



UNIVERSIDAD DE MÁLAGA



GRADO EN INGENIERÍA DEL SOFTWARE

SPORTIFY: TU ZONA DEPORTIVA A UN CLICK

SPORTIFY: WORKING OUT LOCATION IN A CLICK

Realizado por
ANTONIO DAVID LÓPEZ MOSTAZO

Tutorizado por
GABRIEL JESÚS LUQUE POLO
JAMAL TOUTOUH EL ALAMIN

Departamento
LENGUAJE Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

MÁLAGA, FEBRERO DE 2022



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA DEL SOFTWARE

SPORTIFY: TU ZONA DEPORTIVA A UN CLICK

**SPORTIFY: WORKING OUT LOCATIONS IN A
CLICK**

Realizado por

Antonio David López Mostazo

Tutorizado por

Gabriel Jesús Luque Polo

Jamal Toutouh El Alamin

Departamento

Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, FEBRERO DE 2022

Fecha defensa: Febrero de 2022



UNIVERSIDAD
DE MÁLAGA



Resumen

El presente proyecto gira entorno a la creación de una aplicación web, llamada Sportify, la cual sirva de ayuda a las personas que deseen practicar deporte al aire libre utilizando las zonas deportivas que proporciona el Ayuntamiento de Málaga para dicho fin. A través de ella, tanto los usuarios invitados como los registrados podrán consultar qué zonas hay disponibles en la ciudad de Málaga, filtrarlas acorde a diversos parámetros y verlas ubicadas en un mapa. Para los usuarios registrados, tendrá un carácter más típico de una red social en la que, además de lo mencionado anteriormente, podrán configurar un perfil a su gusto, organizarse los entrenamientos, hacer amigos y/o organizar eventos.

Para la elaboración del servidor ha sido utilizado el lenguaje de programación Python [1] y el framework Django [2]; dicho servidor realiza las consultas solicitadas por el cliente a una base de datos alojada en MongoDB [3] y a los datos abiertos, referentes a dichas zonas deportivas, que proporciona el Ayuntamiento de Málaga [4].

Para la elaboración de la parte del cliente ha sido utilizada una biblioteca de Javascript llamada ReactJS [5] y, para su interfaz, el componente React-Bootstrap [6].

Palabras clave:

Málaga, Aplicación web, Deporte y Red social.

Abstract

This project is based on the development of a web application, called Sportify, which uses a client/server architecture. The application will serve as a helpful tool for people who want to work out outside using the sport courts of the Malaga's town hall.

The application shows and marks on a map sport courts in Malaga and allow to apply filters to guests (users without account) and users to see the best courts according to their needs. Furthermore, logged users can use the application as a social network to set a profile, prepare their training sessions, make friends and organize sport events.

The server has been developed using Python [1] software language and Django [2] framework. This server performs the clients' requests in a database hosted in MongoDB [3] and the online open-source data of the Malaga's town hall [4].

ReactJS [5], a Javascript library, and React-Bootstrap [6] have been used to develop the client interface.

Keywords:

Malaga, Web application, Sport and Social network.

Índice

Sportify

Resumen	1
Abstract	1
Índice	1
Introducción	5
1.1 Motivación	5
1.1.1 Análisis de mercado	6
1.2 Objetivos	11
1.3 Estructura de la memoria	12
1.3.1 Introducción	12
1.3.2 Tecnologías y herramientas utilizadas	12
1.3.3 Análisis de requisitos	12
1.3.4 Metodología e implementación	12
1.3.5 Modelado y diseño de la aplicación	13
1.3.6 Pruebas realizadas y despliegue en la nube	13
1.3.7 Conclusiones y líneas futuras	13
1.3.8 Referencias	13
1.3.9 Anexos	14
Análisis de requisitos	15
2.1 Alcance	15
2.2 Requisitos funcionales	16
2.3 Requisitos no funcionales	18
2.4 Casos de uso	19
Tecnologías y herramientas utilizadas	37
3.1 Lado del cliente	37
3.1.1 ReactJS	38
3.1.2 React-Bootstrap	38
3.1.3 Redux-Saga	39
3.2 Lado del servidor	39
3.2.1 Python	40
3.2.2 Django	40
3.3 Otras herramientas	41
3.3.1 Visual Studio Code	41
3.3.2 MongoDB	41
3.3.3 Cloudinary	42

3.3.4	Heroku	42
3.3.5	Datos Abiertos del Ayuntamiento de Málaga.....	43
3.3.6	Balsamiq	43
3.3.7	Postman.....	43
Modelado y diseño de la aplicación		45
4.1 Base de datos		46
4.1.1	Usuario	46
4.1.2	ZonaDeportiva	47
4.1.3	MaquinaDeportiva.....	47
4.1.4	Valoracion.....	47
4.1.5	Entrenamiento.....	48
4.1.6	Evento.....	48
4.1.7	Amistad.....	48
4.2 Bocetos de interfaz de usuario.....		48
Metodología e implementación.....		53
5.1 Metodología.....		54
5.2 Implementación		55
5.2.1	Primera iteración	55
5.2.2	Segunda iteración	57
5.2.3	Tercera iteración.....	58
5.2.4	Cuarta iteración	59
Pruebas realizadas y despliegue en la nube		61
6.1 Pruebas Front-end		62
6.2 Pruebas Back-end		66
6.2.1	Listar zonas deportivas	66
6.2.2	Crear zona deportiva	66
6.2.3	Comprobar Api login.....	67
6.2.4	Listar información de usuario	68
6.2.5	Crear usuario	68
6.2.6	Crear entrenamiento	69
6.2.7	Crear evento	69
6.2.8	Crear amistad	70
6.2.9	Listar máquinas deportivas.....	70
6.2.10	Crear valoración	71
6.3 Pruebas de seguridad		72
6.3.1	Inyección de SQL.....	72
6.3.2	Cross-site scripting.....	73
6.3.3	Fuerza bruta.....	74
6.4 Despliegue en la nube.....		75
Conclusiones y líneas futuras.....		79
7.1 Conclusiones		79
7.2 Líneas futuras		81
7.2.1	Nuevas funcionalidades.....	81
7.2.2	Expansión.....	81
7.2.3	Mejoras conceptuales y de seguridad	82
Referencias		83
Manual de Instalación		85
Manual de usuario		87

B.1 Usuario invitado	87
B.1.1 Ver zonas disponibles	87
B.1.2 Pagar por la lista de zonas	88
B.1.3 Filtrar las zonas	88
B.1.4 Ver información de una zona concreta y filtrarla en el mapa	89
B.1.5 Registrarse en la aplicación	90
B.1.6 Logearse en la aplicación.....	92
B.1.7 Ir a la página de inicio	94
B.2 Usuario autenticado	94
B.2.1 Ver perfil de usuario	94
B.2.2 Modificar perfil de usuario	94
B.2.3 Añadir imagen de perfil	95
B.2.4 Registrar un entrenamiento en una zona deportiva	97
B.2.5 Organizar un evento en una zona deportiva	98
B.2.6 Apuntarse a un evento	99
B.2.7 Añadir una nueva zona deportiva.....	100
B.2.8 Añadir amigos.....	101
B.2.9 Borrar amigo.....	103
B.2.10 Añadir reseña a una zona deportiva.....	103
B.2.11 Cerrar sesión.....	104

1

Introducción

En esta sección se pretende dar al lector una primera toma de contacto con la memoria de este proyecto. En primer lugar se expondrán los motivos que promueven la necesidad de este trabajo, a continuación se presentarán los objetivos que ha proseguido cumplir y, para concluir, se hará una explicación del orden y contenido que contiene este informe.

1.1 Motivación

El panorama internacional y las interacciones humanas han cambiado completamente debido a la pandemia del Covid-19, en el que la facilidad de contagio y el peligro del mismo han llevado a las personas a reducir su círculo de contacto social y cambiar sus hábitos de vida. Todo esto, sumado a un estilo de vida más sedentario que se ha quedado anidado entre la población, provoca que se realice una menor cantidad de actividad física en grupo o en lugares habituales, como gimnasios, provocando un perjuicio en la salud [7].

Para paliar el sedentarismo, los organismos públicos, como ayuntamientos, han instalado en las ciudades y pueblos diferentes zonas para la práctica de actividad física

al aire libre, de uso gratuito y de fácil acceso. Sin embargo, en muchos de los casos dichas zonas se encuentran infrautilizadas porque los propios posibles usuarios no conocen de su existencia y no saben de qué tipo de instalaciones cuentan. Esto no facilita el que se mitigue el sedentarismo de la población.

Con este trabajo se desea apoyar a la población a realizar actividades físicas al aire libre con las garantías sanitarias y sin las restricciones impuestas a este tipo de actividades en pabellones y pistas interiores.

1.1.1 Análisis de mercado

Una parte fundamental que motiva la realización de este trabajo es el análisis de mercado, en el cual se ha realizado un estudio sobre las aplicaciones más utilizadas actualmente con el fin de realizar actividades físicas al aire libre. A través de este análisis se ha podido ver cómo está el mercado en el que se implementará la aplicación y qué puede aportar Sportify respecto a la competencia.

1.1.1.1 Strava



App móvil, a modo de red social, para corredores y ciclistas. Con ella se puede medir el rendimiento de los entrenamientos, trazar rutas o utilizar las rutas que ofrece y competir con tus amigos; además, permite configurar una función de seguimiento a tiempo real por una persona de confianza por si alguna vez se necesita ayuda. Tiene una comunidad ya consolidada de millones de usuarios que la respaldan. La gran mayoría de funciones son a través de una suscripción mensual de 7.99€ o una suscripción anual de 5€ al mes.

URL: <https://www.strava.com/mobile>

1.1.1.2 Runkeeper



Es una aplicación web y móvil que permite llevar el registro de los entrenamientos y, aunque en su principio fue pensada para corredores, permite una gran variedad de actividades como ciclismo, yoga o musculación. Además, en sus últimas actualizaciones, se ha orientado hacia la parte social, creando clubs de corredores y compartiendo logros con amigos para incentivar el progreso deportivo. Cuenta con el apoyo de la conocida marca Asics y ofrece un periodo de prueba de 7 días, después pasa a un precio de suscripción de 39.99\$ anuales.

URL: <https://runkeeper.com/cms/>

1.1.1.3 TrainingPeaks



Se trata de una aplicación web y móvil que ofrece una gran variedad de entrenamientos para diferentes deportes, incluyendo además subcategorías de cada uno de ellos (excepto para musculación).

Aunque principalmente interesa una comparativa referente a las actividades al aire libre, también incluye entrenamientos dentro de casa y gimnasios.

Además de poder planificar los entrenamientos, cuenta con un equipo de monitores con los que asegurar y mejorar la calidad de los ejercicios.

Cuenta con un periodo de prueba de 14 días, tras los cuales se debe adquirir una suscripción mensual (19.95\$), cuatrimestral (16.33\$/mes) o anual (9.92\$/mes).

URL: <https://www.trainingpeaks.com/>

1.1.1.4 Nike Run Club



Es una aplicación móvil que ofrece diferentes tipos de entrenamientos para corredores. Su punto fuerte es que cuenta con el respaldo de Nike, lo que le da un gran auge publicitario. Permite marcar objetivos, compartir los logros con amigos y, sobre todo, se enfoca en que los usuarios se diviertan haciendo running.

Su gran novedad, que no aparece en la mayoría de la competencia, es que permite registrar las zapatillas para ver cuantos kilómetros recorres con ellas.

Por último, otro de sus puntos fuertes es que Nike organiza numerosas carreras semanales en diferentes países, con el fin de promover el uso de su aplicación.

URL: <https://www.nike.com/es/nrc-app>

1.1.1.5 Adidas Running/Runtastic



Tras la unificación de ambas en Adidas Running, tenemos una aplicación móvil pensada para corredores. Su mayor fuerte es que cuenta con el respaldo de grandes empresas, como cabe de esperar una de ellas es Adidas, lo que le da un gran auge publicitario.

Su gran novedad, similar a la aplicación previa, es que cuenta con una sección de registro de zapatillas, con la cual puedes ver cuando kilómetros has recorrido con cada una de ellas.

Tiene una versión gratuita con anuncios y otra de pago (coste único de 5€).

Presenta un problema bastante reportado que es el mal funcionamiento del seguimiento GPS.

URL: <https://www.runtastic.com/es/>

1.1.1.6 Sworkit

SWÖRKIT Se trata de una app móvil gratuita con más de 5 millones de usuarios. Esta sirve de guía para realizar ejercicios en función de que zona del cuerpo se desee ejercitar; cada ejercicio va acompañado de una imagen que lo ilustra.

Aunque no incentiva realmente el hacer ejercicio al aire libre, si promueve alternativas a ir a los gimnasios y es totalmente compatible para realizar la actividad física seleccionada en cualquier lugar.

Aparentemente es intuitiva y fácil de usar, aunque la interfaz deja mucho que desear.

URL: <https://sworkit.com/>

1.1.1.7 Zonas de musculación



Esta aplicación, desarrollada por el Ayuntamiento de Málaga, busca fomentar la práctica deportiva entre los residentes de Málaga, haciendo uso de las instalaciones repartidas por la ciudad.

Es totalmente gratuita y fácil de usar. Además, permite visualizar las zonas deportivas marcadas en un mapa o buscarlas filtrando por barrios de la capital.

Un gran punto que beneficia a esta aplicación es que cuenta con el apoyo del Ayuntamiento de Málaga.

1.1.1.8 Conclusiones

Una vez analizada la mayor competencia que se encuentra en el mercado, se puede ver que la aplicación incorpora novedades respecto a las ya existentes.

Sportify no es solo para corredores, sino que integra varios tipos de entrenamiento funcional según lo que busque cada usuario. Además, ofrece zonas dónde entrenar con una información detallada de qué se puede practicar en cada una de ellas.

Otro gran punto a favor es que no solo estará disponibles para móviles, ya que si un usuario puede organizar cómodamente su entrenamiento desde la pantalla del ordenador y luego llevarlo consigo en el móvil.

Por último, se ha podido ver que la mayoría de las aplicaciones existentes requieren del pago de una suscripción, mientras que con este proyecto se plantea una alternativa totalmente gratuita para la comunidad.

La mayor desventaja con la que cuenta este proyecto es que no tiene grandes patrocinadores que lo respalden y hagan eco del mismo ni con una comunidad de usuarios, pero con el tiempo y una campaña de marketing se puede llegar a conseguir este apoyo.

Para concluir, el proyecto aporta novedades y agrupa algunas de las mejores funcionalidades que ofrece la competencia, además de estar especializada en la ciudad de Málaga. El mayor competidor sería la aplicación desarrollada por el

Ayuntamiento de Málaga, pero dotándola de un carácter social e incorporando más funciones que esta, su implantación en el mercado podría tener una buena acogida por parte de los deportistas malagueños.

Por todo lo expuesto anteriormente, acorde al panorama internacional que ha dejado la pandemia del Covid-19, y el análisis del mercado que ha estudiado el entorno en el que se implantaría la aplicación, se ve que puede resultar bastante beneficioso, provocando una clara motivación para su desarrollo.

1.2 Objetivos

El TFG, como se ha venido comentando a lo largo del informe, consistirá en la elaboración de una aplicación web para fomentar el deporte al aire libre y el descubrimiento de nuevas zonas donde practicarlo.

La aplicación, a alto nivel, ofrecerá a los usuarios un listado de lugares donde poder practicar deportes al aire libre cerca de la ubicación seleccionada; estas zonas de deporte serán obtenidas a través de los datos abiertos que ofrece el Ayuntamiento de Málaga [4], los cuales se consultarán con regularidad para mantener las zonas existentes actualizadas e incorporar las nuevas que se hayan podido añadir. Además, se podrán aplicar diversos filtros (zonas con sombra, fuentes de agua, máquinas en función de que parte del cuerpo queramos entrenar o qué deporte vayamos a practicar y/o localización, ...) y, las pistas que cumplan los requisitos de filtrado, aparecerán marcadas en el mapa.

Además, los visitantes podrán registrarse para ser partícipes de la red social. En ella, una vez registrados, podrán añadir comentarios y valorar las diferentes zonas. También podrán añadir contactos a su lista de amigos y organizar eventos deportivos en una o varias zonas.

Por último, se buscará el objetivo de que la aplicación sea segura frente a los ataques

más comunes que reciben páginas web hoy día (inyección de SQL [8], XSS [9], ataques de fuerza bruta [10] contra formularios de login y/o cualquier otro fallo de seguridad que ponga en riesgo los *assets* de la aplicación).

1.3 Estructura de la memoria

En esta memoria se pretende dejar constancia de las diferentes tareas llevadas a cabo para la realización del proyecto. Se encuentra estructurada en los siguientes bloques:

1.3.1 Introducción

Se trata de un primer acercamiento a la memoria realizada, su objetivo es demostrar la importancia y dar una primera impresión al lector del proyecto que se plantea.

1.3.2 Tecnologías y herramientas utilizadas

Se recopilan las tecnologías y herramientas que se han utilizado para desarrollar la aplicación o de las que hace uso para su correcto funcionamiento. Además, del porqué se han elegido y que problema o posibles problemas han ayudado a solucionar y/o prevenir.

1.3.3 Análisis de requisitos

Se estudia el alcance del proyecto y se presenta una lista con los requisitos, tanto funcionales como no funcionales, que debe presentar la aplicación web para cumplir con las expectativas planteadas.

1.3.4 Metodología e implementación

Se explica la metodología y los pasos que se han seguido durante el desarrollo del Trabajo Final de Grado; además, se fragmenta la implementación en las diferentes

iteraciones por las que ha pasado el proyecto y se cuenta que se ha elaborado en cada una de ellas.

1.3.5 Modelado y diseño de la aplicación

En esta sección se presenta el modelo diseñado para la base de datos y las entidades que lo componen. También se presenta una serie de bocetos que se diseñaron para tener una primera idea de las ventanas que presentará la aplicación, su organización y navegabilidad.

1.3.6 Pruebas realizadas y despliegue en la nube

Se expondrán las pruebas realizadas, tanto al lado del cliente, lado del servidor y seguridad que verifiquen que está preparada para su puesta en producción. Además, se expondrá el proceso de despliegue en la nube.

1.3.7 Conclusiones y líneas futuras

En este apartado se hace recopilación de lo que el proyecto a supuesto a nivel personal; se evalúa tanto las dificultades que se han superado como lo aprendido a nivel técnico y profesional. También se hablará sobre qué le puede deparar a la aplicación en futuras versiones de cara a implementar nuevos requisitos y marcar nuevos objetivos.

1.3.8 Referencias

Se recopila la información que se ha cotejado para desarrollar el proyecto, así como material bibliográfico para ampliar conocimiento sobre los diferentes temas y herramientas comentadas durante el presente proyecto.

1.3.9 Anexos

En esta última sección se abordará el tema del manual de instalación y el manual de usuario, a través de dos apéndices que complementan a la aplicación en cuanto a carácter documental se refiere.

2

Análisis de requisitos

En esta sección vamos a tratar los requisitos que debe cumplir la aplicación antes de su puesta a producción. En primer lugar se analizará el alcance y las partes interesadas/involucradas (*Stakeholders*), los requisitos funcionales y no funcionales y, por último, se elaborarán los casos de uso para los diferentes requisitos funcionales.

2.1 Alcance

Uno de los requisitos básicos en todo proyecto software es fijar el alcance; es decir, tener en cuenta todas las personas y/o organismos que se puedan ver afectados o implicados por el desarrollo del mismo.

El alcance del producto software que se va a desarrollar incluirá al Ayuntamiento de Málaga, a los futuros usuarios de la aplicación web y a las partes interesadas en el proyecto, Antonio David López Mostazo (desarrollador), Gabriel Jesús Luque Polo (tutor) y Jamal Toutouh El Alamin (tutor).

2.2 Requisitos funcionales

La aplicación debe presentar los siguiente requisitos funcionales, es decir, todas las operaciones que se pueden llevar a cabo entre los diferentes tipos de usuario. Todas ellas son de carácter obligatorio y deben estar incorporadas a la hora de su despliegue en la nube. Todos ellos irán identificados por “RF-XX” (donde RF es Requisito Funcional y XX es el número que corresponde a dicho requisito) y acompañados de su descripción.

ID	Descripción del requisito funcional
RF-01	Mostrar la lista de zonas deportivas.
RF-02	Mostrar un mapa marcado por la ubicación de las diferentes zonas deportivas.
RF-03	Ofrecer una opción para visualizar la información de la zona deportiva.
RF-04	Filtrar la lista de zonas deportivas por iluminación.
RF-05	Filtrar la lista de zonas deportivas por zonas con sombra.
RF-06	Filtrar la lista de zonas deportivas por zonas con fuentes.
RF-07	Filtrar la lista de zonas deportivas por zonas con aseos.
RF-08	Filtrar la lista de zonas deportivas por parking.
RF-09	Filtrar la lista de zonas deportivas por taquilla.
RF-10	Filtrar la lista de zonas deportivas por localización.
RF-11	Ofrecer un formulario que permita al usuario invitado registrarse en la aplicación.
RF-12	Ofrecer un formulario que permita al usuario invitado iniciar sesión en la aplicación.
RF-13	Ofrecer una opción que permita al usuario autenticado visualizar su perfil.

RF-14	Ofrecer la posibilidad a cualquier usuario autenticado de modificar su perfil de usuario.
RF-15	Ofrecer un formulario que permita a cualquier usuario autenticado registrar un nuevo entrenamiento.
RF-16	Ofrecer un formulario que permita a cualquier usuario autenticado organizar un nuevo evento.
RF-17	Ofrecer a cualquier usuario autenticado la posibilidad de apuntarse a cualquier evento organizado.
RF-18	Ofrecer un formulario que permita a cualquier usuario autenticado registrar una nueva zona deportiva.
RF-19	Ofrecer a cualquier usuario autenticado un buscador para encontrar otros usuarios.
RF-20	Ofrecer la posibilidad a cualquier usuario autenticado de agregar a otro usuario a la lista de amigos.
RF-21	Ofrecer la posibilidad a cualquier usuario autenticado de eliminar a otro usuario de la lista de amigos.
RF-22	Permitir a un usuario autenticado cerrar su sesión.
RF-23	Permitir al administrador eliminar una cuenta de usuario.
RF-24	Permitir al administrador eliminar un entrenamiento.
RF-25	Permitir al administrador eliminar un evento.
RF-26	Permitir al administrador eliminar la participación a un evento.
RF-27	Permitir al administrador eliminar una zona deportiva.
RF-28	Permitir al administrador eliminar una máquina deportiva.
RF-29	Permitir al administrador eliminar una amistad.
RF-30	Ofrecer un formulario que permita añadir una reseña a una zona deportiva.
RF-31	Permitir al administrador eliminar una reseña

2.3 Requisitos no funcionales

La aplicación debe contar con los requisitos no funcionales, es decir, los criterios arquitectónicos y de calidad que debe cumplir el sistema. Estos deben estar presentes desde la primera versión desplegada en la nube. Todos ellos irán identificados por “RNF-XX” (donde RNF es Requisito No Funcional y XX es el número que corresponde a dicho requisito) y acompañados de su descripción.

ID	Descripción del requisito no funcional
RNF-01	Aplicación web responsive a través de cualquier navegador web con JavaScript.
RNF-02	Implementación de la parte del cliente en un framework basado en Javascript.
RNF-03	Deberán existir tres roles: usuario invitado, usuario autenticado y usuario administrador.
RNF-04	Soportar un mínimo de 4000 peticiones por hora.
RNF-05	Tiempo de respuesta máximo de 5 segundos.
RNF-06	Cualquier elemento de la aplicación será accesible para el usuario en menos de 5 clicks de ratón o pulsaciones de pantalla.
RNF-07	La aplicación continuará operativa, aunque no se pueda acceder a la página de datos abiertos del Ayuntamiento de Málaga.
RNF-08	El sistema debe contar con manual de usuario.
RNF-09	Ofrecer seguridad contra Inyección SQL.
RNF-10	Ofrecer seguridad contra Cross-site Scripting (XSS), garantizando que no pueda provocar una pérdida de identidad del usuario.
RNF-11	Ofrecer seguridad contra ataques de fuerza bruta, detectándose dicho suceso por el administrador de la aplicación.

2.4 Casos de uso

A continuación se expondrán los casos de uso de la aplicación en un entorno de éxito, con el fin de demostrar que todos los requisitos funcionales con los que debe contar la aplicación están presentes.

En ellos se mostrará un identificador “CU-XX” (donde CU significa Caso de Uso y XX es el número que corresponde a dicho caso de uso), un título para saber qué acción se está realizando, el identificador de requisito funcional que se está cubriendo, los actores que intervienen, una o varias precondiciones, la secuencia que se ha seguido y una postcondición.

ID	CU-01
Título	Visualizar lista y marcación en mapa de las zonas deportivas
Requisito funcional	RF-01, RF-02
Actores	Usuario invitado/autenticado, sistema
Precondiciones	El usuario cuenta con un dispositivo con acceso a Internet y tiene abierto un navegador web en el mismo.
Secuencia	<ol style="list-style-type: none">1. El usuario introducirá la dirección web de la aplicación.2. El sistema mostrará la página principal de la aplicación con la lista de zonas deportivas paginadas, varios “checkboxs” y un mapa con la lista de zonas deportivas que aparecen en dicha página marcadas en el mapa.
Postcondición	El usuario verá la lista de las zonas deportivas paginadas y marcadas en el mapa por página.

ID	CU-02
Título	Visualizar información adicional de una zona deportiva
Requisito funcional	RF-03
Actores	Usuario invitado/autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre una zona deportiva. 2. El sistema muestra una ventana emergente con toda la información de la zona seleccionada y el formulario para añadir una nueva reseña.
Postcondición	El usuario visualizará la información completa de la zona deportiva seleccionada

ID	CU-03
Título	Visualizar las zonas deportivas que cuentan con una o varias características
Requisito funcional	RF-04, RF-05, RF-06, RF-07, RF-08, RF-09
Actores	Usuario invitado/autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre una de las características. 2. El sistema muestra por páginas las zonas deportivas que cumplen con la característica seleccionada.

	*Si se desea filtrar por más de una característica a la vez, se vuelve al paso número 1.
Postcondición	El usuario verá la lista de las zonas deportivas, que cumplen la/s condición/es, paginadas y marcadas en el mapa por página

ID	CU-04
Título	Visualizar la ubicación de una zona deportiva concreta
Requisito funcional	RF-10
Actores	Usuario invitado/autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre una zona deportiva concreta. 2. El sistema muestra una ventana emergente con la información de dicha zona. 3. El usuario cierra la ventana emergente. 4. El sistema muestra la zona resaltada y solo mantiene su marca en el mapa.
Postcondición	El usuario visualizará la zona deportiva marcada en el mapa

ID	CU-05
Título	Registro de usuario
Requisito funcional	RF-11

Actores	Usuario invitado, sistema
Precondición	El usuario está en la página principal de la aplicación
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Login”. 2. El sistema muestra un formulario de acceso, un botón de login y un botón de registro. 3. El usuario pulsa el botón “Registrarse”. 4. El sistema muestra el formulario de registro. 5. El usuario rellena todos los campos. 6. El usuario pulsa el botón “Registrarse”. 7. El sistema crea una nueva cuenta de usuario. 8. El sistema redirige a la pantalla principal de la aplicación de un usuario autenticado
Postcondición	El usuario dispone de una nueva cuenta en la aplicación

ID	CU-06
Título	Acceso de usuario
Requisito funcional	RF-12
Actores	Usuario invitado, sistema
Precondición	El usuario está en la página principal de la aplicación
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Login”. 2. El sistema muestra un formulario de acceso, un botón de login y un botón de registro. 3. El usuario rellena el formulario. 4. El usuario pulsa el botón “Login”.

	5. El sistema redirige a la pantalla principal de la aplicación de un usuario autenticado.
Postcondición	El usuario visualiza la pantalla principal de la aplicación de un usuario autenticado

ID	CU-07
Título	Visualización de perfil
Requisito funcional	RF-13
Actores	Usuario autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación de un usuario autenticado
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre “Perfil” en la barra de menú. 2. El sistema muestra toda la información del perfil de dicho usuario.
Postcondición	El usuario verá su perfil de usuario

ID	CU-08
Título	Modificación de perfil
Requisito funcional	RF-14
Actores	Usuario autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación de un usuario autenticado
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre “Perfil” en la barra de menú. 2. El sistema muestra toda la información del perfil de dicho usuario.

	<ol style="list-style-type: none"> 3. El usuario modifica los campos deseados del formulario. 4. El sistema muestra el perfil del usuario con los cambios realizados.
Postcondición	El usuario verá su perfil de usuario modificado

ID	CU-09
Título	Registro de un nuevo entrenamiento
Requisito funcional	RF-15
Actores	Usuario autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación de un usuario autenticado
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre “Registrar entrenamiento” en la barra de menú. 2. El sistema muestra el formulario de registro de un nuevo entrenamiento. 3. El usuario rellena los campos. 4. El sistema habilita el botón “Registrar entrenamiento” 5. El usuario pulsa el botón “Registrar entrenamiento” 6. El sistema registra el entrenamiento. 7. El sistema muestra un mensaje de éxito.
Postcondición	El usuario contará con un nuevo entrenamiento registrado

ID	CU-10
Título	Organización de un nuevo evento
Requisito funcional	RF-16
Actores	Usuario autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación de un usuario autenticado
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre “Organizar evento” en la barra de menú. 2. El sistema muestra el formulario para organizar un nuevo evento. 3. El usuario rellena el formulario. 4. El sistema habilita el botón “Organizar”. 5. El usuario pulsa el botón “Organizar”. 6. El sistema registra un nuevo evento en la aplicación. 7. El sistema muestra un mensaje de éxito.
Postcondición	El usuario habrá organizado un nuevo evento

ID	CU-11
Título	Apuntarse a un evento
Requisito funcional	RF-17
Actores	Usuario autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación de un usuario autenticado
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre “Apuntarse a evento” en la barra de menú.

	<ol style="list-style-type: none"> 2. El sistema muestra una pantalla con la lista de zonas deportivas. 3. El usuario pulsa sobre una zona deportiva. 4. El sistema muestra los eventos disponibles en dicha zona deportiva. 5. El usuario pulsa sobre un evento. 6. El sistema muestra la información del evento. 7. El sistema habilita el botón “Apuntarse”. 8. El usuario pulsa el botón “Apuntarse”. 9. El sistema registra la participación del usuario al evento seleccionado. 10. El sistema muestra un mensaje de éxito.
Postcondición	El usuario se habrá apuntado a un nuevo evento

ID	CU-12
Título	Creación de una nueva zona deportiva
Requisito funcional	RF-18
Actores	Usuario autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación de un usuario autenticado
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre “Añadir zona” en la barra de menú. 2. El sistema muestra el formulario para añadir una nueva zona deportiva. 3. El usuario rellena el formulario. 4. El sistema habilita el botón “Agregar zona”.

	<ol style="list-style-type: none"> 5. El usuario pulsa sobre el botón “Agregar zona”. 6. El sistema añade una nueva zona a la aplicación. 7. El sistema muestra un mensaje de éxito.
Postcondición	El usuario habrá registrado una nueva zona deportiva en el sistema

ID	CU-13
Título	Agregar un usuario a la lista de amigos
Requisito funcional	RF-19, RF-20
Actores	Usuario autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación de un usuario autenticado
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre “Añadir amigos” en la barra de menú. 2. El sistema muestra un buscador, la lista de usuarios, un botón “Atrás” y un botón “Buscar”. 3. El usuario introduce un nombre. 4. El sistema filtra la lista de usuarios por dicho nombre. 5. El usuario pulsa sobre un usuario. 6. El sistema remarca al usuario y habilita el botón “Buscar”. 7. El usuario pulsa sobre el botón “Buscar”

	<p>8. El sistema muestra el perfil del usuario seleccionado, un botón “Borrar amigo” y un botón “Agregar amigo”.</p> <p>9. El usuario pulsa sobre el botón “Agregar amigo”.</p> <p>10. El sistema crea una nueva amistad entre ambos usuarios.</p>
Postcondición	El usuario tendrá un nuevo amigo en su lista de amigos

ID	CU-14
Título	Borrar un usuario a la lista de amigos
Requisito funcional	RF-21
Actores	Usuario autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación de un usuario autenticado
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre “Añadir amigos” en la barra de menú. 2. El sistema muestra un buscador, la lista de usuarios, un botón “Atrás” y un botón “Buscar”. 3. El usuario introduce un nombre. 4. El sistema filtra la lista de usuarios por dicho nombre. 5. El usuario pulsa sobre un usuario. 6. El sistema remarca al usuario y habilita el botón “Buscar”.

	<ol style="list-style-type: none"> 7. El usuario pulsa sobre el botón “Buscar”. 8. El sistema muestra el perfil del usuario seleccionado, un botón “Borrar amigo” y un botón “Agregar amigo”. 9. El usuario pulsa sobre el botón “Borrar amigo”. 10. El sistema elimina la amistad entre ambos usuarios.
Postcondición	El usuario habrá borrado a un amigo de su lista de amigos

ID	CU-15
Título	Cerrar sesión de usuario
Requisito funcional	RF-22
Actores	Usuario autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación de un usuario autenticado
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre “Perfil” en la barra de menú. 2. El sistema muestra el perfil del usuario autenticado. 3. El usuario pulsa sobre el botón “Cerrar sesión”. 4. El sistema cierra la sesión de usuario. 5. El sistema muestra un mensaje de éxito.
Postcondición	El usuario habrá cerrado su sesión de usuario

ID	CU-16
Título	Borrado de un usuario
Requisito funcional	RF-23
Actores	Usuario administrador, sistema
Precondición	El administrador esta autenticado en la pantalla de inicio del panel de Django
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el enlace azul “Usuarios”. 2. El sistema muestra la lista de usuarios. 3. El usuario pulsa el “checkbox” que hay al lado del usuario que se quiere eliminar. 4. El usuario despliega las opciones y pulsa sobre “Delete selected usuarios”. 5. El usuario pulsa el botón “Go”. 6. El sistema elimina al usuario de la aplicación.
Postcondición	El usuario seleccionado será borrado de la aplicación

ID	CU-17
Título	Borrado de un entrenamiento
Requisito funcional	RF-24
Actores	Usuario administrador, sistema
Precondición	El administrador esta autenticado en la pantalla de inicio del panel de Django
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el enlace azul “Entrenamientos”. 2. El sistema muestra la lista de entrenamientos.

	<ol style="list-style-type: none"> 3. El usuario pulsa el “checkbox” que hay al lado del entrenamiento que se quiere eliminar. 4. El usuario despliega las opciones y pulsa sobre “Delete selected entrenamientos”. 5. El usuario pulsa el botón “Go”. 6. El sistema elimina el entrenamiento de la aplicación.
Postcondición	El entrenamiento seleccionado será borrado de la aplicación

ID	CU-18
Título	Borrado de un evento
Requisito funcional	RF-25
Actores	Usuario administrador, sistema
Precondición	El administrador esta autenticado en la pantalla de inicio del panel de Django
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el enlace azul “Eventos”. 2. El sistema muestra la lista de eventos y participaciones. 3. El usuario pulsa el “checkbox” que hay al lado del evento que se quiere eliminar. 4. El usuario despliega las opciones y pulsa sobre “Delete selected eventos”. 5. El usuario pulsa el botón “Go”. 6. El sistema elimina el evento de la aplicación.
Postcondición	El evento seleccionado será borrado de la aplicación

ID	CU-19
Título	Borrado de una participación a evento
Requisito funcional	RF-26
Actores	Usuario administrador, sistema
Precondición	El administrador esta autenticado en la pantalla de inicio del panel de Django
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el enlace azul “Eventos”. 2. El sistema muestra la lista de eventos y participaciones. 3. El usuario pulsa el “checkbox” que hay al lado de la participación a evento que se quiere eliminar. 4. El usuario despliega las opciones y pulsa sobre “Delete selected eventos”. 5. El usuario pulsa el botón “Go”. 6. El sistema elimina la participación a evento de la aplicación.
Postcondición	La participación a evento seleccionada será borrada de la aplicación

ID	CU-20
Título	Borrado de una zona deportiva
Requisito funcional	RF-27
Actores	Usuario administrador, sistema

Precondición	El administrador esta autenticado en la pantalla de inicio del panel de Django
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el enlace azul “Zona deportivas”. 2. El sistema muestra la lista de zonas deportivas. 3. El usuario pulsa el “checkbox” que hay al lado de la zona deportiva que se quiere eliminar. 4. El usuario despliega las opciones y pulsa sobre “Delete selected zona deportivas”. 5. El usuario pulsa el botón “Go”. 6. El sistema elimina la zona de la aplicación.
Postcondición	La zona deportiva seleccionada será borrada de la aplicación

ID	CU-21
Título	Borrado de una máquina deportiva
Requisito funcional	RF-28
Actores	Usuario administrador, sistema
Precondición	El administrador esta autenticado en la pantalla de inicio del panel de Django
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el enlace azul “Maquina deportivas”. 2. El sistema muestra la lista de máquinas deportivas.

	<ol style="list-style-type: none"> 3. El usuario pulsa el “checkbox” que hay al lado de la máquina deportiva que se quiere eliminar. 4. El usuario despliega las opciones y pulsa sobre “Delete selected maquina deportivas”. 5. El usuario pulsa el botón “Go”. 6. El sistema elimina la máquina deportiva de la aplicación.
Postcondición	La máquina deportiva seleccionada será borrada de la aplicación

ID	CU-22
Título	Borrado de una amistad entre dos usuarios
Requisito funcional	RF-29
Actores	Usuario administrador, sistema
Precondición	El administrador esta autenticado en la pantalla de inicio del panel de Django
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el enlace azul “Amistads”. 2. El sistema muestra la lista de amistades entre usuarios. 3. El usuario pulsa el “checkbox” que hay al lado de la amistad que se quiere eliminar. 4. El usuario despliega las opciones y pulsa sobre “Delete selected amistads”. 5. El usuario pulsa el botón “Go”. 6. El sistema elimina la amistad de la aplicación.

Postcondición	La amistad entre los dos usuarios seleccionada será borrada de la aplicación
ID	CU-23
Título	Añadir una reseña a una zona deportiva
Requisito funcional	RF-30
Actores	Usuario autenticado, sistema
Precondición	El usuario está en la página principal de la aplicación de un usuario autenticado
Secuencia	<ol style="list-style-type: none"> 1. El usuario selecciona la zona deportiva a la que quiere añadir la reseña. 2. El sistema muestra la información de la zona y el formulario para añadir una nueva reseña. 3. El usuario selecciona una puntuación y escribe un comentario si lo desea. 4. El sistema habilita el botón “Valorar”. 5. El usuario pulsa el botón “Valorar” 6. El sistema añade una nueva valoración a la zona deportiva seleccionada.
Postcondición	La zona deportiva seleccionada tendrá una nueva reseña

ID	CU-24
Título	Borrado de una reseña
Requisito funcional	RF-31
Actores	Usuario administrador, sistema
Precondición	El administrador esta autenticado en la pantalla de inicio del panel de Django

<p>Secuencia</p>	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el enlace azul “Valoracions”. 2. El sistema muestra la lista de valoraciones. 3. El usuario pulsa el “checkbox” que hay al lado de la valoración que se quiere eliminar. 4. El usuario despliega las opciones y pulsa sobre “Delete selected valoracions”. 5. El usuario pulsa el botón “Go”. 6. El sistema elimina la valoración de la aplicación.
<p>Postcondición</p>	<p>La valoración seleccionada será borrada de la aplicación</p>

3

Tecnologías y herramientas utilizadas

En esta sección se explicarán los dos pilares sobre los que se apoya la aplicación (lado del cliente y lado del servidor), así como las tecnologías que se han utilizado para el desarrollo de cada una de las partes. Por último se expondrán las herramientas auxiliares que, bien no se han usado directamente en la implementación de la aplicación, pero se han utilizado para su desarrollo y su correcto funcionamiento.

3.1 Lado del cliente

Esta parte de la aplicación se centra en el desarrollo del programa que se ejecuta en el navegador de los usuarios finales de la aplicación. En ella, los esfuerzos se centran en elaborar la interfaz que verá el usuario y la navegación entre las diferentes pantallas.

Además, también es la parte encargada de mostrar de una forma clara la información que recibe.

Las tecnologías utilizadas para la implementación de esta parte han sido las siguientes:

3.1.1 ReactJS

ReactJS [5] es una librería de JavaScript desarrollada por Facebook, cuyo objetivo es facilitar la tarea de desarrollo de interfaces de usuario. Parte de la idea de crear SPAs (Single Page Applications), es decir, que la página web desarrollada sea como un cubo, en el que cada ventana que deseamos ver será una cara diferente pero siempre estemos mostrando el mismo cubo. Esto supone una gran ventaja computacional ya que los cálculos se realizan en la parte del cliente y todo el contenido inicial está cargado desde el primer momento, con lo que ayuda al requisito no funcional de que la aplicación no tarde más de 5 segundos en responder.

Otro punto por el que ha sido elegida esta tecnología, es que esta en fuerte demanda en el mercado laboral y hemos querido trabajarla para desarrollar el TFG y tener una primera toma de contacto de cara al futuro profesional.

Por último, ReactJS presenta seguridad por defecto frente a Cross-site scripting por lo que ha servido para cubrir el requisito no funcional que exigía dicha protección. Esta protección se base en escapar las cadenas de texto malicioso y mostrarlas como texto plano, por lo que no tendrán funcionalidad alguna.

3.1.2 React-Bootstrap

React-Bootstrap [6] es una biblioteca basada en componentes que suministra componentes nativos de Bootstrap como si fuesen de ReactJS puros, haciendo que

sean totalmente compatibles y den una apariencia consistente y homogénea en la aplicación.

Se ha elegido esta tecnología porque daba muchas facilidades, al usar ReactJS, para hacer una interfaz agradable para el usuario sin complicar ni ensuciar el código con mero CSS.

3.1.3 Redux-Saga

Redux Saga [11] es una librería que nos permite gestionar las modificaciones de un estado dentro de la aplicación en Redux. A su vez, Redux es un contenedor de estado de aplicaciones desarrolladas en JavaScript que permite crear un comportamiento consistente.

Gracias a su uso es más fácil el manejo de flujos asíncronos (como el proceso desde la petición que hace un cliente hasta que se procesa la respuesta). Además, ha servido para insertar un icono de procesamiento y que el usuario sepa que se está trabajando en mostrarle la información solicitada.

3.2 Lado del servidor

Esta segunda mitad de la aplicación está centrada en responder a las llamadas que realizan los clientes a través del lado del servidor, cotejando la base de datos y otros elementos de Internet (como los datos abiertos del Ayuntamiento de Málaga) para suministrar al cliente la información solicitada. Los clientes finales no interactúan de forma directa con el servidor, por lo que suele carecer de interfaz y son los administradores de la aplicación quienes trabajan directamente sobre él.

Las tecnologías que se han usado para su implementación han sido:

3.2.1 Python

Python [1] es un lenguaje de programación a alto nivel de propósito general; esto quiere decir que sus procedimientos, instrucciones y estructuras de datos están diseñados para resolver todo tipo de problemas.

Uno de los puntos fuertes de este lenguaje es que es un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar aplicaciones, sino que se ejecuta directamente en el dispositivo a través de un intérprete.

La razón por la que se ha elegido para el desarrollo del servidor ha sido su sencillez de entender y escribir código, además de que cuenta con una gran cantidad de bibliotecas y comunidad de respaldo, lo que resulta sumamente útil a la hora de elaborar un proyecto con él.

3.2.2 Django

Django [2] es un framework escrito en el lenguaje Python y está pensado para el desarrollo web, lo que me ha resultado una herramienta sumamente útil para la elaboración del servidor en Python. Gracias a que se encarga de automatizar parte del desarrollo web de forma bastante rápida, me ha permitido evitar tareas repetitivas durante el proyecto.

Adicionalmente a la rapidez de elaborar aplicaciones en Django y su gran versatilidad, cuenta con gran medidas para la seguridad del proyecto. En concreto, la aplicación se ha beneficiado de que construye las consultas utilizando la parametrización de consultas, esto evita que la aplicación pueda ser víctima de inyección SQL, satisfaciendo así uno de los requisitos no funcionales. Especificando un poco sobre dicha parametrización, se encarga de precompilar una sentencia SQL

de forma que una sentencia (por ejemplo SELECT) no pueda ser concatenada con otra sentencia, los datos de inyección serían meramente datos de cadena de texto dentro del parámetro.

3.3 Otras herramientas

El resto de herramientas, como ya se ha comentado, no se encargan de implementar la aplicación directamente, pero sirven de soporte para el desarrollador y para que la aplicación funcione correctamente. Estas herramientas han sido:

3.3.1 Visual Studio Code

Visual Studio Code [12] es un editor de código fuente pensado para facilitar el desarrollo de software. Gracias a su funcionalidad para la depuración de código, la gran cantidad de plugins instalables en función del lenguaje que se está utilizando y la rapidez que otorga tanto a la hora de escribir el código fuente por su predicción como a la de detectar errores resaltándolos en rojo.

Ha sido elegido debido a las facilidades ya comentadas, la gran comunidad que lo respalda con multitud de características para su personalización y a que su uso es totalmente gratuito.

3.3.2 MongoDB

MongoDB [3] es un sistema gestor de bases de datos NoSQL orientado a documentos; en él he alojado la base de datos ya que proporciona una fácil conexión con diversos frameworks y lenguajes de programación como Python.

El punto fuerte que ha llevado a elegir esta plataforma ha sido que en ella se crean bases de datos no relacionales, lo que favorece los tiempos de respuesta al no tener que traer datos innecesarios a la hora de realizar consultas enlazadas y ayudando a cumplir los objetivos de tiempo de respuesta marcados en los requisitos no funcionales.

3.3.3 Cloudinary

Cloudinary [13] es una plataforma que ofrece software como servicio (SaaS, Software as a Service), más concretamente permite alojar contenido multimedia para que sea usado por aplicaciones web y móvil.

Gracias a esta plataforma he podido implementar que los usuarios pongan una foto en su perfil, en lugar de unos anticuados avatares.

El usar Cloudinary en lugar de otro proveedor se ha debido a que resulta muy fácil de aprender a usar y realizar las llamadas desde la aplicación, además que no presenta el problema, como otras similares, de que al cabo de unas horas el enlace a la imagen no esté disponible y no sea realmente funcional para el objetivo.

3.3.4 Heroku

Heroku [14] es una plataforma SaaS, al igual que Cloudinary, pero su servicio es el de computación en la nube. Gracias a Heroku se ha podido desplegar el servidor y el cliente en la nube y que sea accesible a través de cualquier dispositivo con acceso a Internet, satisfaciendo así el requisito de que la aplicación pueda ser utilizada a través de navegador web tanto en móviles como ordenadores. Además, su versión gratuita proporciona hasta 4500 peticiones por hora, cumpliendo con el requisito no funcional propuesto.

El único impedimento que presenta es que al cabo de 30 minutos sin recibir peticiones, deja el servidor en reposo, con lo que retrasa unos segundos la respuesta del servidor si se encuentra en este estado, pero no resulta un problema para cumplir con los requisitos no funcionales.

3.3.5 Datos Abiertos del Ayuntamiento de Málaga

El Ayuntamiento de Málaga ofrece un portal de Internet con datos [4] de multitud de servicios que ofrece y en varios formatos. Gracias a dos de estos documentos (los cuales tienen, ambos, la referencia “EPSG 4326” y están en formato GEOJSON), se ha podido procesar las zonas deportivas, junto a las máquinas que están presentes en cada una de ellas, que hay dentro de la ciudad y dotar de información inicial a la aplicación para el uso de nuestros usuarios.

3.3.6 Balsamiq

Balsamiq [15] es una herramienta online que permite realizar bocetos de interfaz de usuario, con gran número de complementos y un uso muy intuitivo y sencillo. Por esto se ha utilizado para crear una primera idea de cómo se iba a estructurar la interfaz de usuario de la aplicación.

3.3.7 Postman

Postman [16] es una aplicación que permite realizar pruebas a la API de la aplicación mediante HTTP request, permitiendo realizar peticiones del tipo GET, PUT, POST y DELETE.

Ha sido escogida para este proyecto por su interfaz gráfica, la cual facilita su uso, y por ser totalmente gratuita.

4

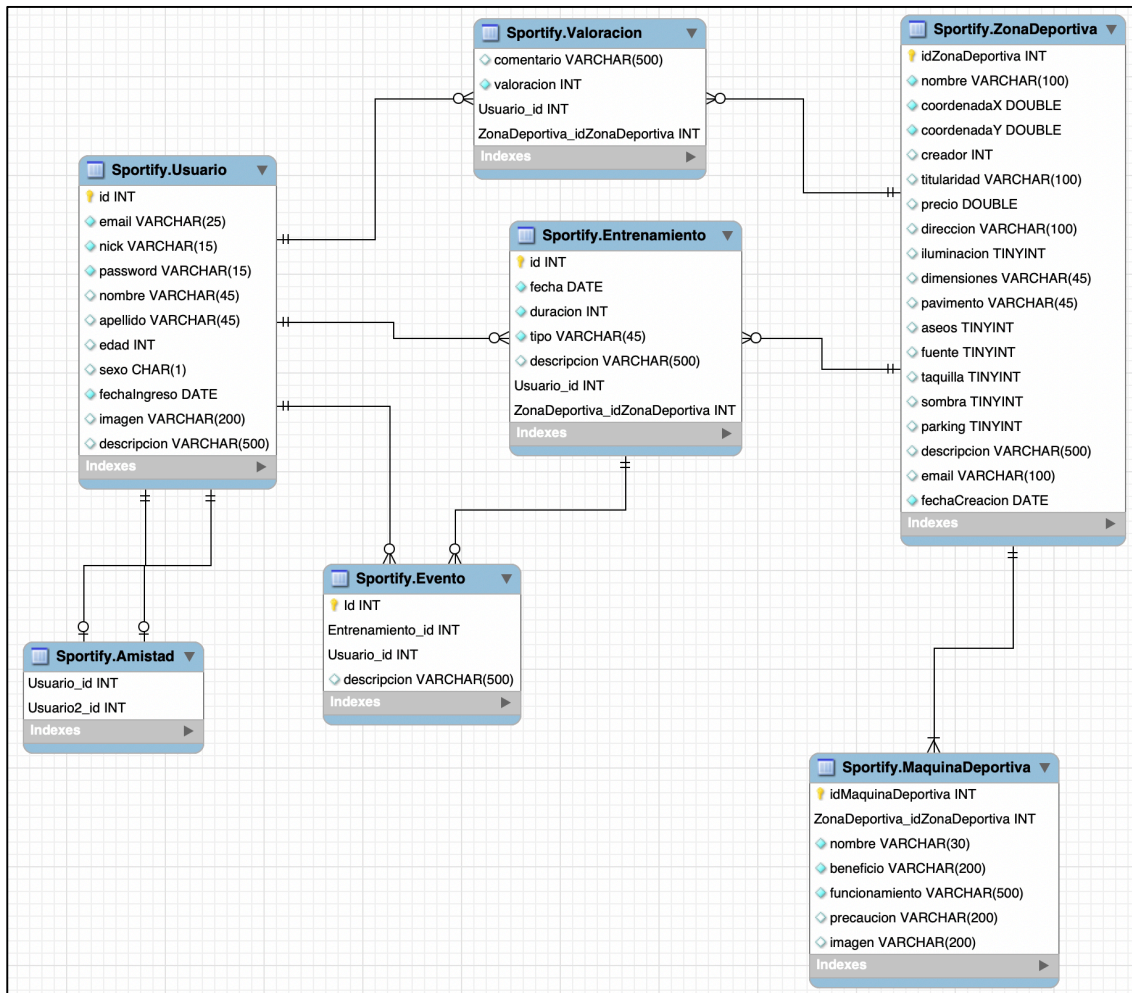
Modelado y diseño de la aplicación

En esta sección se mostrará cómo se empezó a conceptualizar la aplicación. El desarrollo de modelado y diseño se centró en el esquema de la base de datos, identificando cada una de las entidades que serían participes en la aplicación, y en la interfaz de usuario, buscando que fuese amigable e intuitiva.

4.1 Base de datos

En este apartado se presentará el esquema lógico de la base de datos y se comentarán las diferentes entidades de las que se compone.

El esquema de la base de datos es el que se presenta a continuación.



En la imagen superior se pueden ver las entidades que componen la base de datos y las claves foráneas de las entidades que son dependientes de otras. Se van a explicar a continuación una a una:

4.1.1 Usuario

“Usuario” es la entidad central ya que, como en toda red social, lo más importante son los usuarios con los que contará la aplicación. Cada usuario se identifica por un número entero y se autentifica en la aplicación a través de su email (el cual es

único) y su contraseña. El resto de atributos se destinan a la personalización de su perfil.

4.1.2 ZonaDeportiva

La entidad “ZonaDeportiva” es la otra gran entidad de la aplicación. Cada zona deportiva se identifica con un número entero. Además cuenta con atributos que sirven para proporcionar información sobre la misma (como las coordenadas o el nombre) y con otros pensados para el filtrado de zona por ciertas características.

4.1.3 MaquinaDeportiva

La entidad “MaquinaDeportiva” depende necesariamente de la entidad “ZonaDeportiva”, ya que toda máquina deportiva pertenece a una única zona deportiva, con lo cual se identifica por dos números enteros, uno propio y otro que hace referencia al identificador de la zona a la que pertenece. El resto de atributos sirven para complementar información sobre la máquina.

Cabe destacar que, a pesar de tener un atributo imagen, las imágenes proporcionadas por los datos abiertos del Ayuntamiento de Málaga son un enlace a un mapa, por lo que no se han utilizado en la aplicación.

4.1.4 Valoracion

“Valoracion” es la entidad utilizada para crear reseñas. Es una entidad débil que depende de “Usuario” y de “ZonaDeportiva” y cuenta con dos atributos que son los usados para realizar la reseña; valoración para la nota y comentario, el cual se puede dejar en blanco, para explicar el porqué de esa calificación.

4.1.5 Entrenamiento

La entidad “Entrenamiento” se identifica por un número entero que le pertenece y, además, depende de las entidades “Usuario” y “ZonaDeportiva”, por lo que cuenta con dos identificadores más, de tipo entero, que pertenecen al usuario que registra el entrenamiento y a la zona deportiva en la que lo hace. El resto de atributos sirven para aportar información al usuario.

4.1.6 Evento

“Evento” es una entidad que depende de las entidades “Entrenamiento” (ya que un evento es un entrenamiento configurado para todo el que desee asistir) y “Usuario” (que es quien se apunta a dicho evento, ya que el organizador es el creador del entrenamiento). Además, de dos valores enteros, los cuales hacen referencia a las dos entidades de las que depende, también se identifica por un tercer valor entero propio. Cuenta también con un atributo “descripción” que es opcional, por añadir más información sobre cada evento.

4.1.7 Amistad

La entidad “Amistad” representa, valga la redundancia, la amistad entre dos usuarios. Depende de la entidad “Usuario” y se identifica por dos números enteros que referencian al identificador de los dos usuarios implicados.

4.2 Bocetos de interfaz de usuario

En este apartado se mostrarán y comentarán imágenes, en las cuales se podrá ver las primeras ideas conceptuales y de navegación que se tenían sobre la interfaz que vería

el usuario final de la aplicación. Como todo boceto, dista del resultado final, pero fue de gran ayuda para comprobar que se tenía la misma idea común.

En esta primera imagen se puede ver cómo sería la pantalla principal de la aplicación. En ella, un usuario invitado vería información sobre las zonas disponibles y se marcarían en el mapa. Además, se podrían filtrar acorde a unos parámetros.



Ilustración 1: Pantalla principal de invitado

La siguiente ilustración muestra cómo sería la pantalla principal de un usuario autenticado. En ella podría ver y filtrar las zonas deportivas y navegar por los diferentes elementos de la aplicación desde un menú del tipo “navbar”.

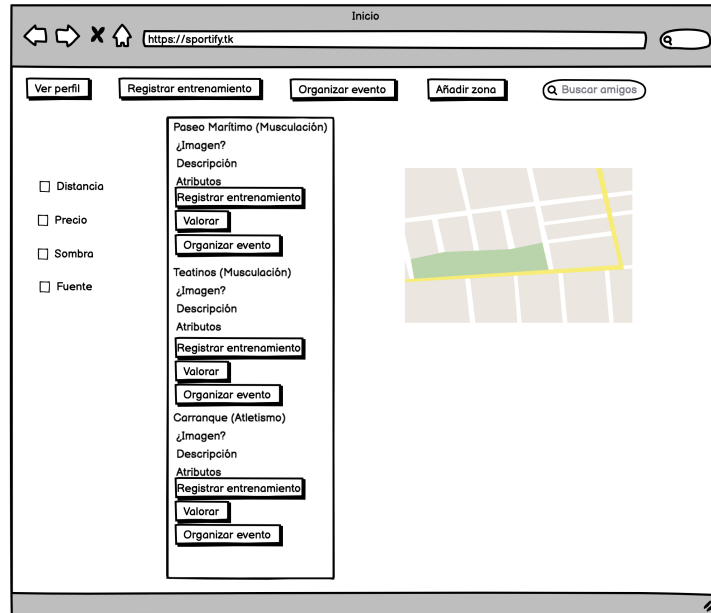


Ilustración 2: Pantalla principal de usuario

En el siguiente boceto se muestra la pantalla de perfil del usuario autenticado; desde ella podría modificar su perfil y/o cerrar sesión.

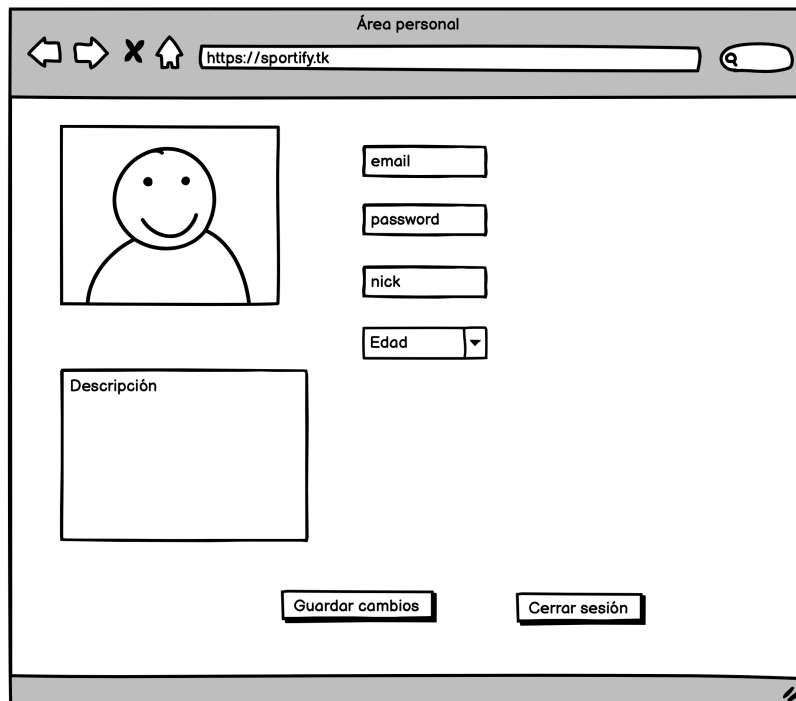


Ilustración 3: Pantalla de perfil de usuario

En la próxima imagen se puede ver el perfil de otro usuario diferente al que está autenticado. En ella se ofrece la posibilidad de añadirlo como amigo o, si ya está en la lista de amigos, borrarlo como tal.

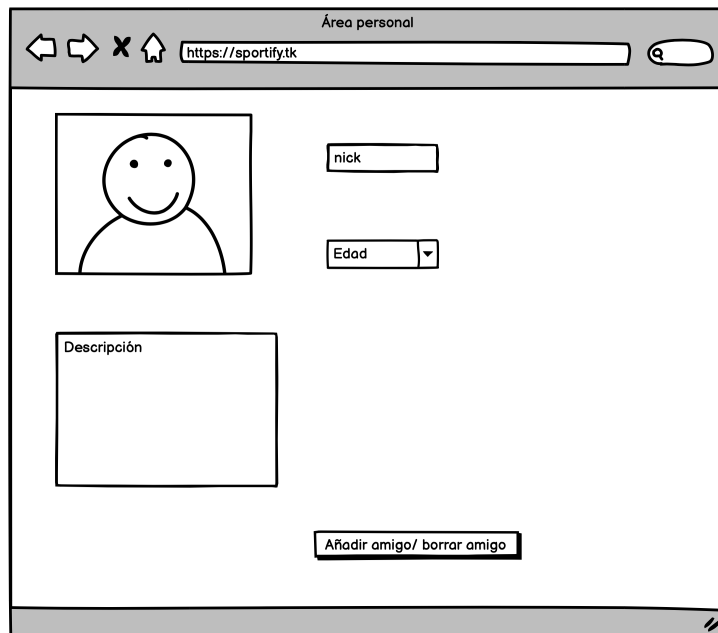


Ilustración 4: Pantalla de perfil de otro usuario

En el siguiente boceto se muestra cómo sería la pantalla para añadir una nueva zona a la aplicación a través de un formulario.

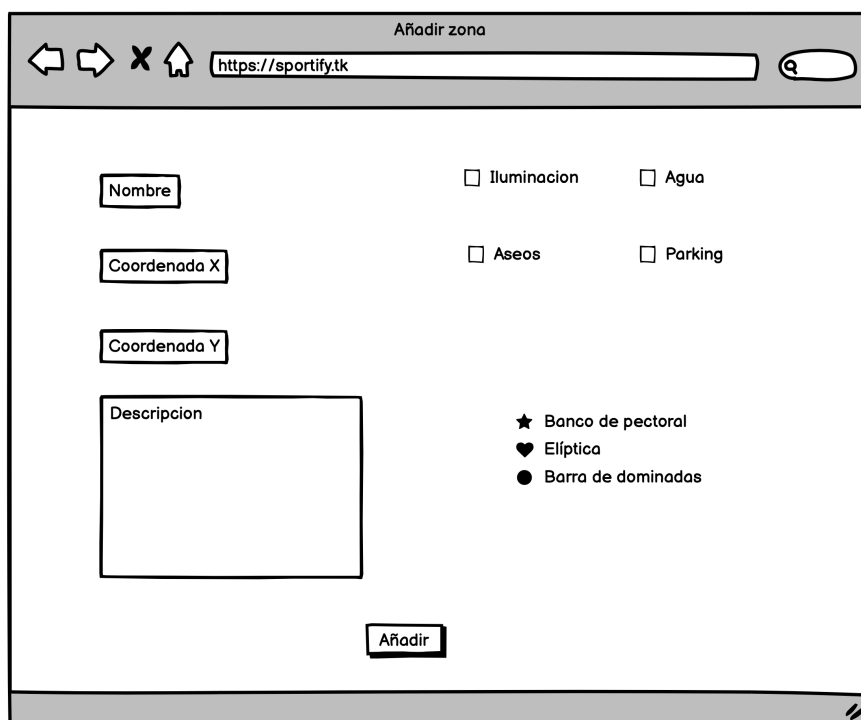


Ilustración 5: Pantalla para añadir una nueva zona

La próxima ilustración muestra la pantalla que ofrece un formulario para registrar un nuevo entrenamiento.

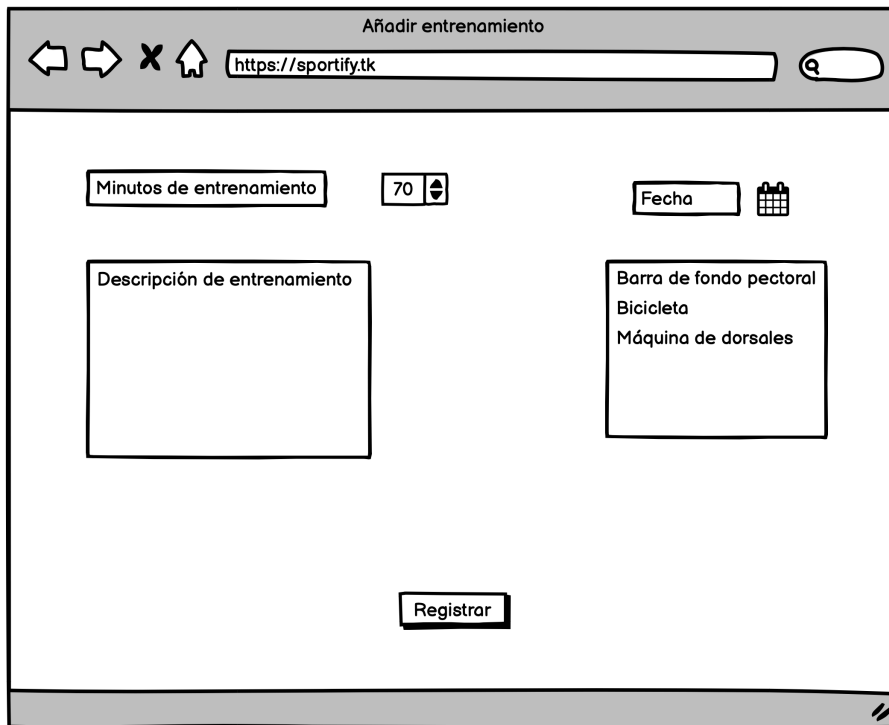


Ilustración 6: Pantalla para registrar un entrenamiento

En la siguiente imagen se muestra la pantalla referente a los eventos. En ella un usuario puede organizar un nuevo evento y/o apuntarse a alguno ya existente.

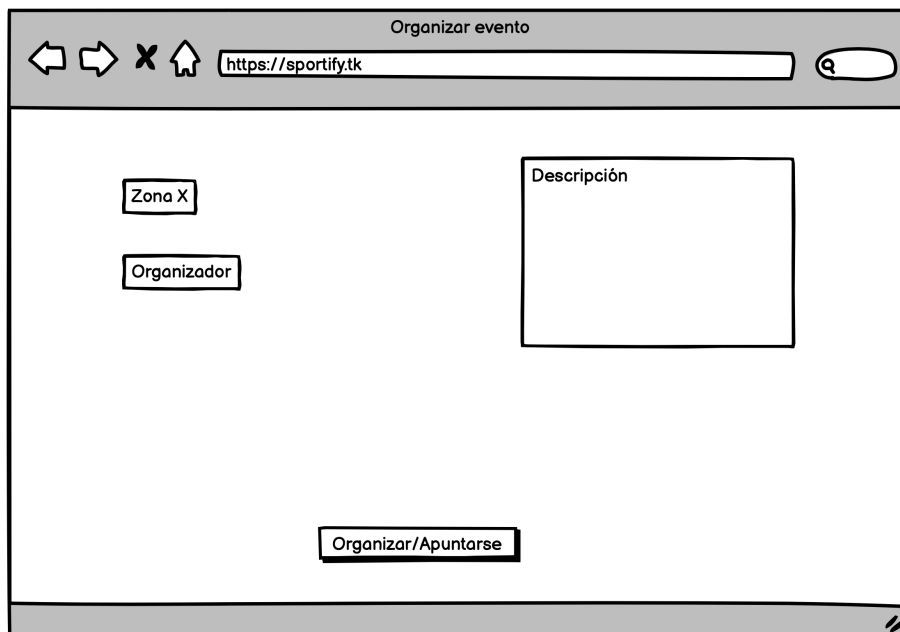


Ilustración 7: Pantalla sobre los eventos

5

Metodología e implementación

Este trabajo consta en el desarrollo de una aplicación web con diversas funcionalidades y características, la cual sigue una estructura del tipo cliente/servidor. Esta estructura se caracteriza porque el usuario final, a través del uso de la interfaz del programa que se ejecuta en el lado del cliente (como ya hemos comentado anteriormente), realiza peticiones HTTP (GET, POST, PUT y DELETE) para interactuar con la aplicación. Estas peticiones llegan al servidor quien, haciendo llamadas a diferentes componentes como la base de datos u otras páginas web, responde a la petición del usuario enviándole la información solicitada.

Esto proporciona dos grandes ventajas, una es que varios usuarios pueden interactuar simultáneamente con la aplicación y, la segunda, que la lógica de la aplicación queda

totalmente desacoplada; de esta forma, si hubiese más de un servidor para satisfacer las peticiones de los usuarios, se podría mejorar los tiempos de respuesta y disminuir la probabilidad de sufrir una denegación de servicios.

A partir de lo comentado, se ha seleccionado una metodología y se han seguido unos pasos concretos para la implementación de la aplicación, en busca de optimizar el desarrollo del proyecto.

5.1 Metodología

Se ha optado por seguir una metodología iterativa, la cual aporta un gran beneficio al desarrollo de la misma, ya que permite desacoplar funcionalidades y modularizarla en diferentes partes. Al final de cada iteración, se ha realizado una verificación de que el desarrollo se ha hecho correctamente y cumple con las expectativas esperadas.

En primer lugar, se realizó la captura de todos los requisitos funcionales y no funcionales. A continuación, y una vez estaban bien definidos, se llevaron a cabo tres iteraciones. En cada una de dichas iteraciones se siguieron los siguientes pasos:

1. Selección de varios requisitos funcionales.
2. Análisis formal de los requisitos funcionales elegidos y desarrollo de los diagramas y documentación oportuna.
3. Realización de pruebas para comprobar que funcionen correctamente los nuevos requisitos implementados y pruebas de integración con las funcionalidades ya existentes.
4. Incorporación de los aspectos concretos, de la parte implementada, en el manual de usuario y dossier final del proyecto para llevar un control diario del desarrollo.

En una fase adicional, se ha llevado a cabo el estudio y desarrollo de medidas de seguridad para evitar los ataques más conocidos a día de hoy a aplicaciones web y las pruebas correspondientes. Además en esta última fase se ha llevado a cabo el despliegue de la aplicación en la nube.

5.2 Implementación

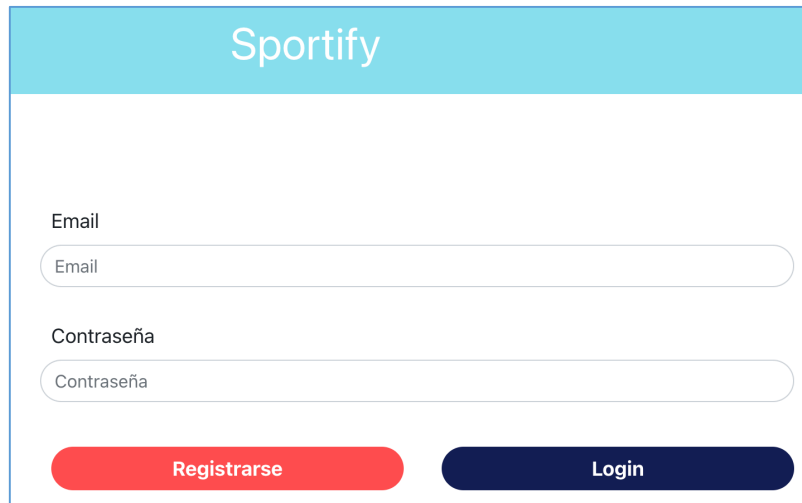
Siguiendo la metodología definida en el anterior epígrafe, la aplicación se ha implementado de forma incremental a través de cuatro iteraciones. La primera de ellas aborda el desarrollo de la interfaz del cliente, la segunda y tercera se centran en la funcionalidad de la aplicación satisfaciendo los requisitos funcionales y la cuarta, en poner a prueba la seguridad, además de desplegar la aplicación final en la nube.

5.2.1 Primera iteración

En primer lugar, se decidió implementar la parte del cliente, por lo que no se abordaron requisitos funcionales. El propósito de empezar por la interfaz de usuario fue tener el esqueleto de lo que sería la aplicación a la hora de incorporar la funcionalidad y hacer las primeras pruebas no formales para ver la correcta implementación.

Se optó por hacer una interfaz agradable e intuitiva, agregando en este punto la navegabilidad de la aplicación, y, así, satisfacer el requisito no funcional de poder realizar cualquier operación con la aplicación en menos de 5 clicks de ratón o pulsaciones de pantalla.

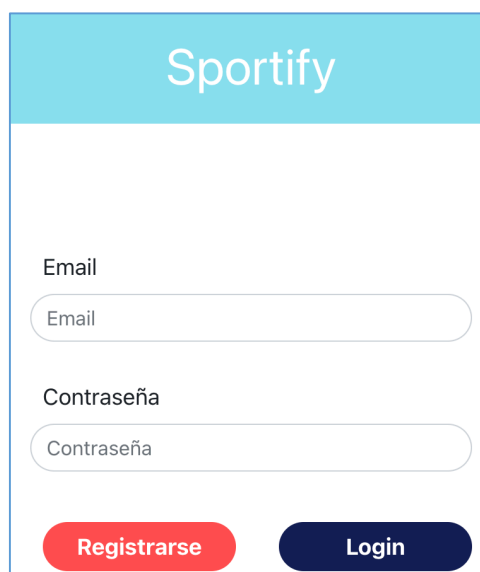
Aquí se muestra un ejemplo de cómo sería la interfaz de usuario desde un navegador web en un ordenador.



En esta etapa se incorporó la tecnología Redux-saga, para añadir una capa intermedia que controlase los estados por los que pasa la aplicación en las peticiones del usuario.

También se ajustó la interfaz de usuario para que fuese *responsive* con los navegadores web desde un dispositivo móvil.

Aquí se muestra un ejemplo de cómo sería la interfaz de usuario desde un móvil.



Por último, se pasaron las pruebas al *front-end* elaborado durante esta primera iteración, probando que la navegabilidad de la interfaz fuese la esperada y se realizó la documentación referente a lo elaborado en esta etapa.

Además, se tuvo una reunión entre tutores y alumno para confirmar que la primera iteración había concluido y se habían alcanzado los objetivos.

5.2.2 Segunda iteración

En esta segunda iteración se empezó a trabajar sobre el lado del servidor, no solo en la implementación, sino también en los recursos de los que haría uso, como la base de datos.

Como ya se ha comentado, se eligió una base de datos NoSQL y se realizó el esquema de las entidades que contendría con los atributos necesarios. Una vez estaba el esquema finalizado, se empezó a implementar en código Python, a través del framework Django y se migró a MongoDB. Gracias a la incorporación de Django al proyecto, el panel de administración de los datos de la aplicación se elaboró de forma automática a partir del esquema de la base de datos.

A continuación se seleccionaron los requisitos funcionales a implementar; estos fueron todos los referentes a manejo de usuarios (registro, login, perfil y lista de amigos) y a la visualización, filtrado y marcación en el mapa de las diferentes zonas deportivas.

Una vez implementados, se pasaron las pruebas formales y se elaboró la documentación pertinente a esta segunda iteración.


Para concluir, se produjo una reunión por parte de los tutores y el alumno para verificar el proceso y dar por concluida la iteración.

Filtrar

- Iluminación
- Aseos
- Fuente
- Taquilla
- Sombra
- Parking

Zonas disponibles

- PARQUE FUENTE OLLETAS
- INST. DEP. EN BDA. LOS CASINIS
- INS. DEP. AL AIRE LIBRE EN BARRIADA LAS VIRREINAS 1
- INST. DEPORT. AL AIRE LIBRE EN BDA. PARQUE ARROYO DEL CUARTO
- INST. DEP. EN BDA. JARDÍN DE MÁLAGA
- INST. DEPORT. AL AIRE LIBRE EN PARQUE DE LA HIGUERETA
- INST. DEP. EN BDA. EL PALO
- INST. DEP. AL AIRE LIBRE JUNTO A LA FAROLA



1
Atr
Sig

5.2.3 Tercera iteración

La tercera iteración se centró en los requisitos que pretendían cumplir el objetivo final de la aplicación, incentivar a las personas a hacer deporte y organizar sus entrenamientos.

Se implementaron los requisitos para registrar entrenamientos, organizar eventos y apuntarse a eventos. Además, también se implementaron el resto de requisitos funcionales de la aplicación agregar nuevas zonas, comentar y valorar zonas y se mejoró el filtro de las zonas, haciéndolo más eficiente y reutilizable.

Sportify

Duración de entrenamiento

Fecha

Tipo de entrenamiento

Descripción del entrenamiento

- 3 x 12 barras paralelas verticales
- 3 x 15 barras paralelas
- 3 x 12 poleas
- 4 x 10 barra a media altura

Máquinas disponibles:

disco de muñecas, escalera de dedos, giro de cintura, movilidad de dedos, poleas, serpiente, silla con pedales, volante, barra a media altura, barra giro muñecas, barras paralelas verticales, barras paralelas, base inestable, disco de dedos y muñecas, asiento con apoyo, plataforma de estiramiento, disco de muñecas, escalera de dedos, giro de cintura, movilidad de dedos, poleas, serpiente, silla con pedales, volante, barra a media altura, barra giro muñecas, barras paralelas verticales, barras paralelas, base inestable, disco de dedos y muñecas, asiento con apoyo, plataforma de estiramiento.

INST. DEP. EN BDA. JARDÍN DE MÁLAGA

58

A continuación se pasaron las pruebas a los requisitos implementados y, además, gracias a que Postman permite guardar las pruebas realizadas y volverlas a repetir, se volvieron a realizar las pruebas de la iteración anterior con el fin de que al incorporar nuevas funcionalidades no se hubiesen visto afectadas las que ya existían.

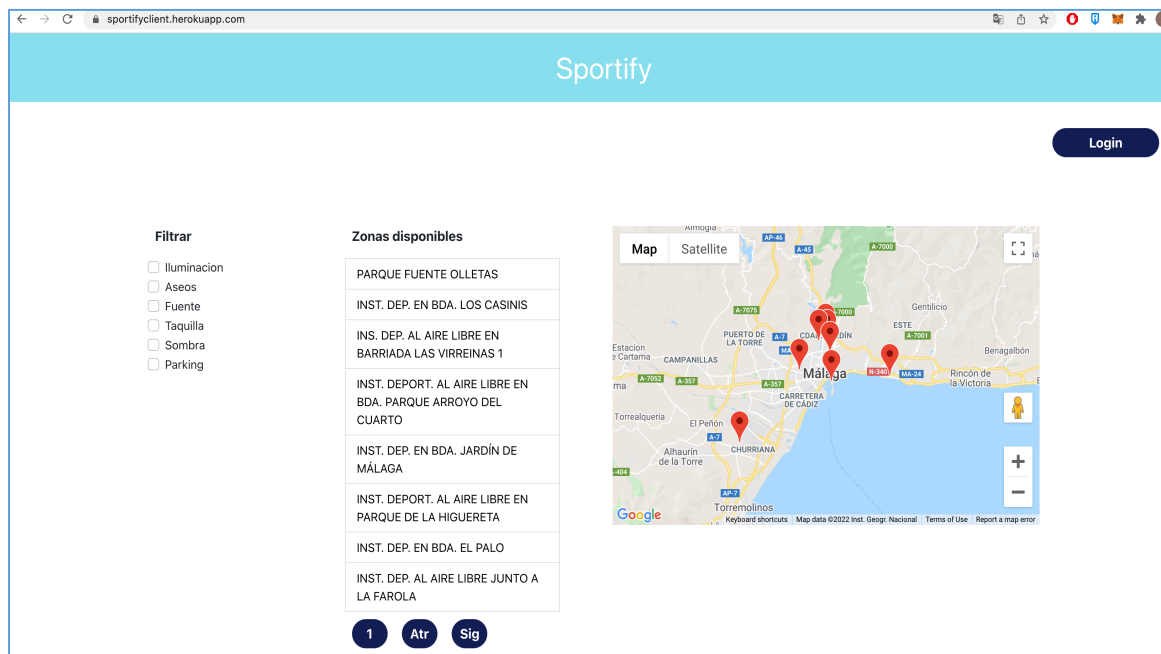
Para concluir, se produjo una reunión entre tutores y alumno para verificar el correcto desarrollo de la tercera iteración y dar la misma por concluida.

5.2.4 Cuarta iteración

En esta cuarta, y última iteración, se llevaron a cabo las tareas y pruebas de seguridad de la aplicación, abordando la inyección de SQL, los ataques de Cross-site scripting y los ataques de fuerza bruta contra el formulario login de usuario.

Adicionalmente se corrigieron algunos errores menores y se realizaron algunos cambios sobre la privacidad (como por ejemplo buscar amigos por su nick en lugar de por su dirección de email) y, como comentaremos posteriormente, se llevó a cabo el despliegue de la aplicación en la nube.

Más adelante se contará con mayor detenimiento cómo se ha realizado el despliegue, pero se incorpora una imagen que muestra el cliente alojado en el dominio “<https://sportifyclient.herokuapp.com>”.



Por último, esta última etapa tuvo un carácter más documental, ya que, además de realizar la documentación al igual que en las otras iteraciones, se unificó toda la documentación en el presente dossier, se estructuró y organizó siguiendo la plantilla establecida para tal fin.

A lo largo de esta iteración se produjeron varias reuniones ya que no solo se debía verificar que se hubiesen cumplido los objetivos, al igual que en las otras iteraciones, sino que también había que dejar la documentación de la forma más formal y correcta posible, lo que requirió varias revisiones.

6

Pruebas realizadas y despliegue en la nube

En esta sección se van a describir las pruebas realizadas y, brevemente, cómo se ha realizado el despliegue de la aplicación. Las pruebas realizadas a la aplicación se han dividido en tres partes.

En primer lugar se ha comprobado la navegabilidad de la aplicación, probando únicamente la interfaz de usuario y que todas las ventanas se mostrasen como se esperaba, además de que no hubiese ningún punto desde el que no se pudiese navegar a otra ventana a través de un botón. Estas se han registrado en una tabla y se han llevado a cabo sobre la misma aplicación.

En segundo lugar se ha comprobado que las llamadas realizadas por el servidor devolviesen el contenido esperado, sin fugas de información ni fallos en las llamadas

y/o la confirmación de la petición. Para esta tarea se ha utilizado la herramienta Postman.

Para concluir se han añadido unas pruebas de seguridad para comprobar que la aplicación presente unos mínimos frente a un ataque por parte de algún cibercriminal.

Una vez validado que el sistema pasaba todas las pruebas realizadas, se ha puesto en producción en Heroku.

6.1 Pruebas Front-end

Las pruebas de la aplicación del lado del cliente se han realizado durante la primera iteración por parte del desarrollador y cumpliendo todas el resultado esperado. Para ilustrar dichas pruebas, se ha creado una tabla con los siguientes campos:

- **Fecha:** Día, mes y año en el que se pasó la prueba.
- **Desarrollador:** Persona encargada de realizar la prueba.
- **Estado:** Pantalla en la que se encuentra la aplicación antes de la prueba.
- **Acción:** Interacción del usuario con la aplicación, la cual provocará un cambio de pantalla.
- **Resultado:** Pantalla que se espera que se muestre tras la interacción del usuario.
- **Validación:** Indica si la prueba se ha pasado con éxito (tick verde) o si, por el contrario, ha ocurrido un error y no se ha pasado la prueba (cruz roja).

A continuación se muestra el resultado de las pruebas que se han realizado.

Fecha	Desarrollador	Estado	Acción	Resultado	Validación
30/11/2021	Antonio David López Mostazo	Pantalla inicial	Pulsar botón "Login"	Redirige a la pantalla con el formulario de login	✓
30/11/2021	Antonio David López Mostazo	Pantalla de login	Pulsar botón "Registrarse"	Redirige a la pantalla con el formulario de registro	✓
30/11/2021	Antonio David López Mostazo	Pantalla de registro	Pulsar botón "Registrarse"	Redirige a la pantalla de inicio autenticado	✓
30/11/2021	Antonio David López Mostazo	Pantalla de login	Pulsar botón "Login"	Redirige a la pantalla de inicio autenticado	✓
30/11/2021	Antonio David López Mostazo	Pantalla de inicio autenticado	Pulsar link "Perfil"	Redirige a la pantalla de perfil del usuario autenticado	✓
30/11/2021	Antonio David López Mostazo	Pantalla de perfil del usuario autenticado	Pulsar botón "Cerrar sesión"	Redirige a la pantalla de inicio	✓
30/11/2021	Antonio David López Mostazo	Pantalla de perfil del usuario autenticado	Pulsar botón "Guardar cambios"	Redirige a la pantalla de perfil del usuario autenticado	✓

30/11/2021	Antonio David López Mostazo	Pantalla de inicio autenticado	Pulsar link "Registrar entrenamiento"	Redirige a la pantalla con el formulario de registro de entrenamiento	✓
30/11/2021	Antonio David López Mostazo	Pantalla del formulario de registro de entrenamiento	Pulsar botón "Registrar entrenamiento"	Redirige a la pantalla de inicio autenticado	✓
01/12/2021	Antonio David López Mostazo	Pantalla de inicio autenticado	Pulsar link "Organizar evento"	Redirige a la pantalla con el formulario de organización de evento	✓
01/12/2021	Antonio David López Mostazo	Pantalla del formulario de organización de evento	Pulsar botón "Organizar"	Redirige a la pantalla de inicio autenticado	✓
01/12/2021	Antonio David López Mostazo	Pantalla de inicio autenticado	Pulsar link "Apuntarse a evento"	Redirige a la pantalla con la lista de eventos disponibles	✓
01/12/2021	Antonio David López Mostazo	Pantalla con la lista de eventos disponibles	Pulsar botón "Apuntarse"	Redirige a la pantalla de inicio autenticado	✓

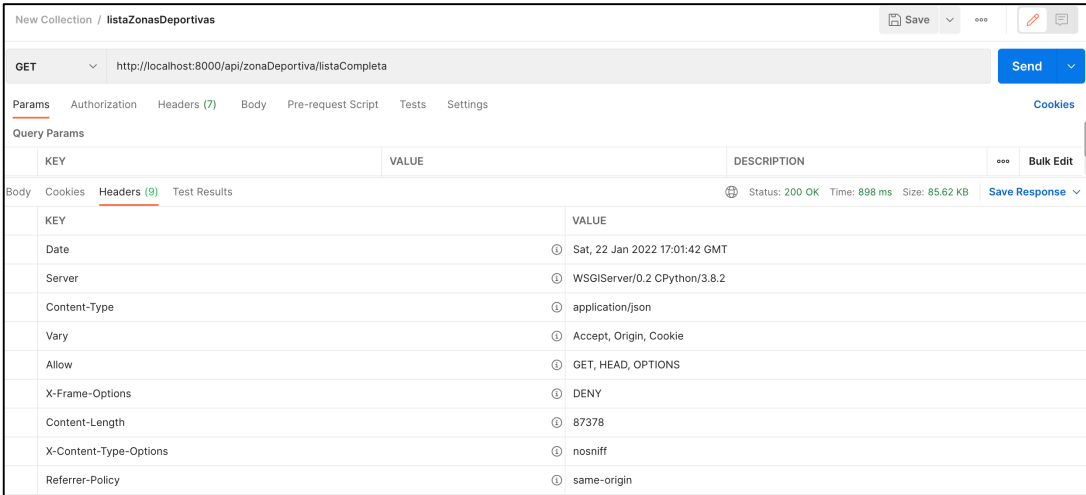
01/12/2021	Antonio David López Mostazo	Pantalla de inicio autenticado	Pulsar link "Añadir zona"	Redirige a la pantalla con el formulario para añadir una nueva zona	✓
01/12/2021	Antonio David López Mostazo	Pantalla con el formulario para añadir una nueva zona	Pulsar botón "Agregar zona"	Redirige a la pantalla de inicio autenticado	✓
01/12/2021	Antonio David López Mostazo	Pantalla de inicio autenticado	Pulsar link "Añadir amigos"	Redirige a la pantalla con el buscador de usuarios	✓
01/12/2021	Antonio David López Mostazo	Pantalla con el buscador de usuarios	Pulsar botón "Atrás"	Redirige a la pantalla de inicio autenticado	✓
01/12/2021	Antonio David López Mostazo	Pantalla con el buscador de usuarios	Pulsar botón "Buscar"	Redirige a la pantalla de perfil del usuario buscado	✓
01/12/2021	Antonio David López Mostazo	Pantalla de perfil de un usuario	Pulsar botón "Borrar amigo"	Se mantiene en la pantalla de perfil del usuario	✓
01/12/2021	Antonio David López Mostazo	Pantalla de perfil de un usuario	Pulsar botón "Agregar amigo"	Se mantiene en la pantalla de perfil del usuario	✓

6.2 Pruebas Back-end

Las pruebas de la aplicación en el lado del servidor se han realizado durante la segunda y tercera iteración, aunque las imágenes muestran las pruebas realizadas al final de la tercera iteración, ya que las de la segunda iteración, como ya se ha comentado, se repitieron con la aplicación ya terminada para comprobar que ninguna funcionalidad se hubiese visto afectada.

6.2.1 Listar zonas deportivas

En la siguiente prueba se puede ver la cabecera de la respuesta del servidor a la petición de listar las zonas deportivas, la cual es correcta y devuelve el contenido en formato JSON. Se ha elegido mostrar la cabecera porque contiene gran cantidad de objetos en formato JSON y mostrar su contenido sería poco útil.



The screenshot shows a REST client interface with the following details:

- Method: GET
- URL: http://localhost:8000/api/zonaDeportiva/listaCompleta
- Status: 200 OK
- Time: 898 ms
- Size: 85.62 KB

KEY	VALUE
Date	Sat, 22 Jan 2022 17:01:42 GMT
Server	WSGIServer/0.2 CPython/3.8.2
Content-Type	application/json
Vary	Accept, Origin, Cookie
Allow	GET, HEAD, OPTIONS
X-Frame-Options	DENY
Content-Length	87378
X-Content-Type-Options	nosniff
Referrer-Policy	same-origin

6.2.2 Crear zona deportiva

En esta prueba se crea una zona deportiva a través de una llamada del tipo POST al servidor, el cual devuelve un mensaje de confirmación.

POST http://localhost:8000/api/zonaDeportiva/crear

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "nombre": "Zona Postman18",
3   "coordenadaX": 36.73613878,
4   "coordenadaY": -4.415189,

```

Body Cookies Headers (9) Test Results Status: 200 OK Time: 769 ms Size: 291 B Save Response

KEY	VALUE
Date	Sun, 23 Jan 2022 08:49:42 GMT
Server	WSGIServer/0.2 CPython/3.8.2
Content-Type	text/html; charset=utf-8
Vary	Accept, Origin, Cookie
Allow	POST, OPTIONS
X-Frame-Options	DENY
Content-Length	0
X-Content-Type-Options	nosniff
Referrer-Policy	same-origin

6.2.3 Comprobar Api login

En esta prueba se hace un login contra el panel de administrador de la aplicación.

En la imagen se puede ver la cabecera de respuesta de éxito del servidor.

GET ApiLogin

Sportify / ApiLogin

GET http://localhost:8000/api/usuario/login

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Type Basic Auth

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Username admin

Password

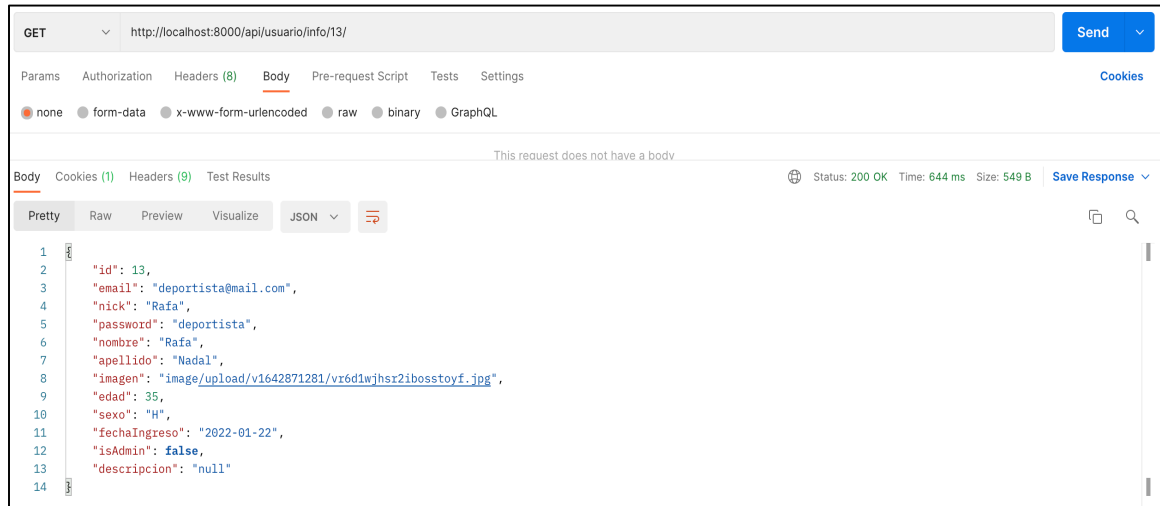
Show Password

Body Cookies Headers (9) Test Results Status: 200 OK Time: 888 ms Size: 290 B Save Response

KEY	VALUE
Date	Sun, 23 Jan 2022 10:36:55 GMT
Server	WSGIServer/0.2 CPython/3.8.2
Content-Type	application/json
Vary	Accept, Origin, Cookie
Allow	GET, HEAD, OPTIONS
X-Frame-Options	DENY
Content-Length	2
X-Content-Type-Options	nosniff
Referrer-Policy	same-origin

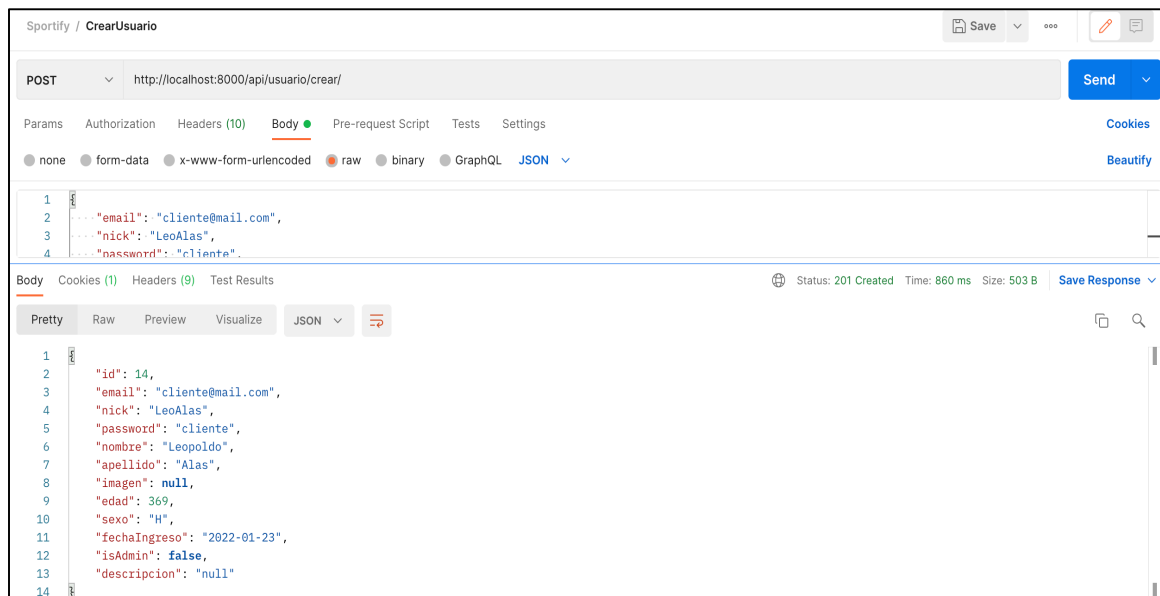
6.2.4 Listar información de usuario

En esta prueba se hace una petición de tipo GET al servidor, con el fin de obtener la información de un usuario a partir de su identificador (id). El servidor devuelve el usuario en formato JSON.



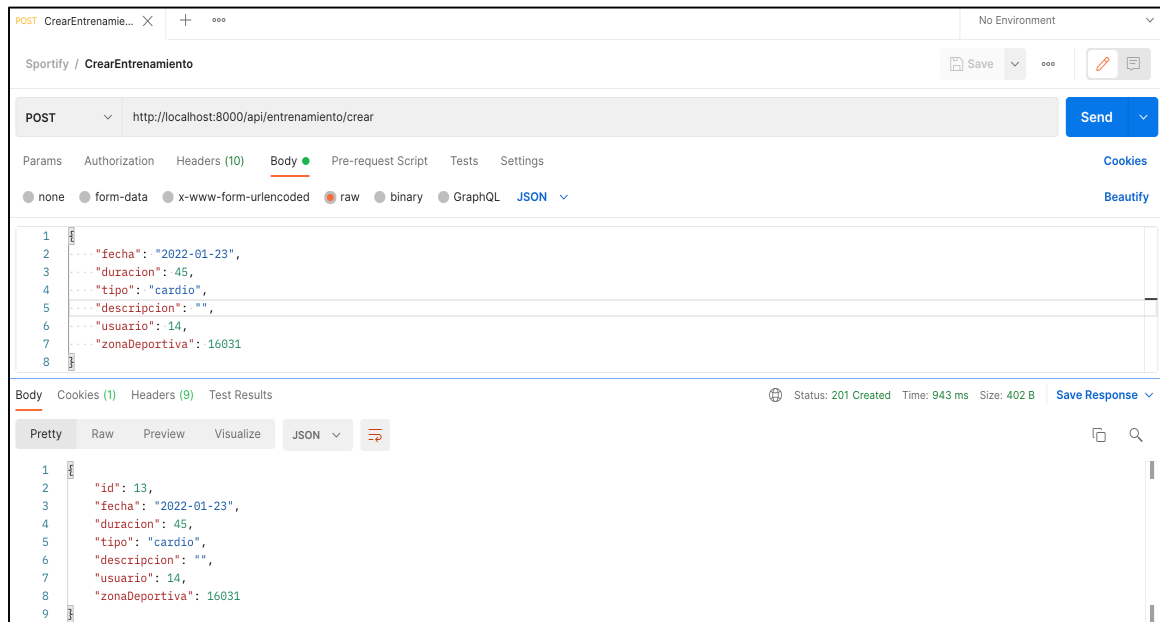
6.2.5 Crear usuario

En esta prueba se envía una petición POST al servidor con el fin de crear un nuevo usuario, pasando los atributos del mismo en formato JSON. El servidor responde confirmando la creación del usuario.



6.2.6 Crear entrenamiento

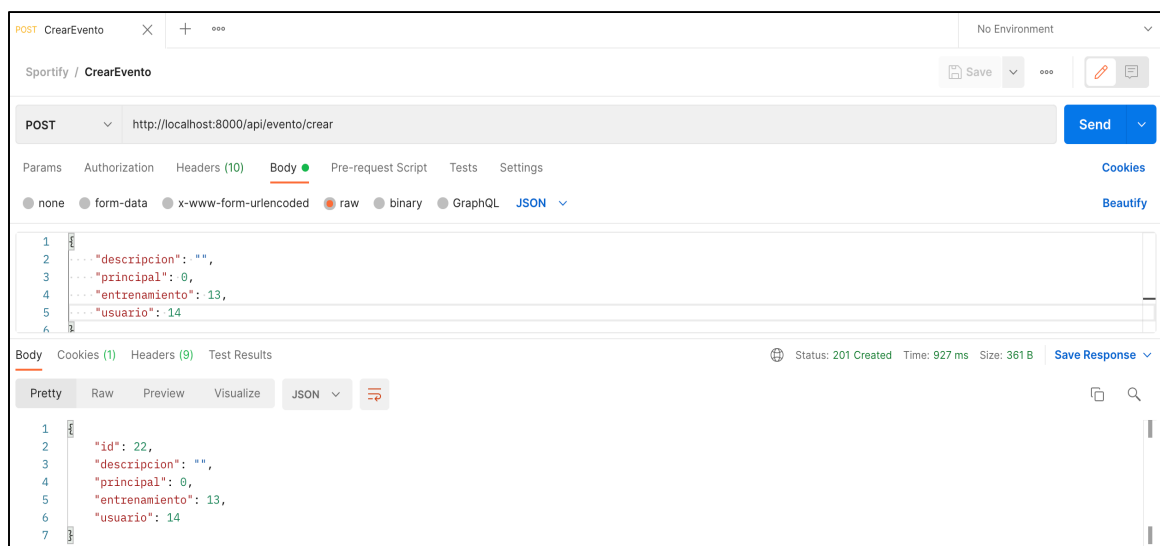
En esta prueba se envía una petición POST al servidor con el fin de crear un nuevo entrenamiento, pasando los atributos del mismo en formato JSON. El servidor responde confirmando el registro del entrenamiento.



The screenshot shows a REST client interface for a POST request to `http://localhost:8000/api/entrenamiento/crear`. The request body is a JSON object with the following fields: `fecha` (2022-01-23), `duracion` (45), `tipo` (cardio), `descripcion` (empty string), `usuario` (14), and `zonaDeportiva` (16631). The response status is 201 Created, with a time of 943 ms and a size of 402 B. The response body is a JSON object with the following fields: `id` (13), `fecha` (2022-01-23), `duracion` (45), `tipo` (cardio), `descripcion` (empty string), `usuario` (14), and `zonaDeportiva` (16631).

6.2.7 Crear evento

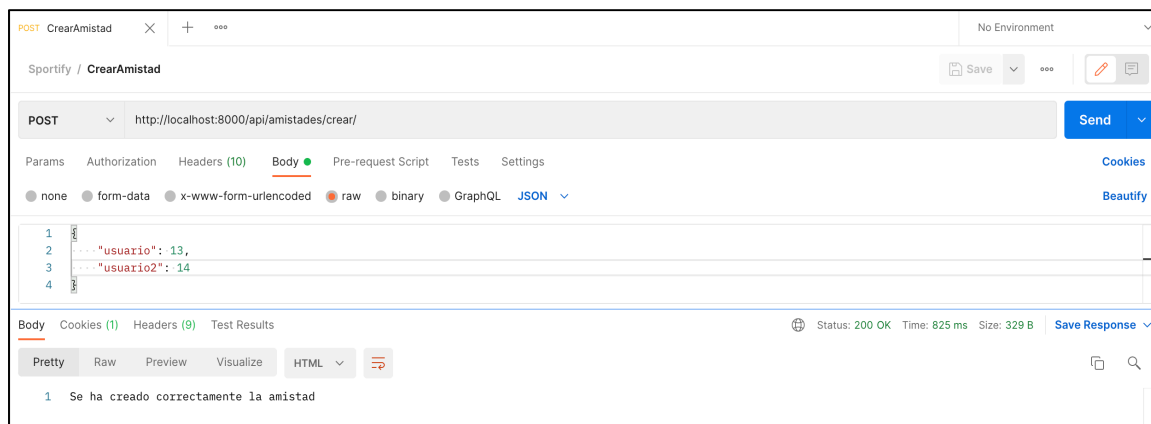
En esta prueba se envía una petición POST al servidor con el fin de crear un nuevo evento en la aplicación, pasando los atributos del mismo en formato JSON al servidor. Este responde confirmando la creación del evento.



The screenshot shows a REST client interface for a POST request to `http://localhost:8000/api/evento/crear`. The request body is a JSON object with the following fields: `descripcion` (empty string), `principal` (0), `entrenamiento` (13), and `usuario` (14). The response status is 201 Created, with a time of 927 ms and a size of 361 B. The response body is a JSON object with the following fields: `id` (22), `descripcion` (empty string), `principal` (0), `entrenamiento` (13), and `usuario` (14).

6.2.8 Crear amistad

En esta prueba se ha comprobado la creación de una amistad entre dos usuarios diferentes, pasando el identificador de los usuarios en formato JSON en una petición POST al servidor. Este muestra un mensaje confirmando la creación de la amistad.



6.2.9 Listar máquinas deportivas

En esta prueba se ha realizado una petición del tipo GET al servidor con el fin de listar todas las máquinas deportivas con las que cuenta la aplicación. El servidor devuelve en formato JSON las máquinas deportivas almacenadas en la base de datos.

Sportify / ListaMaquinasDeportivas

GET http://localhost:8000/api/maquinaDeportiva/listaGuardar

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies (1) Headers (8) Test Results Status: 200 OK Time: 5 m 9.66 s Size: 801.77 KB Save Response

Pretty Raw Preview Visualize HTML

```

1 [{"zonaDeportiva": "1", "precaucion": "Se debe abandonar el ejercicio si aparecen mareos o dolor de cuello.Los
2 movimientos deben ser lentos y controlados.", "beneficio": "Mejorar limitaciones de la articulaci\u00f3n del
3 hombro.Aumentar flexibilidad de la mu\u00f1eca y movilidad del hombro", "funcionamiento": "Colocados debajo del arco,
4 perpendicularmente a \u00e9ste, con la espalda erguida y piernas separadas agarramos la anilla con una mano y la
5 desplazamos desde un extremo a otro; repetimos con la otra mano. Cambiamos de posici\u00f3n, para situarnos paralelos al
6 arco, y realizamos el ejercicio con las dos manos a la vez.", "imagen":
7 "http://geoportal.malaga.eu/img/deportes/fm_1_1.jpg", "nombre": "Arco para brazos", "idMaquinaDeportiva":
8 "da_deportesMaquinasZonasMusculacion.fid-678917ee-17e84126ff4-51ee"}, {"zonaDeportiva": "1", "precaucion": "Evitar
9 permanecer con la barbilla hacia arriba, para prevenir molestias cervicales.", "beneficio": "Aumentar la amplitud de
10 movimientos del hombro.Ejercitar los dedos.", "funcionamiento": "Nos colocamos frente a la m\u00e1quina, con la espalda
11 erguida, piernas separadas y ligeramente flexionadas. Ponemos un dedo \u00edndice en el escal\u00f3n situado a la altura
12 del pecho, y vamos subiendo lenta y consecutivamente, intentando tocar con cada dedo de la mano un escal\u00f3n
13 diferente. Una vez que alcancemos la altura m\u00e1xima, repetiremos el ejercicio, esta vez de arriba hacia abajo.Y
14 despu\u00e9s, repetimos el ejercicio con los dedos de la otra mano.", "imagen":
15 "http://geoportal.malaga.eu/img/deportes/fm_11_1.jpg", "nombre": "Escalera de dedos", "idMaquinaDeportiva":
16 "da_deportesMaquinasZonasMusculacion.fid-678917ee-17e84126ff4-51ed"}, {"zonaDeportiva": "1", "precaucion": "Los
17 movimientos deben ser lentos y controlados.", "beneficio": "Aumentar, o mantener, la flexibilidad y movilidad de la
18 mu\u00f1eca y del codo.", "funcionamiento": "Situados frente al aparato, y agarrando el asidero con una mano,
19 desplazamos \u00e9ste de un extremo a otro de la espiral. Repetimos con la otra mano.", "imagen":
20 "http://geoportal.malaga.eu/img/deportes/fm_12_1.jpg", "nombre": "Espiral", "idMaquinaDeportiva":
21 "da_deportesMaquinasZonasMusculacion.fid-678917ee-17e84126ff4-51ec"}, {"zonaDeportiva": "1", "precaucion": "No
22 recomendado a personas con problemas lumbares.Evitar las rotaciones bruscas.", "beneficio": "Mejorar la musculatura
23 abdominal.", "funcionamiento": "De pie, en la plataforma giratoria, con las manos en las empu\u00f1aduras y manteniendo
24 el tronco derecho, se realizan rotaciones de la pelvis, de un lado a otro, manteniendo los hombros in\u00f3viles."}]

```

6.2.10 Crear valoración

En esta prueba se envía una petición POST al servidor con el fin de crear una valoración, pasando los atributos del mismo en formato JSON. El servidor responde confirmando la creación de dicha valoración.

Sportify / CrearValoracion

POST http://localhost:8000/api/valoracion/crear

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 [{"comentario": "",
2 "valoracion": 4,
3 "usuario": 14,
4 "zonaDeportiva": 15842}]]

Body Cookies (1) Headers (9) Test Results Status: 201 Created Time: 742 ms Size: 364 B Save Response

Pretty Raw Preview Visualize JSON

```

1 [{"id": 17,
2 "comentario": "",
3 "valoracion": 4,
4 "usuario": 14,
5 "zonaDeportiva": 15842}]]

```

6.3 Pruebas de seguridad

En este último apartado se va a realizar unas pruebas para comprobar que la aplicación es segura frente a ataques del tipo inyecciónSQL, Cross-site scripting y fuerza bruta contra formularios de login.

6.3.1 Inyección de SQL

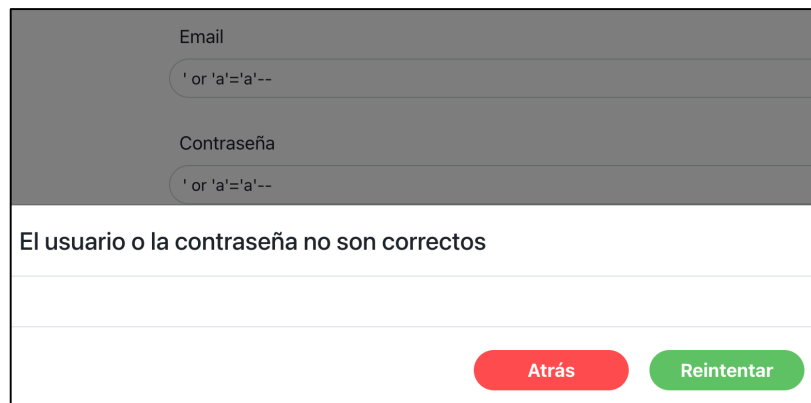
Aunque, como ya se ha comentado Django parametriza las *queries*, se van a realizar pruebas para garantizar que no sufra vulnerabilidades.

En primer lugar se introduce el comando “ or ‘a’=’a” tanto en el email como en la contraseña.



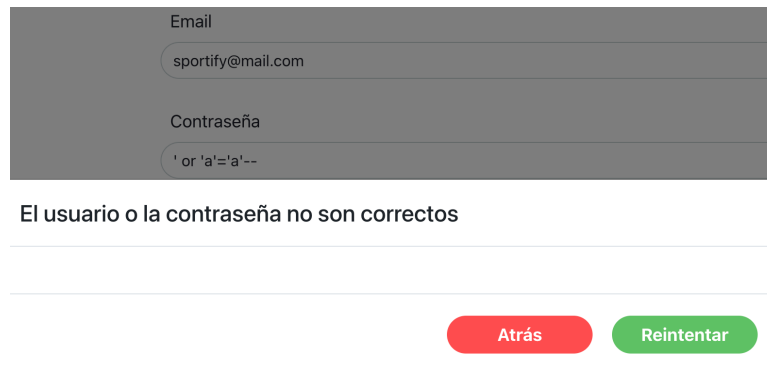
The screenshot shows a login form with two input fields: "Email" and "Contraseña". Both fields contain the SQL injection payload "' or 'a'='a". Below the fields, a message reads "El usuario o la contraseña no son correctos". At the bottom right, there are two buttons: "Atrás" (red) and "Reintentar" (green).

La segunda prueba es parecida a la primera, pero el comando será “' or 'a'='a'--” tanto en el campo de email como en el de contraseña.

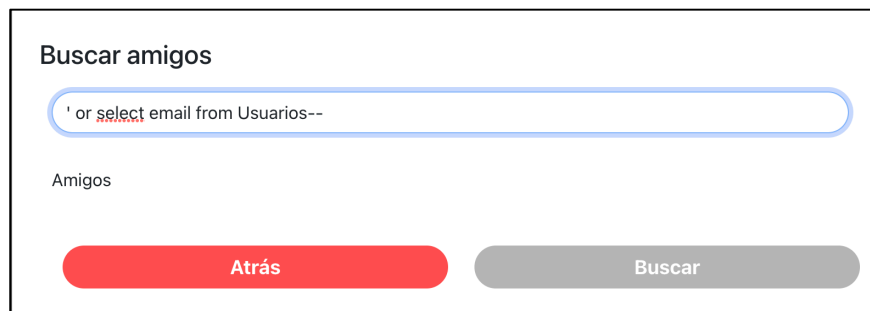


The screenshot shows a login form with two input fields: "Email" and "Contraseña". Both fields contain the SQL injection payload "' or 'a'='a'--". Below the fields, a message reads "El usuario o la contraseña no son correctos". At the bottom right, there are two buttons: "Atrás" (red) and "Reintentar" (green).

Para la tercera prueba se introduce un email correcto y se trata de inyectar código en la contraseña.



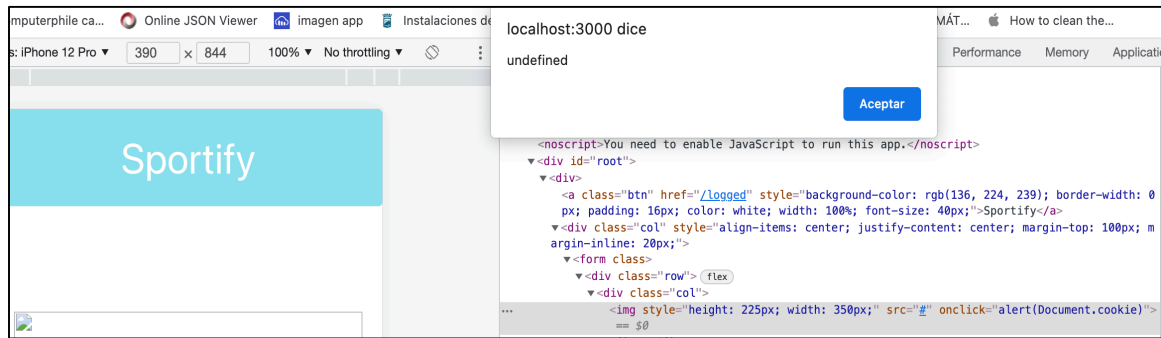
En la última prueba se ha tratado de inyectar código en el buscador de usuarios dentro de la aplicación, con el fin de obtener las direcciones de email de los usuarios de la aplicación.



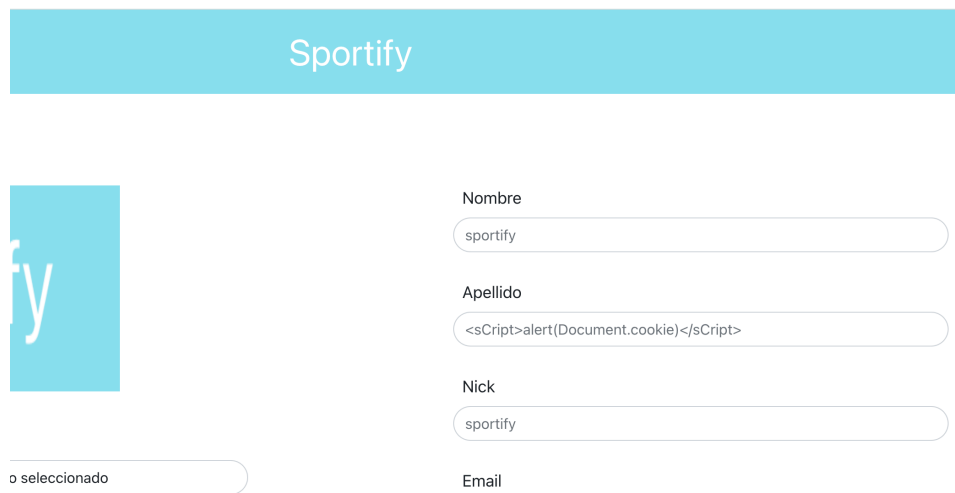
6.3.2 Cross-site scripting

Como ya se ha comentado, ReactJs presenta protección frente a Cross-site scripting, aun así se van a realizar dos pruebas para comprobar su correcto funcionamiento.

La primera prueba realizada en cuanto a ataques XSS se ha basado en intentar capturar la cookie de sesión del usuario a través de cambiar el formato de la imagen de perfil a “” lo cual carga una imagen rota debido a que el enlace no existe y, al pulsar sobre ella, no muestra la cookie de sesión del usuario.



La segunda prueba ha consistido en ingresar un script dentro de un campo de texto y comprobar si cada vez que se accediese a la pantalla que lo contiene, carga el script. Como se puede comprobar, lo muestra como texto plano y no lo ejecuta como un script.



6.3.3 Fuerza bruta

Para este requisito no se han realizado pruebas formales como tal ya que la aplicación está limitada por Heroku a 4500 peticiones por hora, así que, incluso partiendo de que se conozca el email de un usuario registrado, para obtener una contraseña real el atacante debería provocar varias denegaciones de servicio (DOS), por lo que el administrador de la aplicación vería la dirección IP del atacante y la bloquearía en los ajustes.

Concretando un poco más lo comentado, partiendo de un alfabeto de 27 caracteres como es el español, para obtener una contraseña de tan solo 3 caracteres serían

necesarios 27^3 intentos (19683 peticiones) y eso sin incluir números, caracteres especiales o mayúsculas. Con lo que partiendo de una contraseña tan poco segura como la palabra “sol”, un atacante realizando 4500 intentos cada hora, provocaría una denegación de servicios durante 3 horas, por lo que sería detectado mucho antes de conseguir una contraseña.

6.4 Despliegue en la nube

Tanto para el despliegue del lado del servidor como para el lado del cliente se ha utilizado, como ya se ha ido comentando, Heroku.

En primer lugar se ha desplegado el servidor, para lo cual se ha tenido que incorporar al proyecto un archivo llamado “Procfile”; dicho archivo incluye la herramienta Gunicorn, que es un servidor HTTP que se sirve de la puerta de enlace del proyecto (el fichero “wsgi” al que apunta dicho “Procfile”) para especificar que se trata de un servidor realizado en Python. Con él, Heroku identifica que se trata de un servidor destinado a alojarse en la web y prepara el entorno para su correcta ejecución en la nube. Adicional a este proceso, se crea un fichero llamado “requirements.txt” a través del comando “pip freeze > requirements.txt”, el cual incorpora todas las dependencias del proyecto para que, a la hora de desplegarse en la nube, se instalen.

La dirección DNS resultante que ha proporcionado Heroku para el servidor ha sido <https://sportifyserver.herokuapp.com/>.

Aquí se muestra una imagen para visualizar que el servidor responde a una petición HTTP de tipo GET en la dirección DNS comentada anteriormente.

```
← → ↻ sportifyserver.herokuapp.com/api/usuario/info/13/

Django REST framework

• admin
• Usuario Info

GET 

• json
• api

OPTIONS

Usuario Info

GET /api/usuario/info/13/

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "id": 13,
  "email": "deportista@mail.com",
  "nick": "Rafa",
  "password": "deportista",
  "nombre": "Rafa",
  "apellido": "Nadal",
  "imagen": "image/upload/https://res.cloudinary.com/sportifytfg/image/upload/v1642941061/xuuo5gdlpuhvwm1jhbyo.jpg",
  "edad": 35,
  "sexo": "H",
  "fechaIngreso": "2022-01-22",
  "isAdmin": false,
  "descripcion": "Vamos Rafa!"
}
```

En segundo lugar se ha desplegado el cliente. Este despliegue se ha realizado con posterioridad para incorporar la nueva dirección con la que cuenta el servidor en la nube a las llamadas que se realizan desde el lado del cliente.

A la hora de inicializar el directorio “.git” para poder realizar el despliegue, se crea un archivo llamado “yarn.lock”, el cual ha sido suprimido ya que el proyecto contaba con el fichero “package-lock.json”. Ambos sirven para que Heroku sepa todas las dependencias que debe instalar para el correcto despliegue de una aplicación realizada con ReactJS, pero, dado que como el proyecto se generó utilizando el sistema de gestión de paquetes “npm”, el cual genera por defecto “package-lock.json”, se ha elegido utilizar el fichero nativo.

Aquí una imagen que muestra como el cliente está alojado en el dominio DNS proporcionado por Heroku, el cual es “sportifyclient.herokuapp.com”.

Login

Filtrar

- Iluminacion
- Aseos
- Fuente
- Taquilla
- Sombra
- Parking

Zonas disponibles

PARQUE FUENTE OLLETAS
INST. DEP. EN BDA. LOS CASINIS
INS. DEP. AL AIRE LIBRE EN BARRIADA LAS VIRREINAS 1
INST. DEPORT. AL AIRE LIBRE EN BDA. PARQUE ARROYO DEL CUARTO
INST. DEP. EN BDA. JARDÍN DE MÁLAGA
INST. DEPORT. AL AIRE LIBRE EN PARQUE DE LA HIGUERETA
INST. DEP. EN BDA. EL PALO
INST. DEP. AL AIRE LIBRE JUNTO A LA FAROLA

1 Atr Sig



7

Conclusiones y líneas futuras

7.1 Conclusiones

Una vez terminado el desarrollo de la aplicación, haber probado su correcto funcionamiento y estar ante una de las últimas secciones de la presente memoria, es hora de hacer balance.

En primer lugar, viendo el resultado obtenido, se puede decir con orgullo que se han cumplido los requisitos de la aplicación, tanto de carácter funcional como técnico y arquitectónico.

En segundo lugar, el haber trabajado con un framework como ReactJS en JavaScript y un lenguaje como Python, llevo a la conclusión que, sin lugar a dudas, me llevo un enriquecimiento profesional muy grande. Ambos están muy demandado hoy día en el mercado laboral y el haberlos trabajado y practicado a base de quebrarme la cabeza

para encontrar soluciones a problemas como entender el flujo, manejo de pantalla y estilos de ReactJs o la carencia de una declaración condicional tan usada como es un “switch-case” en la versión 3.8 del lenguaje Python, me ha enriquecido muchísimo a nivel de pensamiento lateral.

En tercer lugar, abandonar los trabajos en grupo y desarrollar por mí mismo todas las etapas del proceso de desarrollo de software, aunque sí es cierto que respaldado por mis tutores, ha supuesto un precedente moral muy importante.

Cuando empiece a buscar trabajo y quieran contratarme, esta experiencia es un extra de ánimos, para pensar que sí que soy apto, he aprendido y soy capaz de aportar mucho a un proyecto software. Desde luego tras esta experiencia, el síndrome de impostor se ha alejado de mi mente.

Por último, yo mismo también me he hecho ver que en esta profesión nunca se termina de aprender y debemos estar en continuo avance y estudio. El poder poner en práctica técnicas de pentesting que he aprendido paralelamente en el máster sobre ciberseguridad ofensiva que estoy cursando me ha hecho ver que el trabajo de un ingeniero de software no se limita a pensar cómo debe ser la aplicación en un entorno utópico. Se debe tener una visión global y tener en cuenta que no siempre se le va a dar un buen uso al software que desarrollemos, por lo que, si valoramos el trabajo que realizamos y a los usuarios finales, estamos en la obligación de protegerlo frente a las amenazas y eso pasa por encargarnos de hacerlo lo más seguro posible.

Sin lugar a dudas, me llevo mucho más que conocimiento de este proyecto, me llevo lecciones que recordaré cada vez que lleve días pensando cómo dar solución a un

problema o escuche que algo no es posible porque aun nadie lo ha hecho. Ahí estará mi mejor esfuerzo para conseguirlo.

7.2 Líneas futuras

El futuro inmediato de Sportify se podría bifurcar por tres líneas diferentes de trabajo. Dado que la aplicación estaría lista para su puesta en producción, estas líneas serían tratadas para el mantenimiento evolutivo:

7.2.1 Nuevas funcionalidades

Se le podrían añadir nuevas funcionalidades para hacer más completa la aplicación, estas podrían ser:

- Incorporar un sistema de mensajería entre la lista de amigos.
- Poder previsualizar la zona en 3D y ver el estado de las máquinas y su organización
- Vender productos deportivos.
- Retransmitir los entrenamientos y eventos en directo.

7.2.2 Expansión

Otra línea de trabajo podría ser llevar Sportify a nivel nacional y hacer uso de las diferentes zonas deportivas públicas que se encuentran en todo el territorio español, no solo conformarse con el público malagueño.

En un futuro en el que la situación sanitaria por el COVID-19 lo permita, incluso podrían incorporarse las pistas privadas (aunque sean cerradas) y un sistema de reservas enlazado y no quedarse meramente en las que son de uso público.

7.2.3 Mejoras conceptuales y de seguridad

Por último se podría enfocar el trabajo en mejorar los tiempos de respuesta de la aplicación y su capacidad de usuarios. Esta última idea está estrechamente enlazada con la expansión de la aplicación ya que necesitaríamos no solo soportar más de 4500 peticiones por hora, sino poder tramitar más peticiones de forma simultánea sin producir una denegación de servicios.

También podría trabajarse sobre la ciberseguridad de la misma, ya que la seguridad que presenta a día de hoy se basa únicamente en las tareas de pentesting de una única persona, la cual también es el desarrollador de la misma y puede haber pasado detalles importantes por alto.

Algunos de los problemas de seguridad que se conocen a día de hoy a los que se podría dar respuesta pero salen del ámbito de este proyecto son:

- Guardar las contraseñas hasheadas en la base de datos en lugar de hacerlo como texto plano.
- Proteger la información del usuario almacenada de forma temporal en el almacenamiento local del navegador con un JSON Web Token (JWT) para impedir su modificación y, por ende, corromper la integridad de la aplicación.
- Limitar el envío masivo de peticiones desde una determinada dirección IP con el fin de evitar que se produzca una denegación de servicios (DOS) debido a que, como ya hemos comentado, Heroku, en su versión gratuita, solo acepta hasta 4500 peticiones por hora.

8

Referencias

- [1] Python, <https://devguide.python.org/>, **Último acceso: 15/01/2022**
- [2] Django, <https://docs.djangoproject.com/en/4.0/>, **Último acceso: 15/01/2022**
- [3] MongoDB, <https://docs.mongodb.com/guides/>, **Último acceso: 15/01/2022**
- [4] Datos Abiertos del Ayuntamiento de Málaga, <https://datosabiertos.malaga.eu/dataset>, **Último acceso: 15/01/2022**
- [5] ReactJS, <https://es.reactjs.org/docs/getting-started.html>, **Último acceso: 15/01/2022**
- [6] React-Bootstrap, <https://react-bootstrap.github.io/getting-started/introduction>, **Último acceso: 15/01/2022**
- [7] Problemas del sedentarismo, https://www.lasexta.com/noticias/ciencia-tecnologia/demencia-diabetes-cancer-oms-alerta-que-1400-millones-personas-practican-deporte-poniendo-riesgo-salud_201809055b900ff30cf26cfa43c8b614.html,

Último acceso: 15/01/2022

[8] Inyección SQL, <https://www.incibe.es/protege-tu-empresa/blog/ataques-inyeccion-sql-amenaza-tu-web>, **Último acceso: 15/01/2022**

[9] Cross-site scripting (XSS), https://developer.mozilla.org/es/docs/Glossary/Cross-site_scripting, **Último acceso: 15/01/2022**

[10] Ataques de fuerza bruta, <https://www.kaspersky.es/resource-center/definitions/brute-force-attack>, **Último acceso: 15/01/2022**

[11] Redux-Saga, <https://redux-saga.js.org/docs/introduction/GettingStarted>, **Último acceso: 15/01/2022**

[12] Visual Studio Code, <https://code.visualstudio.com/learn>, **Último acceso: 15/01/2022**

[13] Cloudinary, <https://cloudinary.com/documentation>, **Último acceso: 15/01/2022**

[14] Heroku, <https://devcenter.heroku.com/categories/working-with-django>, **Último acceso: 15/01/2022**

[15] Balsamiq, <https://balsamiq.com/>, **Último acceso: 15/01/2022**

[16] Postman, <https://learning.postman.com/docs/getting-started/introduction/>, **Último acceso: 15/01/2022**

Apéndice A

Manual de Instalación

Dado que es el caso de una aplicación web desplegada en Internet, no necesita instalación local.

Cualquier modificación que se desee incorporar en el servidor de la aplicación, no requerirá de una instalación, sino de que se desplieguen estos cambios en la plataforma del proveedor de servicios. En este caso, recreando el despliegue en Heroku (la cual ha sido la plataforma para alojar la aplicación), se deberán seguir los siguientes pasos:

1. `heroku login`
2. `heroku git:remote -a sportifyservidor`
3. `git add .`
4. `git commit -m "<texto del commit>"`
5. `git push heroku master`

Cualquier modificación que se desee incorporar en el cliente de la aplicación se deberá desplegar también en Heroku siguiendo estos pasos:

1. `heroku login`

2. heroku git:remote -a sportifycliente
3. git add .
4. git commit -m "<texto del commit>"
5. git push heroku master

De cara a los usuarios finales de la aplicación, no necesitarán más que un dispositivo electrónico con acceso a Internet para poder hacer uso de la misma, sin necesidad de instalación o procesos adicionales.

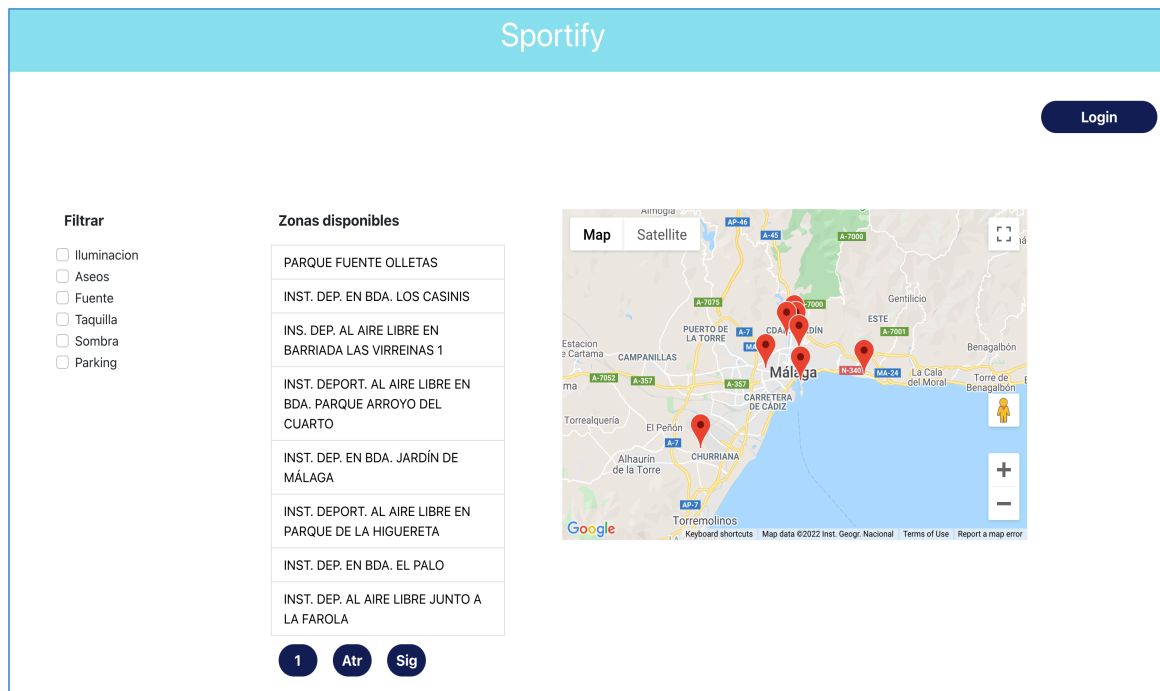
Apéndice B

Manual de usuario

B.1 Usuario invitado

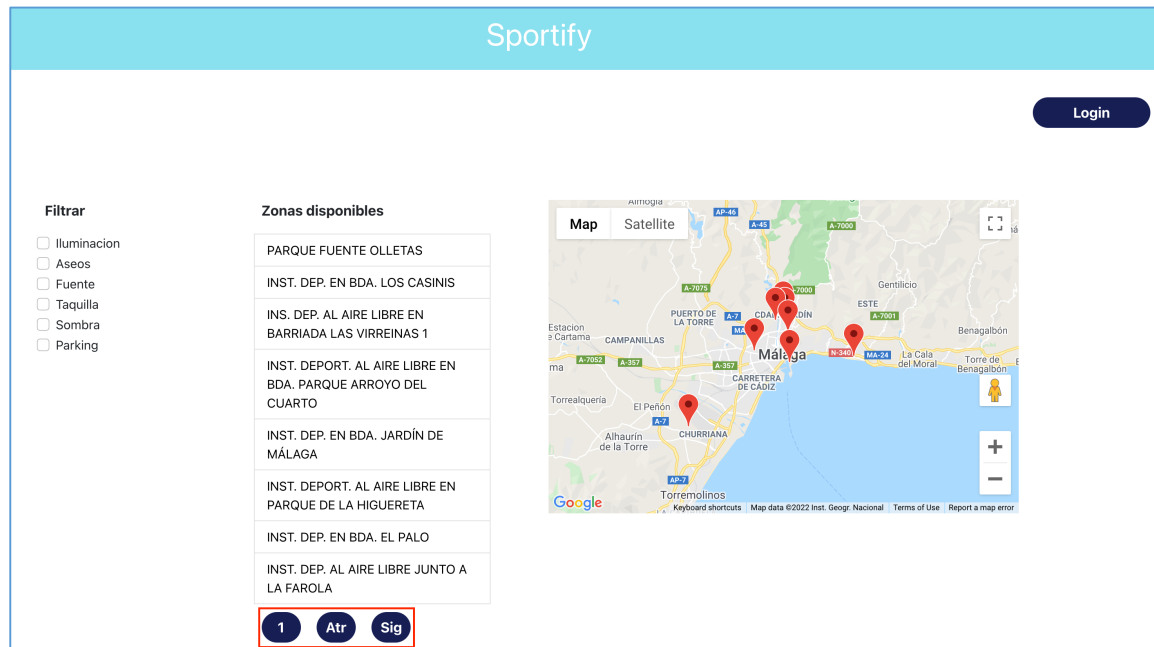
B.1.1 Ver zonas disponibles

No requiere ninguna acción más allá que acceder a la aplicación a través de un navegador.



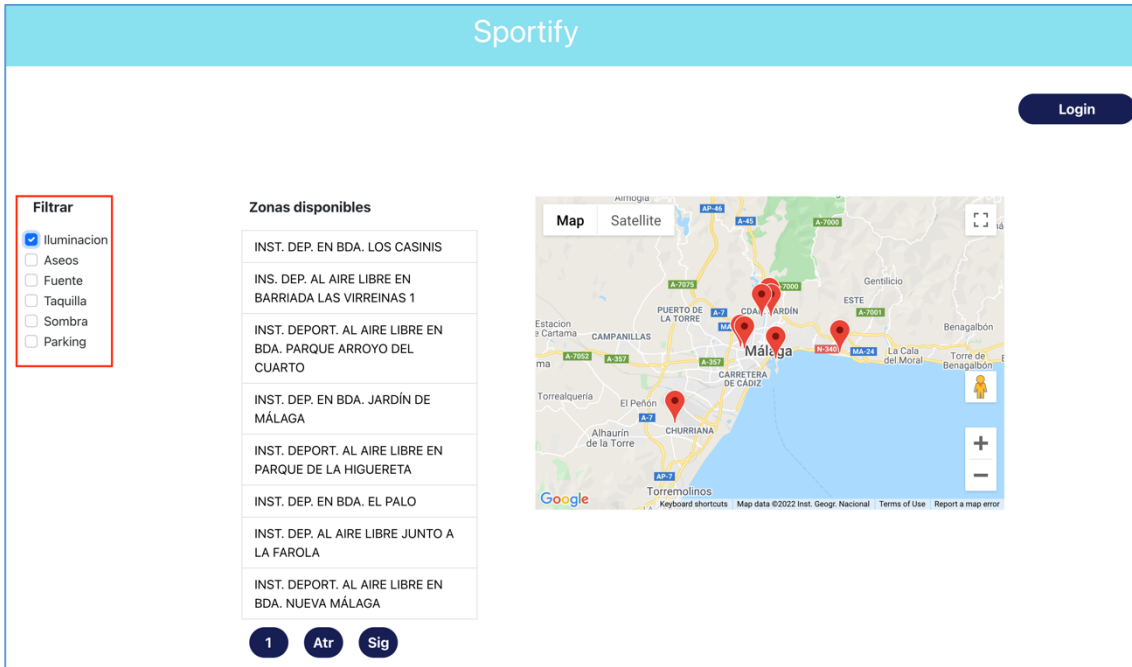
B.1.2 Pagar por la lista de zonas

Pulsando con el ratón sobre los botones, se navega por las zonas. El botón “1” lleva a la primera página de zonas, el botón “Atr” vuelve a la página anterior, si la hay, y el botón “Sig”, a la siguiente página, si la hay.



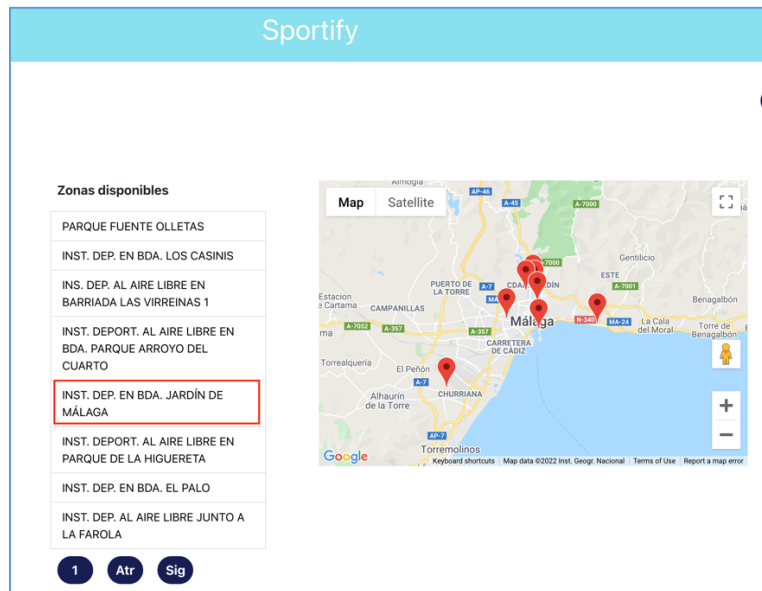
B.1.3 Filtrar las zonas

Se selecciona el filtro o filtros que se deseen aplicar y solo se mostrarán en las páginas las zonas que cumplan con esos requisitos.



B.1.4 Ver información de una zona concreta y filtrarla en el mapa

Pulsando sobre el nombre de la zona de la que se desea ver más información aparecerá una ventana emergente con dicha información.



Pulsando sobre el botón “Cerrar”, se verá dicha zona marcada en el mapa.

INST. DEP. EN BDA. JARDÍN DE MÁLAGA

Dirección: CALLE ALCALDE NICOLÁS MAROTO, 1

Coordenadas: 36.74913092, -4.41977299

Dimensiones: 18x8

Máquinas: disco de muñecas, escalera de dedos, giro de cintura, movilidad de dedos, poleas, serpiente, silla con pedales, volante, barra a media altura, barra giro muñecas, barras paralelas verticales, barras paralelas, base inestable, disco de dedos y muñecas, asiento con apoyo, plataforma de estiramiento.

Agregar valoración

Puntuación ▾

Comentario

Valorar

Valoraciones:
- Puntuación: 1, Comentario: demasiadas maquinas

Cerrar

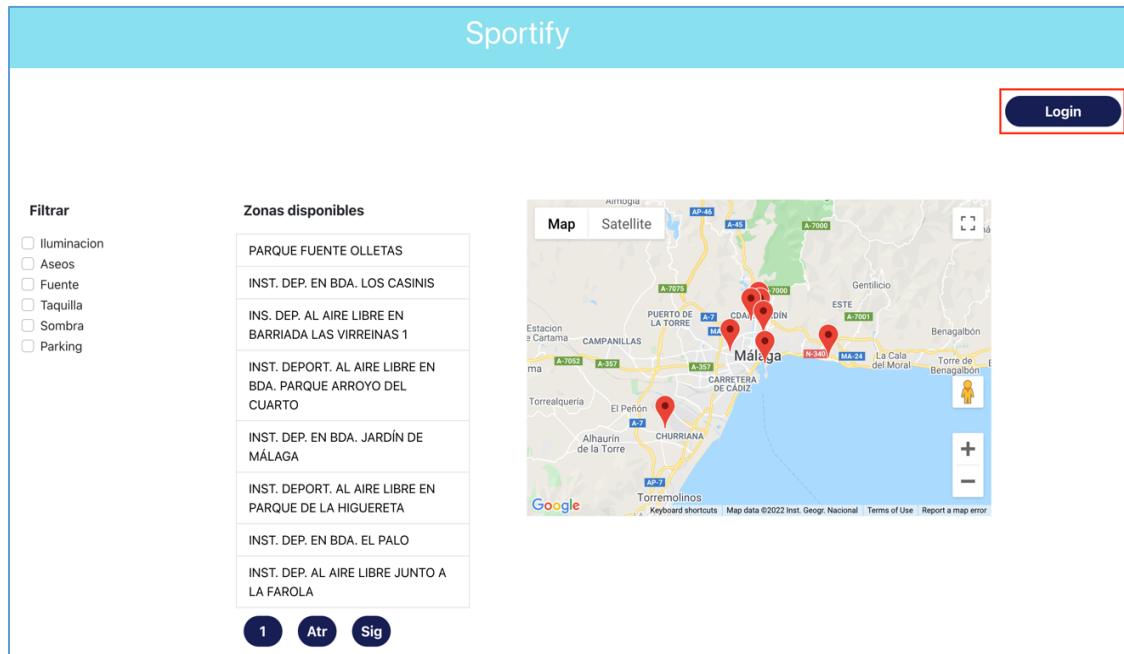
Para volver a ver todas las zonas en el mapa, se debe pulsar sobre la zona seleccionada.

Zonas disponibles

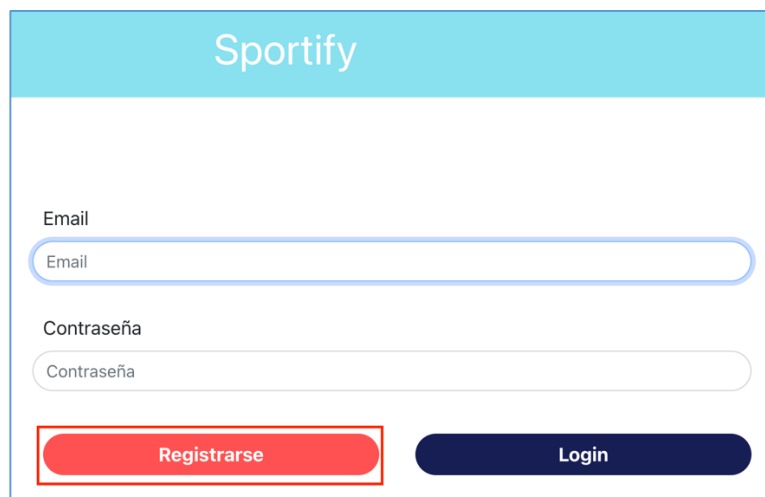
PARQUE FUENTE OLLETAS
INST. DEP. EN BDA. LOS CASINIS
INS. DEP. AL AIRE LIBRE EN BARRIADA LAS VIRREINAS 1
INST. DEPORT. AL AIRE LIBRE EN BDA. PARQUE ARROYO DEL CUARTO
INST. DEP. EN BDA. JARDÍN DE MÁLAGA

B.1.5 Registrarse en la aplicación

En primer lugar hay que pulsar el botón “Login”.



A continuación, hay que pulsar el botón “Registrarse”.



Por último, se deben rellenar todos los campos y pulsar el botón “Registrarse”.

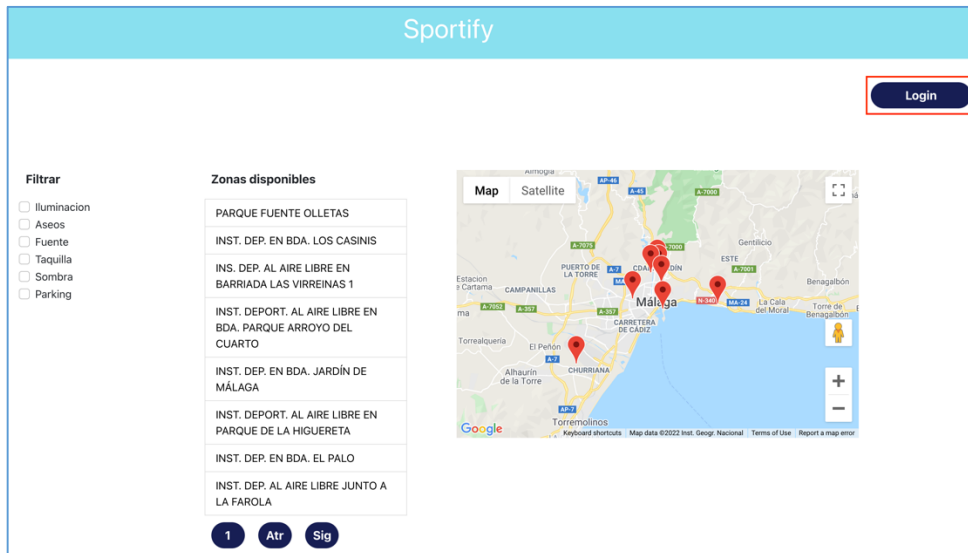
The image shows a registration form for 'Sportify'. At the top, there is a teal header with the word 'Sportify' in white. Below the header, the form consists of several input fields: 'Email' (containing 'sportify@mail.com'), 'Nombre' (containing 'sportify'), 'Apellido' (containing 'sportify'), 'Nick' (containing 'sportify'), 'Edad' (containing '1997-04-29'), a gender selection button (showing 'H'), and 'Contraseña' (containing '.....'). At the bottom of the form, there is a dark blue button with the text 'Registrarse' in white, which is highlighted with a red rectangular border.

Por último se mostrará un mensaje de éxito si todo ha ido bien.

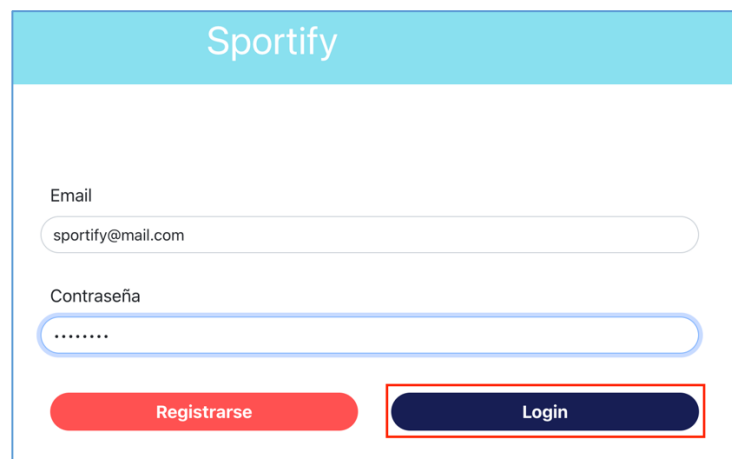
The image shows a success message overlay on a dark grey background. The message is contained in a white box with the text 'El usuario se ha registrado correctamente'. Below the text is a green button with the text 'Continuar'. The background shows the registration form from the previous image, but it is dimmed and partially obscured by the success message.

B.1.6 Logearse en la aplicación

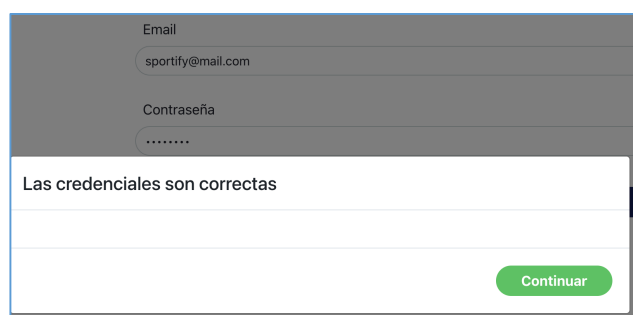
En primer lugar hay que pulsar el botón “Login”.



A continuación, hay que rellenar los campos y pulsar sobre el botón “Login”.



Por último se mostrará un mensaje de éxito si los datos han sido introducidos correctamente.



B.1.7 Ir a la página de inicio

Desde cualquier pantalla, pulsando sobre el logo “Sportify”, se redirigirá al usuario a la pantalla de inicio.



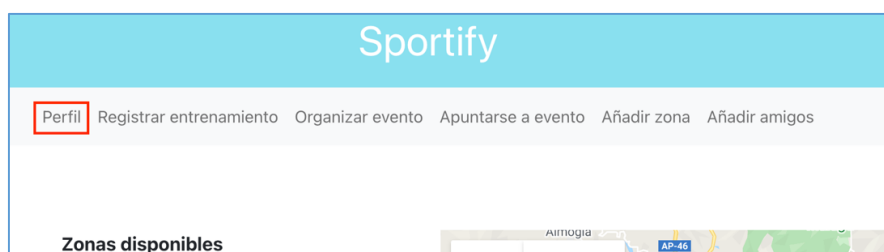
The screenshot shows the Sportify login interface. At the top, there is a light blue header with the 'Sportify' logo in white text, enclosed in a red rectangular box. Below the header, there are three input fields stacked vertically. The first field is labeled 'Email' and contains the placeholder text 'Email'. The second field is labeled 'Nombre' and contains the placeholder text 'Nombre'. The third field is labeled 'Apellido' and contains the placeholder text 'Apellido'.

B.2 Usuario autenticado

Para este tipo de usuario, el funcionamiento para ver y filtrar las zonas deportivas y volver a la pantalla de inicio es igual que el del usuario invitado, por lo que si se desea ver cómo realizar estos procesos, hay que dirigirse a dichos apartados en este manual de usuario.

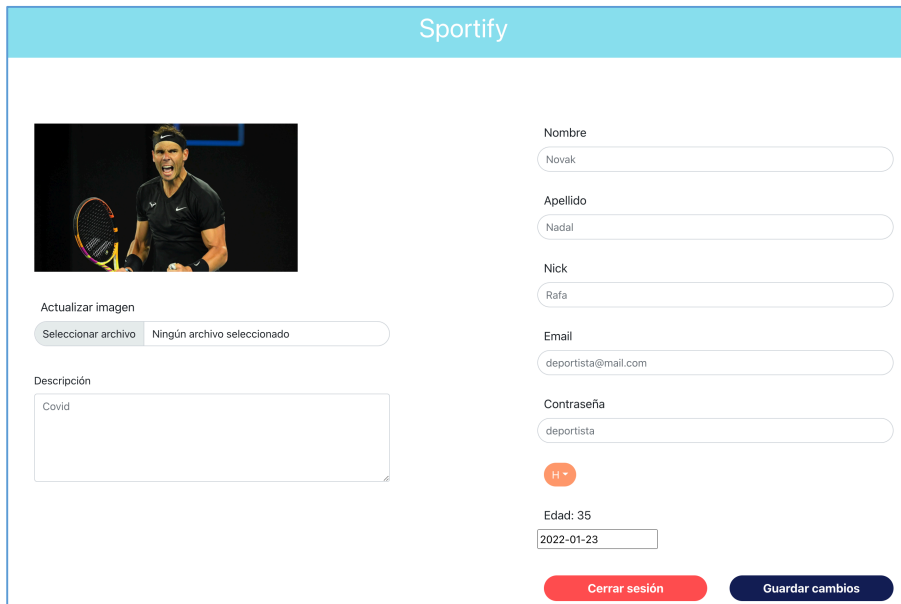
B.2.1 Ver perfil de usuario

Desde la pantalla de inicio, hay que pulsar sobre el enlace “Perfil”



B.2.2 Modificar perfil de usuario

Desde la pantalla de perfil de usuario:



Sportify

Actualizar imagen

Seleccionar archivo Ningún archivo seleccionado

Descripción

Covid

Nombre

Novak

Apellido

Nadal

Nick

Rafa

Email

deportista@mail.com

Contraseña

deportista

H

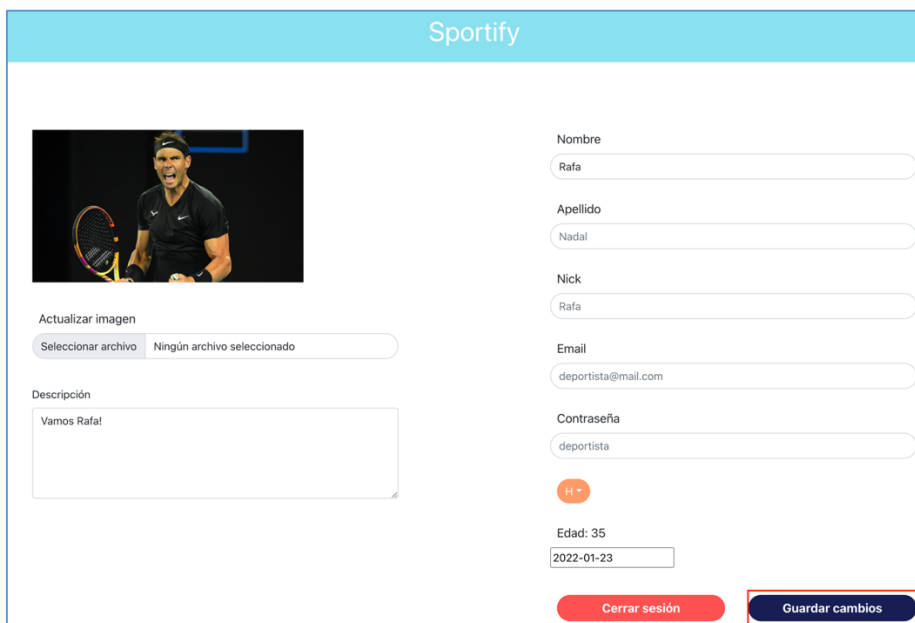
Edad: 35

2022-01-23

Cerrar sesión

Guardar cambios

Se debe rellenar o modificar el campo o campos que se quiera modificar y pulsar sobre el botón “Guardar cambios”.



Sportify

Actualizar imagen

Seleccionar archivo Ningún archivo seleccionado

Descripción

Vamos Rafa!

Nombre

Rafa

Apellido

Nadal

Nick

Rafa

Email

deportista@mail.com

Contraseña

deportista

H

Edad: 35

2022-01-23


Cerrar sesión

Guardar cambios

B.2.3 Añadir imagen de perfil

Desde la pantalla de perfil de usuario, se debe pulsar sobre el botón “Seleccionar archivo”.

Sportify



Actualizar imagen

Seleccionar archivo Ningún archivo seleccionado

Descripción

Vamos Rafa!

Nombre
Rafa

Apellido
Nadal

Nick
Rafa

Email
deportista@mail.com

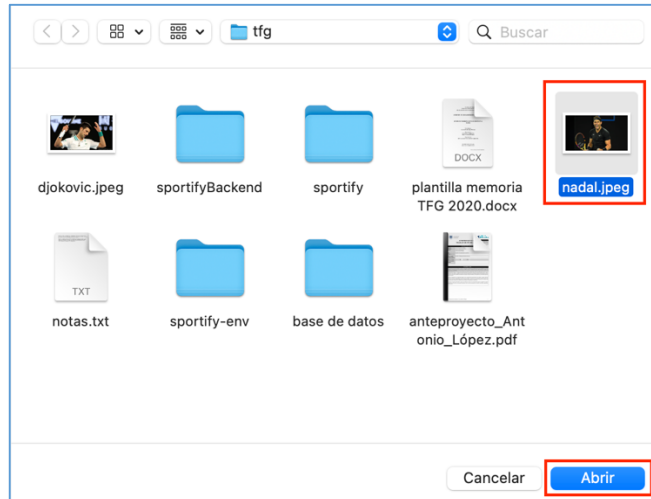
Contraseña
deportista

H ▾

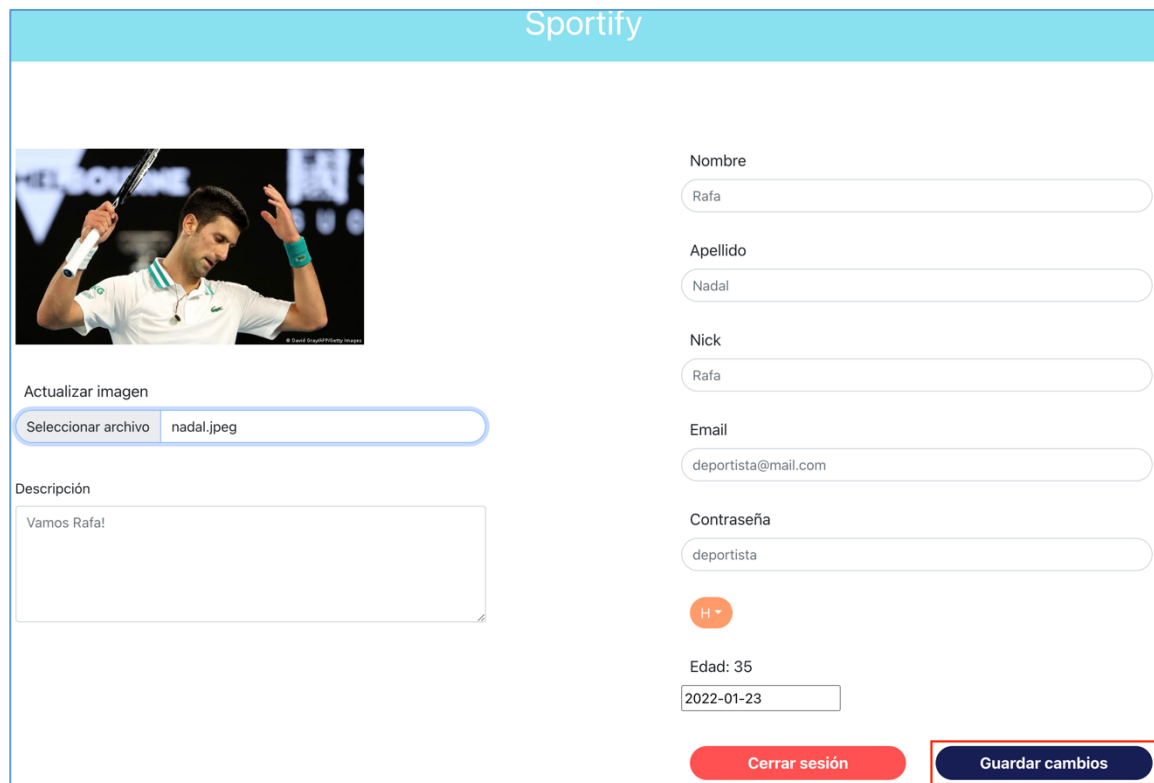
Edad: 35
2022-01-23

Cerrar sesión Guardar cambios

Se abrirá un explorador de archivos, seleccionamos la nueva imagen deseada y pulsamos en “abrir”.



Por último, se pulsa sobre el botón “Guardar cambios”.



Sportify

Actualizar imagen

Seleccionar archivo

Descripción

Vamos Rafa!

Nombre

Apellido

Nick

Email

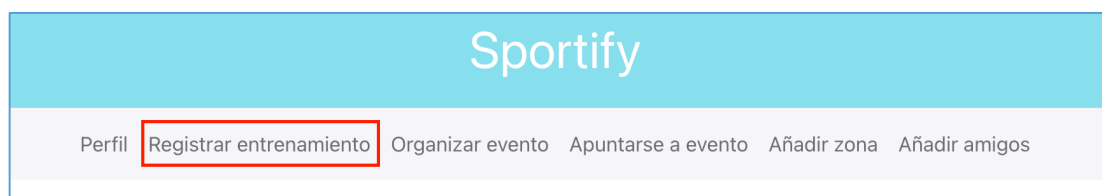
Contraseña

H ▾

Edad: 35

B.2.4 Registrar un entrenamiento en una zona deportiva

Desde la pantalla de inicio, hay que pulsar sobre el enlace “Registrar entrenamiento”.



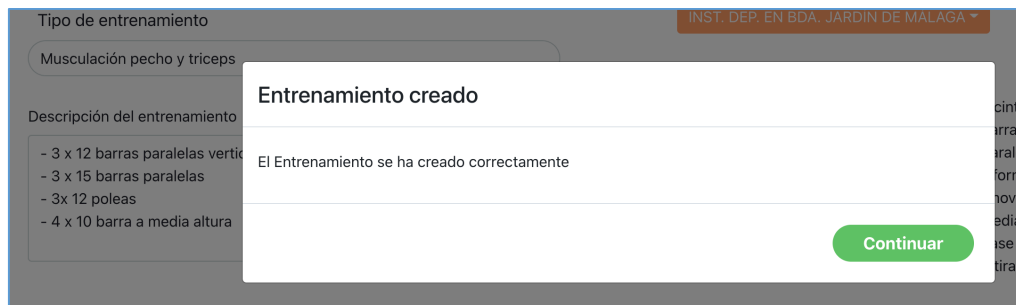
Se debe seleccionar la fecha del entrenamiento, la zona deportiva en la que se va a realizar y rellenar todos los campos para poder registrar el entrenamiento.

Una vez hecho, se pulsa sobre el botón “Registrar entrenamiento”.

The screenshot shows the 'Sportify' registration form. It includes the following fields and information:

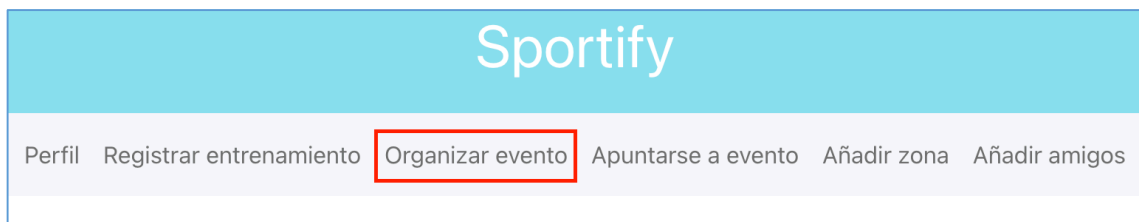
- Duración de entrenamiento:** 75
- Fecha:** 2022-02-02
- Tipo de entrenamiento:** Musculación pecho y tríceps
- Descripción del entrenamiento:**
 - 3 x 12 barras paralelas verticales
 - 3 x 15 barras paralelas
 - 3 x 12 poleas
 - 4 x 10 barra a media altura
- Máquinas disponibles:** disco de muñecas, escalera de dedos, giro de cintura, movilidad de dedos, poleas, serpiente, silla con pedales, volante, barra a media altura, barra giro muñecas, barras paralelas verticales, barras paralelas, base inestable, disco de dedos y muñecas, asiento con apoyo, plataforma de estiramiento, disco de muñecas, escalera de dedos, giro de cintura, movilidad de dedos, poleas, serpiente, silla con pedales, volante, barra a media altura, barra giro muñecas, barras paralelas verticales, barras paralelas, base inestable, disco de dedos y muñecas, asiento con apoyo, plataforma de estiramiento.
- Botón:** Registrar entrenamiento

Si todo ha ido bien, se mostrará un mensaje de confirmación.



B.2.5 Organizar un evento en una zona deportiva

Desde la pantalla de inicio, hay que pulsar sobre el enlace “Organizar evento”.

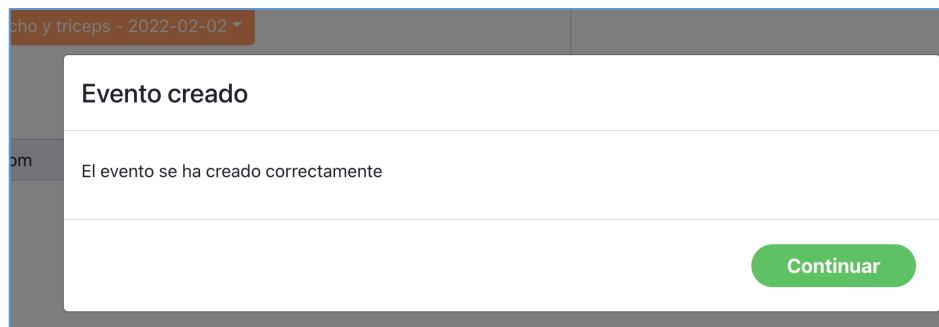


A continuación, se elige la zona y un entrenamiento registrado (si no se posee de ningún entrenamiento, registrar uno siguiendo los pasos del apartado anterior). De forma opcional se podrá añadir una descripción.

Cuando todo esté listo, se debe pulsar sobre el botón “Organizar”.

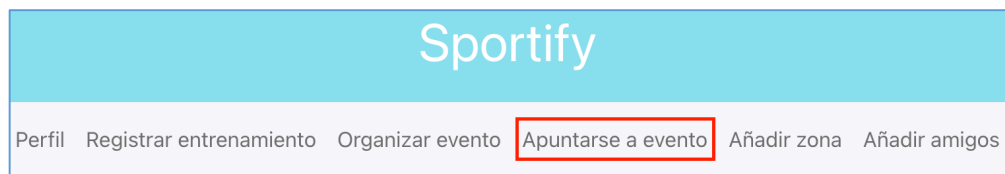
The screenshot shows the 'Sportify' event creation interface. At the top, the 'Sportify' logo is displayed in a light blue header. Below the header, there are several input fields: a location dropdown menu showing 'INST. DEP. EN BDA. JARDÍN DE MÁLAGA', a date dropdown menu showing 'Musculación pecho y triceps - 2022-02-02', and an organizer text field containing 'sportify@mail.com'. To the right of these fields is a larger text area for the event description, which contains the text 'Vamos a quemar calorías'. At the bottom center of the form, there is a dark blue button labeled 'Organizar' with a white border.

Si todo ha ido correctamente, se mostrará un mensaje de confirmación.



B.2.6 Apuntarse a un evento

Desde la pantalla de inicio, hay que pulsar sobre el enlace “Apuntarse a evento”.



Paso siguiente, se elige la zona de la que se desea ver los eventos disponibles y se selecciona sobre uno.

Se mostrará el organizador y la descripción (si tiene) y se podrá pulsar sobre el botón “Apuntarse”.

Si ha tenido éxito, se mostrará un mensaje de confirmación.

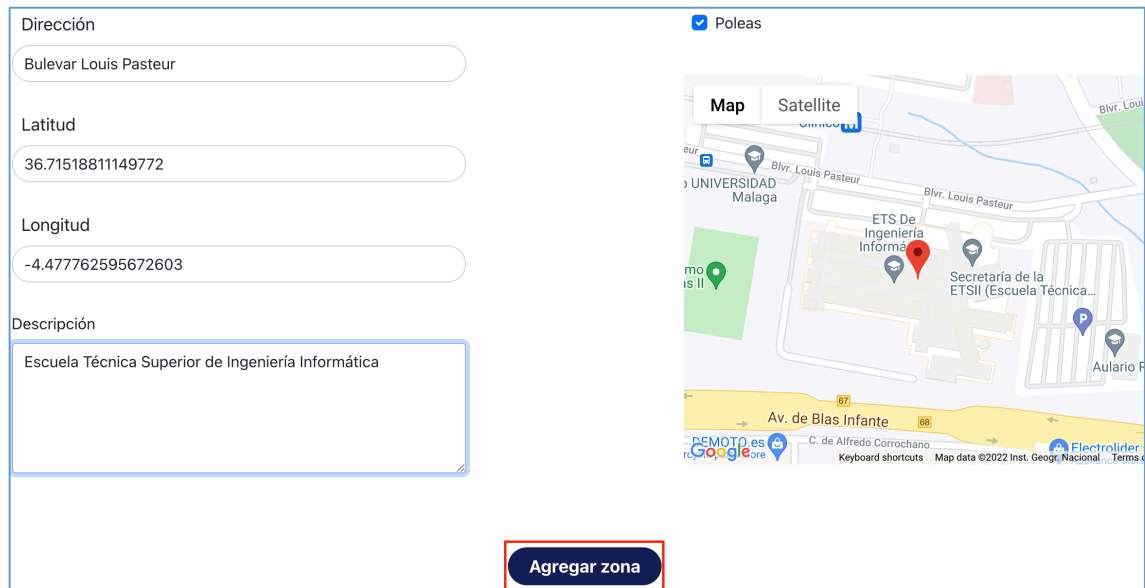
B.2.7 Añadir una nueva zona deportiva

Desde la pantalla de inicio, hay que pulsar sobre el enlace “Añadir zona”.

Se debe seleccionar en el mapa la ubicación de la zona y los campos “Latitud” y “Longitud” se autorrellenarán.

Además, se debe marcar las características con las que cuenta la zona (si cuenta con alguna de ellas), así como con las máquinas que tiene.

Por último, se rellenan los campos (cabe destacar que, si el acceso a una zona es gratuito, se podrá poner “gratis” o 0) y se pulsa sobre el botón “Agregar zona”.

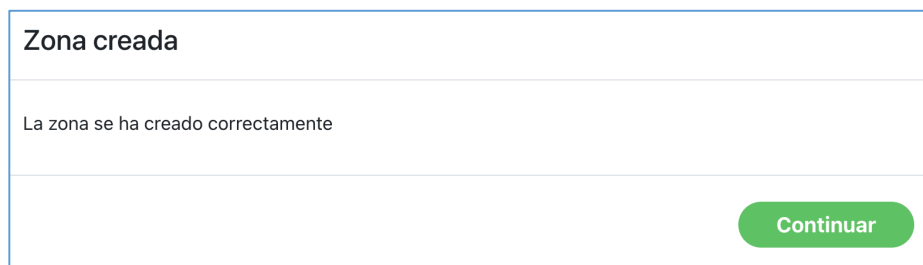


The screenshot shows a form with the following fields:

- Dirección:** Bulevar Louis Pasteur
- Latitud:** 36.71518811149772
- Longitud:** -4.477762595672603
- Descripción:** Escuela Técnica Superior de Ingeniería Informática

There is a checked checkbox for "Poleas" and a map showing the location of the "ETS De Ingeniería Informática" in Malaga. A red box highlights the "Agregar zona" button at the bottom.

Si todo ha ido correctamente, se mostrará el siguiente mensaje de confirmación.



The confirmation message is displayed in a white box with a blue border:

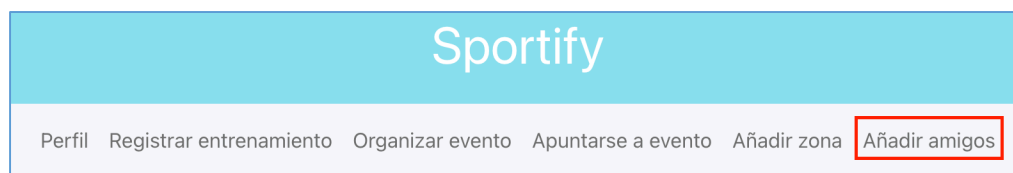
Zona creada

La zona se ha creado correctamente

[Continuar](#)

B.2.8 Añadir amigos

Desde la pantalla de inicio, hay que pulsar sobre el enlace “Añadir amigos”.



The screenshot shows the Sportify app home screen with a navigation bar at the bottom containing the following items:

- Perfil
- Registrar entrenamiento
- Organizar evento
- Apuntarse a evento
- Añadir zona
- Añadir amigos** (highlighted with a red box)

Se mostrará un buscador con una lista de usuarios y conforme se vaya escribiendo se irán filtrando.

Buscar amigos

Introduzca el email

Amigos

Email: sportify@mail.com

Email: deportista@mail.com

Atrás Buscar

Cuando encontremos al usuario, pulsamos sobre su email y se hace click en el botón “Buscar”.

Buscar amigos

depo


Amigos

Email: deportista@mail.com

Atrás Buscar

Se mostrará el perfil del usuario buscado y se debe pulsar sobre el botón “Agregar amigo”.

Sportify



Descripción

Vamos Rafa!

Email

deportista@mail.com

Nick

Rafa

Nombre

Rafa

Edad

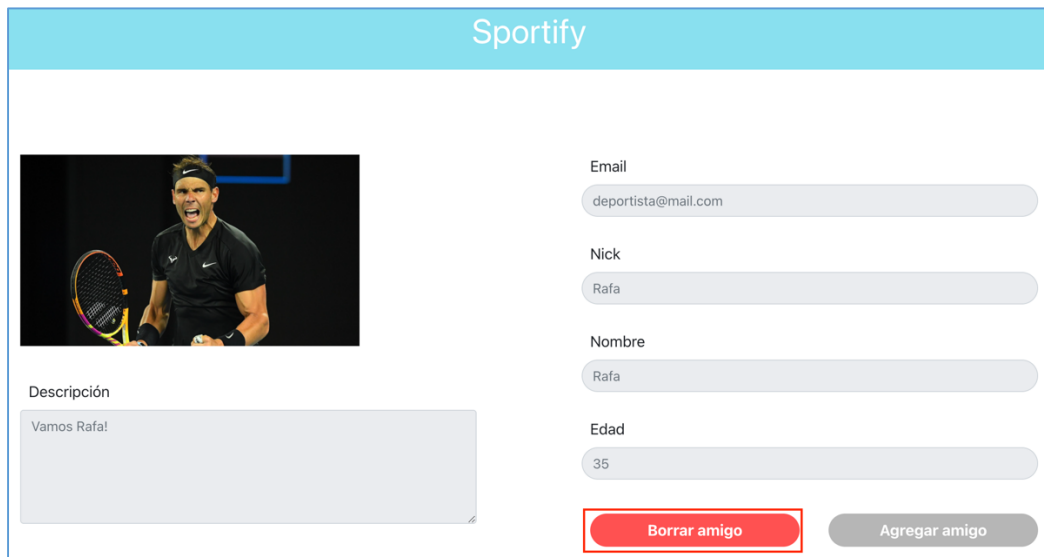
35

Borrar amigo Agregar amigo

B.2.9 Borrar amigo

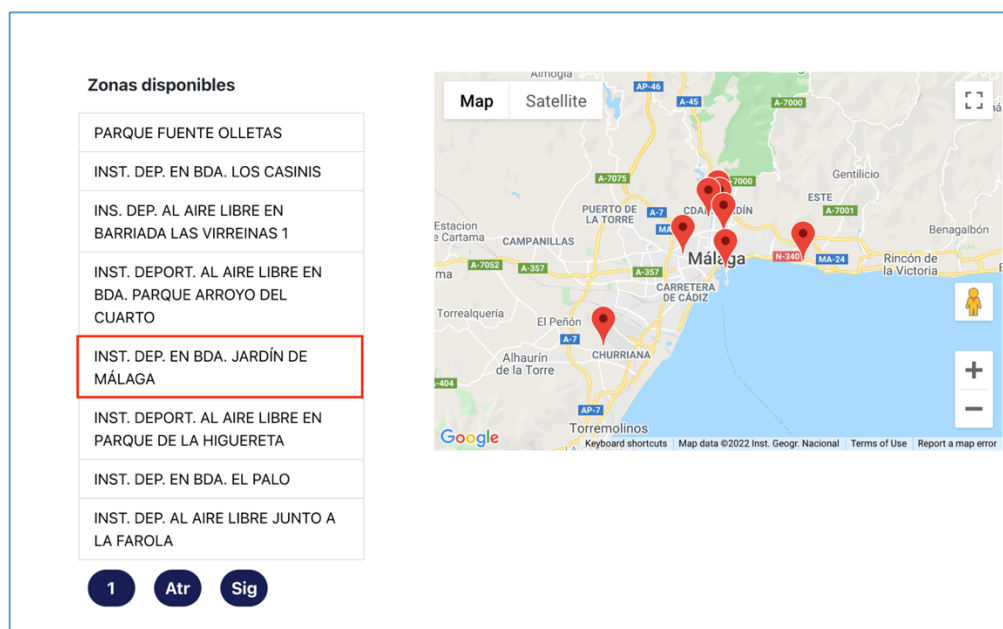
Siguiendo los pasos del apartado anterior, se busca al usuario que se desea eliminar de los amigos.

Una vez en el perfil de dicho usuario, se pulsa sobre el botón “Borrar amigo”.



B.2.10 Añadir reseña a una zona deportiva

Desde la pantalla de inicio, se pulsa sobre la zona deportiva deseada.



En la ventana emergente, se pulsa sobre el desplegable “Puntuación” y se elige la calificación (el campo de comentario es opcional). Una vez terminado, se pulsa sobre el botón “Valorar”.

INST. DEP. EN BDA. JARDÍN DE MÁLAGA

Dirección: CALLE ALCALDE NICOLÁS MAROTO, 1

Coordenadas: 36.74913092, -4.41977299

Dimensiones: 18x8

Máquinas: disco de muñecas, escalera de dedos, giro de cintura, movilidad de dedos, poleas, serpiente, silla con pedales, volante, barra a media altura, barra giro muñecas, barras paralelas verticales, barras paralelas, base inestable, disco de dedos y muñecas, asiento con apoyo, plataforma de estiramiento.

Agregar valoración

5 ▾

Comentario

Gran variedad de máquinas

Valorar

Valoraciones:
- **Puntuación:** 1, **Comentario:** demasiadas maquinas

Cerrar

B.2.11 Cerrar sesión

Desde la pantalla de perfil, se pulsa sobre el botón “Cerrar sesión”.

Seleccionar archivo Ningún archivo seleccionado

Descripción

Vamos Rafa!

Email

deportista@mail.com

Contraseña

deportista

H ▾

Edad: 35

2022-01-23

Cerrar sesión **Guardar cambios**

Se mostrará un mensaje de confirmación para que el usuario sepa que se ha cerrado sesión correctamente.

Se ha cerrado sesión
Salir



UNIVERSIDAD
DE MÁLAGA

| uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA