.
- Diseño del modelo de datos. Para diseñar el modelo de datos fue imprescindible tener el conocimiento del funcionamiento de la base de datos.
- Implementación de la aplicación. En esta tarea se comenzó a utilizar la metodología que hemos mencionado en el apartado anterior y a desarrollar la aplicación.

A continuación, se prosiguió con la siguiente fase que residió en verificar el funcionamiento de la herramienta. Para ello se realizaron las tareas siguientes:

- Pruebas de funcionamiento y verificación del funcionamiento de la aplicación móvil. Estas tareas se realizaron juntas debido a que consistía en realizar lo mismo.

Para acabar este proyecto, se finalizó con la última fase:

- Realización de la memoria y de la presentación. Consistía en documentar lo que se ha realizado en este proyecto y elaborar una presentación para el día de la defensa.

Durante el desarrollo del proyecto no ha surgido ningún imprevisto que permita replantearse la planificación. Hay que señalar que la fase de implementación ha resultado más compleja de lo esperado debido a Firebase, porque tenía una versión nueva en la que sus funciones eran completamente diferentes a las anteriores.

# Capítulo 6. Herramientas y tecnologías utilizadas.

En la actualidad existen una gran variedad de herramientas y tecnologías que nos pueden ser útiles para el desarrollo y diseño de aplicaciones móviles. Antes de comenzar a desarrollar la aplicación, explicaremos de forma breve aquellas que hemos seleccionado y nos han sido de mayor utilidad.

## 6.1. Herramientas de diseño

- **Figma** <sup>[12]</sup>. Es una herramienta de creación de prototipos web que se centra en el diseño de interfaces de usuario y la experiencia de usuario.
- **StartUML** <sup>[13]</sup>. Se trata de una aplicación que permite modelar diagramas UML.

## 6.2. Herramientas para el desarrollo

### 6.2.1. Lenguajes y librerías

- **HTML (HyperText Markup Language)** <sup>[14]</sup>. Es un lenguaje de marcado para el desarrollo de sitios web, que permite indicar la estructura de un documento mediante etiquetas.
- **CSS (Cascading Style Sheets)** <sup>[15]</sup>. Es un lenguaje de estilos para la definición de la presentación de un documento estructurado escrito en un lenguaje de marcado.
- **TypeScript** <sup>[16]</sup>. Es un lenguaje de programación open-source que traduce el código a JavaScript mediante un compilador. Además de las características de JavaScript, tiene el tipado estático, tipos genéricos, tipos enumerados y objetos basados en clases, entre otro.

### 6.2.2. IDE y frameworks

- **Ionic y Ionic CLI** <sup>[17]</sup> <sup>[18]</sup>. Es un kit de desarrollo de software, conocido como SDK, open-source para el desarrollo de aplicaciones móviles híbridas basado en tecnologías web (HTML, CSS y JS). Ionic CLI es la interfaz de línea de

comando para el desarrollo de aplicaciones basadas en Ionic que facilita la creación de proyectos y módulos dentro de un proyecto.

- **Angular** <sup>[19]</sup> <sup>[20]</sup>. Es un framework que se utiliza para crear y mantener aplicaciones de una sola página. Está desarrollado en TypeScript y es open-source. Su objetivo es aumentar las aplicaciones basadas en navegador utilizando el patrón de diseño Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.
- **Firebase** <sup>[21]</sup>. Se trata de una plataforma para el desarrollo de aplicaciones web y móviles. Ofrece muchos servicios, entre ellos, está el acceso a información en tiempo real ubicado en la nube de Firebase.
- **Realtime Database** <sup>[22]</sup>. Es una base de datos en tiempo real, que ofrece como servicio Firebase. Esta base de datos permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firebase.
- **Authentication** <sup>[23]</sup>. Es uno de los servicios que ofrece Firebase y que consisten en la administración de los usuarios.
- **Storage** <sup>[23]</sup>. Consiste en un sistema de almacenamiento de archivos que permite guardar las imágenes o vídeos que los usuarios generan en la aplicación.
- **Microsoft Visual Studio Code** <sup>[24]</sup>. Es un editor de código fuente multiplataforma que destaca por diversas características: soporte de depuración, control de versiones, resaltado de sintaxis y autocompletado del código.
- **Android Studio** <sup>[25]</sup>. Es un entorno de desarrollo, usualmente llamado IDE, para la plataforma Android basado en IntelliJ IDEA. Destaca principalmente por su emulador para probar aplicaciones, renderizar en tiempo real, consola de desarrollador, refactorización de código, sistema de compilación flexible basado en Gradle, control de versiones con GitHub, renderizar aplicaciones en ejecución, entre otros.
- **Bootstrap** <sup>[26]</sup>. Es un framework open-source que consiste en diseñar la parte front-end de sitios web y aplicaciones de manera cómoda y sencilla.

### 6.3. Herramientas organizativas

- **Git y GitHub** <sup>[27]</sup> <sup>[28]</sup>. Git es un software open-source que consiste en guardar todos los cambios que se van realizando sobre un repositorio de un proyecto. También permite la coordinación y organización de varias personas que desarrollan un mismo proyecto. GitHub es una plataforma de desarrollo colaborativo que permite almacenar proyectos utilizando un sistema de control de versiones, como es Git.
- **Microsoft Office 365** <sup>[7]</sup>. Es un conjunto de aplicaciones que permite crear, acceder, editar y compartir documentos.
- **Trello** <sup>[29]</sup> . Es una herramienta que facilita la organización de proyecto mediante un tablón virtual donde se pueden colgar ideas, tareas, imágenes o enlaces.

# Capítulo 7. Análisis.

## 7.1. Roles

En esta aplicación se distinguen diferentes actores:

- **Organizador:** Es quién gestiona los eventos y sus respectivas tareas. Puede enviar mensajes a los voluntarios para comunicar alguna incidencia que ocurre durante el evento y, además, tiene permitido escoger los voluntarios que se encargarán de realizar las tareas.
- **Voluntarios:** Este actor puede inscribirse en eventos y elegir qué tareas se encargará de hacer. Dentro de este rol, existen dos grupos importantes a la hora de registrarse:
  - ❖ **Responsable de un grupo de voluntarios:** Es la persona que se encarga de registrar a un grupo de voluntarios para que se inscriban en un evento.
  - ❖ **Individual:** Es un usuario que se registra.

## 7.2. Casos de uso

En la siguiente ilustración, podemos observar el diagrama de casos de uso que tendrá la aplicación móvil.

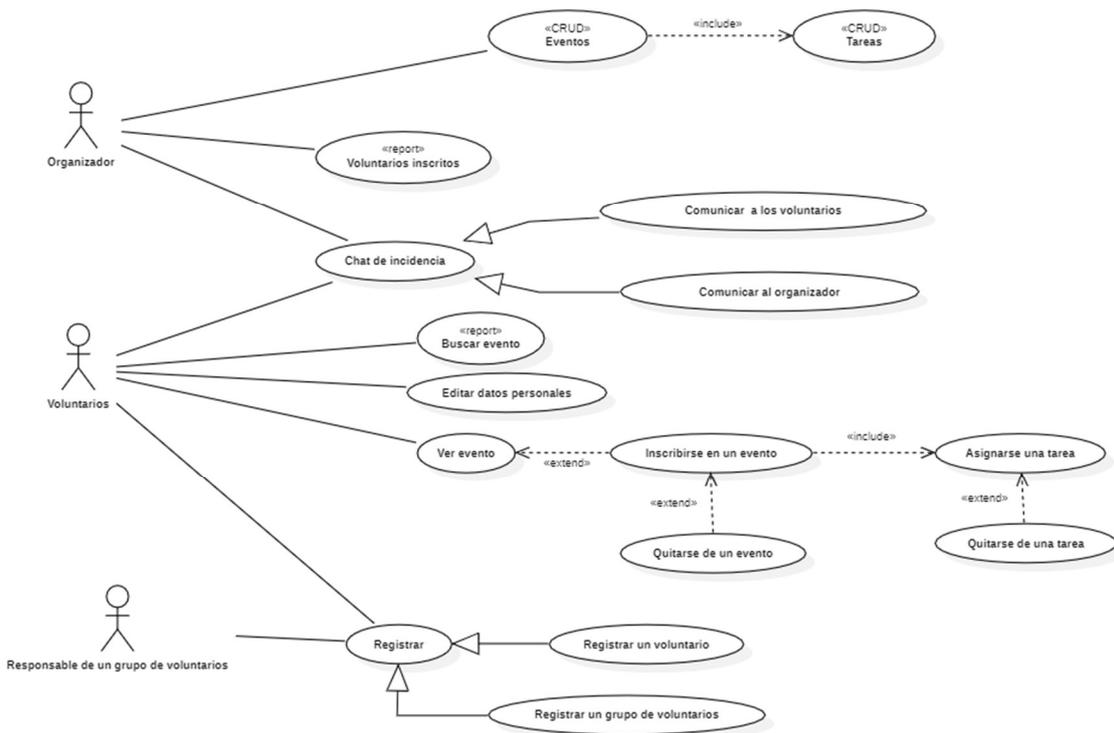


Ilustración 7-1 Diagrama de casos de uso.

Como se puede apreciar, hay cuatro casos de uso importantes. Son los siguientes:

- **<<CRUD>> Eventos.** El organizador podrá crear, ver, modificar y eliminar los eventos.
- **<<CRUD>> Tareas.** El organizador podrá crear, ver, editar y borrar las tareas.
- **Chat de incidencias.** El organizador puede chatear con un voluntario o con todos los voluntarios que participan en el evento. Por otro lado, el voluntario podrá enviar mensajes al organizador o a otros voluntarios.
- **Registrar.** Si el actor es un voluntario individual simplemente introducirá sus datos para registrarse. En cambio, si se tratará de un responsable tendrá que introducir todos los datos de los voluntarios.
- **Ver evento.** El voluntario puede ver el evento. Tiene la opción de inscribirse en él y le permitirá asignarse una tarea.

Una vez definidos los casos de usos, es necesario realizar la especificación de los casos de usos que permitirán aclarar algunos aspectos relacionados con el funcionamiento de la aplicación. (Están descritos en el Anexo I: Especificación de casos de uso).

### **7.3. Requisitos funcionales**

Los requisitos funcionales que tendrá esta aplicación son los siguientes:

- El sistema debe permitir al organizador crear, publicar, modificar y eliminar un evento con sus respectivas tareas.
- El sistema permitirá a los voluntarios inscribirse en una tarea.
- El sistema debe permitir a los voluntarios quitarse de una tarea.
- El sistema permitirá al organizador bloquear y asignar tareas a los voluntarios tanto dentro como fuera de plazo.
- El sistema debe permitir al voluntario editar sus datos personales.
- El sistema permitirá a los voluntarios asignarse a una tarea y a otra como suplente.
- El sistema debe permitir a los voluntarios comunicar incidencias al organizador y los voluntarios.
- El sistema permitirá que el organizador pueda enviar mensajes a un voluntario y a todos los voluntarios.
- El sistema debe permitir al organizador fijar un plazo en las tareas para que se inscriban los voluntarios.
- El sistema debe permitir que un responsable de un grupo de voluntarios pueda registrar a todos los voluntarios.
- El sistema enviará un correo de confirmación al voluntario una vez que se registre.
- El sistema permitirá al voluntario y al organizador buscar un evento.
- El sistema notificará a los voluntarios y al organizador que tiene mensajes sin leer.

## 7.4. Requisitos no funcionales

Esta aplicación tendrá como requisitos no funcionales:

- El sistema podrá deshabilitar la opción asignar una tarea al voluntario una vez que está fuera de plazo.
- El sistema muestra un mensaje de aviso si un dato es insertado de forma incorrecta.
- El sistema cerrará un evento una vez que se cumpla el plazo establecido por el organizador.
- El sistema asignará un identificador único a todos los usuarios.
- El sistema debe permitir que un responsable de un grupo de voluntarios registre un número considerable de voluntarios.

# Capítulo 8. Diseño.

## 8.1. Objetivos del diseño

El objetivo principal de este proyecto es cubrir una necesidad existente, como es la gestión de voluntarios y la inscripción de estos. Por este motivo, diseñar una aplicación que sea útil, sencilla, organizada, intuitiva, cómoda y agradable a la vista es fundamental.

El diseño facilita a los usuarios poder interactuar con la aplicación. De ahí que es importante diseñar prototipos que permitan describir la idea principal de nuestro proyecto. Utilizando la herramienta conocida como Figma, podemos desarrollar modelos que reflejen las funcionalidades y los requisitos que se han mencionado en el capítulo del Análisis. Por eso, se ha diseñado un prototipo que cumpla con todas las especificaciones y transmita esos aspectos relacionados con la interacción del usuario. Tener una paleta de colores hace que la aplicación tenga una misma temática. Esta paleta la podemos ver en la siguiente ilustración 8-1.

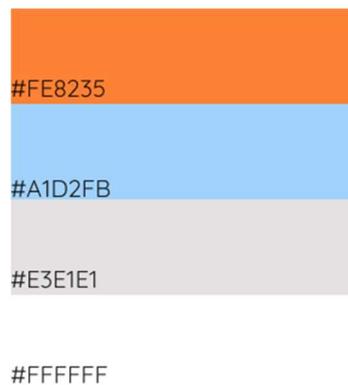


Ilustración 8-1. Paleta de color de la aplicación.

## 8.2. Diseño de mockups

La aplicación móvil comenzará con el inicio de sesión. En ella, podrá el usuario iniciar sesión (Ilustración 8-2).



Ilustración 8-2. Mockups: Iniciar sesión.

También podrá registrarse. Al pulsar el botón de registrarse, nos mostraría una opción donde se indica si el usuario es un responsable o un usuario individual. (Ilustración 8-3)



Ilustración 8-3. Mockups: Seleccionar quien va a registrarse.

Si se selecciona que es un usuario individual se mostrará un formulario (Ilustración 8-4) que requiere de los siguientes datos:

- Nombre y Apellidos.
- Correo electrónico. Para enviar correos relacionados a un evento inscripto.
- Número de teléfono. Para llamar si fuera necesario al voluntario.
- D.N.I. / N.I.F. / Pasaporte del usuario.
- Talla de la camiseta. Para el día del evento.
- Contraseña.

The image shows a mobile application interface for a registration form. At the top, the status bar displays the time as 5:13 PM and various system icons. Below the status bar is a navigation bar with a back arrow and the title 'Registrarse'. The form itself is contained within a light blue rounded rectangle and consists of several sections, each with a label and a corresponding input field:

- Nombre y Apellidos:** A text input field containing the placeholder text 'Nombre y Apellidos'.
- E-mail:** A text input field containing the placeholder text 'nombre@example.com'.
- DNI/NIF/Pasaporte:** A text input field containing the placeholder text '12345678A'.
- Teléfono:** A text input field containing the placeholder text 'Tif.: 000000000'.
- Talla de camiseta:** A dropdown menu currently showing 'S'. Below the dropdown is a list of options: 'Talla XS', 'Talla S', 'Talla M', and 'Talla L'.
- Contraseña:** A text input field containing the placeholder text 'Contraseña'.
- Confirmar contraseña:** A text input field containing the placeholder text 'Confirmar contraseña'.

At the bottom of the form is a large orange button with the text 'Registrarse' in white.

Ilustración 8-4. Mockups: Registrarse.

En caso de que fuera un responsable que quiere registrar a un grupo de voluntarios, mostraría el formulario que se puede apreciar en la Ilustración 8-5.

5:13 PM

← Responsable de voluntarios

**Nombre y Apellidos:**  
Nombre y Apellidos

**E-mail:**  
nombre@example.com

**DNI/NIF/Pasaporte:**  
12345678A

**Teléfono:**  
Tlf.: 000000000

**Contraseña:**  
Contraseña

**Confirmar contraseña:**  
Confirmar contraseña

**Siguiete**

Ilustración 8-5. Mockups: Formulario para el responsable.

Una vez rellenado todos los datos correspondientes, podrá añadir a los voluntarios mediante el formulario de la ilustración 8-4. Una vez que se han introducido todos los datos pertinentes y pulsado el botón ‘Registrarse’, se redirigirá a la página de inicio en la que se mostrarán los próximos eventos que se van a realizar (Ilustración 8-6).

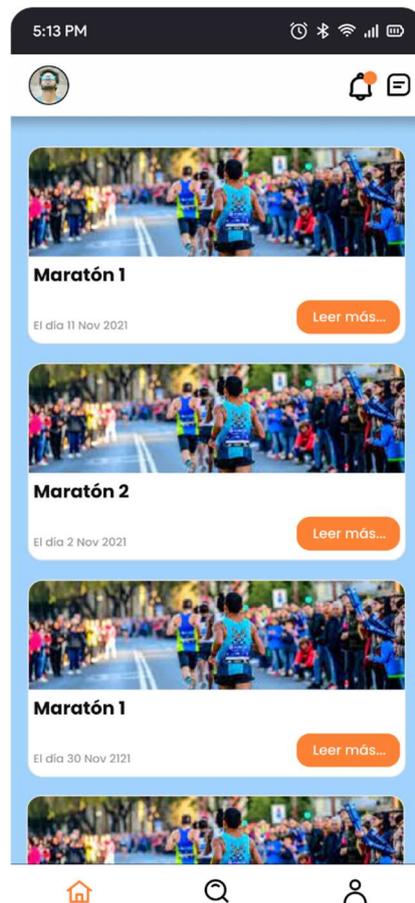


Ilustración 8-6. Mockups: Inicio.

El usuario puede ver un evento, ir al perfil, a la sección de chats y buscar un evento en concreto. Si accedemos a un evento cualquiera, nos mostraría su descripción y las tareas que esta tiene (Ilustración 8-7). En ella, los voluntarios pueden inscribirse o ser suplente en la tarea. Este último caso, solo ocurriría cuando el número de voluntario está completo.



Ilustración 8-7. Mockups: Evento.

Si un voluntario se inscribe mostrará un mensaje (Ilustración 8-8) para confirmar la inscripción.

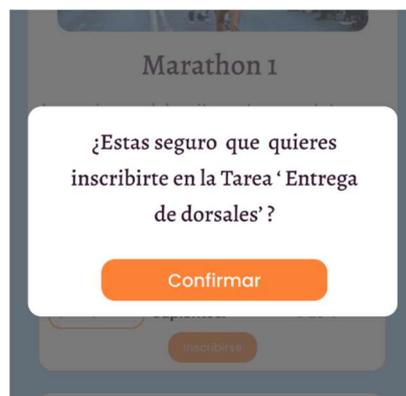


Ilustración 8-8. Mockups: Confirmar inscripción en una tarea.

Un usuario puede buscar un evento en concreto utilizando el buscador que se muestra en la ilustración 8-9.

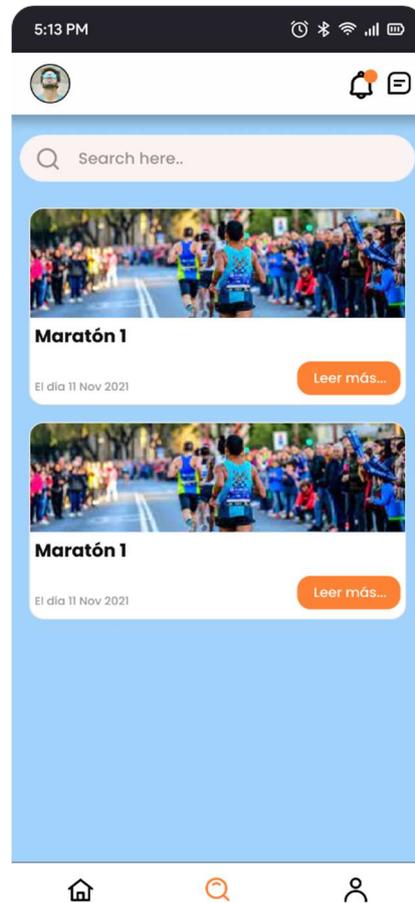


Ilustración 8-9. Mockups: Buscar un evento.

En perfil, cualquier usuario podrá editar todos los datos y su foto. Esto se puede apreciar en la siguiente ilustración.

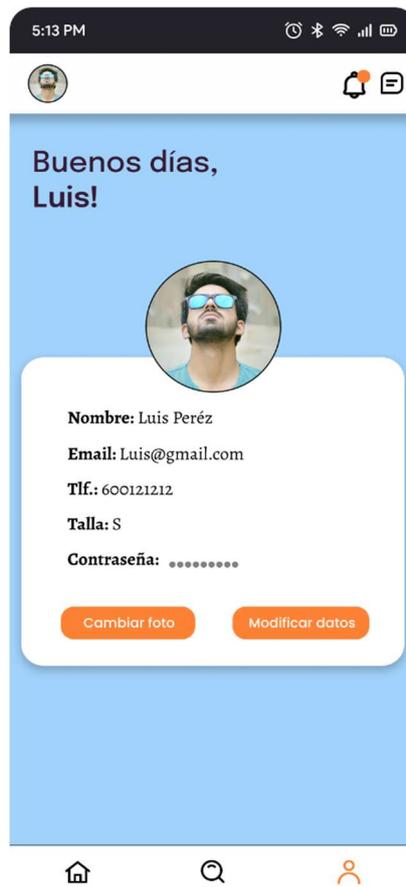


Ilustración 8-10. Mockups: Perfil.

La aplicación necesita que los voluntarios y el/los organizador/es se comuniquen entre ellos para el día del evento. Por este motivo, el usuario tendrá una interfaz que permita mostrar un listado de diferentes conversaciones (Ilustración 8-11).

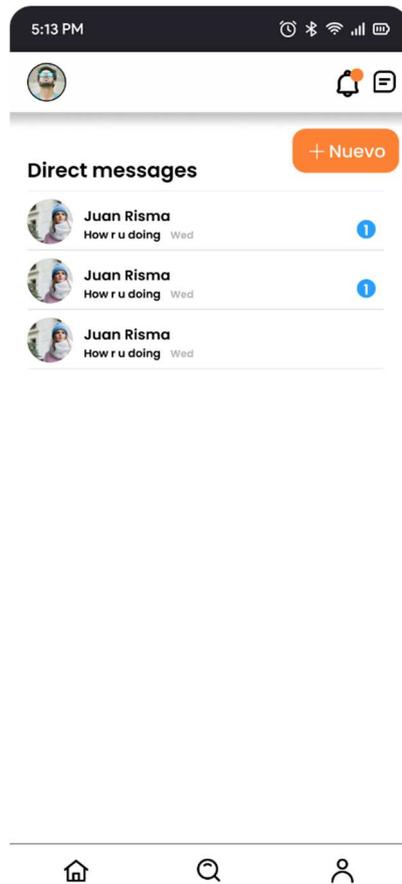


Ilustración 8-11. Mockups: Chats.

Para comunicarse con otro usuario, simplemente necesitará pulsar en el nombre del usuario para entrar en su chat común, que corresponde a la siguiente ilustración.

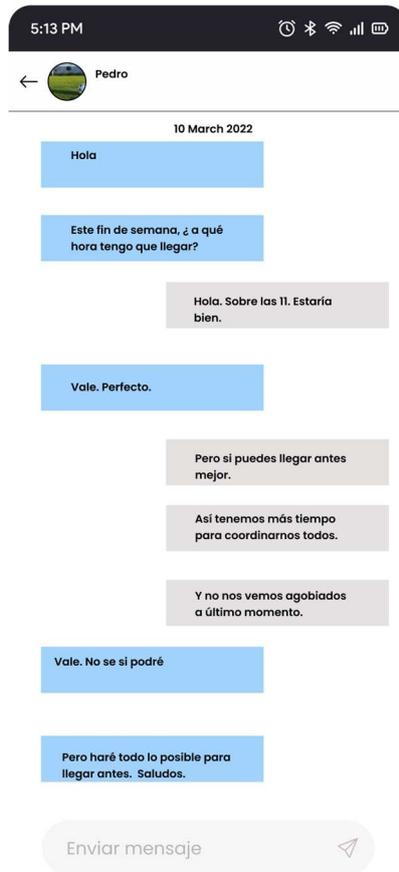


Ilustración 8-12. Mockups: Chat con una persona.

En caso de que hasta ese momento no haya sido necesaria la comunicación con un usuario, podrá crear un nuevo chat a través del botón ‘Nuevo’ de la ilustración 1-11. Este botón mostrará un mensaje para que el usuario indique el/los destinatario/s, como podemos ver en la ilustración 8-13.

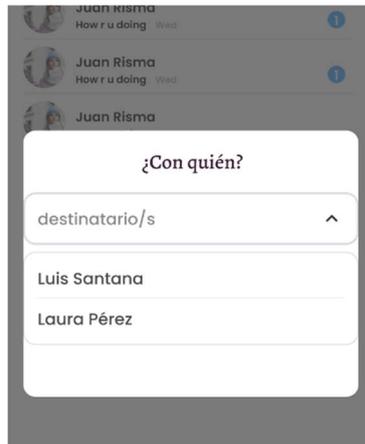


Ilustración 8-13. Mockups: Nuevo chat indicando destinatario.

El organizador tendrá un menú que le permita crear evento, ver los organizadores que tiene la aplicación o ver los diferentes eventos que ha ido creado (Ilustración 8-14).

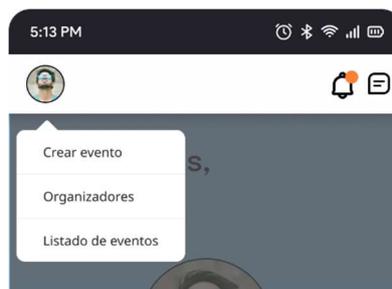


Ilustración 8-14. Mockups: Menú del organizador.

Si el organizador pulsa 'Crear evento', le mostrará un formulario en el que tiene que rellenar los siguientes campos:

- Título del evento.
- Descripción del evento.
- Plazo de inscripción. Esto se refiere al plazo que tienen los voluntarios para inscribirse.

5:13 PM

← Crear evento

**Título del evento:**  
Título del evento

**Descripción:**  
Breve descripción...

**Organizador:**  
Nombre del organizador

**Plazo de inscripción:**  
Plazo de inscripción

Siguiete

Ilustración 8-15. Mockups: Crear evento.

Rellenado el formulario y pulsando el botón ‘Siguiete’ de la ilustración anterior, el organizador podrá crear y eliminar tareas. Esta característica se puede observar en la ilustración siguiente. Cada tarea tendrá:

- Nombre de la tarea.
- Número de voluntarios. Nos indicará el máximo de voluntarios que podrá tener.
- Número de suplentes. Al igual que el anterior, corresponde al máximo de suplentes que tendrá permitido la tarea.



Ilustración 8-16. Mockups: Crear tareas.

En cuanto al organizador se refiere, podrá ver a otros organizadores o dar de alta a un nuevo organizador (Ilustración 8-17). Para crear un organizador se tendrá que rellenar el formulario de la ilustración 8-4.

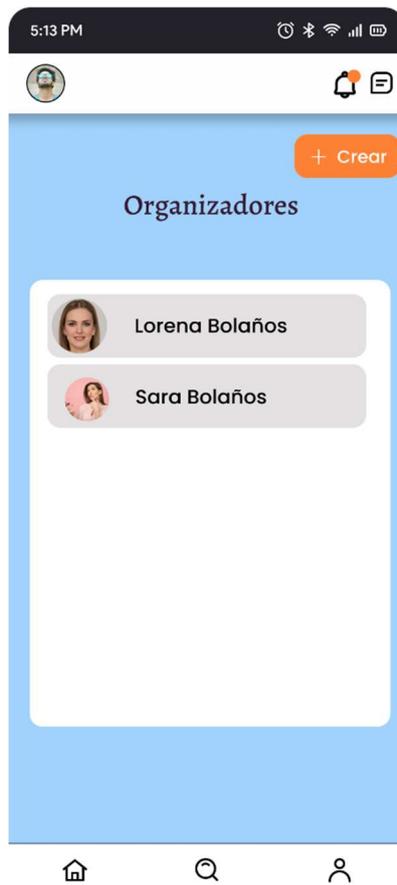


Ilustración 8-17. Mockups: Listado de organizadores.

El organizador tendrá un listado de los eventos que ha ido creando para poder modificar y eliminar. Del mismo modo, tiene los privilegios necesarios para ver el listado de voluntarios que están inscriptos en él (Ilustración 8-18).



Ilustración 8-18. Mockups: Eventos del organizador.

En cada evento, podrá ver a los voluntarios que se han ido inscribiendo y tendrá la opción de ver en cada tarea quien se ha inscrito. Esto se puede observar en las dos ilustraciones siguientes:

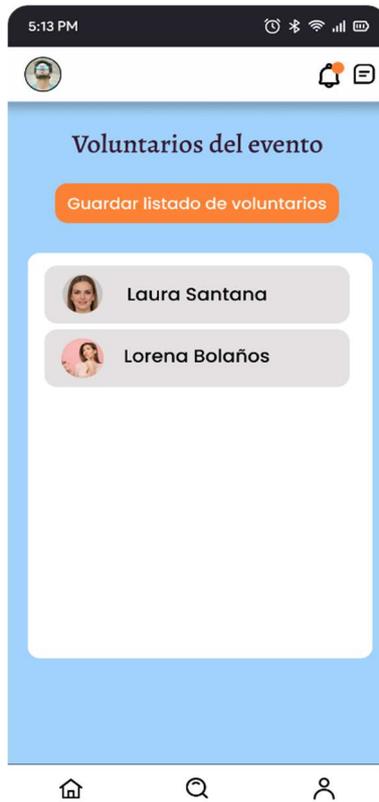


Ilustración 8-19. Mockups: Voluntarios del evento.



Ilustración 8-20 Mockups: Listado de voluntario de una tarea

## Capítulo 9. Desarrollo.

Para el desarrollo de esta aplicación utilizamos la herramienta Ionic CLI que nos ofrece una interfaz de comandos. Esta interfaz nos permite crear proyectos instalando y configurando todas las dependencias necesarias y facilitar a la hora de implementar un proyecto de esta envergadura. Además, necesitamos de la instalación de Firebase ya que la usaremos como base de datos para nuestro proyecto.

### 9.1. Estructura del proyecto

Una de las ventajas que tiene la interfaz Ionic CLI es la estructuración que tiene al crear un proyecto. De ahí que, esté organizado de esta forma:

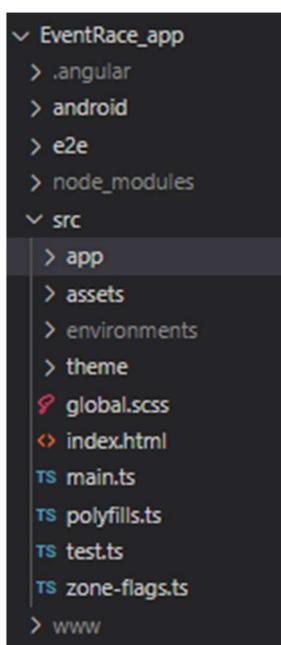


Ilustración 9-1 Estructura del proyecto

Como podemos apreciar en esta ilustración, en la carpeta “/src” se encuentra todo el código de nuestro proyecto. Dentro de esta carpeta existen diferentes directorios que describiremos a continuación:

- **/src/app** → Se encuentra todo el código que se ha ido desarrollando. En ella, se organiza de la siguiente forma:
  - ❖ **/app/components** → Contiene todos los componentes que se han implementado en la aplicación.

- ❖ **/app/model** → Se encuentran los diferentes modelos de datos que mencionaremos más adelante.
- ❖ **/app/service** → Es donde están los servicios que se han desarrollado para la aplicación.
- ❖ **AppModule** y **AppComponent** → Se trata del módulo y componente principal de proyecto.
- **/src/assets** → Esta carpeta contiene las imágenes que vamos a utilizar en el proyecto.
- **/src/environments** → Contiene la configuración a la base de datos de Firebase.
- **/src/theme** → En este directorio están los estilos comunes de toda la aplicación, como son las clases y las variables globales.

## 9.2. Modelo de datos

Antes de comenzar a implementar, era necesario saber qué datos eran importantes para almacenarlos en la base de datos permitiendo que el usuario pueda interactuar con la aplicación. Como podemos observar en la siguiente ilustración, los modelos de datos fundamentales son: Chats, Messages, Users, Events y Tasks.

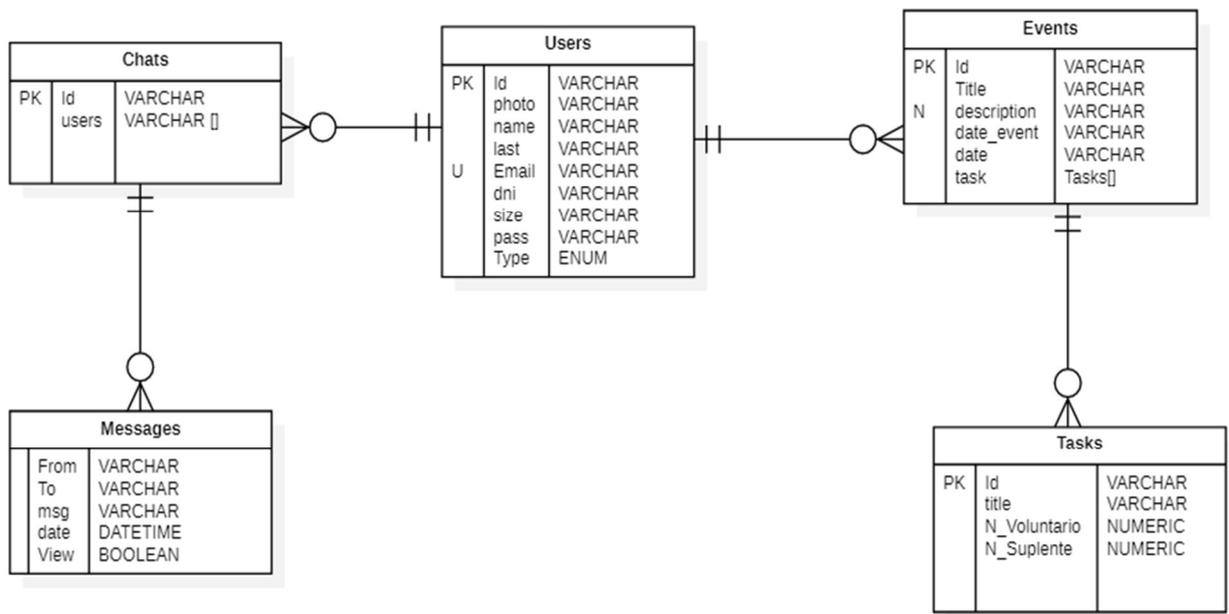


Ilustración 9-2 Modelo de datos de la aplicación.

### 9.2.1. Chats

Este modelo de datos permite guardar los diferentes chats que tienen los usuarios. En él, se almacenan las siguientes propiedades:

- id: es el identificador del chat.
- users: es un array con los números de identificación de los usuarios que forman el chat.

### 9.2.2. Messages

Corresponde a los mensajes que los usuarios envían a través de los chats, cuyos datos son:

- from: esta propiedad permite guardar el identificador del usuario que ha enviado el mensaje.
- to: está relacionada con la anterior y aquí se guarda el identificador del/de los receptor/es del mensaje.
- msg: corresponde con el mensaje que el usuario quiere enviar.
- date: hace referencia a la fecha del envío del mensaje.
- view: se trata de un token cuya utilidad es indicar que hay nuevos mensajes que no han sido leídos por los usuarios receptores.

### 9.2.3. Users

El modelo de datos 'users' guarda los datos de los usuarios. Sus propiedades son:

- id: el número identificativo del usuario.
- photo: corresponde con la foto que tiene el usuario.
- name: el nombre del usuario.
- last: el/los apellido/s que tiene el usuario.
- email: en este campo se guarda el correo electrónico que el usuario proporciona para poder enviarle correos.
- size: la talla de la camiseta para el día del evento. Esta propiedad no la tienen todos los usuarios, solamente los voluntarios.

- dni: corresponde al número identificativo que está en el Documento Nacional de Identidad, que comúnmente es conocido como D.N.I. en el caso de España. En el caso de que el usuario no tuviera este documento, puede proporcionar el número del Pasaporte o el NIF. Esto es imprescindible ya que los organizadores necesitan dar de alta a los voluntarios con este dato.
- pass: contraseña del usuario para poder acceder a la aplicación.
- type: esta propiedad hace referencia al rol que tiene el usuario, que ya hemos mencionado en el capítulo de análisis.

#### 9.2.4. Events

Como se puede aludir, ‘events’ corresponde con el modelo de datos eventos, cuyas propiedades son las siguientes:

- id: es el identificador del evento.
- title: es el título de la carrera deportiva.
- description: se trata de las características del evento.
- date\_event: se corresponde con el plazo de inscripción de los voluntarios en una tarea.
- date: es la fecha del evento.
- tasks: es un array con las diferentes tareas que tiene la carrera deportiva.

#### 9.2.5. Tasks

Es un modelo de datos que contiene las propiedades requeridas de una tarea.

- id: se trata del número Identificativo de la tarea.
- title: es el título de la tarea.
- n\_voluntarios: hace referencia al número máximo de voluntarios que puede haber.
- n\_suplentes: al igual que la anterior propiedad, se refiere al máximo de suplentes que puede tener.

## 9.3. Firebase

Como ya se mencionó en el capítulo de las herramientas, utilizaremos la Realtime Database de Firebase como base de datos para almacenar todos los datos en un árbol JSON. Una de las características que posee es la baja latencia que permite tener la sincronización de los datos y el estado, cuya propiedad no posee Cloud Firestore (la otra base de datos que ofrece Firebase). Este servicio de Firebase tiene estas particularidades de ahí que podamos obtener de forma inmediata los datos. Por este motivo, se ha considerado imprescindible la utilización de este servicio ya que es útil para la recopilación de mensajes en los chats de los usuarios.

Es importante señalar que Firebase nos ofrece otros servicios y nos facilita a la hora de desarrollar un proyecto como este. Uno de ellos es Authentication que nos permite el registro (mediante correo y contraseña) y la identificación de los usuarios para poder interactuar con la aplicación. Además, sirve también para el almacenamiento de archivos, y nos ofrece el servicio Storage con el que podemos almacenar las imágenes.

### 9.3.1. Estructura de Storage y Realtime Database

Para el almacenamiento de archivos se ha creado un directorio raíz, llamado 'images', donde se almacenan todas las imágenes que contiene la aplicación. Dentro de esta, existen otras dos carpetas: 'users' y 'events', que se corresponden respectivamente con las imágenes de los usuarios y de los eventos.

En cuanto a la estructura que posee la base de datos Realtime Database, podemos observar cómo está organizado con la siguiente ilustración:

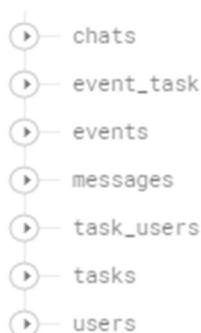


Ilustración 9-3 Estructura de la base de datos de Realtime Database.

Como se comentó anteriormente en el modelo de datos, tenemos seis nodos principales que se corresponden a:

- Chats. Los diferentes chats que tienen los usuarios en la aplicación.
- Event\_task. Las tareas que tienen los eventos.
- Events. Los eventos que se han ido creado.
- Messages. Los mensajes que se han enviado en los distintos chats.
- Task\_users. Los voluntarios que se han inscrito en las tareas de los eventos.
- Tasks. Las tareas que se han creado.
- Users. Los usuarios que existen en la aplicación.

Dentro de cada uno de estos, se utilizan otros nodos como identificadores únicos (id). Estos últimos nodos, permiten almacenar los datos que hemos indicado en el modelo de datos.

## 9.4. Servicios

Para el desarrollo de la aplicación, se han necesitado cinco servicios que son los encargados de acceder a los datos y entregarlos a los componentes pertinentes. Son los siguientes:

### 9.4.1. User service

Contiene todos los métodos y propiedades que corresponden a los datos de los usuarios. A continuación enumeramos las funciones que tiene:

- Iniciar sesión y cerrar sesión (**login** y **logout**).
- Comprobar la sesión y guardar el usuario que ha iniciado sesión (**session** y **setUser**). La función 'session' comprueba el estado de la aplicación., es decir, se encarga de verificar si el usuario ha iniciado sesión anteriormente o no.
- Registrar un voluntario, un organizador y un responsable (**createUser**, **createOrganizador** y **createResponsable**).
- Devolver los datos del usuario, la foto de perfil, un usuario en concreto, el id del usuario, el rol del usuario que ha iniciado sesión, el listado de

organizadores, los voluntarios y todos los usuarios (`getUsers`, `getPhoto`, `getUsers`, `getId`, `getType`, `getOrganizador`, `getVoluntarios`, `getAllUsers`).

- Editar una foto y editar un usuario (`editPhoto`, `setPhotodevolver` y `editUser`).

### 9.4.2. Images service

Este servicio consiste en la subida de imágenes al servicio Storage. Tenemos dos métodos:

- `setPhoto`. Consiste en almacenar las imágenes de los usuarios en la carpeta `images/users`.
- `editPhotoEvent`. Al igual que la función anterior, esta guarda las imágenes de los eventos en la carpeta `images/events`.

### 9.4.3. Chat service

En este servicio se encuentra todo lo relacionado con los chats y mensajes que los usuarios se envían. Las funciones que posee son las siguientes:

- Crear chat y mensajes (`createChat` y `createMessage`).
- Devolver los chats que tiene un usuario y los usuarios con su id, nombre y apellidos (`getUserChat` y `getUsers`).
- Devolver un chat y sus mensajes (`getChat` y `getMessages`).
- Devolver los id de los usuarios del chat (`getChatUser`).
- Obtener el número de mensajes nuevos sin leer de un chat (`getNewMessages`).
- Obtener el último mensaje que se ha enviado en el chat (`getLastMessage`).
- Cambiar el mensaje como visto (`setView`).

### 9.4.4. Event service

El event service contiene las funciones y las propiedades necesarias para la gestión de los eventos, entre las que cabe destacar:

- Crear evento (**createEvent**).
- Devolver todos los eventos y un evento en concreto (**getEvents** y **getEvent**).
- Editar un evento (**editEvent** y **setEvent**).
- Crear la relación de evento y tareas (**eventTasks**).
- Devolver todos los eventos que tiene un organizador (**getEventsOrganizador**).
- Obtener el plazo de inscripción de un evento (**getDateEvent**).
- Eliminar un evento (**deleteEvent**).

#### 9.4.5. Task service

En este servicio se encuentran todas las propiedades y métodos relacionados con el manejo de las tareas, que son:

- Crear una tarea (**createTask**)
- Obtener una tarea (**getTask**)
- Editar una tarea (**editTask**)
- Eliminar una tarea y la inscripción de los usuarios (**deleteTask** y **deleteInscription**)
- Devolver todas las tareas que tiene un evento con sus respectivas propiedades (**getTaskEvent**)
- Obtener la tarea con sus propiedades y los voluntarios inscritos (**getTaskUser**)
- Inscribir un voluntario a una tarea (**addUserTask**)
- Devolver la tarea con todos los voluntarios y suplentes (**getuserTask**)
- Obtener todos los voluntarios y suplentes del evento (**getUserTasksEvent** y **get VoluntarioSuplente**)
- Eliminar una inscripción (**deleteUserTask**)
- Obtener el número máximo de voluntarios y suplentes que se pueden inscribir en una tarea (**getMaxInscription**)

## 9.5. Principales componentes y funcionalidades

Como mencionamos en el apartado Estructura del proyecto, en la carpeta ‘app/components’ existen diferentes componentes que permiten el funcionamiento de la aplicación móvil. En este directorio podemos encontrar las funcionalidades, estilos de cada componente y la estructura que tienen. A su vez, esta carpeta se ha estructurado en seis subcarpetas, que son la que se pueden apreciar en la siguiente ilustración:

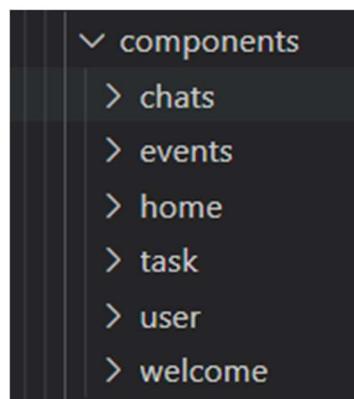


Ilustración 9-4 Estructura de los componentes.

Para crear un software eficaz, escalable, reutilizable y mantenible se ha aplicado la modularización en los componentes, motivo por el que se han estructurado así los componentes.

### 9.5.1. Iniciar sesión y registrar

Para poder iniciar sesión tenemos el componente ‘login’ ubicado en la carpeta ‘user’, cuya funcionalidad es permitir que el usuario se identifique. Este inicio de sesión cuenta con una validación en el que se verifica si el correo electrónico y la contraseña son las correctas. Si tenemos un usuario o responsable que no tenga ninguna cuenta puede pulsar el botón ‘Registrarse’ que le permitirá crear una nueva.

Como se señaló en el capítulo de Diseño, hay una fase previa para identificar el usuario que quiere registrarse (Ilustración 8-3). Esta vista corresponde con el componente ‘sign\_up’ que está en la carpeta ‘user’. Una vez que se ha identificado, el usuario podrá rellenar el/los formulario/s pertinentes. Al igual que la funcionalidad de iniciar sesión, registrarse posee varias validaciones. En este caso, son necesarias

dos tipos de validaciones: por un lado, ha sido la fase previa, en la que el usuario debe aceptar los términos, las condiciones de uso y la política de privacidad. Por otro lado, la otra validación se corresponde con los formularios que tendrán que rellenar los usuarios (el voluntario y el responsable del grupo de un voluntario).

Estas verificaciones han sido posibles gracias a una de las librerías que tiene Angular. Esta librería es '@angular/forms' y posee las clases 'FormBuilder', 'FormGroup' y 'Validators'. Estas han permitido que las funcionalidades 'iniciar sesión' y 'registrarse' se hallan implementado correctamente. Además, en el caso de registrarse, fue necesario crear un método que verificase la contraseña y su confirmación. En las siguientes ilustraciones, podemos observar cómo se ha realiza la validación de registrar a un voluntario:

```
form:FormGroup;
constructor(private FBBuilder: FormBuilder, private userService: UserService, private router: Router) {
  this.form= this.FBBuilder.group({
    name: ['', Validators.required],
    last: ['', Validators.required],
    email: ['', [Validators.required, Validators.email]],
    dni: ['', [Validators.required, Validators.minLength(9), Validators.maxLength(9)]],
    tlf: ['', [Validators.required, Validators.pattern('(6|7|9)[ -]*([0-9][ -]*){8}')]],
    size: ['', Validators.required],
    pass: ['', [Validators.required, Validators.minLength(6)]],
    repeat_pass: ['', [Validators.required, Validators.minLength(6)]]
  });
}
```

Ilustración 9-5 Creación de la validación para poder registrar a un voluntario.

```
validate(pass:string, repeat_pass:string){
  return (pass.localeCompare(repeat_pass)===0)? true: false;
}
```

Ilustración 9-6 Método que valida las contraseñas al registrarse.

Normalmente en los formularios, se suelen mostrar mensajes y avisos de los datos que se han introducido incorrectamente. En toda la aplicación, cuando existe una validación en estos casos, hemos implementado métodos o utilizado los métodos que nos ofrece este plugin. Esto último, se puede apreciar en la ilustración 9-7:

```
<span *ngIf="this.form.get('name')?.errors?.['required']">
  <ion-icon name="information-circle-outline"></ion-icon>
  No puede dejar nombre vacío
</span>
```

Ilustración 9-7 Mensaje de la validación al intentar registrarse el voluntario.

### 9.5.2. Perfil del usuario.

En la vista de perfil del usuario, la funcionalidad que destaca es la de editar los datos de usuarios que está en el componente 'profile' del directorio 'user'. En ella, podemos editar la foto del perfil, el nombre, los apellidos, la talla de camiseta, el teléfono, el correo electrónico y la contraseña. Como requisito para modificar dichos datos es necesario introducir la contraseña para confirmar que es el usuario. Esto último lo podemos apreciar en el Anexo II: Manual de usuario. Para editar los datos que posee el usuario, también fue necesario emplear la librería, que mencionamos en el apartado anterior, debido a que era preciso introducir la contraseña para editar esta información.

### 9.5.3. Inicio

El inicio se muestra cuando un usuario acaba de iniciar sesión o si la sesión sigue activa. Este componente, que está ubicado en las carpetas 'welcome' y 'home', tiene la peculiaridad de mostrar los distintos privilegios que tienen los usuarios y mostrar todos los eventos que tiene la aplicación. Con esta página, podemos ir a casi todas las vistas que tiene la aplicación. Para identificar el tipo de usuario, implementamos un método que comprobará el rol antes de cargar la página. Este método lo podemos ver en la siguiente ilustración.

```

async profile(): Promise<void>{
  this.userService.getType().then( (res)=>{
    if(res== '1' || res=='2'){
      this.user=true;
    }else{
      this.user=false;
    }
  });
  this.photo=await this.userService.getPhoto();
}

```

Ilustración 9-8 Método que comprueba el rol del usuario.

Hay que mencionar que en este método ha sido necesario cargar la foto de perfil del usuario ya que se trataba del mismo menú.

#### 9.5.4. Chats y chat

La interfaz de chats tiene la funcionalidad de mostrar el listado de chats que tiene el usuario. Es más, este componente ‘chats’ situado en el directorio ‘chats’, nos permite mostrar los chats que tienen mensajes nuevos sin leer, los nombres de los usuarios, el último mensaje que se ha enviado en el chat y crear uno nuevo.

Para mostrar la notificación de mensajes nuevos sin leer, desarrollamos un método con el que podemos verificar los estados de los mensajes. Al tener la propiedad ‘view’ que comentamos en el apartado de modelo de datos, podemos identificar que mensajes no se han leído. Este método podemos verlo en la siguiente ilustración:

```

countNewMessage(){
  for (let i = 0; i < this.list_chats.length; i++) {
    let id_chat = this.list_chats[i].id;
    this.chatService.getNewMessages(id_chat).then((res)=>{
      this.message_new.push(res);
    });
  }
}

```

Ilustración 9-9 Método que cuenta el número de mensajes sin leer.

Debemos destacar que para crear un nuevo chat, el usuario tiene que seleccionar el/los usuario/s a los que quiere enviar un mensaje. También, hemos implementado un buscador de usuario para que el usuario tenga una mayor facilidad a la hora de seleccionar a los usuarios.



Ilustración 9-10 Vista: Seleccionas usuario/s para crear un nuevo chat.

También existe una validación mediante la que se verifica si los usuarios que se han seleccionado están todos en un mismo chat. Con esto se garantiza que no se creen nuevos chats con los mismos usuarios. Ello es posible gracias a un método que identifica los chats que tiene el usuario, lo podemos ver en la ilustración 9-11.

```

checkChat(users:string[]){
  let check=true;
  for (let i = 0; i < this.list_chats.length; i++) {
  let chat_users = this.list_chats[i].users;
  if(chat_users.length==users.length){
    for (let j = 0; j < chat_users.length; j++) {
      let user = chat_users[j];
      if(!users.includes(user.id)){
        check=false;
      }else {
        check=true;
      }
    }
    if(check){
      break;
    }
  }else{
    check=false;
  }
  }
  return check;
}

```

Ilustración 9-11 Verificación de chats

El chat contiene todos los mensajes que se han enviado y las fechas de estas. La funcionalidad principal de este componente ‘chat’, que se halla en la carpeta ‘chats’, es la de enviar mensaje al usuario o al resto de usuarios.

Para eliminar el aviso de mensajes nuevos que se muestra en la vista de chats, se modifica la propiedad ‘view’ de los mensajes al entrar al chat. Con esto, se consigue que los usuarios puedan ver en el listado de chats si existen nuevos mensajes que no ha sido leídos todavía. Esta modificación se realizó gracias a un método que se ejecuta nada más entrar en el chat. Lo podemos observar en la siguiente ilustración:

```

async setView(id_chat:string){
  let id_user=this.userService.getId();
  for (let i = 0; i < this.messages.length; i++) {
    let msg = this.messages[i];
    if(!msg.view && msg.to==id_user){
      msg.view=true;
    }
    if(!msg.view && msg.to.includes(id_user)){
      msg.view=true;
    }
  }
  console.log(this.messages)
  await this.chatService.setView(id_chat, this.messages);
}

```

Ilustración 9-12 Cambiar el estado de los mensajes

### 9.5.5. Evento

La vista del evento cuyo componente es ‘event’ en el directorio ‘events’, muestra los datos que tienen el evento y sus respectivas tareas, que se corresponden con el componente ‘task’ de la carpeta ‘task’. Además, para mayor comodidad, los usuarios tienen un buscador de tareas para encontrar una tarea en concreto. Esta vista destaca principalmente por la funcionalidad de inscripción de los voluntarios, que permite al voluntario tanto inscribirse como eliminar la inscripción de dicha tarea. Para su implementación, utilizamos uno de los componentes que ofrece Ionic que es ‘ion-alert’. Nos permite mostrar un mensaje de confirmación a través del cual el usuario puede confirmar la inscripción o la eliminación de la tarea en cuestión.

Esta confirmación solo se mostrará cuando existen plazas libres. En el caso que no existiera ninguna plaza, hemos empleado otro componente de Ionic (ion-toast) que nos enseña un mensaje temporal en el que indica al usuario que no puede realizar la inscripción en dicha tarea.

Por otro lado, los organizadores tienen el privilegio de editar o ver los voluntarios que se han inscrito en una tarea en concreto permitiendo que los organizadores no tengan que editar todas las tareas y el evento. Esta funcionalidad la podemos observar en el anexo II: Manual del usuario.

```

if(this.checkPlaza(delet)){
  const alert = await this.alertController.create({
    cssClass: 'my-custom-class',
    header: 'Confirmar',
    mode: 'ios',
    message: msg,
    buttons: [
      {
        text: 'Cancelar',
        cssClass: 'cancel-button'
      },
      {
        text: 'Confirmar',
        cssClass: 'confirm-button',
        handler: () => {
          let user=this.userService.getId();
          if(delet==''){
            this.taskService.addUserTask(user,id).then((res)=>{
              if(res !=='error'){
                if(res.voluntarios.length!=undefined){
                  this.task.n_voluntarios=res.voluntarios.length;
                }
                if(res.suplentes.length!=undefined){
                  this.task.n_suplentes=res.suplentes.length;
                }
                this.inscrito=true;
              }
            })
          }else{
            if(this.voluntario) {
              this.taskService.deleteUserTask(user,id, this.i, 'voluntarios').then( ()=>{
                this.task.n_voluntarios--;
              });
            }
            if(this.suplente) {
              this.taskService.deleteUserTask(user,id, this.i, 'suplentes').then( ()=>{
                this.task.n_suplentes--;
              });
            }
            this.inscrito=false;
          }
        }
      }
    ]
  });
  await alert.present();
}else{
  this.alertMessage(title);
}
}

```

Ilustración 9-13 Confirmación de inscripción y eliminación

Podemos decir que los organizadores tienen el privilegio de editar o ver los voluntarios que se han inscrito en una tarea en concreto permitiendo así que los organizadores no tengan que editar todas las tareas y el evento. Esta funcionalidad la podemos observar en el anexo II: Manual del usuario.

### 9.5.6. Crear evento y tareas

El componente 'create\_event' ubicado en la carpeta 'events', tiene implementado un formulario para crear los eventos. Al crear, el usuario no tendrá permitido añadir una imagen al evento, pero al editar podrá cambiar la imagen que

tiene por defecto, que se podrá apreciar en el anexo II: Manual del usuario. Una vez rellenado todos los campos del formulario, podrá ir a la vista de crear tareas siempre y cuando los datos que se han introducido son los correctos. En el caso de que los datos sean incorrectos nos mostrará los mensajes correspondientes. Esta funcionalidad es prácticamente igual que las de los formularios de iniciar sesión y registrarse.

En la interfaz de crear tareas, el organizador tendrá que rellenar el formulario de la tarea e indicar el número máximo de voluntarios y suplentes que se pueden inscribir en la tarea. Hay que destacar que el usuario podrá bloquear dicha tarea y asignar que voluntarios la realizarán. Esto último, lo podemos ver en el anexo II: Manual de usuario donde se puede apreciar dicha funcionalidad. En esta situación la única validación que existe es la de introducir el título de la tarea y los números de plazas que tendrá. No consideramos que en este formulario fuera necesario mostrar algún mensaje ya que simplemente consistía en introducir la información.

### **9.5.7. Listado de voluntarios**

El componente listado de voluntarios, situado en la carpeta ‘events’, es una vista del organizador y se caracteriza por la funcionalidad de recopilar los datos de todos los voluntarios que tiene un evento en concreto. Esta característica es fundamental en la aplicación ya que se trata de mejorar la gestión de inscripciones de los voluntarios y de proporcionar una mayor comodidad al organizador a la hora de recopilar la información de los voluntarios.

En el Manual del usuario, podemos apreciar esta funcionalidad donde se genera un fichero Excel recogiendo todos los datos de los voluntarios inscritos. Esto es posible gracias a dos plugins: ‘xlsx’ para crear un fichero Excel y ‘File’, que es un plugin oficial de Ionic. Con estas librerías hemos conseguido generar un fichero Excel y guardarlo en el sistema de fichero del móvil, cuyo método es el que se muestra en la siguiente ilustración.

```

download(){
  let datos =XLSX.utils.json_to_sheet(this.getDatos());
  let date= this.datePipe.transform(Date.now(), 'dd-MM-yyyy')
  let fileName= date.toString()+'.xlsx';
  var wb={Sheets: {'data': datos}, SheetNames: ['data']};
  var buffer = XLSX.write(wb, {bookType:'xlsx', type:'array'})
  var fileType='application/vnd.openxmlformats-officedocument.spreadsheetml.sheet';
  var data:Blob=new Blob([buffer],{type:fileType});
  this.file.writeFile("file:///storage/emulated/0/documents", fileName,data)
}

```

Ilustración 9-14 Crear y guardar fichero excel

### 9.5.8. Listado de organizadores

El listado de los organizadores permite registrar nuevos organizadores. Este componente 'list\_organizadores' se encuentra en el directorio 'user'. Su principal funcionalidad es la de listar todos los organizadores y crear nuevos usuarios como organizadores. Al igual que todos los formularios relacionados con los usuarios (iniciar sesión, registrar y editar el perfil), crear un nuevo organizador tiene validación similar a la de estos.

### 9.5.9. Buscador de eventos

Para mayor comodidad hemos considerado necesario que una de las funcionalidades fuera la de un buscador de eventos. Este componente 'search', ofrece a todos los usuarios la opción de buscar un evento en concreto.

## Capítulo 10. Pruebas.

Las pruebas son una fase crucial que se han realizado una vez terminada la implementación. Este proceso es fundamental para depurar y detectar los errores que los programadores no vemos a simple vista. Para comprobar el correcto funcionamiento de este proyecto, se han realizado varias pruebas.

Antes de comenzar con las principales pruebas, hemos realizado comprobaciones en diferentes dispositivos virtuales que se han podido instalar y compilar sin tener ningún inconveniente. Esto ha sido posible gracias a la herramienta Android Studio.

La primera de las pruebas ha consistido en garantizar que todas las funcionalidades desarrolladas se ejecutasen adecuadamente en los diferentes dispositivos móviles. Esto se ha podido efectuar gracias a la colaboración de familiares y amigos, sin ellos, no hubiera sido lo mismo.

La segunda corresponde a la parte de diseño, que ha sido la más delicada ya que en cada dispositivo había diferentes tamaños de pantalla. Esta prueba permite comprobar que los estilos y la estructura de los componentes no se descoquen. Por eso, hemos probado con diferentes tamaños para poder verificar que esta herramienta se ajustaba a distintos tipos de pantallas.

La última de las pruebas estaba relacionada con los tests de usabilidad. Esta técnica es comúnmente conocida por los desarrolladores web y permiten conocer con mayor precisión los bugs que poseen. Esta es la más importante de todas las pruebas que se han realizado ya que con ella se evalúa la aplicación con la interacción de los usuarios. Esto nos ha permitido poder ofrecer una mayor información de errores que en las anteriores pruebas.

Una vez terminado de realizar todas las comprobaciones, enumeramos los diferentes bugs que se encontraron para posteriormente resolverlos. Así pues, volvimos a repetir las pruebas para comprobar de nuevo la aplicación móvil.

# Capítulo 11. Resultados, conclusiones y futuras mejoras.

## 11.1. Resultados y conclusiones

Para concluir, podemos afirmar que se han conseguido todos los objetivos que se nos hemos propuesto al comienzo del mismo. Se ha implementado una herramienta que mejore la gestión de los voluntarios, sus inscripciones, tener la capacidad de asignarles tareas, poder hacer un seguimiento y que entre ellos se pueda establecer una correcta coordinación.

Por un lado, la aplicación que hemos creado ofrece a los organizadores las siguientes funcionalidades:

- Bloquear las tareas con la finalidad de poder asignar a los voluntarios que se desea añadir.
- La opción de poder modificar los eventos y sus respectivas tareas, así como la de poder eliminarlas.
- Añadir a nuevos usuarios con los mismos privilegios que posee un organizador.
- Gestionar los eventos y sus tareas. Así mismo, poder ver el listado de voluntarios inscriptos en un evento y una tarea en un momento dado.
- Generar y descargar ficheros Excel con la información de los voluntarios de los eventos.

Y por otro, los voluntarios pueden realizar las siguientes acciones:

- Ver los próximos eventos que se desarrollarán.
- Modificar los datos personales.
- Inscribirse y eliminarse de tareas.

Cabe destacar que los actores tienen en común la funcionalidad de chatear entre ellos con el objetivo de poder comunicarse en cualquier momento.

Todos esto, hace que los organizadores recopilen la información de los voluntarios desde la aplicación sin necesidad de emplear diversas herramientas, como por ejemplo a través de correos electrónicos, mensajes de WhatsApp, llamadas telefónicas, entre otras opciones, facilitando de esta forma toda la labor de coordinación a los organizadores.

Por todo ello, podemos decir que hemos cumplido con los objetivos y que además hemos optimizado la gestión de las inscripciones de una forma más eficiente y cómoda para estos usuarios.

Otra de las ventajas que tiene es poder permitir que todos (organizadores, responsables de un grupo y voluntarios) tengan un sistema de chat que les facilite su coordinación y no mediante otras herramientas para realizarlo, como pueden ser llamadas, a través de correo electrónico, mensajes, etc.

A nivel personal quiero destacar que este proyecto me ha resultado enriquecedor en todos los aspectos. Debido a que he podido realizar un proyecto que comenzó siendo una idea para realizar un Trabajo de Fin de Título y terminó en una aplicación que puede ser de gran utilidad para alguna empresa o algún profesional que se dedique a la realización de carreras deportivas.

Realizar las diferentes fases que hemos mencionado a lo largo de este documento, me ha enseñado lo importante que es seguir aprendiendo y conociendo las diferentes tecnologías que existen actualmente en el mundo laboral. Es importante conocer las herramientas que hoy en día son muy necesarias para poder desarrollar aplicaciones como esta.

## **11.2. Futuras mejoras**

Hoy en día, cualquier proyecto software puede tener nuevas funcionalidades e incluso estas se pueden mejorar. Esta aplicación no deja de ser una de ellas, ya que en ella se pueden incorporar diferentes funcionalidades.

Entre ellas, podemos describir algunas características nuevas y posibles mejoras a las funcionalidades que hemos implementado con el objetivo de innovar y perfeccionar este sistema:

- Enviar notificaciones al correo electrónico con el objetivo de recordar al usuario que se ha inscrito en un evento.
- Crear un recordatorio en el calendario como aviso.
- Tener una interfaz mostrando un listado de las diferentes notificaciones pendientes.
- Permitir que los usuarios creen en ese mismo instante fotos al modificar su foto de perfil.
- Editar una tarea en el proceso de creación de un evento.
- Implementar un sistema de recuperación de contraseña.
- Desarrollar un control de asistencia con la finalidad de localizar a los diferentes voluntarios y organizadores el día del evento.
- Añadir unas propiedades nuevas que permitan que el voluntario pueda indicar si tiene el carnet de conducir o qué idiomas habla.

## Capítulo 12. Bibliografía.

- [1] Evenbrite, «EventBrite Organizador - Tienda Online Google Play,» [En línea]. Disponible:  
<https://play.google.com/store/apps/details?id=com.eventbrite.organizer>.  
[Último acceso: 10 Marzo 2022].
- [2] L. S. Inc., «TeamHub - Tienda Online Google Play,» [En línea]. Disponible:  
<https://play.google.com/store/apps/details?id=com.teamhub>. [Último acceso:  
15 Marzo 2022].
- [3] RunnApp, «RunnApp Buscador de Carreras Populares - Tienda Online Google Play,» [En línea]. Disponible:  
[https://play.google.com/store/apps/details?id=appinventor.ai\\_alejandro\\_perez.runn](https://play.google.com/store/apps/details?id=appinventor.ai_alejandro_perez.runn). [Último acceso: 19 Marzo 2022].
- [4] Wikipedia, «GNU GPL License,» [En línea]. Disponible:  
[https://es.wikipedia.org/wiki/GNU\\_General\\_Public\\_License](https://es.wikipedia.org/wiki/GNU_General_Public_License). [Último acceso:  
17 Julio 2022].
- [5] Wikipedia, «Licencia MIT,» Wikipedia, [En línea]. Disponible:  
[https://es.wikipedia.org/wiki/Licencia\\_MIT](https://es.wikipedia.org/wiki/Licencia_MIT). [Último acceso: 17 Julio 2022].
- [6] Wikipedia, «Apache License,» Wikipedia, [En línea]. Disponible:  
[https://es.wikipedia.org/wiki/Apache\\_License](https://es.wikipedia.org/wiki/Apache_License). [Último acceso: 18 Julio 2020].
- [7] Microsoft, «Microsoft Office 365,» Microsoft, [En línea]. Disponible:  
<https://support.microsoft.com/es-es/office/-qu%C3%A9-es-microsoft-365-847caf12-2589-452c-8aca-1c009797678b>. [Último acceso: 31 Mayo 2022].
- [8] BOE, «Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales,» BOE, [En línea]. Disponible:

<https://www.boe.es/buscar/pdf/2018/BOE-A-2018-16673-consolidado.pdf>.

[Último acceso: 18 Julio 2022].

[9] Wikipedia, «Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales,» Wikipedia, [En línea]. Disponible: [https://es.wikipedia.org/wiki/Ley\\_Org%C3%A1nica\\_de\\_Protecci%C3%B3n\\_de\\_Datos\\_Personales\\_y\\_garant%C3%ADa\\_de\\_los\\_derechos\\_digitales](https://es.wikipedia.org/wiki/Ley_Org%C3%A1nica_de_Protecci%C3%B3n_de_Datos_Personales_y_garant%C3%ADa_de_los_derechos_digitales). [Último acceso: 18 Julio 2022].

[10] Wikipedia, «Reglamento General de Protección de Datos,» Wikipedia, [En línea]. Disponible:

[https://es.wikipedia.org/wiki/Reglamento\\_General\\_de\\_Protecci%C3%B3n\\_de\\_Datos](https://es.wikipedia.org/wiki/Reglamento_General_de_Protecci%C3%B3n_de_Datos). [Último acceso: 18 Julio 2022].

[11] Wikipedia, «SCRUM (desarrollo de software),» Wikipedia, [En línea]. Disponible: [https://es.wikipedia.org/wiki/Scrum\\_\(desarrollo\\_de\\_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)). [Último acceso: 18 Julio 2022].

[12] Wikipedia, «Figma,» Wikipedia, [En línea]. Disponible: <https://es.wikipedia.org/wiki/Figma>. [Último acceso: 30 Mayo 2022].

[13] StarUML, «Introduction,» StarUML, [En línea]. Disponible: <https://docs.staruml.io/>. [Último acceso: 30 Mayo 2022].

[14] Wikipedia, «HTML,» Wikipedia, [En línea]. Disponible: <https://es.wikipedia.org/wiki/HTML>. [Último acceso: 30 Mayo 2022].

[15] Wikipedia, «CSS,» Wikipedia, [En línea]. Disponible: <https://es.wikipedia.org/wiki/CSS>. [Último acceso: 30 Mayo 2022].

[16] Wikipedia, «TypeScript,» Wikipedia, [En línea]. Disponible: <https://es.wikipedia.org/wiki/TypeScript>. [Último acceso: 30 Mayo 2022].

[17] Ionic, «Installing Ionic,» Ionic, [En línea]. Disponible: <https://ionicframework.com/docs/intro/cli>. [Último acceso: 30 Mayo 2022].

- [18] Wikipedia, «Ionic (mobile app framework),» Wikipedia, [En línea]. Disponible: [https://en.wikipedia.org/wiki/Ionic\\_\(mobile\\_app\\_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework)). [Último acceso: 30 Mayo 2022].
- [19] Angular, «Introduction to the Angular Docs,» Angular, [En línea]. Disponible: <https://angular.io/docs>. [Último acceso: 03 Junio 2022].
- [20] Wikipedia, «Angular (framework),» Wikipedia, [En línea]. Disponible: [https://es.wikipedia.org/wiki/Angular\\_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework)). [Último acceso: 03 Junio 2022].
- [21] Firebase, «Elige una base de datos: Cloud Firestore o Realtime Database,» Firebase, [En línea]. Disponible: <https://firebase.google.com/docs/database/rtdb-vs-firestore>. [Último acceso: 30 Mayo 2022].
- [22] Wikipedia, «Firebase,» Wikipedia, [En línea]. Disponible: <https://es.wikipedia.org/wiki/Firebase>. [Último acceso: 30 Mayo 2022].
- [23] Firebase, «Documentación,» Firebase, [En línea]. Disponible: <https://firebase.google.com/docs>. [Último acceso: 30 Mayo 2022].
- [24] Wikipedia, «Microsoft Visual Studio Code,» Wikipedia, [En línea]. Disponible: [https://es.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://es.wikipedia.org/wiki/Visual_Studio_Code). [Último acceso: 30 Mayo 2022].
- [25] A. Studio, «Introducción a Android Studio,» Android Studio, [En línea]. Disponible: <https://developer.android.com/studio/intro?hl=es-419>. [Último acceso: 30 Mayo 2022].
- [26] Wikipedia, «Bootstrap,» Wikipedia, [En línea]. Disponible: [https://es.wikipedia.org/wiki/Bootstrap\\_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework)). [Último acceso: 30 Mayo 2022].

- [27] Git, «Git,» Git, [En línea]. Disponible: <https://git-scm.com/>. [Último acceso: 30 Mayo 2022].
- [28] Wikipedia, «GitHub,» Wikipedia, [En línea]. Disponible: <https://es.wikipedia.org/wiki/GitHub>. [Último acceso: 30 Mayo 2022].
- [29] Wikipedia, «Trello,» Wikipedia, [En línea]. Disponible: <https://es.wikipedia.org/wiki/Trello>. [Último acceso: 31 Mayo 2022].
- [30] M. Office, «Office 365 Education,» Microsoft Office, [En línea]. Disponible: <https://www.microsoft.com/es-es/education/products/office#:~:text=Los%20alumnos%20y%20educadores%20de,v%C3%A1lida%20y%20empiece%20hoy%20mismo..> [Último acceso: 17 Julio 2022].

# Anexos

## Anexo I: Especificación de casos de uso

CU - 01	<<CRUD>> Eventos
<b>Descripción</b>	El organizador podrá crear, visualizar, modificar o eliminar eventos.
<b>Precondiciones</b>	El organizador debe haber iniciado sesión.
<i>Flujo normal</i>	
1	El organizador crea un evento.
1.a	El organizador rellena los campos necesarios y se pulsa el crear evento.
2	El organizador va a el evento que quiere modificar.
2.a	Modifica los campos del evento y lo actualiza pulsando en el botón actualizar.
3	El organizador visualiza el evento.
4	El organizador elimina el evento.
<i>Extensiones</i>	
1	El organizador decide que CU-2 (Tareas) tareas tendrá el evento
<b>Postcondiciones</b>	La base de datos almacena los eventos.
<i>Excepciones</i>	
1.a.1	Se pierde la conexión a la base de datos durante el proceso de creación, modificación o eliminación del evento.
1.a	Se manda una notificación indicando el error.
1.b	Los datos no se guardan en la base de datos y el evento no se crea o actualiza.
1.c	Se mantiene en la base de datos el evento.
2.a.1	No se modifica ningún dato o se dejan datos en blanco
2.a	Se manda una notificación indicando que los datos son incorrectos.

CU - 02		<<CRUD>> Tareas	
<b>Descripción</b>	El organizador puede crea, visualiza, modifica o eliminar tareas.		
<b>Precondiciones</b>	Crear un evento.		
<i>Flujo normal</i>			
1	El organizador crea la tarea.		
	1.a	Se rellena los campos y se pulsa el botón crear tarea	
2	El organizador modifica la tarea.		
	2a	Se modifica los campos pertinentes y se pulsa en actualizar tarea.	
3	El organizador visualiza las tareas que tiene el evento.		
4	El organizador elimina una tarea.		
<i>Variantes</i>			
1	Crear tareas		
	1.a	Se bloquea la inscripción de voluntarios y el organizador escoge los voluntarios para las tareas.	
	1.b	Se establece una fecha límite para que los voluntarios se inscriban	
2	Modificar tareas.		
	2.a	Desbloquear la inscripción de voluntarios.	
	2.b	Se cambia la fecha límite para la inscripción.	
	2.c	Se aumenta o disminuye el número de voluntarios que se puede inscribir.	
<b>Postcondiciones</b>	La base de datos guarda las tareas que tiene cada evento.		
<i>Excepciones</i>			
1	No se han colocado todos los datos necesarios.		
	1.a	Se manda una notificación indicando el error.	

	1.b	Los datos no se guardan en la base de datos y la tarea no se crea.
2		Se pierde la conexión a la base de datos durante el proceso de creación, modificación o eliminación de la tarea.
	1.a	Se manda una notificación indicando el error.
	1.b	Los datos no se guardan en la base de datos y la tarea no se crea o actualiza.
	1.c	Se mantiene en la base de datos la tarea
3		No se modifica ningún dato o se dejan datos en blanco
	3.a	Se manda una notificación indicando que los datos son incorrectos.

CU - 03		Chat de incidencias	
<b>Descripción</b>	Se comunicará las incidencias mediante mensajes de texto.		
<b>Precondiciones</b>	El organizador y el/los voluntario/s son del mismo evento.		
<i>Flujo normal</i>			
1	Enviar un mensaje de texto.		
	1.a	Se escribe el mensaje y se pulsa el botón de enviar.	
<i>Variantes</i>			
1	Enviar un mensaje de texto		
	1.a	El organizador envía un mensaje a un voluntario o a todos.	
	1.b	Un voluntario envía un mensaje al organizador o a otros voluntarios.	
<b>Postcondiciones</b>	El organizador y el/los voluntario/s están informados.		
<i>Excepciones</i>			
1	El mensaje está vacío y no se envía el mensaje.		
2	Se ha perdido la conexión.		
	2.a	Se manda una notificación indicando el error.	
	2.b	No se envía el mensaje.	

CU - 04 Registrar	
<b>Descripción</b>	Se registran los voluntarios.
<b>Precondiciones</b>	
<i>Flujo normal</i>	
1	Registrar.
	1.a. Se rellenan los campos y se pulsa el botón registrar.
<i>Variantes</i>	
1	Registrar
	1.a. Un voluntario se registra
	1.b. Un responsable de un grupo de voluntarios registra este grupo.
<b>Postcondiciones</b>	La base de datos añade a los voluntarios.
<i>Excepciones</i>	
1	No se han colocado todos los datos necesarios.
	1.a. Se manda una notificación indicando el error.
	1.b. Los datos no se guardan en la base de datos y el/los voluntarios no sé registra/n.
2	Se pierde la conexión a la base de datos durante el proceso de creación, modificación o eliminación de la tarea.
	Se manda una notificación indicando el error.
	Los datos no se guardan en la base de datos y la tarea no se crea o actualiza.
	Se mantiene en la base de datos la tarea

CU - 05 <<report>> Voluntarios inscritos	
<b>Descripción</b>	El organizador visualiza un listado de los voluntarios inscritos.
<b>Precondiciones</b>	Tener voluntarios inscritos al evento.
<i>Flujo normal</i>	
1	El organizador visualiza a todos los voluntarios inscritos con sus respectivos datos.
<b>Postcondiciones</b>	El organizador tiene un listado con todos los voluntarios del evento.
<i>Excepciones</i>	
1	Se ha perdido la conexión a la base de datos.
1.a	Se manda una notificación indicando el error.
1.b	Los datos no se muestran.

CU - 06 <<report>> Buscar evento	
<b>Descripción</b>	El organizador busca un evento.
<b>Precondiciones</b>	Tener un evento creado y almacenado en la base de datos.
<i>Flujo normal</i>	
1	El organizador busca un evento.
<b>Postcondiciones</b>	El organizador localiza el evento.
<i>Excepciones</i>	
1	Se ha perdido la conexión a la base de datos.
1.a	Se manda una notificación indicando el error.
1.b	El/los evento/s no se muestran.
2	El evento no existe en la base de datos.
2.a	Se manda una notificación indicando que los datos introducidos no coinciden con ningún evento existente.

<b>CU - 07</b>	Editar datos personales
<b>Descripción</b>	El organizador y el/los voluntario/s puede modificar sus datos personales.
<b>Precondiciones</b>	Estar registrado.
<i>Flujo normal</i>	
1	Se pulsa el botón perfil
	1.a Se muestra los datos personales y se modifican.
	1.b. Se pulsa el botón actualizar datos.
<b>Postcondiciones</b>	La base de datos se actualiza.
<i>Excepciones</i>	
1	Se ha perdido la conexión a la base de datos.
	1.a Se manda una notificación indicando el error.
	1.b Los datos personales no se actualizan en la base de datos.
2	No se modifica ningún dato o se dejan datos en blanco
	2.a Se manda una notificación indicando que los datos son incorrectos.

## Anexo II: Manual del usuario

Para poder utilizar la aplicación móvil todos los usuarios deben iniciar sesión con el formulario de la ilustración 1-1.



The image shows a mobile application login screen with a light blue background. At the top, the text 'Iniciar sesión' is centered. Below this are two white input fields with rounded corners. The first field is labeled 'Email' and the second is labeled 'Contraseña'. Underneath the second field, there is a link that reads '¿No tienes cuenta? Registrarse'. The word 'Registrarse' is highlighted with a purple oval. At the bottom of the screen, there is a prominent orange button with the text 'INICIAR SESIÓN' in white capital letters.

Ilustración 1-1 Iniciar sesión.

Si un usuario todavía no tiene una cuenta para poder entrar tiene la posibilidad de registrarse. Para ello debe pulsar el botón 'Registrarse' (Ilustración 1-1, círculo de color lila). Con esta acción, se le redirige a una vista en la que debe seleccionar el rol que es y aceptar los términos, las condiciones de uso y la política de privacidad. Una vez que ha realizado el paso anterior, podrá seguir con el proceso. Esto último se muestra en la siguiente ilustración:



Ilustración 1-2 Identificación del usuario y aceptación de condiciones.

Las opciones que tiene el usuario para identificarse son las que se muestran en la ilustración 1-3.

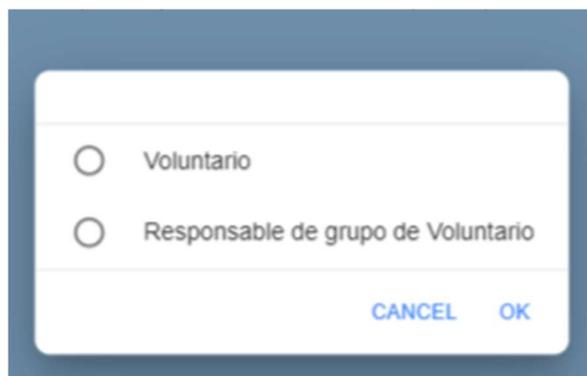


Ilustración 1-3 Opciones para identificarse.

Si se escoge la opción ‘voluntario’, el usuario tendrá un formulario en el que debe rellenar todos los campos que se observan en la siguiente captura. No se puede dejar ningún campo sin rellenar ni introducir los datos incorrectamente ya que no le permitirá registrarse. Así mismo, si se insertan erróneamente aparecerán uno/s mensaje/s comunicando la información que es incorrecta y la forma en que se debe introducir (Ilustración 1-4, círculo de color verde). Posteriormente podrá registrarse pulsando el botón ‘Registrarse’ que le llevará a la interfaz de inicio.

← Registrarse

Nombre:  
Nuevo

Apellidos:  
nuevo

E-mail:  
nuevo@gmail.com

DNI/NIF/Pasaporte:  
ⓘ Es necesario introducir el DNI/NIF/Pasaporte

Tel.:  
12  
ⓘ El formato del teléfono es incorrecto

Selecciona una talla de camiseta  
S

Contraseña:  
\*\*\*\*\*

Confirmar contraseña:  
12356  
ⓘ No puede dejar la repetir contraseña vacía ⓘ  
Deben coincidir

Registrarse

Ilustración 1-4 Formulario para registrarse un voluntario.

Si la opción elegida es 'Responsable de un grupo de voluntarios', se le redirige al formulario de la ilustración 1-5. Al igual que en el caso anterior, el usuario debe rellenar los campos adecuadamente y, también, le avisará al usuario de los datos que ha introducido mal.

← Responsable de un grupo de voluntarios

Nombre:  
respuesta 2

Apellidos:  
apellidos

E-mail:  
resp2@gmail.com

DNI/NIF/Pasaporte:  
12323232r

Tif.:  
900983932

Contraseña:  
\*\*\*\*\*

Confirmar contraseña:  
\*\*\*\*\*

SIGUIENTE

Ilustración 1-5 Formulario de registro del responsable.

Una vez que termine y pulse el botón ‘siguiente’, el usuario tendrá que rellenar los formularios de los voluntarios (Ilustración 1-6). Como en los anteriores, tendrá que añadir correctamente la información para poder registrarlos.

← Registrar voluntario

Nombre:  
voluntario 1

Apellidos:  
apel

E-mail:  
vol1@gmail.com

DNI/NIF/Pasaporte:  
34343321s

Tif.:  
909889892

Selecciona una talla de camiseta  
XS

Contraseña:  
\*\*\*\*\*

Confirmar contraseña:  
\*\*\*\*\*

**CREAR VOLUNTARIO**

Ilustración 1-6 Formulario de registro de un grupo de voluntarios.

Después de pulsar el botón ‘Crear voluntario’ (Ilustración 1-6, círculo de color verde), obtendrá el listado de los voluntarios que ha ido añadiendo. Este listado le permitirá editarlos (Ver siguiente ilustración, recuadro de color lila), eliminarlos pulsando el botón de la ‘X’ y también añadir a otros voluntarios haciendo clic en el botón ‘Crear’ (círculo de color verde).



Ilustración 1-7 Listado para registrar a los voluntarios.

Para acabar de registrar, el usuario tendrá que pulsar el botón 'Registrar' y este le llevará a la vista de inicio (Ilustración 1-8).

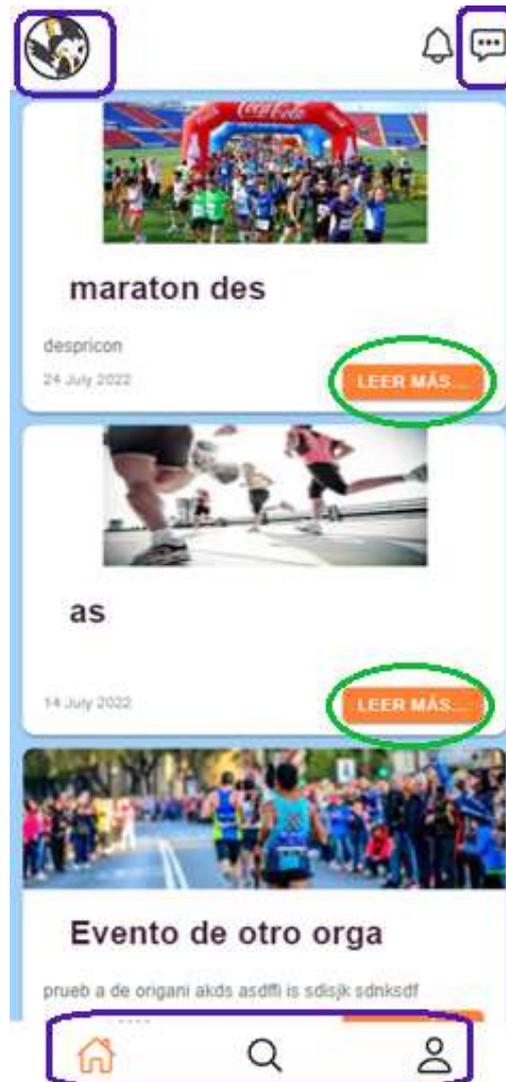


Ilustración 1-8 Inicio.

El inicio tiene diferentes acciones que se corresponden con los círculos de color lila y verde de la ilustración anterior. Si el usuario no ha cerrado la sesión, la aplicación le llevará a la interfaz de inicio. Los de verde conducen a la información del evento. En la foto se aprecia en la esquina superior izquierda: se trata de un menú en el que se enseñan los diferentes privilegios que tiene un usuario. Estos privilegios se pueden observar en las siguientes ilustraciones.

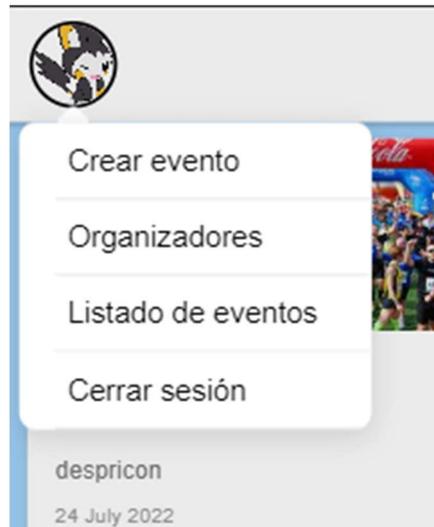


Ilustración 1-9 Menú de un organizador.

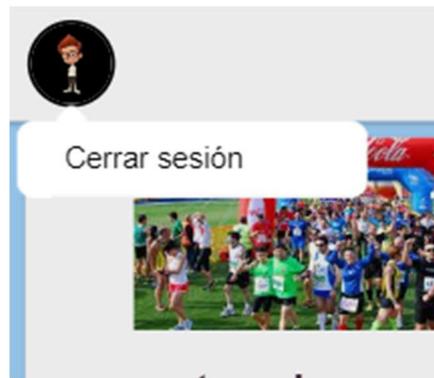


Ilustración 1-10 Menú de un voluntario.

En la ilustración 1-9, se observan cuatro opciones que son las que posee un organizador. En la anterior solo hay una opción que corresponde a la de un voluntario o un responsable de un grupo.

Si nos fijamos en la esquina superior derecha de la ilustración 1-8, hay un botón que permite a todos los usuarios entablar una conversación con otros usuarios. Este botón redirige a la vista que se muestra en la siguiente captura.



Ilustración 1-11 Crear un nuevo chat.

Como se puede observar, el usuario tiene la posibilidad de crear nuevos chats mediante el botón 'Nuevo' (Ilustración 1-11, círculo de color verde). Al pulsarlo, muestra una sección en la que se permite buscar un/os usuario/s y seleccionarlos. Esto se observa en la siguiente captura.

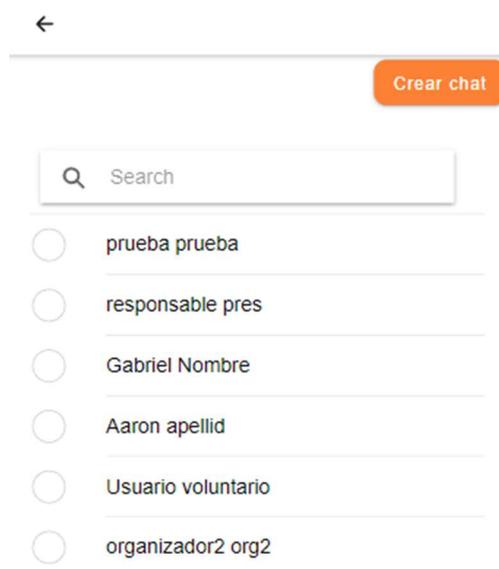


Ilustración 1-12 Seleccionar los usuarios para crear un chat.

En cambio si se elige a un/os usuario/s en el que ya existe un chat formado por estos, la aplicación enseñará al usuario un mensaje de aviso y no le permitirá crear el chat (Ilustración 1-13, círculos de color verde).

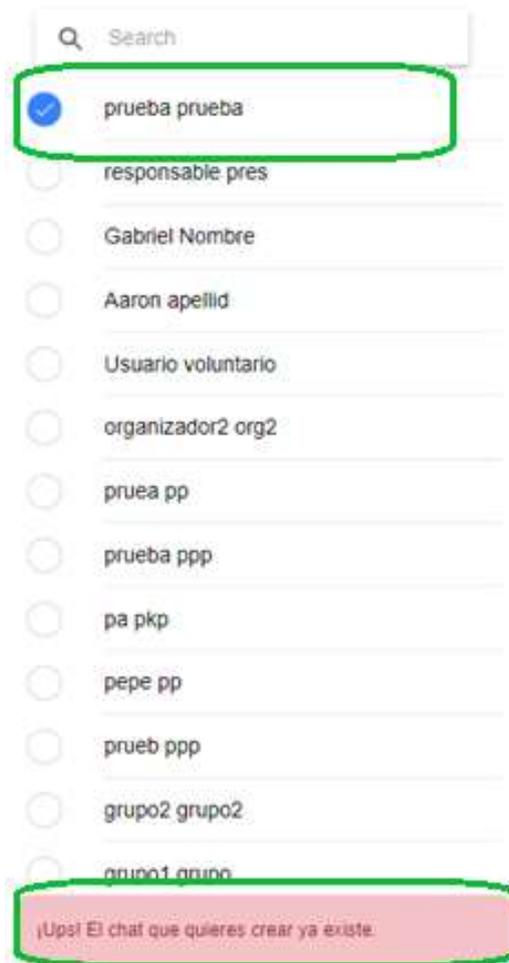


Ilustración 1-13 Aviso: el chat existe.

Un chat puede contener nuevos mensajes que todavía los usuarios receptores no han leído. Para ello, en la vista chats se le avisa mediante un punto de color azul con el número de mensajes nuevos. Esto lo podemos observar en el recuadro lila de la siguiente ilustración.

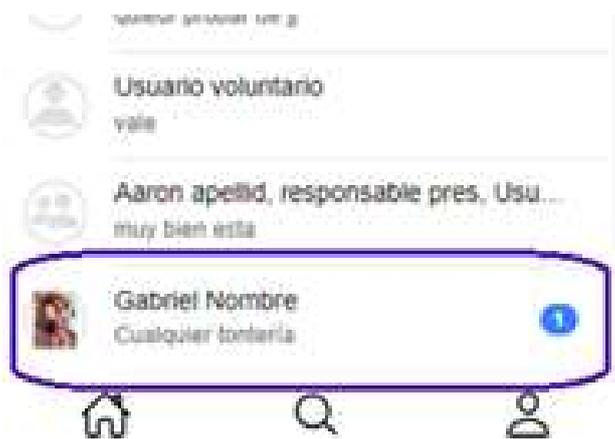


Ilustración 1-14 Aviso: Mensajes nuevos sin leer.

Si ya se ha creado el chat, para comunicarse con un usuario o un grupo de personas, solo se hacer clic en el propio chat. Este le dirige a la conversación que tiene con los otros usuarios y los mensajes que han escrito con las fechas en las que se han enviado (Ilustración 1-15). De la misma manera, para escribir un mensaje y enviarlo hay pulsar el botón de color naranja como se apreciar en la siguiente captura:



Ilustración 1-15 Chat.

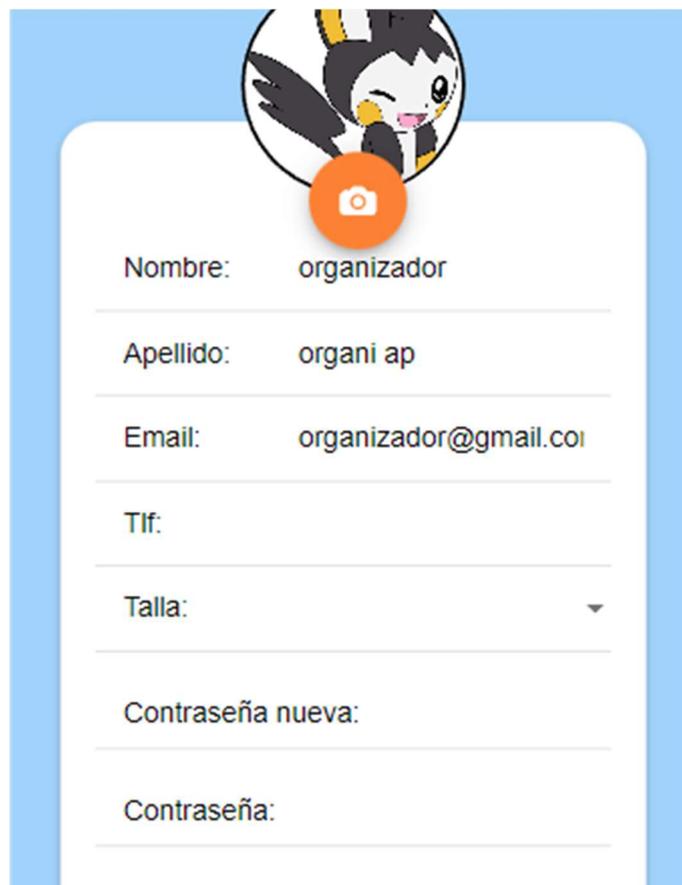
Si retrocedemos a Chats, el usuario verá que ya no se muestra el aviso de mensajes nuevos (Ilustración 1-16, círculo rojo).



Ilustración 1-16 Aviso: no existe mensajes nuevos sin leer.

Otra de las acciones que existe es la de ver el perfil del usuario. Para llegar a esta interfaz hay que pulsar el botón que se encuentra en la esquina inferior derecha (Ilustración 1-8, círculo inferior lila).

El perfil permite a un usuario ver y actualizar los datos que posee. En el caso de que fuera un organizador, se aparecerá la siguiente información:



The image shows a user profile form for an organizer. At the top, there is a circular profile picture of a cartoon character with black and white fur, yellow accents, and a camera icon overlaid on it. Below the profile picture, the form contains several input fields with labels and values:

Nombre:	organizador
Apellido:	organi ap
Email:	organizador@gmail.co
Tlf:	
Talla:	▼
Contraseña nueva:	
Contraseña:	

Ilustración 1-17 Perfil de un organizador.

Para modificar los datos es imprescindible introducir la contraseña en el campo 'Contraseña' para garantizar que es el usuario y no otra persona que modifique esta información. Si no lo realiza, no se le modifican los datos cambiados y el sistema avisará con un mensaje en el que le advierte de que debe insertar la contraseña para que se actualice (Ilustración 1-18, recuadro lila).

The image shows a user profile form with a light blue background. At the top is a circular profile picture of a cartoon character with black and white fur and yellow accents. Below the picture is a camera icon. The form contains several input fields: 'Nombre:' with the value 'organizador', 'Apellido:' with 'apellidos del organiz', 'Email:' with 'organizador@gmail.com', 'Tlf:', 'Talla:', and 'Contraseña nueva:'. A red asterisk and the word 'Contraseña:' are visible in the password field. An orange button labeled 'Modificar datos' is at the bottom. A red warning message at the bottom of the form reads: 'ⓘ Para modificar los datos tiene que escribir la contraseña'.

Ilustración 1-18 Aviso: no se ha introducido la contraseña para modificar la información.

Una persona que tiene el rol de voluntario se le permite ver los demás campos: el número de teléfono y la talla de la camiseta. Así se observa en el recuadro lila de la siguiente captura.



Ilustración 1-19 Perfil de un voluntario.

Además, cualquier usuario puede cambiar la foto de perfil pulsando en el botón con el símbolo de una cámara. Como se puede ver en la ilustración 1-20, en la que el voluntario anteriormente tenía una imagen por defecto (Ilustración 1-19, círculo verde) en la siguiente captura se le mostrará una foto:

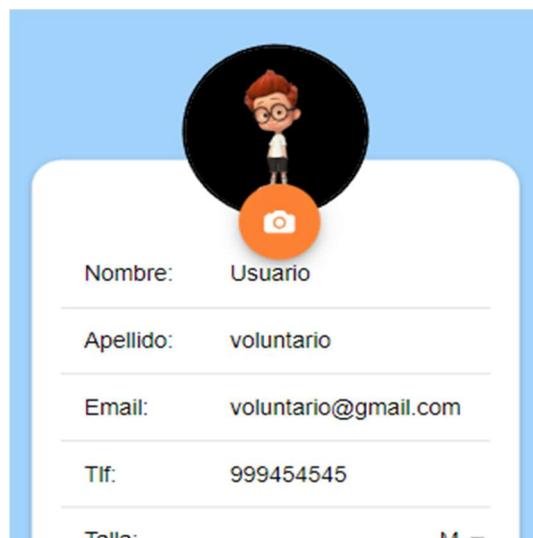


Ilustración 1-20 Modificar la foto de perfil.

Para mayor comodidad, existe una sección en la que el usuario puede buscar un/os evento/s. Esta vista se puede ver pulsando en el botón con el símbolo de una lupa situada en la parte inferior.

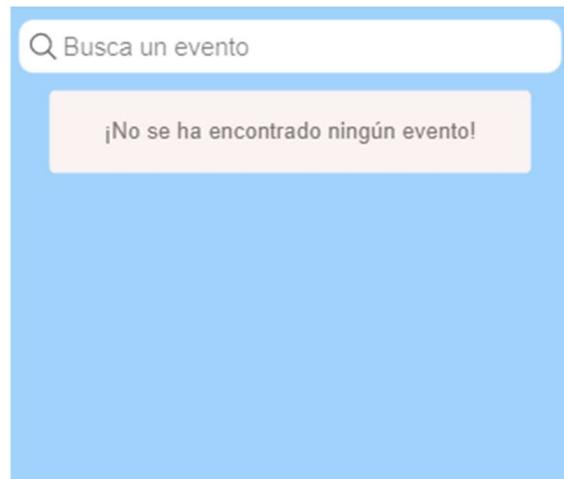


Ilustración 1-21 Buscador de eventos.

Este buscador encuentra los eventos cuyo título contenta la/s palabra/s clave (Ilustración 1-22, círculos rojos).



Ilustración 1-22 Buscar un evento que coincida con la palabra clave.

En la interfaz del evento detallado existen diferentes acciones. Para facilitar la inscripción o encontrar una tarea en concreto, hay un buscador que les permite a los usuarios localizar tareas.

Si la persona que ha iniciado sesión es un organizador y también, es quién ha creado el evento, tendrá las opciones de ver a los voluntarios inscritos en ellas y modificar las tareas. Esto último se puede ver en los círculos lilas de la siguiente captura:



Ilustración 1-23 Privilegio de un organizador en la descripción del evento.

Para que el organizador pueda ver un listado de voluntarios inscritos en una tarea, simplemente hay que pulsar el botón 'ver' en el que mostrará este grupo de voluntarios y suplentes. Existe la posibilidad de retroceder al evento haciendo clic en el botón 'volver'.



Ilustración 1-24 Ver los voluntarios inscriptos.

También, este usuario puede editar una tarea en concreto pulsando en el botón ‘editar’ de la ilustración 1-23. Este botón redirige al organizador a un formulario con los datos de la tarea como se observa en la siguiente captura.

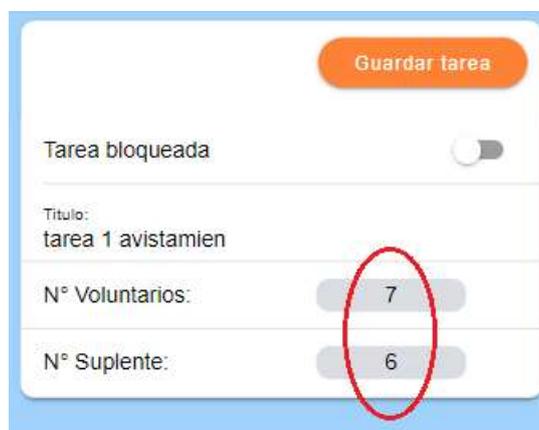


Ilustración 1-255 Modificar una tarea.

Si se modifican los campos, por ejemplo, el número de voluntarios y suplentes, se puede ver que la tarea ha sido modificada en el evento (Ilustración 1-26, círculo rojo).



Ilustración 1-266 Actualización de una tarea.

De la misma manera, el voluntario tiene las opciones de inscribirse o eliminar la inscripción en la tarea. (Ilustración 1-27, círculos verde y lila).



Ilustración 1-277 Privilegio de un voluntario en la descripción del evento.

Para confirmar la acción de estas dos opciones, aparecerá un mensaje en el que se tiene que confirmar dicha operación. Dependiendo de la que sea, se mostrará un mensaje u otro. Esto se observa en las dos siguientes capturas resaltando los mensajes con colores.



Ilustración 1-288 Confirmación de la inscripción.

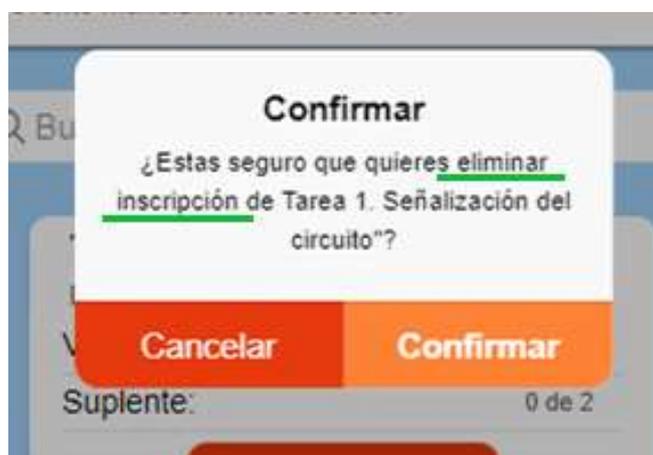


Ilustración 1-299 Confirmación de la eliminación de la inscripción.

Hay que señalar que un organizador puede inscribirse en otros eventos que no ha creado. Esto se corresponde con el círculo verde de la siguiente captura.

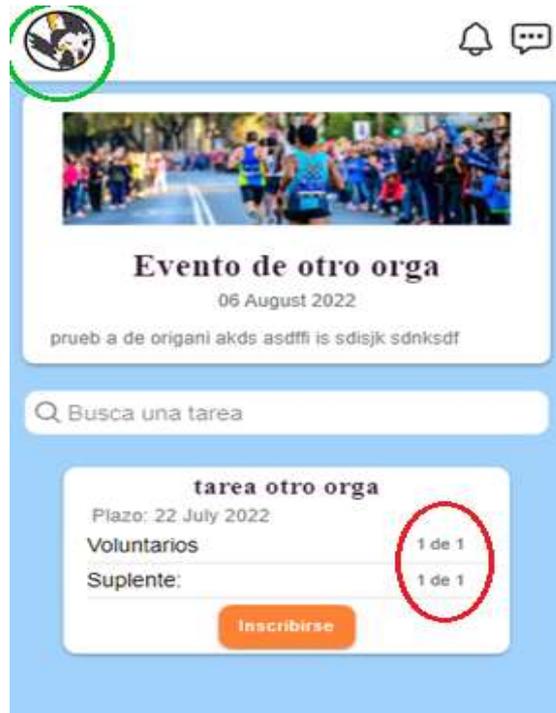


Ilustración 1-30 Tarea sin plazas libres.

Si un usuario quiere inscribirse en una tarea en la que no hay plazas disponibles, como por ejemplo se observa en la anterior captura, no le permitirá realizar dicha acción y, además, le mostrará un mensaje de aviso que le indicará que no se ha podido inscribir en dicha tarea. Esto se corresponde con el recuadro lila de la ilustración 1-31.



Ilustración 1-31 Aviso: no se puede inscribir.

De los diferentes privilegios que tiene el organizador, uno de ellos es el de gestionar los eventos que ha creado. Este se puede ver pulsando en el botón ‘listado de eventos’ que se encuentra en el menú de la ilustración 1-9.



Ilustración 1-32 Listado de eventos.

En esta sección, el usuario puede eliminar (círculos verdes), modificar y ver los voluntarios inscriptos en ese evento (círculos lilas). Pulsando en el botón ‘Editar’ de un evento, la herramienta redirige a un formulario con los datos del evento (Ilustración 1-33).

Titulo:  
as

Descripción:

Seleccionar archivo Ninguno archivo selec.

Fecha del evento:  
14/07/2022

Organizador: organizador

Plazo de inscripción:  
13/07/2022

Ilustración 1-33 Modificar un evento.

Al igual que en otros formularios, si dejamos los campos vacíos, a excepción del de la foto, mostrará los mensajes en los que avisa al usuario de que debe insertar la información (Ilustración 1-34, recuadro de color lila).

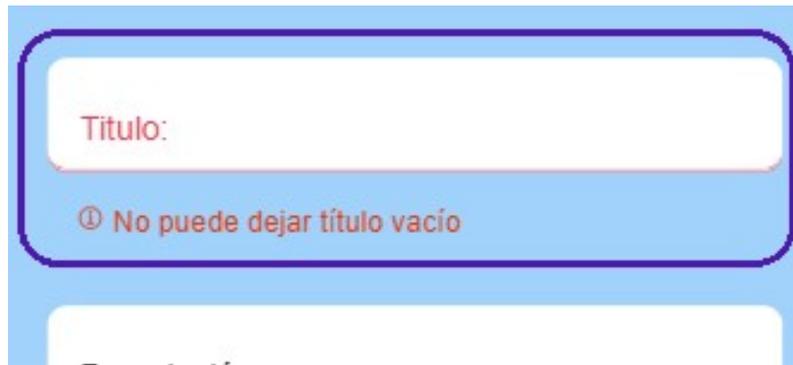


Ilustración 1-34 Aviso: no se puede dejar el título vacío.

Otra operación que tiene un organizador es la de listar a los voluntarios que se han inscrito. Para ello, se pulsa en el botón 'Ver voluntarios', esta nos conduce a una sección que muestra a todos los voluntarios. Esto se puede ver en la siguiente ilustración.

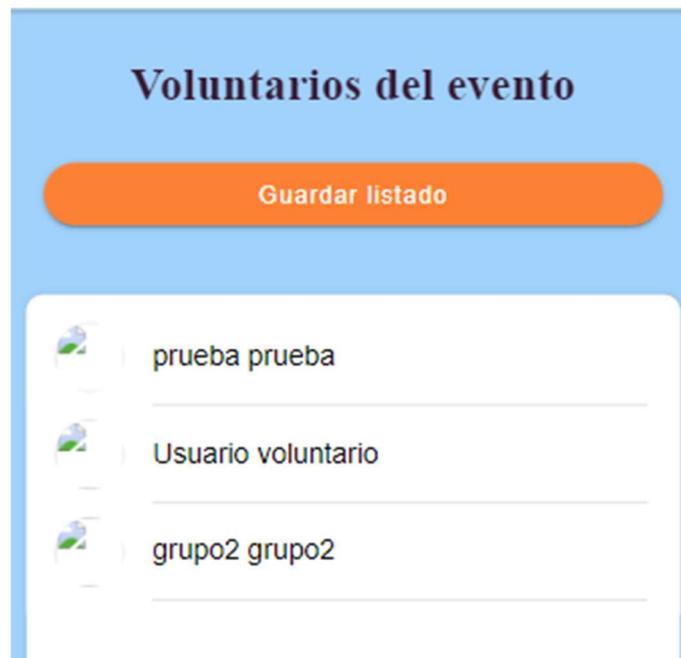


Ilustración 1-35 Listado de voluntarios inscritos en un evento.

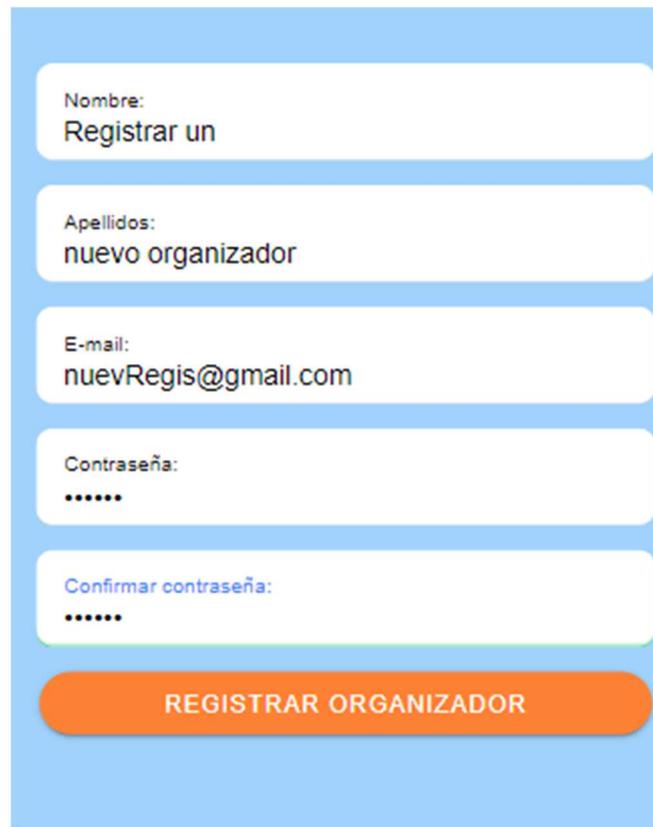
Hay que señalar que esta vista, haciendo clic en el botón 'Guardar listado', crea y guarda un fichero Excel en la carpeta 'Documentos' del sistema de ficheros del dispositivo móvil. Este archivo contiene la información de estos usuarios.

Para crear un nuevo organizador, se necesita ir al menú (Ilustración 1-9) y pulsar en el botón 'Organizador'. Esta sección lista a los organizadores que existen en la aplicación haciendo clic en el botón 'Crear' que está señalado con un círculo lila en la siguiente captura:



Ilustración 1-36 Crear un nuevo organizador.

Como en los demás formularios, hay que introducir correctamente los datos ya que mostrará los avisos con los datos incorrectos, Esto se observa en las dos siguientes ilustraciones.



Nombre:  
Registrar un

Apellidos:  
nuevo organizador

E-mail:  
nuevRegis@gmail.com

Contraseña:  
\*\*\*\*\*

Confirmar contraseña:  
\*\*\*\*\*

**REGISTRAR ORGANIZADOR**

Ilustración 1-37 Formulario de registro de un organizador.

← Organizador

Nombre:  
ⓘ No puede dejar nombre vacío

Apellidos:  
ⓘ No puede dejar apellido vacío

E-mail:  
ⓘ No puede dejar el email vacío

Contraseña:  
ⓘ No puede dejar la contraseña vacía

Confirmar contraseña:  
ⓘ No puede dejar la repetir contraseña vacía

REGISTRAR ORGANIZADOR

Ilustración 1-38 Aviso: los campos no pueden quedar vacío.

Una vez rellenado el formulario, pulsamos el botón ‘Registrar organizador’ y mostrará un mensaje en el que hay que introducir la contraseña del organizador que ha iniciado sesión. Esto último se puede apreciar en la siguiente ilustración.

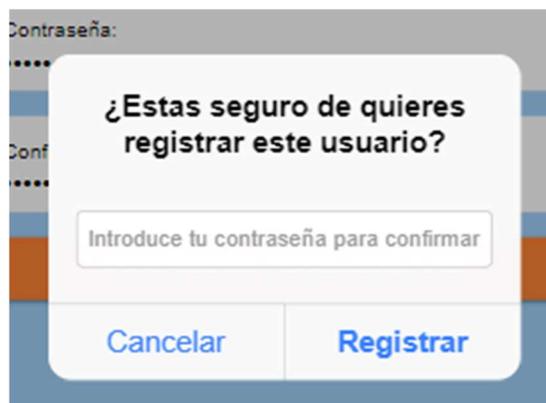


Ilustración 1-39 Confirmación de registro.

Al confirmar dicha operación, el organizador nuevo aparecerá en el listado de organizadores como se muestra en el círculo verde de la ilustración 1-40.



Ilustración 1-40 Nuevo organizador.

Para crear un evento, es necesario ir al menú que muestra en la ilustración 1-9 y hacer clic en el botón 'Crear evento' que redirige a un formulario.

Descripción detallada. Prueba de descripción de un maratón. Se celebrará las 15:00. Consiste en un evento mundialmente conocido.

Seleccionar archivo Ninguno archivo selec.

Fecha del evento:  
06/08/2022

Organizador: organizador

Plazo de inscripción:  
02/08/2022

Siguiente

Ilustración 1-41 Crear un evento.

En esta vista hay que señalar que existe una validación de los datos, sobre todo de las fechas introducidas. En el caso de indicar que el plazo es posterior al día del evento, no se podrá seguir con el proceso de creación hasta corregirlo (Ilustración 1-42, recuadros de color lila).

← Crear evento

Título:  
Crear un nuevo evento

Descripción:  
Descripción detallada. Prueba de descripción de un maratón. Se celebrará las 15:00. Consiste en un evento mundialmente conocido.

Seleccionar archivo Ninguno archivo selec.

Fecha del evento:  
06/08/2022

Organizador: organizador

Plazo de inscripción:  
11/08/2022

ⓘ No puede introducir un plazo posterior al día del evento.

Siguiente

Ilustración 1-42 Aviso: el plazo no puede ser posterior a la fecha del evento.

Una vez rellenado los campos correctamente, podrá continuar con la siguiente fase que consiste en crear las tareas pertinentes del evento. El formulario de tareas obliga al organizador a no dejar los campos.

+ Crear

Tarea bloqueada

---

Título:  
 Tarea 1. Señalización del circuito

---

N° Voluntarios:

---

N° Suplente:

Ilustración 1-43 Crear tarea.

Si un organizador quiere asignar voluntarios a una tarea, es necesario seleccionar el campo 'Tarea bloqueada' y cuando haga clic en el botón 'Crear', le conducirá al usuario a otra vista.

← Crear tareas

+ Crear

Tarea bloqueada

---

Título:  
 Tarea 2. Entrega de premios

---

N° Voluntarios:

---

N° Suplente:

---

Tarea 1. Señalización del circuito

Ilustración 1-44 Bloquear y crear tarea.

En la sección de asignar voluntarios, previamente se mostrará un mensaje en el que avisará al organizador de la selección de estos.

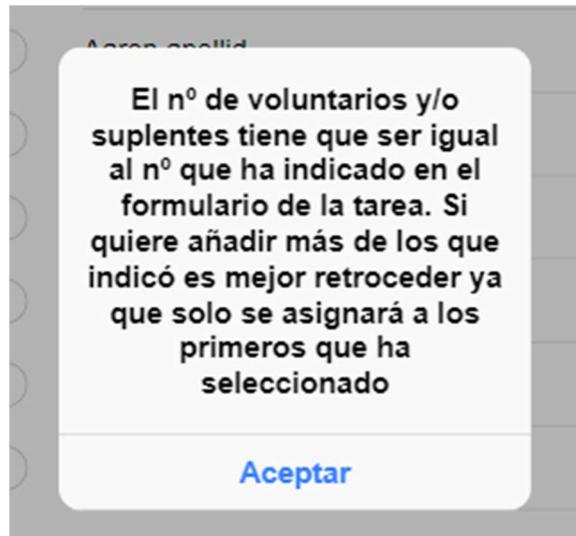


Ilustración 1-45 Aviso: número de usuarios tiene que ser igual al indicado.

Pulsando en aceptar, podrá escoger los voluntarios o retroceder a la anterior interfaz.

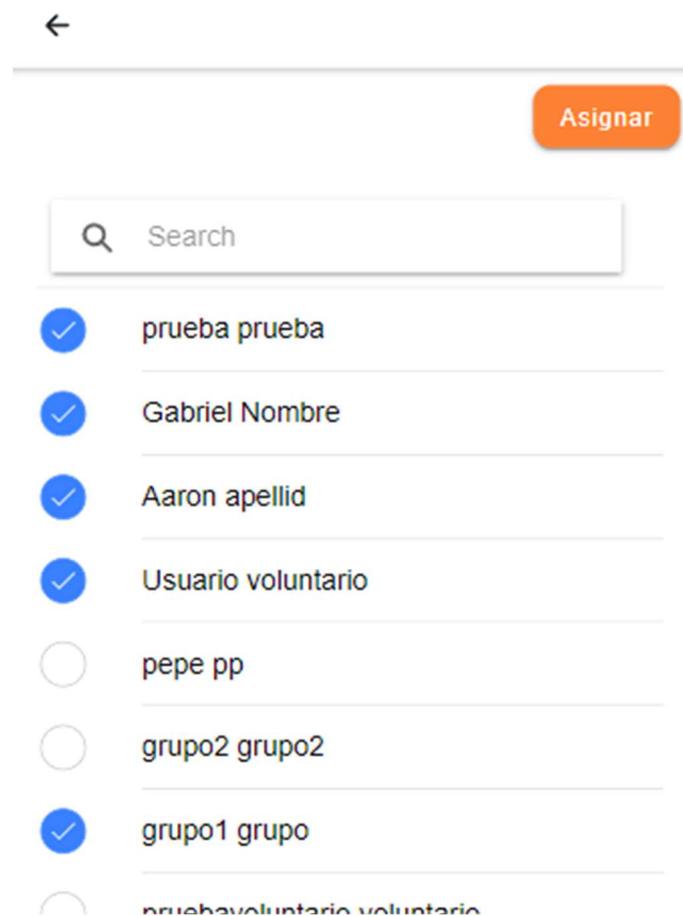


Ilustración 1-46 Asignación de voluntarios en una tarea.

Además, se pueden eliminar las tareas que se encuentran en el listado pulsando el botón de la 'X'.

← Crear tareas

+ Crear

Tarea bloqueada

Titulo:  
titulo del evento

N° Voluntarios:

N° Suplente:

Tarea 1. Señalización del circuito ✕

Tarea 2. Entrega de premios ✕

Crear evento

Ilustración 1-47 Crear evento y tareas.

Pulsando el botón 'Crear evento', se comienza a crear un nuevo evento en la aplicación móvil que se puede apreciar en el recuadro verde de la siguiente captura.

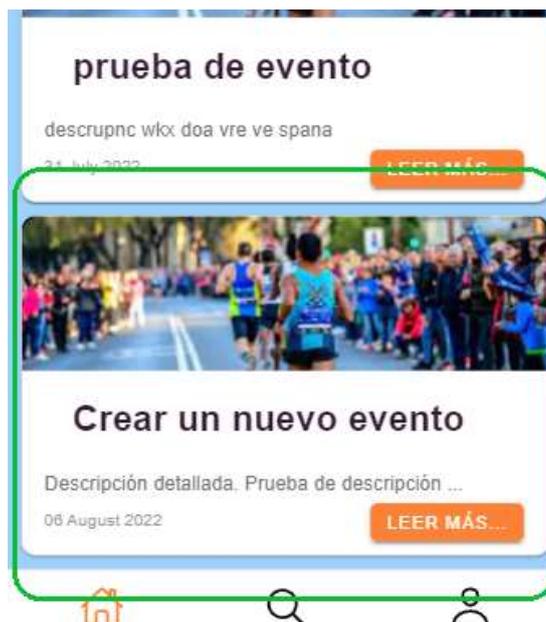


Ilustración 1-48 Inicio con el nuevo evento.

Finalmente, para comprobar que se ha creado correctamente el evento, el organizador puede entrar en el evento y ver la información que ha rellenado en los formularios. Del mismo modo, le permite ver a los voluntarios que ha asignado en la tarea bloqueada. Esto se corresponde con las dos ilustraciones siguientes:

The screenshot displays a user interface for event management. At the top, there is a photograph of a marathon. Below it, the event title is "Crear un nuevo evento" with a date of "06 August 2022". A description follows: "Descripción detallada. Prueba de descripción de un maratón. Se celebrará las 15:00. Consiste en un evento mundialmente conocido." Below the description is a search bar labeled "Busca una tarea". Two task cards are shown: "Tarea 1. Señalización del cir..." with a deadline of "02 August 2022", 0 volunteers, and 0 substitutes; and "Tarea 2. Entrega de premios" with a deadline of "02 August 2022", 3 volunteers, and 2 substitutes. Each task card has "Ver" and "Editar" buttons.

**Crear un nuevo evento**  
06 August 2022

Descripción detallada. Prueba de descripción de un maratón. Se celebrará las 15:00. Consiste en un evento mundialmente conocido.

Q Busca una tarea

**Tarea 1. Señalización del cir...**  
Plazo: 02 August 2022  
Voluntarios: 0 de 8  
Suplente: 0 de 2  
Ver Editar

**Tarea 2. Entrega de premios**  
Plazo: 02 August 2022  
Voluntarios: 3 de 3  
Suplente: 2 de 2  
Ver Editar

Ilustración 1-49 Evento con sus tareas.

## Tarea 2. Entrega de premios

### Voluntarios:



prueba prueba



Gabriel Nombre



Aaron apellid

### Suplentes:



Usuario voluntario



grupo1 grupo

Ilustración 1-50 Tarea bloqueada: listado de voluntarios elegidos.