



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA DEL SOFTWARE

**Aplicación móvil para el seguimiento y evaluación del
progreso en entrenamientos físicos**

**Mobile application for monitoring and evaluating
progress in physical training**

Realizado por
Valentín Nahuel García Rosas

Tutorizado por
Eduardo Guzmán De los Riscos

Departamento
Lenguas y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, FEBRERO DE 2024

Fecha defensa: marzo de 2024

Resumen

La actividad física es muy importante para la salud y bienestar de las personas, sin embargo, muchas personas tienen dificultades para mantenerse en forma y activas de forma regular. En España, durante el 2022, según la encuesta de hábitos publicada en España y que se resume en [1], solo un 23,8% de las personas practicaron algún deporte todos los días, y menos de un 53% practicó deporte al menos una vez a la semana.

Ante este panorama, surge la necesidad de ofrecer soluciones accesibles y efectivas que faciliten la incorporación de hábitos saludables. En este contexto, una aplicación diseñada para gestionar entrenamientos y seguimiento de progreso se convierte en una herramienta idónea. Esta aplicación no solo está dirigida a aquellos que buscan iniciar una rutina de ejercicios, sino que también ofrece recursos útiles para aquellas personas que ya son activas físicamente.

De esta forma, este proyecto intenta proporcionar una plataforma intuitiva y eficaz, permite a los usuarios planificar y registrar sus sesiones de entrenamiento, asociar ejercicios a sus rutinas, y gestionar días de descanso de manera organizada. Asimismo, los usuarios pueden modificar fácilmente el número de repeticiones, series y peso de los ejercicios realizados para adaptarse a sus necesidades y objetivos de entrenamiento.

La plataforma también permite visualizar gráficas que representan la progresión, con el propósito de que los usuarios sean conscientes de su avance en cada ejercicio y sesión de entrenamiento de forma clara y visual.

Por último, se proporciona la opción de detectar automáticamente los ejercicios a través de los sensores y la cámara del dispositivo, lo que simplifica el conteo preciso de los ejercicios realizados. Además, ofrece la posibilidad de cambiar el idioma tanto en dispositivos iOS como Android para mejorar la experiencia de usuario.

Palabras clave: seguimiento de la actividad física, React Native, Red Neuronal Convolutacional, MySQL, Python.

Abstract

Physical activity is very important for people's health and well-being, however, many people find it difficult to keep fit and active on a regular basis. In Spain, during 2022, according to the habits survey published in Spain and summarized for us [1], only 23.8 % of people practiced some sport every day, and less than 53 % practiced sport at least once a week.

Given this scenario, the need arises to offer accessible and effective solutions that facilitate the incorporation of healthy habits. In this context, an application designed to manage training and progress tracking becomes an ideal tool. This application is not only aimed at those looking to start an exercise routine, but also offers useful resources for those who are already active.

This project provides an intuitive and effective platform, allowing users to plan and record their training sessions, associate exercises to their routines, and manage rest days in an organised way. In addition, users can easily modify the number of repetitions, sets and weight of the exercises performed to suit their training needs and goals.

It also offers the opportunity to display graphs representing progression, with the purpose of making users aware of their progress in each exercise and training session in a clear and visual way.

Finally, it provides the option of automatically detecting exercises through the device's sensors and camera, which simplifies the accurate counting of exercises performed. In addition, it offers the ability to change the language on both iOS and Android devices to enhance the user experience.

Keywords: physical activity tracking, React Native, Convolutional Neural Network, MySQL, Python

Índice

Resumen	1
Abstract	1
Índice	1
Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Estructura de la memoria	2
Tecnologías y librerías	5
2.1 Tecnologías utilizadas	5
2.1.1 React Native.....	5
2.1.2 Expo	5
2.1.3 Visual Studio Code.....	6
2.1.4 Justinmind	6
2.1.5 Git	6
2.1.6 Python.....	6
2.1.7 MySQL.....	6
2.2 Librerías utilizadas	7
2.2.1 Flask [2].....	7
2.2.2 Axios [3]	7
2.2.3 Expo Camera [4]	7
2.2.4. Expo Sensors [5].....	7
2.2.5 I18Next [6]	7
2.2.6 React-Native-Gifted-Charts [7]	7
2.2.7 NumPy [8]	7
2.2.8 Torch [9].....	7
2.2.9 SQLAlchemy [10]	8
Análisis previo	9
3.1 Requisitos funcionales	9
3.2 Requisitos no funcionales	10
Diseño	13
4.1. Base de datos	13
4.2. Arquitectura general	14
4.3. Servidor - Arquitectura REST	15
4.3.1 Arquitectura Modelo-Vista-Controlador	16
4.3.2 Uso de <i>Blueprints</i>	17
4.4 Aplicación móvil - Estructura	18
Desarrollo	19
5.1 Metodología	19

5.1.1 Descripción del proceso iterativo	20
5.1.2 Ventajas de la metodología iterativa incremental	20
5.2 Primera iteración	21
5.2.1 Organización	21
5.2.2 Servidor	21
5.2.3 Aplicación móvil.....	21
5.2.4 Casos de uso implementados	22
5.2.5 Evaluación del resultado.....	27
5.3 Segunda iteración	27
5.3.1 Organización	27
5.3.2 Servidor.....	27
5.3.3 Aplicación móvil.....	28
5.3.4 Casos de uso implementados	28
5.3.5 Evaluación del resultado.....	34
5.4 Tercera iteración.....	35
5.4.1 Organización	35
5.4.2 Servidor.....	35
5.4.3 Aplicación móvil.....	35
5.3.4 Casos de uso implementados	36
5.4.5 Evaluación del resultado.....	39
5.5 Cuarta iteración	39
5.5.1 Organización	39
5.5.2 Servidor.....	39
5.5.3 Aplicación móvil.....	39
5.5.4 Casos de uso implementados	40
5.5.5 Evaluación del resultado.....	44
Nuevas funcionalidades basadas en Ciencia de Datos	45
6.1 Introducción	45
6.2 Elección y obtención de los datos	46
6.2.1 Acelerómetro y giroscopio	46
6.2.1 Cámara.....	47
6.3 Aprendizaje Profundo	48
6.4 Representación de los datos	49
6.4.1 Representación de los sensores	49
6.4.2 Representación de la cámara	50
6.5 Toma de datos.....	53
6.6 Entrenamiento y reconocimiento	55
6.7 Integración en la aplicación.....	56
6.7.1 Cambios en el servidor	57
6.7.2 Cambios en la aplicación móvil.....	58
6.8 Pruebas en la aplicación.....	59
6.8.1 Tiempo de detección	60
6.8.2 Reconocimiento desde diversos ángulos	60
6.8.3 Reconocimiento en cuerpos diferentes.....	61
Conclusiones.....	63
7.1 Posibles líneas futuras	64
Manual de instalación.....	67
Manual de usuario	71
Referencias.....	89

Índice de figuras	93
-------------------------	----

1

Introducción

1.1 Motivación

La motivación de este proyecto reside en la creación de una aplicación móvil con un enfoque utilitario, destinada a aquellas personas que desean mantener un registro detallado de su progreso en el ámbito del ejercicio físico.

No solo se busca permitir a los usuarios ingresar información sobre los días en que llevaron a cabo sus entrenamientos, sino también proporcionar la capacidad de registrar y ofrecer seguimiento a los ejercicios realizados, incluyendo repeticiones, series y peso empleado.

Con el propósito de hacer que la aplicación sea altamente efectiva y brinde un apoyo significativo al usuario en su camino hacia la constancia y mejora física, se aportan gráficos visuales que ofrecen una representación clara y rápida del progreso alcanzado.

Finalmente, se pretende enriquecer la experiencia del usuario mediante la aplicación de técnicas de ciencia de datos relacionadas con los ejercicios realizados. Este enfoque permite identificar automáticamente los ejercicios sin requerir una entrada manual por parte del usuario, lo cual contribuye a una experiencia más fluida y eficiente.

1.2 Objetivos

El objetivo principal es el diseño, desarrollo e implementación de una aplicación móvil, útil tanto para usuarios de iOS como de Android y orientada a la gestión y seguimiento personalizado de entrenamientos y ejercicios. Se ha realizado un desarrollo *full-stack* donde el *front-end* ha sido implementado empleando React Native y el *back-end* en Python utilizando el framework Flask.

La aplicación permite, de manera sencilla, la planificación de rutinas de ejercicio. Busca proporcionar a los usuarios una herramienta integral que fomente la adherencia a hábitos saludables y promueva un estilo de vida sano.

Algunas de las características principales que ofrece esta aplicación son:

- **Gestión de entrenamientos y días de descanso:** Incluye un sistema intuitivo que permite a los usuarios registrar sus entrenamientos o días de descanso de manera flexible y personalizada.
- **Gestión de los ejercicios realizados:** Los usuarios pueden agregar ejercicios a sus rutinas de entrenamiento y ajustar las repeticiones, series y peso correspondientes según sea necesario.
- **Seguimiento de progreso a través de gráficos:** Se integran herramientas de visualización de datos que generan gráficos informativos que permiten a los usuarios evaluar su progreso en distintos aspectos, como el incremento de carga en los ejercicios o la frecuencia de entrenamientos a lo largo del tiempo.
- **Reconocimiento mediante imágenes y sensores de los ejercicios realizados:** Esta última característica involucra el estudio y la aplicación de técnicas de ciencia de datos, más concretamente, de aprendizaje profundo. El objetivo es, haciendo uso de los elementos de los que dispone el usuario en su dispositivo móvil (cámara del dispositivo y sensores), reconocer el ejercicio que está realizando, de forma automática.

1.3 Estructura de la memoria

En la memoria primero se introduce al lector el entorno y las tecnologías utilizadas, y después se avanza progresivamente en función de las fases que se han llevado a cabo en el desarrollo de la aplicación para finalizar con una explicación más detallada del proceso de la ciencia de datos.

1. Introducción

Se pone en contexto al usuario sobre lo que va a tratar la memoria: La motivación, los objetivos y la estructura de la misma.

2. Tecnologías y librerías

Se presentan las tecnologías y librerías que han tenido mayor relevancia durante el proceso de desarrollo y de ciencia de datos del proyecto.

3. Análisis previo

Se exponen los requisitos, funcionales y no funcionales que se han recopilado y que se deben implementar para el pleno funcionamiento de la aplicación, incluyendo una descripción de estos.

4. Diseño

En la sección de diseño, se detallan las decisiones tomadas con respecto al diseño de la base de datos, el servidor y la aplicación móvil, así como su estructuración correspondiente.

5. Desarrollo

En el apartado de desarrollo se desglosa el trabajo realizado por iteraciones, y se indican para cada una de ellas los cambios realizados en el servidor y en la aplicación móvil, además de los casos de uso y las evaluaciones finales de cada una de ellas.

6. Ciencia de datos

Llegados a este apartado ya se ha terminado la fase de desarrollo y se estudia la ciencia de datos relacionada al reconocimiento de ejercicios a través de sensores y otras fuentes de datos. Además, se realizan distintas pruebas en los modelos generados para comprobar la eficacia de estos.

7. Conclusiones

Se detallan las conclusiones que han surgido durante y una vez finalizado el proyecto y, además, se habla de posibles mejoras que se pueden implementar o añadir al producto final.

8. Apéndice A: Manual de instalación

En este primer apéndice, se especifican los requisitos previos necesarios para el despliegue adecuado de la aplicación, así como los pasos detallados para desplegar tanto la aplicación móvil como el servidor, lo que permite el uso pleno de la aplicación.

9. Apéndice B: Manual de usuario

En este apéndice, se brinda una guía sobre cómo utilizar la aplicación para usuarios sin experiencia previa, con el propósito de optimizar y mejorar su experiencia. Asimismo, sirve como referencia para consultar en caso de duda, sobre el funcionamiento de la aplicación.

2

Tecnologías y librerías

2.1 Tecnologías utilizadas

En esta sección se presentan las tecnologías y entornos que han tenido mayor relevancia en el desarrollo del proyecto.

2.1.1 React Native

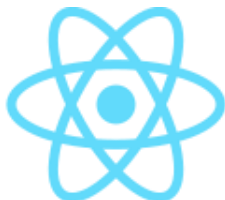


Figura 1. Logo React Native [17]

React Native es un marco de desarrollo de código abierto que permite a los desarrolladores construir aplicaciones móviles utilizando *JavaScript* y *React*.

La principal ventaja de *React Native* es que permite a los desarrolladores crear aplicaciones móviles para *iOS* y *Android* usando un único código base. *React Native* utiliza componentes que se traducen directamente a elementos de la Interfaz de usuario nativa.

2.1.2 Expo

Expo es un conjunto de herramientas y servicios que simplifican el desarrollo de aplicaciones móviles con *React Native*. *Expo Go* permite a los desarrolladores previsualizar y probar sus aplicaciones *React Native* en dispositivos móviles de forma rápida y sencilla.



Figura 2. Logo de Expo [18]

2.1.3 Visual Studio Code



Visual Studio Code es un entorno de desarrollo Integrado (*IDE*) liviano y de código abierto desarrollado por *Microsoft*.

Incluye soporte para diversas funciones, como refactorización, resaltado de sintaxis o finalización inteligente de código.

Figura 3. Logo de *Visual Studio Code* [19]

2.1.4 Justinmind

Justinmind es una herramienta de prototipado y diseño de Interfaces de usuario que ayuda a los diseñadores y desarrolladores a crear prototipos interactivos para aplicaciones web y móviles.



Figura 4. Logo de *Justinmind* [20]

2.1.5 Git



Git es un sistema de control de versiones sumamente utilizado en el desarrollo de software, diseñado con un enfoque en la confiabilidad, compatibilidad y la eficiencia.

Figura 5. Logo de *Git* [21]

Ofrece una sólida infraestructura para rastrear y gestionar cambios en el código fuente a lo largo del tiempo.

2.1.6 Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Es conocido por su sintaxis clara y legible, lo que facilita la estructura del código y la comprensión.



Figura 6. Logo de *Python* [22]

2.1.7 MySQL



Figura 7. Logo de *MySQL* [23]

MySQL es un sistema de gestión de base de datos relacional de código abierto ampliamente utilizado. Realiza un manejo eficiente de grandes cantidades de datos, evita la redundancia y facilita la escalabilidad.

2.2 Librerías utilizadas

En esta sección se presentan las librerías más destacadas de *Python* y de *React Native* que han sido empleadas en el desarrollo del proyecto.

2.2.1 *Flask* [2]

Flask es un *framework* (marco de trabajo) web escrito en *Python* que permite crear aplicaciones web de forma sencilla y rápida. *Flask* proporciona una serie de funciones y clases que facilitan el desarrollo de aplicaciones web basadas en el patrón de diseño arquitectónico *MVC* (Modelo-vista-controlador).

2.2.2 *Axios* [3]

Axios es una biblioteca de *JavaScript* que facilita la realización de solicitudes *HTTP* desde navegadores o entornos *Node.js*. Su objetivo principal es simplificar y mejorar la gestión de peticiones y respuestas *HTTP* en aplicaciones web y en el lado del servidor.

2.2.3 *Expo Camera* [4]

Expo Camera es un módulo proporcionado por *Expo* que facilita la integración y el uso de la cámara en aplicaciones móviles desarrolladas con *Expo*.

2.2.4. *Expo Sensors* [5]

Expo Sensors es un módulo proporcionado por *Expo*, una plataforma de desarrollo para aplicaciones móviles con *React Native*. Este módulo facilita el acceso a una variedad de sensores del dispositivo desde aplicaciones *React Native*.

2.2.5 *I18Next* [6]

I18next es una biblioteca de *JavaScript* muy utilizada para la internacionalización de aplicaciones. Aborda el proceso de adaptar una aplicación para que pueda ser utilizada por usuarios de diferentes regiones y culturas.

2.2.6 *React-Native-Gifted-Charts* [7]

React-Native-Gifted-Charts es una biblioteca de gráficos para *React Native* que proporciona una amplia gama de tipos de gráficos, incluyendo diagramas de barras, de líneas o de áreas.

2.2.7 *NumPy* [8]

NumPy es una de las herramientas más importantes en el ecosistema de *Python* para computación científica y numérica. Proporciona soporte para matrices y operaciones matemáticas de manera eficiente y rápida.

2.2.8 *Torch* [9]

La biblioteca *PyTorch* es una biblioteca de aprendizaje profundo de código abierto desarrollada principalmente por *Facebook's AI Research Lab* (FAIR). *PyTorch* es conocida por su flexibilidad y facilidad de uso, y se utiliza mucho en la investigación y la implementación de modelos de aprendizaje profundo.

2.2.9 SQLAlchemy [10]

SQLAlchemy es una biblioteca de *Python* que facilita la interacción con bases de datos relacionales de una manera más orientada a objetos. Destaca por ser un *ORM* (Mapeo objeto-relacional) que da soporte para múltiples bases de datos.

3

Análisis previo

3.1 Roles de usuario

La aplicación se centra exclusivamente en un único rol de usuario, el cual engloba a cualquier individuo que accede y utiliza la plataforma. Este rol permite todas las acciones sin importar el nivel de experiencia, propósito de uso o necesidades específicas del usuario.

Dentro de este rol, los usuarios tienen acceso completo a todas las características y funcionalidades de la aplicación, sin privilegios o restricciones adicionales.

3.2 Requisitos funcionales.

En este apartado se detallan cada uno de los requisitos funcionales que se han propuesto para poder abarcar toda la funcionalidad esperada. Los requisitos, con sus identificadores y descripciones, pueden observarse en la Tabla 1 y Tabla 2.

Identificador del requisito	Nombre	Descripción
RF1	Registro del usuario	El usuario podrá registrarse con un usuario y contraseña, además podrá añadir otros datos como el número de teléfono o la fecha de nacimiento
RF2	Identificación del usuario	El usuario podrá acceder a la aplicación con un usuario y contraseña válidos
RF3	Cerrar sesión	Los usuarios identificados podrán cerrar la sesión para poder registrarse/ identificarse con otra cuenta
RF4	Cambiar datos del usuario	El usuario identificado podrá cambiar datos varios de su cuenta como la dirección, el número de teléfono o la fecha de nacimiento

Tabla 1. Requisitos Funcionales (primera parte)

RF5	Añadir un entrenamiento	El usuario podrá añadir un nuevo entrenamiento para una fecha determinada
RF6	Eliminar un entrenamiento	El usuario podrá eliminar un entrenamiento registrado
RF7	Modificar un entrenamiento	El usuario podrá modificar un entrenamiento registrado
RF8	Añadir un día de descanso	El usuario podrá añadir un día de descanso para una fecha determinada
RF9	Modificar un día de descanso	El usuario podrá modificar un día de descanso registrado
RF10	Eliminar un día de descanso	El usuario podrá eliminar un día de descanso registrado
RF11	Añadir un ejercicio a un entrenamiento	El usuario podrá añadir un ejercicio a un entrenamiento, indicando las series, repeticiones y peso
RF12	Eliminar un ejercicio de un entrenamiento	El usuario podrá eliminar un ejercicio registrado en un entrenamiento
RF13	Modificar un ejercicio de un entrenamiento	El usuario podrá modificar las series, repeticiones y peso de un ejercicio registrado en un entrenamiento
RF14	Ver entrenamientos anteriores	El usuario podrá ver los entrenamientos realizados
RF15	Ver gráfica de progresión	El usuario podrá ver la progresión de sus entrenamientos
RF16	Cambiar el objeto de la gráfica	El usuario podrá cambiar el aspecto que se desea comparar en la gráfica, por número de entrenamientos o por ejercicios. Puede seleccionar qué ejercicio y aspectos comprar
RF17	Cambiar el número de entrenamientos de la gráfica	El usuario podrá cambiar la gráfica para que se muestre con la información recogida en los entrenamientos de los últimos 6 días o todos los días

Tabla 2. Requisitos Funcionales (segunda parte)

3.3 Requisitos no funcionales.

En este apartado se detallan casa uno de los requisitos no funcionales que se han identificado. Los requisitos, con sus tipología, identificadores y descripciones pueden observarse en la Tabla 3.

Identificador del requisito	Tipo de requisito no funcional	Nombre	Descripción
RNF1	Requisito de seguridad	Datos tratados de forma segura	Los datos en la aplicación serán usados y almacenados de forma segura
RNF2	Requisito de producto	Desarrollo en React Native	La aplicación se desarrollará en React Native y se podrá utilizar en Android y iOS
RNF3	Requisito de arquitectura	Servidor	La aplicación móvil realizará las peticiones a un servidor externo que interactúa con la base de datos
RNF4	Requisito de usabilidad	Información para los errores	Cuando haya un error en la recogida de datos de la base de datos se advertirá al usuario
RNF5	Requisito de internacionalización	Múltiples idiomas	La aplicación permite cambiar el idioma, las opciones disponibles son inglés y español

Tabla 3. Requisitos No Funcionales

4

Diseño

En este apartado se va a documentar las decisiones tomadas a la hora de diseñar la aplicación, tanto de la base de datos, como del servidor y de la aplicación.

4.1. Base de datos

La primera decisión es la elección del método de almacenamiento de datos. En este caso, se ha decidido usar MySQL para la base de datos. MySQL es un sistema de bases de datos relacional de código abierto que permite almacenar, organizar y gestionar grandes volúmenes de datos de manera eficiente. Se ha elegido por los siguientes motivos:

- **Estructura de datos relacionada:** Los datos de la aplicación tienen una estructura claramente definida y relacionada.
- **Consultas complejas con *Joins* (*Uniones*):** MySQL es especialmente eficiente para realizar consultas que involucran *Joins*, gracias a su relación entre las tablas.

Una vez aclarados los requisitos y el funcionamiento de la aplicación, se ha especificado el diseño de la base de datos (Figura 8).

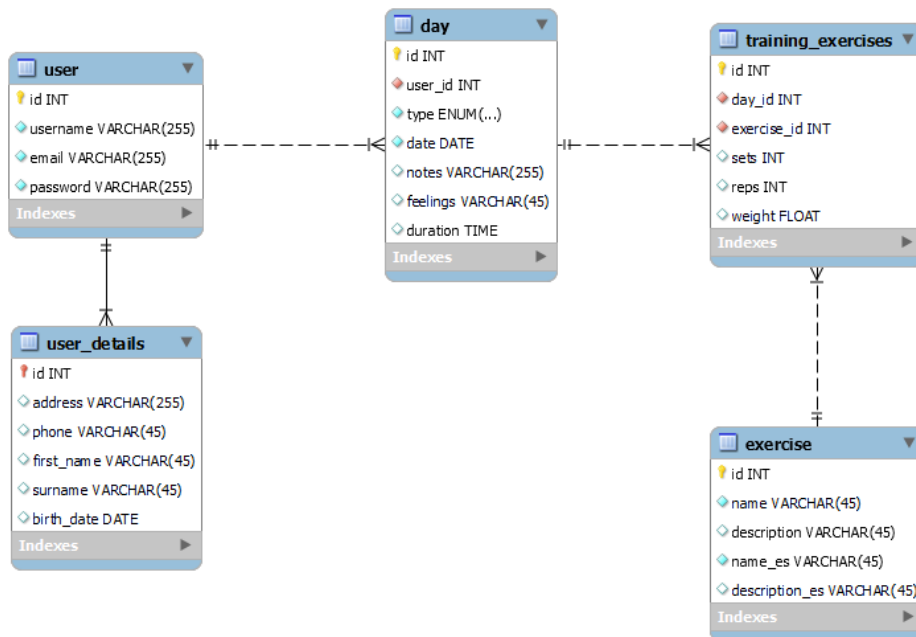


Figura 8. Diseño de la base de datos

Para los usuarios se siguen los principios de normalización de bases de datos. Por un lado, se cuenta con la tabla «user» para almacenar los datos del usuario de la identificación del usuario. Por otro lado, existe la tabla «user_details» que guarda la información extra de un usuario que no es necesaria para su identificación.

Además, cada usuario mantiene una relación de uno a muchos con la tabla «day», la cual representa un día entrenamiento o de descanso. Esta diferencia se representa con la columna «type».

La tabla «day», además de guardar información importante como la fecha o la duración, también mantiene una relación de uno a muchos con la tabla «training_exercises». Esta última representa cada uno de los ejercicios realizados en ese entrenamiento junto con sus respectivos sets, repeticiones y peso.

4.2. Arquitectura general

En la aplicación se seguirá una arquitectura de aplicación de dos niveles, también conocida como arquitectura cliente-servidor. Es uno de los estilos arquitectónicos más utilizados y conocidos. Las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. En la Figura 9 se puede observar la simplicidad de esta arquitectura.

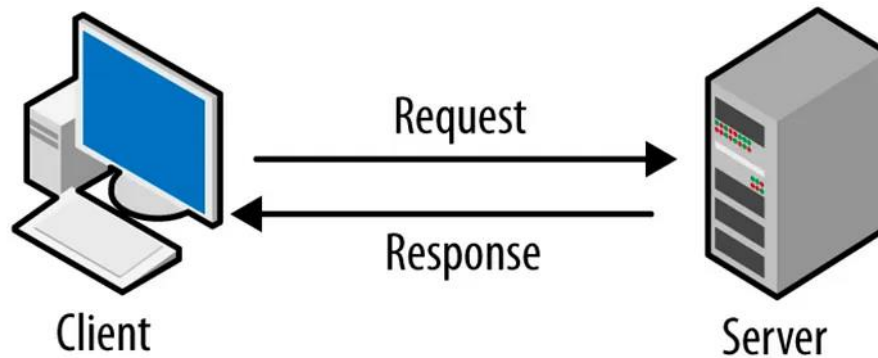


Figura 9. Arquitectura Cliente-Servidor [11]

Algunas de las ventajas que ofrece la arquitectura cliente-servidor son:

- **Escalabilidad:** La arquitectura cliente-servidor es escalable, lo que significa que puede adaptarse a un incremento en la demanda.
- **Reusabilidad:** Permite reutilizar el código y los recursos, lo que reduce los costes de desarrollo y mantenimiento.
- **Seguridad:** Puede ayudar a mejorar la seguridad de los datos, ya que los datos sensibles pueden almacenarse en los servidores.

4.3. Servidor - Arquitectura REST

La arquitectura *REST* (Transferencia de Estado Representacional) es un estilo arquitectónico para el diseño de sistemas de software distribuido. En un sistema *RESTful*, los recursos (datos o servicios) son identificados por localizadores de recursos uniformes (*URLs*).

Las operaciones sobre estos recursos se realizan utilizando los métodos estándar para obtener (*GET*), publicar (*POST*), actualizar (*SET*) o eliminar datos (*DELETE*).

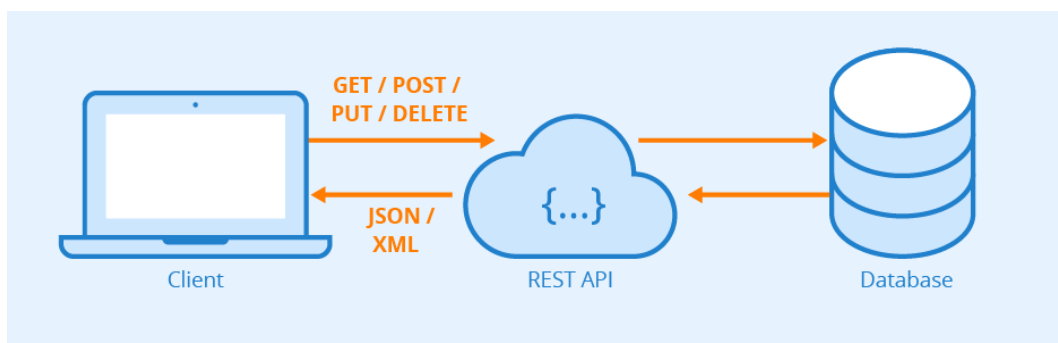


Figura 10. Funcionamiento general arquitectura REST [12]

Además, se usarán los códigos de estado HTTP para indicar el resultado de una solicitud entre el cliente y el servidor. Los códigos se dividen en distintas clases, cada una de estas con un significado general.

Aquí alguna de las clases más comunes y sus significados generales:

- **1xx** - Respuestas informativas: Indican que la solicitud ha sido recibida y el servidor está procesando la solicitud.
- **2xx** - Respuestas satisfactorias: Indican que la solicitud fue recibida, entendida y aceptada satisfactoriamente.
- **3xx** - Redirecciones: Para indicar que el cliente necesita realizar más acciones para completar la solicitud.
- **4xx** - Errores del cliente: Indican que hubo un error por parte del cliente al realizar la solicitud.
- **5xx** - Errores en el servidor: Indican que hubo un error por parte del servidor al procesar la solicitud.

El servidor por tanto seguirá una arquitectura REST por las ventajas que nos aporta:

- **Escalabilidad:** Es muy sencillo escalar horizontalmente, ya que no tiene estados y cada solicitud contiene toda la información necesaria para que el servidor sea capaz de resolverla.
- **Interoperabilidad:** Son independientes de la plataforma y se puede implementar en cualquier lenguaje de programación. Si se quisiese crear una página web para la aplicación se podría usar el mismo servidor sin problema.
- **Capacidad de almacenamiento en caché:** Permite a los clientes almacenar en caché las respuestas. Esto mejora el rendimiento y reduce la carga en el servidor.

4.3.1 Arquitectura Modelo-Vista-Controlador

En el servidor se ha utilizado un esquema que sigue un enfoque basado en el patrón de diseño MVC (Modelo-Vista-Controlador). Este patrón divide las responsabilidades en:

- **Modelo:** Es la capa encargada de trabajar con los datos, tiene los mecanismos para acceder a la base de datos y actualizar los datos. Para este proyecto se utiliza la librería *SQLAlchemy*, un ORM (*Object-RelationalMapping*) que mapea las entidades directamente a la base de datos.
- **Vista:** Es la capa encargada de generar la visualización de las interfaces de usuario. En el caso de este proyecto la vista no se devuelve ni es representada directamente en el servidor, sino a través de peticiones y respuestas a estas en formato REST.
- **Controlador:** Es capaz de responder a las acciones que se solicitan realizando una validación previa. Actúa como puente entre el modelo y la vista. En este caso, los controladores vienen definidos por las funciones que son capaces de recibir una petición y devolver lo que el usuario necesita.

Todos los ficheros relacionados con las peticiones se encuentran en el directorio «routes», mientras que en el directorio «utils» se encuentran ficheros adicionales para facilitar el funcionamiento de los *Blueprints*.

4.4 Aplicación móvil - Estructura

La aplicación móvil sigue una estructura por el tipo de fichero, es decir, los ficheros pertenecerán a la carpeta que corresponde con el tipo de fichero. Es una estructura muy sencilla y útil para proyectos de poca envergadura.

Además, se diferencia entre componentes y pantallas, de forma que los componentes son aquellos que son utilizados en varias pantallas o que no tienen representación gráfica sin necesidad de duplicar código. En la Figura 13 se puede observar la estructura de la aplicación móvil en este proyecto.

Finalmente, en el directorio «context» está todo lo relacionado con la autenticación del usuario, y, en el directorio «utils» existen archivos convenientes para otros ficheros, como por ejemplo el archivo «api.js». En este archivo se encuentran todas las peticiones a la base de datos.

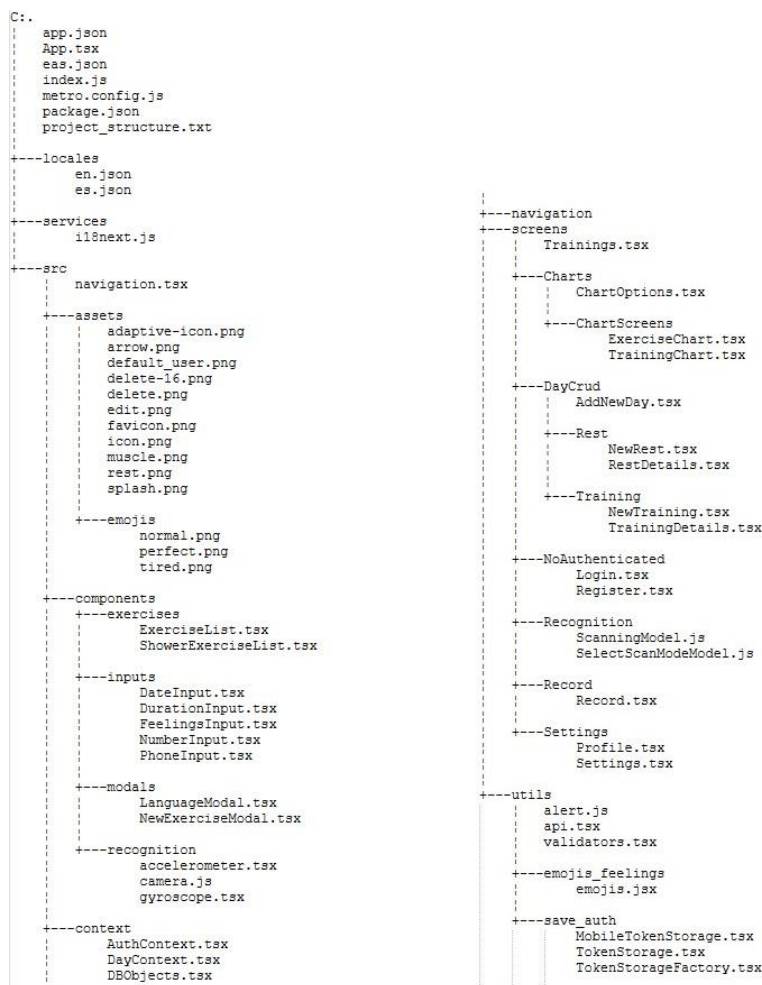


Figura 13. Organización de aplicación *React Native* por carpetas

5

Desarrollo

5.1 Metodología

En el marco de este proyecto, se ha adoptado una metodología iterativa incremental que ha permitido una evolución sistemática y controlada del desarrollo. Esta metodología es especialmente adecuada para proyectos como este, en los que los requisitos pueden cambiar o necesitan una validación continua. En la Figura 14 se puede observar cómo se estructuran las iteraciones siguiendo esta metodología.

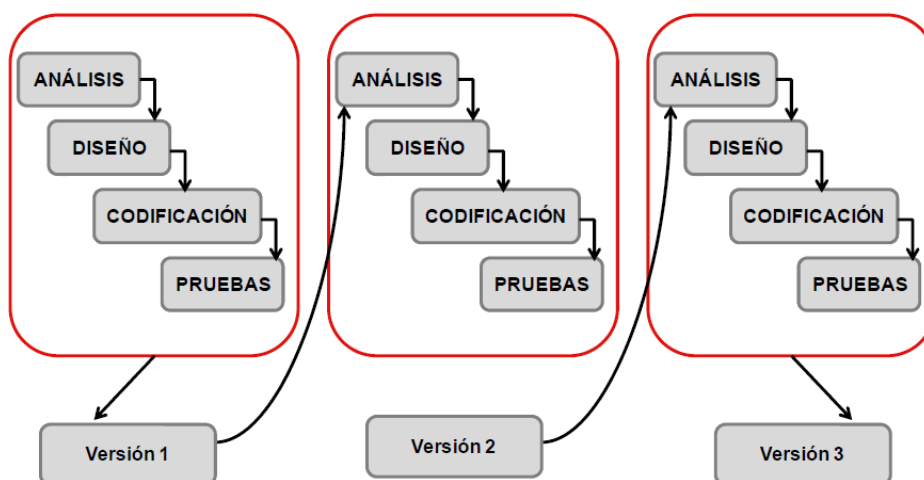


Figura 14. Metodología iterativa e incremental [13]

5.1.1 Descripción del proceso iterativo

El proyecto se ha dividido en iteraciones (*sprints*) con una duración de dos semanas cada una. Cada iteración tiene objetivos específicos y proporciona un marco temporal bien definido para el desarrollo.

Durante cada iteración se han incorporado nuevas funcionalidades al sistema y mejorado las anteriores. Este enfoque ha permitido un crecimiento escalonado de la aplicación y una reacción ágil a las necesidades fluctuantes del proyecto.

Al concluir cada iteración, se ha llevado a cabo una revisión exhaustiva para evaluar el progreso y planificar la siguiente iteración. Esta revisión ha sido útil para asegurarse de que el desarrollo es completo y no hay ningún requisito restante por implementar. Además, ha proporcionado una visión integral del estado del proyecto y ha facilitado la toma de decisiones fundamentadas para las iteraciones siguientes.

5.1.2 Ventajas de la metodología iterativa incremental

- **Adaptabilidad:** La metodología ha resultado ser altamente adaptable a los cambios en los requisitos en el proyecto. La flexibilidad inherente ha permitido ajustes continuos, lo que ha asegurado que el desarrollo siempre estuviera alineado con las expectativas del proyecto. Esta ha sido una de las mayores diferencias con otras metodologías, como con el modelo en cascada.
- **Retroalimentación continua:** Gracias a la incorporación de revisiones periódicas y la retroalimentación constante al final de las iteraciones se han podido realizar ajustes proactivos, que han contribuido a la mejora constante del producto.
- **Gestionar la complejidad del proyecto:** Los requisitos se han organizado de forma que los primeros que se realizan son los más importantes y, por tanto, si el tiempo de desarrollo no hubiese sido suficiente, los requisitos más importantes estarían implementados.

5.2 Primera iteración

5.2.1 Organización

En la primera iteración se ha querido implementar todo lo relacionado con el usuario, lo que servirá de base para que en las próximas iteraciones las aportaciones que el usuario realice se relacionen directamente con él.

Por tanto, en la primera iteración se han implementado los siguientes requisitos:

- RF1 - Registro de usuario
- RF2 - Identificación del usuario
- RF3 - Cerrar sesión
- RF4 - Cambiar datos del usuario

Para ser capaces de implementar todos los requisitos, primero se ha creado la base de datos siguiendo el esquema definido en la etapa de diseño de la aplicación.

También se ha considerado el requisito no funcional RNF1. Este recoge que los datos del usuario serán tratados de forma segura. Además, también se ha tenido en cuenta el requisito no funcional RNF5, para dar opción al usuario de cambiar el idioma de la aplicación.

5.2.2 Servidor

En el servidor se ha realizado un *endpoint* (punto de acceso) para hacer la identificación a través de un nombre de usuario y contraseña. Este devuelve un *token* (señal) que se usará para mantenerse identificado durante el resto de las operaciones.

También se ha implementado el *endpoint* de registro, que recibe parámetros obligatorios como el nombre de usuario y contraseña, y otros opcionales como el número de teléfono o la dirección.

Además, la contraseña del usuario se guarda con un *hash* con el algoritmo *HS256* y una duración de siete días. La implementación se ha realizado a través de la librería *jwt* de *Python*, que permite calcular los *hashes* fácilmente.

Por otro lado, una función se ejecutará antes de aquellas en las que la identificación del usuario sea necesaria, la cual verificará que el *token* que recibe sea correcto y no esté caducado, además de devolver el usuario al cual le pertenece ese *token*.

Por último, han sido necesarios los *endpoints* para obtener la información de un usuario específico y para modificarlo realizando las validaciones pertinentes.

5.2.3 Aplicación móvil

En la aplicación móvil, se han implementado pantallas para realizar la identificación del usuario, para hacer el registro de un nuevo usuario y, por último, una pantalla donde es necesario que el usuario esté identificado y en la cual se pueda cerrar sesión.

El *token* de autenticación del usuario se guarda de forma segura localmente en la aplicación, de forma que, si el usuario sale de la aplicación y vuelve a entrar, se iniciará sesión automáticamente si el *token* no ha caducado.

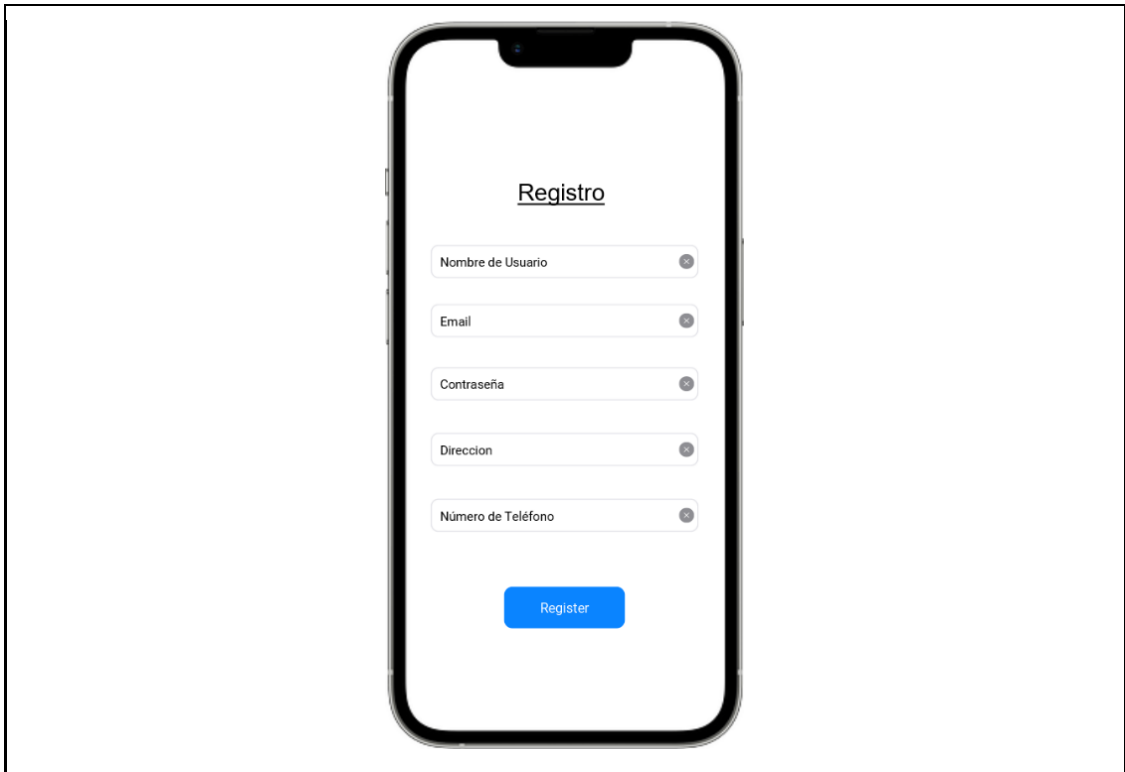
Se ha creado un navegador de *React Native* que, dependiendo de si el usuario está identificado o no, lo redirigirá a la pantalla de identificación o a la del usuario ya identificado.

En las pantallas de identificación y registro se hace una comprobación de que los campos han sido rellenados correctamente: el formato del correo electrónico, el formato del número de teléfono, etc. En caso contrario se mostrará un mensaje de error.

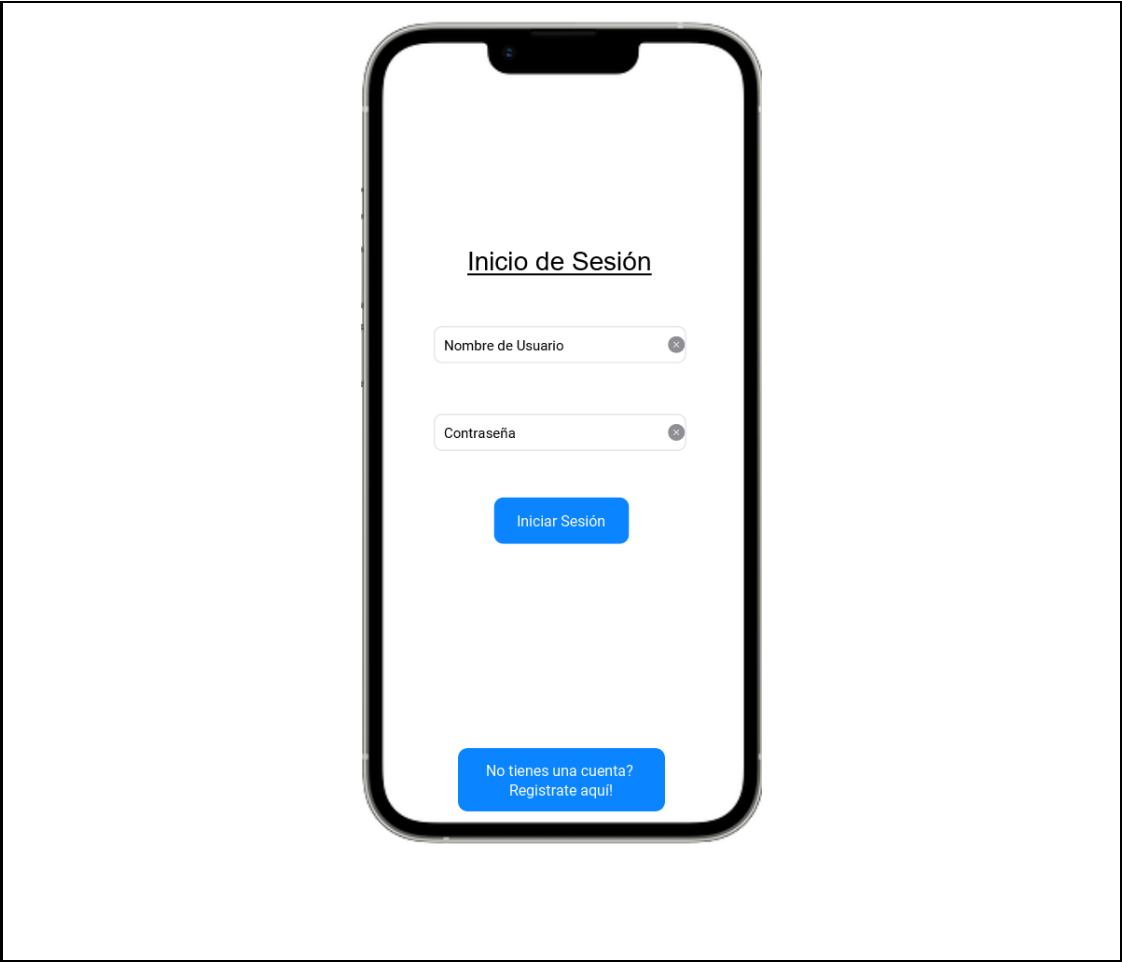
En cuanto al cambio de datos del usuario y el cambio de idioma, se ha implementado una pantalla de ajustes con las opciones de cambiar el idioma y de ir a la pantalla de cambio de datos del usuario. El idioma también es posible cambiarlo desde el menú de identificación, de forma que el usuario no tiene por qué estar identificado para hacer el cambio de idioma.

5.2.4 Casos de uso implementados

Título	Registro del usuario
Descripción	El usuario podrá registrarse con un usuario y contraseña, además podrá añadir otros datos como el número de teléfono o la fecha de nacimiento
Pre-condición	El usuario no está identificado en la aplicación
Post-condición	Se añade un usuario nuevo
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario abre la aplicación 2. El usuario selecciona la opción de «No tengo cuenta, regístrate aquí» 3. Se muestra una nueva pantalla donde poner los datos necesarios para el registro 4. El usuario inserta los datos necesarios y presiona el botón «Registrarse» 5. El sistema muestra un mensaje de éxito, guarda el usuario en la base de datos y redirige al usuario a la pantalla de Inicio de sesión 	
Escenario alternativo	
<ol style="list-style-type: none"> 4.b. El usuario no inserta los datos correctamente 5.b. El sistema muestra un mensaje de error 6.b. Vuelta al punto 4 	
Maquetas de interfaz	



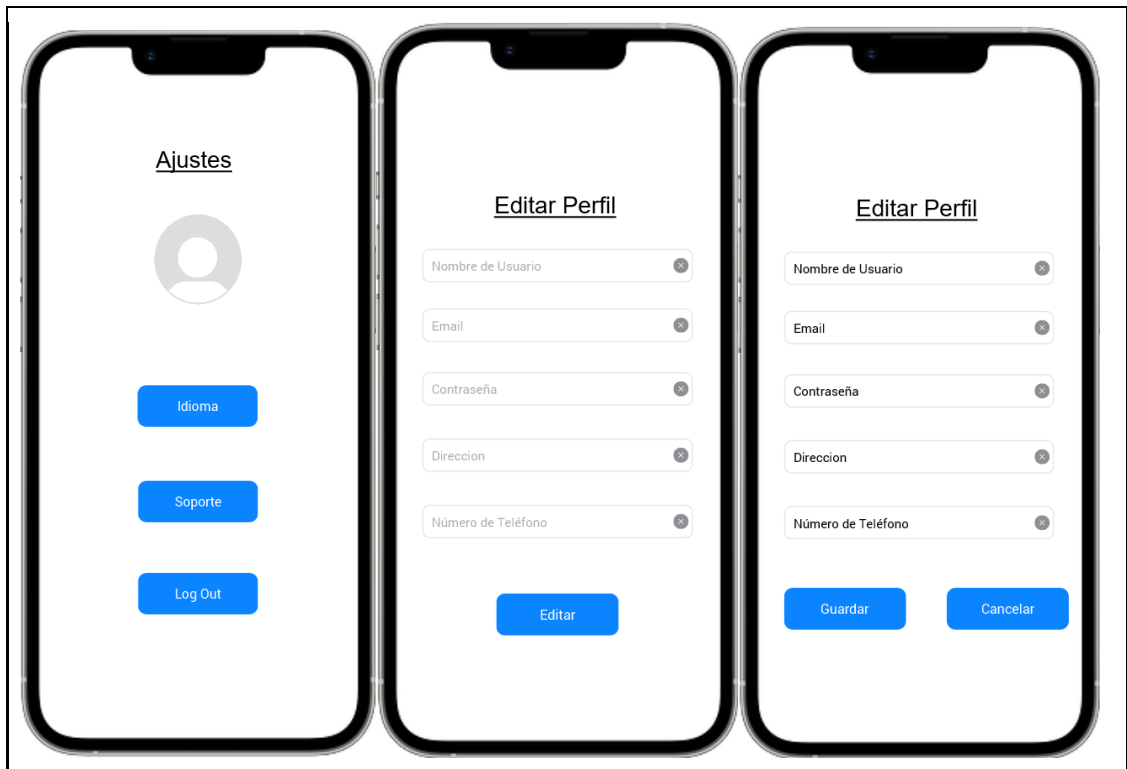
Título	Identificación del usuario
Descripción	El usuario podrá acceder a la aplicación con un usuario y contraseña válidos
Pre-condición	El usuario no está identificado en la aplicación
Post-condición	El usuario está identificado en la aplicación
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario abre la aplicación 2. El usuario indica un nombre de usuario y contraseña correctos 3. El sistema le redirige a la página principal con el usuario identificado 	
Escenario alternativo	
<ol style="list-style-type: none"> 2.b. El usuario indica un nombre de usuario y contraseña incorrectos 3.b. El sistema muestra un mensaje indicando la identificación fallida 4.b. Vuelta al punto 2 	
Maquetas de interfaz	



Título	Cerrar sesión
Descripción	Los usuarios identificados podrán cerrar la sesión para poder registrarse/identificarse con otra cuenta
Pre-condición	El usuario está identificado y en la pestaña de ajustes
Post-condición	El usuario no está identificado en la aplicación
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario presiona el botón para cerrar sesión 2. El sistema muestra una pantalla de confirmación 3. El usuario selecciona aceptar 4. El sistema cierra la sesión y redirecciona a la pantalla de inicio de sesión 	
Maquetas de interfaz	



Título	Cambiar datos del usuario
Descripción	El usuario identificado podrá cambiar datos varios de su cuenta como la dirección, el número de teléfono o la fecha de nacimiento
Pre-condición	El usuario está identificado y en la pestaña de ajustes
Post-condición	Los datos del usuario han cambiado
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario presiona el botón para cambiar información de usuario 2. El sistema redirecciona al usuario a la página para cambiar la información, pero los campos no son editables 3. El usuario presiona el botón de editar 4. El sistema habilita la edición de los campos y el botón guardar y cancelar 5. El usuario cambia los campos necesarios y presiona el botón para guardar 6. El sistema realiza los cambios y vuelve a habilitar el botón editar 	
Escenarios alternativos	
<p>5.b. El usuario no inserta los datos correctamente</p> <p>6.b. El sistema muestra un mensaje de error</p> <p>7.b. Vuelta al punto 5</p> <p>5.c. El usuario presiona el botón cancelar</p> <p>6.c. El sistema no cambia ningún dato y vuelve a habilitar el botón de editar</p>	
Maquetas de interfaz	



5.2.5 Evaluación del resultado

Al completar la primera iteración se ha conseguido implementar todo lo establecido en la planificación inicial, aunque no se ha conseguido que la entrada del número de teléfono y la fecha de nacimiento sea atractiva para el usuario y, por tanto, se propone como mejora para la siguiente iteración.

5.3 Segunda iteración

5.3.1 Organización

En la segunda iteración se han implementado los requisitos relacionados con los entrenamientos y los días de descanso. Esto comprende los siguientes requisitos:

- RF5 - Añadir un entrenamiento
- RF6 - Eliminar un entrenamiento
- RF7 - Modificar un entrenamiento
- RF8 - Añadir un día de descanso
- RF9 - Modificar un día de descanso
- RF10 - Eliminar un día de descanso

5.3.2 Servidor

En el servidor se han añadido nuevos *endpoints* para completar la funcionalidad. Un *endpoint* que devuelve el entrenamiento o día de descanso en un día específico al recibir una fecha, y en el caso de que no exista ningún entrenamiento o descanso en esa fecha, devuelve una respuesta con el código de error 404 (*Not Found*).

Además, un *endpoint* que recibe los parámetros necesarios para crear un día, valida que los datos son correctos y los inserta en la base de datos.

Por último, existen los *endpoints* para modificar un entrenamiento o día de descanso y para eliminarlos a partir de sus identificadores.

Todos los *endpoints* de esta iteración utilizan la función desarrollada en la iteración anterior para verificar el *token* del usuario antes de realizar cualquier operación. Así hay certeza de que el usuario está identificado y que los datos a modificar son los suyos.

5.3.3 Aplicación móvil

Se ha cambiado el modelo de la navegación actual para que exista una barra inferior que permita alternar entre las pantallas principales de la aplicación.

En la página principal de la aplicación se ha creado una pantalla para decidir qué tipo de día se quiere añadir, si un entrenamiento o un día de descanso. Una vez seleccionada una opción, se abrirá la página para rellenar todos los detalles.

Si para ese día ya existe un entrenamiento o descanso se mostrará una página de información que muestre los datos de ese día, además de un botón de modificar para cambiar los campos de ese día.

Por último, se ha cambiado la entrada de número de teléfono y de fecha de nacimiento de la primera iteración para que sean más atractivas y usables.


5.3.4 Casos de uso implementados

Título	Añadir un entrenamiento
Descripción	El usuario podrá añadir un nuevo entrenamiento para una fecha determinada
Pre-condición	El usuario está identificado y en la pantalla para añadir un día
Post-condición	El sistema ha registrado un nuevo entrenamiento para el usuario identificado
Escenario principal	
<ol style="list-style-type: none">1. El usuario presiona el botón para añadir un nuevo entrenamiento2. El sistema muestra un formulario con la información respecto al entrenamiento3. El usuario rellena los datos y presiona el botón para añadir4. El sistema añade el entrenamiento y muestra un mensaje de éxito	
Escenario alternativo	

- 3.b. El usuario no inserta los datos correctamente y presiona el botón para añadir
- 4.b. El sistema muestra un mensaje de error
- 5.b. Vuelta al punto 3

Maquetas de interfaz



Título	Eliminar un entrenamiento
Descripción	El usuario podrá eliminar un entrenamiento registrado
Pre-condición	El usuario está identificado, en la pantalla de entrenamientos y con un entrenamiento registrado
Post-condición	El sistema ha eliminado el entrenamiento
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario presiona el botón para eliminar el entrenamiento 2. El sistema muestra un mensaje de confirmación 3. El usuario presiona el botón para aceptar la eliminación 4. El sistema borra el entrenamiento 	
Escenario alternativo	
<ol style="list-style-type: none"> 3.b. El usuario presiona el botón para cancelar la eliminación 4.b. Volvemos al punto 1 	
Maquetas de interfaz	
	

Título	Modificar un entrenamiento
Descripción	El usuario podrá modificar un entrenamiento registrado
Pre-condición	El usuario está identificado, en la pestaña de entrenamientos y con un entrenamiento registrado
Post-condición	El sistema ha modificado el entrenamiento

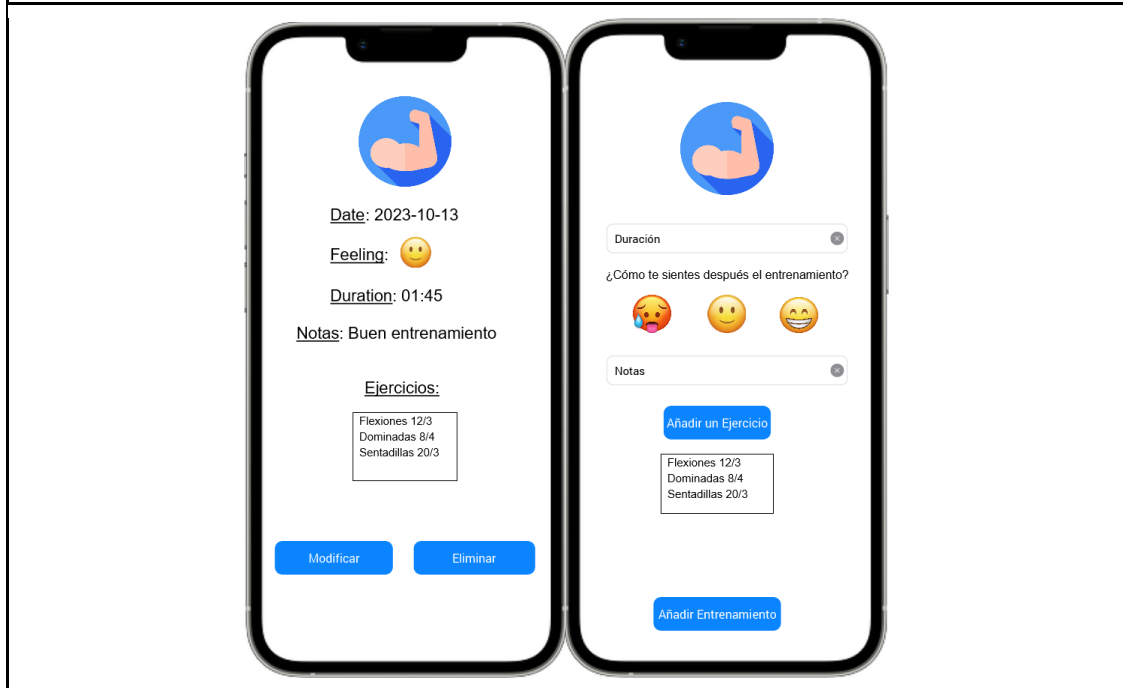
Escenario principal

1. El usuario presiona el botón para modificar el entrenamiento
2. El sistema muestra una ventana con los campos del entrenamiento modificables
3. El usuario cambia los datos que desee
4. El usuario presiona el botón para modificar
5. El sistema modifica el entrenamiento

Escenario alternativo


- 4.b. El usuario presiona el botón para cancelar la modificación
- 5.b. Volvemos al punto 1

Maquetas de interfaz



Título	Añadir un día de descanso
Descripción	El usuario podrá añadir un nuevo descanso para una fecha determinada

Pre-condición	El usuario está identificado y en la pantalla de entrenamientos
Post-condición	El sistema ha registrado un nuevo descanso para el usuario identificado
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario presiona el botón para añadir un nuevo descanso 2. El sistema muestra un formulario con la información respecto al descanso 3. El usuario rellena los datos y presiona el botón para añadir 4. El sistema añade el descanso y muestra un mensaje de éxito 	
Maquetas de interfaz	
	

Título	Eliminar un día de descanso
Descripción	El usuario podrá eliminar un día de descanso existente
Pre-condición	El usuario está identificado y en la pantalla de entrenamientos
Post-condición	El sistema ha eliminado el descanso seleccionado
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario presiona el botón para eliminar 2. El sistema muestra un mensaje para confirmar 3. El usuario selecciona el botón para confirmar 4. El sistema elimina el descanso seleccionado 	
Maquetas de interfaz	
	

Título	Modificar un día de descanso
Descripción	El usuario podrá modificar un día de descanso existente
Pre-condición	El usuario está identificado y en la pestaña de entrenamientos
Post-condición	El sistema ha eliminado el descanso seleccionado
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario presiona el botón para modificar el descanso 2. El sistema muestra una ventana con los campos del descanso modificables 3. El usuario cambia los datos que desee 4. El usuario presiona el botón para modificar 5. El sistema modifica el descanso 	
Maquetas de interfaz	

5.3.5 Evaluación del resultado

En la segunda iteración se ha implementado de manera exitosa todo lo que se estableció en la organización. El aspecto visual de la aplicación no es el esperado y por tanto se propone investigar y mejorar la apariencia de esta para la siguiente iteración.

5.4 Tercera iteración

5.4.1 Organización

En la tercera iteración se ha implementado lo relacionado con los ejercicios que se realizan en cada entrenamiento. Se pueden indicar las repeticiones, series y peso para cada uno de los ejercicios. Los requisitos correspondientes son, por tanto:

- RF11 - Añadir un ejercicio a un entrenamiento
- RF12 - Eliminar un ejercicio de un entrenamiento
- RF13 - Modificar un ejercicio de un entrenamiento

5.4.2 Servidor

Como resultado de esta iteración, en el servidor, al añadir o modificar un entrenamiento se tiene en cuenta que se puede recibir una lista de ejercicios asociados a ese entrenamiento.

Los ejercicios asociados se insertan en la base de datos relacionados con el entrenamiento correcto. De esta forma se unifica el *endpoint* para que todo se realice en una sola petición al servidor.

También se ha añadido un nuevo *endpoint* que devuelve todos los ejercicios disponibles que el usuario puede añadir a su entrenamiento.

Por último, en la base de datos los nombres de los ejercicios se guardan en inglés y en español, y por tanto, el servidor devuelve el nombre de los ejercicios en función de lo que el usuario especifica en la cabecera «*Language*», por defecto en inglés.

5.4.3 Aplicación móvil

La aplicación móvil se ha modificado para que en las páginas de agregar o modificar un entrenamiento exista un botón para añadir ejercicios. En esta nueva pantalla se puede indicar el número de repeticiones, sets y el peso.

Por otro lado, se mostrará la lista de ejercicios ya añadidos en el entrenamiento, en la que se podrá seleccionar alguno de los ejercicios ya añadidos para modificar cualquiera de sus campos o eliminarlos.

Si un ejercicio ya se ha realizado en el entrenamiento, entonces ya no aparecerá entre los ejercicios disponibles.

Por último, se han investigado librerías para que la aplicación mantenga un buen aspecto visual.

5.3.4 Casos de uso implementados

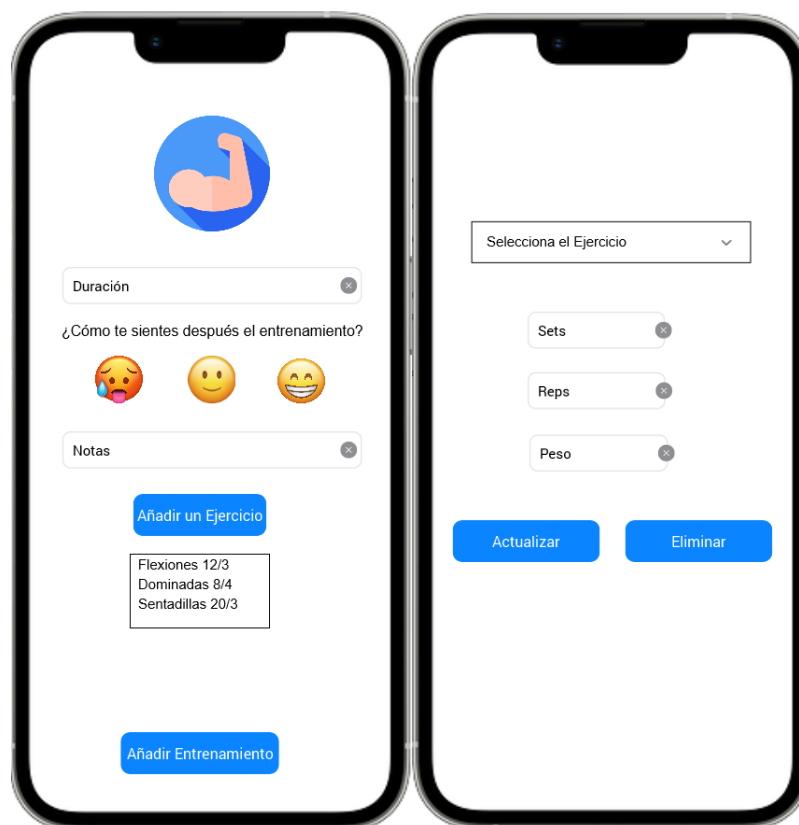
Título	Añadir un ejercicio a un entrenamiento
Descripción	El usuario podrá añadir un ejercicio a un entrenamiento, indicando las series, repeticiones y peso
Pre-condición	El usuario está identificado y en la pantalla del entrenamiento al que quiere añadir un ejercicio
Post-condición	El sistema ha añadido el ejercicio al entrenamiento
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario presiona el botón para modificar 2. El sistema muestra una ventana con los campos modificables y hay un botón para añadir un ejercicio 3. El usuario presiona el botón para añadir un ejercicio 4. El sistema muestra un formulario con la información del ejercicio 5. El usuario rellena el formulario y presiona el botón para añadir 6. El sistema añade el ejercicio al entrenamiento 	
Maquetas de interfaz	

Título	Eliminar un ejercicio de un entrenamiento
Descripción	El usuario podrá eliminar un ejercicio registrado en un entrenamiento
Pre-condición	El usuario está identificado y en la pantalla del entrenamiento al que desea eliminar el ejercicio
Post-condición	El sistema ha eliminado el ejercicio al entrenamiento

Escenario principal

1. El usuario presiona el botón para modificar
2. El sistema muestra una ventana con los campos modificables y la lista de ejercicios
3. El usuario presiona en el ejercicio que quiere eliminar
4. El sistema muestra un formulario con la información del ejercicio y un botón para eliminarlo
5. El usuario presiona el botón para eliminar el ejercicio
6. El sistema elimina el ejercicio del entrenamiento y ya no es visible en la lista de ejercicios realizados

Maquetas de interfaz



Título	Modificar un ejercicio de un entrenamiento
Descripción	El usuario podrá modificar un ejercicio de un entrenamiento
Pre-condición	El usuario está identificado y en la pantalla del entrenamiento al que modificar el ejercicio
Post-condición	El sistema ha modificado el ejercicio
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario presiona el botón para modificar 2. El sistema muestra una ventana con los campos modificables y la lista de ejercicios 3. El usuario presiona en el ejercicio que quiere modificar 4. El sistema muestra un formulario con la información del ejercicio 5. El usuario cambia los campos del formulario y presiona el botón de guardar 6. El sistema modifica el ejercicio y vuelve a la ventana del entrenamiento 	
Maquetas de interfaz	

5.4.5 Evaluación del resultado

En la tercera iteración se ha conseguido mejorar el aspecto visual de la aplicación, que era el propósito proveniente de la iteración anterior. Además, se han implementado los nuevos objetivos propuestos y mejorado el código ya existente en la aplicación móvil.

5.5 Cuarta iteración

5.5.1 Organización

Los buenos resultados obtenidos en esta cuarta iteración, que es la última de la planificación inicial, han permitido plantear una segunda fase del proyecto, centrada en la introducción en la aplicación de funcionalidades basadas en ciencia de datos. Por tanto, se han implementado los siguientes requisitos:

- RF13 - Ver entrenamientos anteriores
- RF14 - Ver gráfica de progresión
- RF15 - Cambiar el objeto de la gráfica
- RF16 - Cambiar el número de entrenamientos de la gráfica

5.5.2 Servidor

En el servidor ha sido necesario añadir el *endpoint* para obtener los entrenamientos con la posibilidad de añadir filtros, de forma que se pueden establecer las fechas de inicio y de fin como parámetro de consulta. Así el *endpoint* devuelve todos los días del usuario identificado entre esas fechas.

Los parámetros de consulta (*query parameters*) son los datos adicionales que se envían como parte de una solicitud *HTTP* para solicitar recursos específicos. En este caso se han añadido los parámetros de consulta «*groupby*», «*start_date*» y «*end_date*». Con estos tres parámetros se han podido obtener los datos necesarios para la implementación de las gráficas.

Por último, se ha añadido un *endpoint* para obtener los datos (repeticiones, series o peso) de ejercicios específicos. Esta información se podrá utilizar en la aplicación para ver el progreso de cada usuario en cada ejercicio. Se han añadido los mismos parámetros de consulta para este *endpoint*.

5.5.3 Aplicación móvil

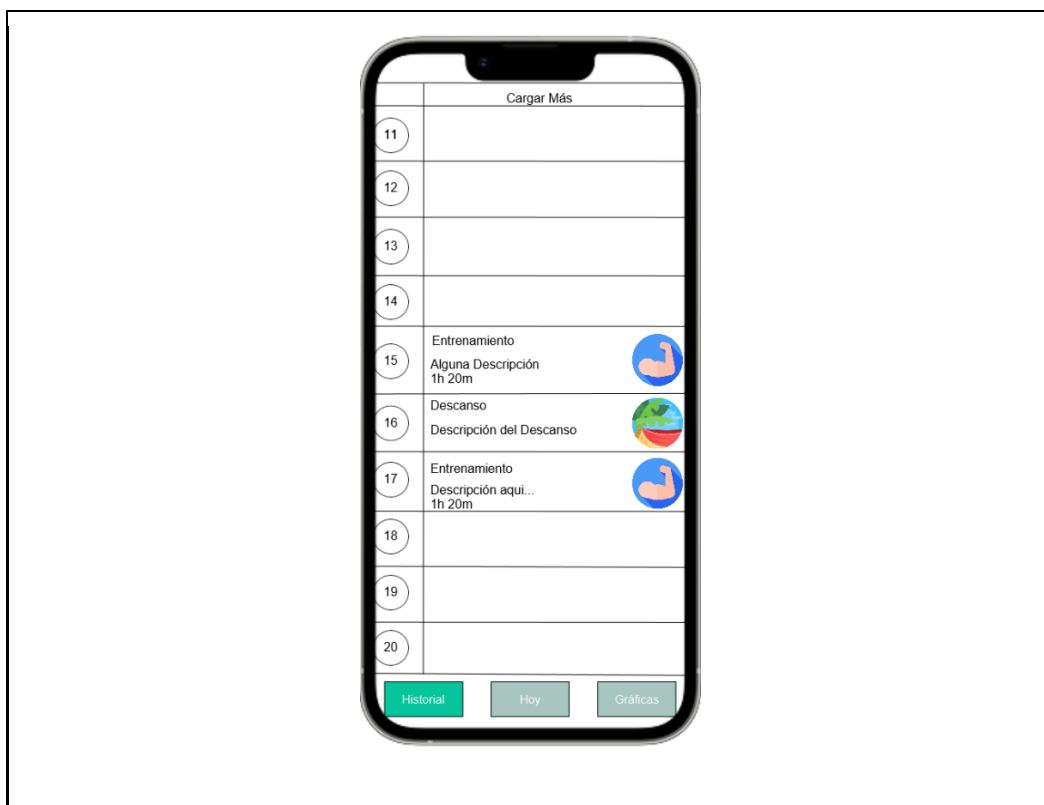
En la aplicación móvil se ha añadido una pantalla para mostrar el historial de días. Esta pantalla es accesible desde la barra inferior de la aplicación. En este historial, se diferencia muy bien si en ese día se realizó un entrenamiento o un descanso gracias a que los entrenamientos y los días tienen distintas imágenes.

En el historial, por cuestiones de rendimiento no se muestran todos los días, sino que aparece un número de días específico y el usuario tendrá la opción de cargar más días si así lo desea. A su vez, desde el historial, es posible acceder a la información del entrenamiento para poder verlo, modificarlo o borrarlo.

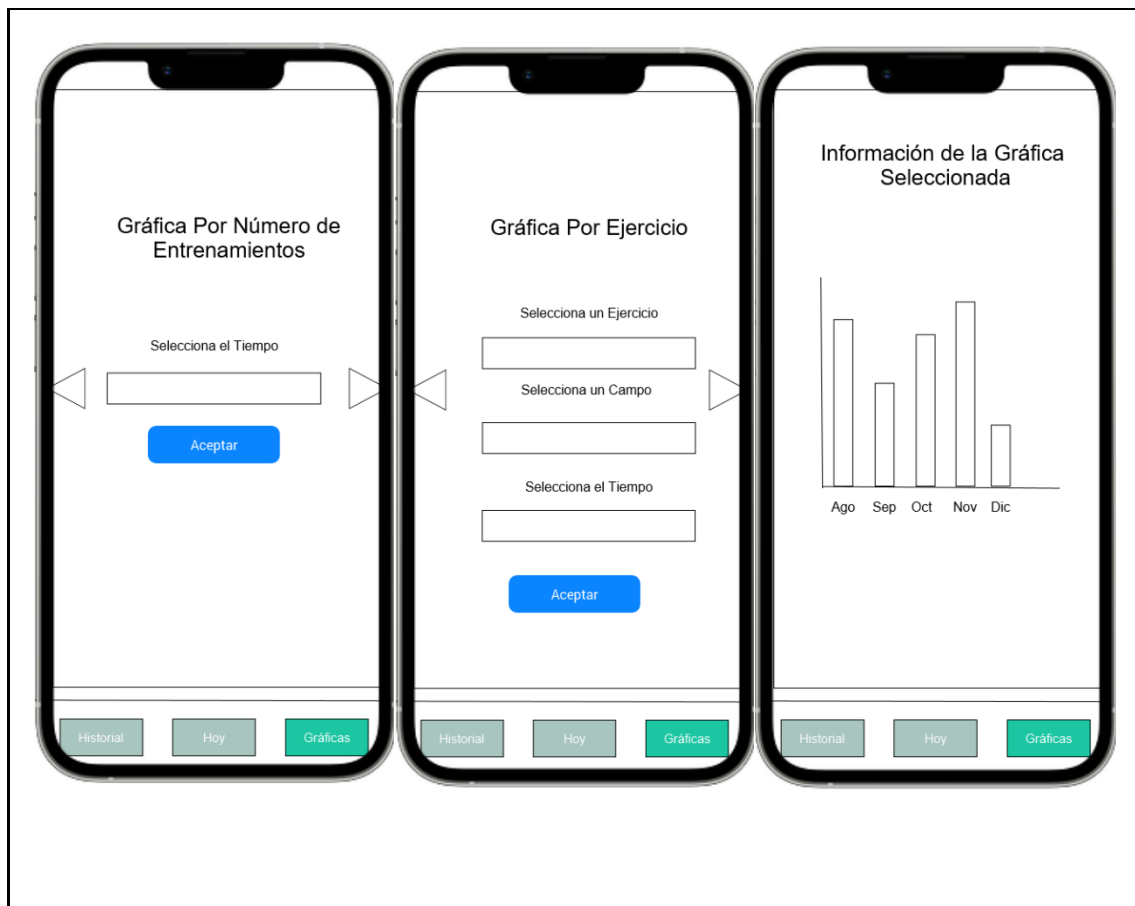
Por último, se han implementado las funcionalidades relacionadas con las gráficas. En la primera página el usuario puede seleccionar los aspectos que desea ver en la gráfica. Una vez el usuario ha seleccionado los criterios, al presionar un botón podrá ver la gráfica e incluso cambiar algunos criterios desde esa pantalla.

5.5.4 Casos de uso implementados

Título	Ver entrenamientos anteriores
Descripción	El usuario podrá ver el historial de sus últimos entrenamientos
Pre-condición	El usuario está identificado en la aplicación
Post-condición	El sistema mostrará los entrenamientos y descansos anteriores
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona en el menú inferior la opción del historial 2. El sistema muestra el historial con los últimos entrenamientos 	
Escenario alternativo	
<ol style="list-style-type: none"> 3.a. El usuario desea ver más entrenamientos de los mostrados y presiona el botón para cargar más 4.a El sistema muestra entrenamientos más antiguos 	
Maquetas de interfaz	



Título	Ver gráfica de progresión
Descripción	El usuario podrá ver una gráfica con la progresión en sus entrenamientos
Pre-condición	El usuario está identificado
Post-condición	Se muestran las gráficas con lo que el usuario ha seleccionado
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona en el menú inferior la opción de gráficas 2. El usuario selecciona entre las distintas opciones de la gráfica 3. El usuario presiona el botón para aceptar 4. El sistema muestra la gráfica según las opciones elegidas por el usuario 	
Escenario alternativo	
<ol style="list-style-type: none"> 2.b. El usuario no selecciona las opciones correctamente 3.b. El sistema muestra un mensaje de error 4.b. Vuelta al punto 2 	
Maquetas de interfaz	



Título	Cambiar el objeto de la gráfica
Descripción	El usuario podrá cambiar el aspecto que se desea comparar en la gráfica: por número de entrenamientos o por ejercicios. Puede seleccionar qué ejercicio comparar y que información de ese ejercicio
Pre-condición	El usuario está identificado
Post-condición	El objeto de la gráfica es aquel seleccionado por el usuario
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona en el menú inferior la opción de gráficas 2. El usuario elige en un carrusel el objeto que desea comparar 3. El usuario indica el resto de información que desea comparar 4. El usuario presionó el botón para aceptar 5. El sistema muestra la gráfica según el objeto elegido por el usuario 	
Escenario alternativo	

- 4.b. El usuario no ha insertado los datos correctamente
- 5.b. El sistema muestra un mensaje de error
- 6.b. Vuelta al punto 2

Maquetas de interfaz



Título	Cambiar el número de entrenamientos de la gráfica
Descripción	El usuario podrá cambiar la gráfica para que se muestre con la información recogida en los entrenamientos de los últimos 6 días o todos los días
Pre-condición	El usuario está identificado
Post-condición	La gráfica corresponde a lo seleccionado por el usuario

Escenario principal

1. El usuario selecciona en el menú inferior la opción de gráficas
2. El usuario elige en un carrusel el objeto a comparar
3. El usuario indica el tiempo a comparar
4. El usuario presiona el botón para aceptar
5. El sistema muestra la gráfica según el objeto elegido por el usuario

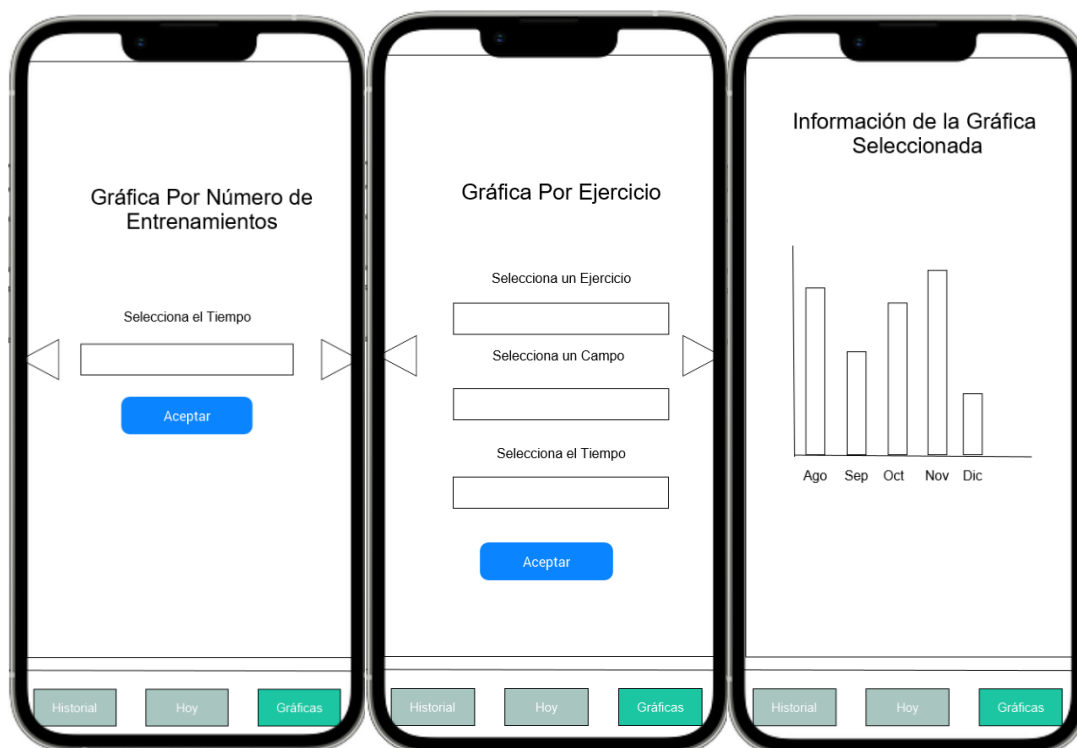
Escenario alternativo

4.b. El usuario no ha insertado los datos correctamente

5.b. El sistema muestra un mensaje de error

6.b. Vuelta al punto 2

Maquetas de interfaz



5.5.5 Evaluación del resultado

Para finalizar con esta cuarta iteración se ha verificado que todos los requisitos propuestos se han implementado y que por tanto la aplicación queda finalizada según el diseño inicial.

A partir de ahora, se ha planteado realizar un estudio, a través del uso de técnicas de la ciencia de datos, para el reconocimiento automático de la actividad física realizada, en base a la información recogida a través de de distintos sensores y dispositivos. Si este estudio resulta exitoso, se continuará con una última iteración que implemente la opción de reconocimiento en la aplicación.

6

Nuevas funcionalidades basadas en Ciencia de Datos

6.1 Introducción

La ciencia de datos ha emergido, en las últimas décadas, como un campo crucial en la era actual, y desempeña un papel fundamental en la extracción de conocimiento significativo a partir de conjuntos de datos.

Una vez la aplicación está desarrollada y probada al completo, en esta siguiente fase del proyecto, se ha podido estudiar la viabilidad del uso de técnicas de ciencia de datos para mejorar la aplicación.

Primero, hay que enfocar la ciencia de datos a lo que realmente pueda ser útil para el usuario. En este proyecto se ha decidido orientar la ciencia de datos al reconocimiento del tipo de ejercicio físico que el usuario está realizando en un momento concreto y comparar qué datos son más útiles a la hora de reconocer estos ejercicios.

Una vez establecido el objetivo, se ha obtenido información de estudios previos sobre este campo, para posteriormente establecer cuáles son los datos necesarios para el análisis y definir la metodología para obtenerlos.

6.2 Elección y obtención de los datos

Para la recolección de datos, se ha considerado utilizar las opciones más comúnmente utilizadas, es decir, aquellas que puedan simplificar la experiencia del usuario. Se prefiere aprovechar los datos que pueden obtenerse fácilmente del usuario en lugar de depender de *hardware* adicional que pueda complicar el proceso.

Inicialmente, se contempló la posibilidad de emplear relojes y pulseras inteligentes, ya que almacenan información sobre su posición y movimiento mediante sensores como el acelerómetro o el giroscopio. Sin embargo, se han presentado diversos problemas que han llevado a descartar esta opción, la manera en la que estos dispositivos realizaban las conexiones difería entre sí. A pesar de que todos utilicen *Bluetooth* o *Bluetooth Low Energy (BLE)*, la forma de enviar y recibir los datos es distinta para cada uno.

En un principio, se intentó emparejar el reloj inteligente *Samsung Galaxy Gear S3*, sin embargo, la conexión a través de BLE no estaba disponible de manera directa. A pesar de este obstáculo, se descubrió una aplicación que podía instalarse en el reloj para almacenar datos de varios sensores localmente en el dispositivo. No obstante, esta solución no resultaba viable para su integración con la aplicación principal.

Ante esta situación, se exploró la posibilidad de conectar un *Xiaomi Mi Band 4* siguiendo las indicaciones del artículo [14], el cual detallaba el proceso para la *Xiaomi Mi Band 3*. No obstante, la autenticación de la pulsera varía entre versiones, lo que dificultó la conexión y, en última instancia, no se logró establecer.

Por lo tanto, la decisión final se centró exclusivamente en aprovechar los datos que pueden ser obtenidos a través del dispositivo móvil.

6.2.1 Acelerómetro y giroscopio

El acelerómetro es un sensor que incluyen la mayoría de los dispositivos móviles que permite medir la orientación del dispositivo con su posición en los tres ejes para obtener el movimiento en un espacio tridimensional.

El giroscopio también se encarga de medir la aceleración y sirve como complemento del acelerómetro. Es capaz de medir los cambios angulares o giros que se realizan en el dispositivo, lo que proporciona información adicional.

Por tanto, el acelerómetro y giroscopio del dispositivo móvil aportan datos que nos pueden dar información suficiente para saber cómo varía la posición del dispositivo en un cierto intervalo de tiempo.

El principal inconveniente de esta opción radica en la necesidad de mantener el dispositivo en una posición constante durante cada captura de datos. Los valores pueden fluctuar si el dispositivo cambia su ubicación, como por ejemplo, al pasar del bolsillo de un pantalón al de una sudadera. Por tanto, se ha decidido utilizar un soporte

para móviles, comúnmente utilizado, que se ajusta al brazo, similar al que se muestra en la Figura 15.



Figura 15. Soporte para sujetar el móvil en el brazo [14]

Podemos asegurarnos de que los datos de los sensores del dispositivo móvil se capturan desde una posición consistente. El principal problema proveniente de esta fuente de datos son los ejercicios que involucran el tren inferior pero no el tren superior, ya que el dispositivo se encuentra en el brazo y solo detecta el movimiento en este.

Para obtener estos datos del dispositivo, se utiliza la librería *expo-sensors* que ya está integrada en el entorno *Expo*. Esta librería nos permite acceder a datos de varios sensores del dispositivo móvil, como el acelerómetro y el giroscopio. Además, nos permite especificar la frecuencia de captura y es compatible tanto con dispositivos *Android* como *iOS*.

6.2.1 Cámara

La cámara, un componente crucial de los dispositivos móviles, permite a los usuarios tomar fotos y vídeos. Este proyecto se centrará en la opción de grabación de vídeos para detectar los ejercicios realizados por el usuario.

Al analizar un vídeo del usuario haciendo ejercicio, se pone énfasis en qué información puede ser relevante para el reconocimiento de dicho ejercicio. No es necesario examinar todo el contenido del vídeo, como el fondo o el entorno en el que se realiza el ejercicio. Lo importante es observar cómo el usuario mueve cada parte de su cuerpo.

Por tanto, se van a estudiar los datos a partir de la información de la posición del usuario extraída de un vídeo, para ello hay que obtener la posición de puntos clave anatómicos del usuario. Con puntos clave anatómicos se hace referencia a partes del cuerpo que pueden ser útiles para el reconocimiento, como las rodillas o los codos.

Mediapipe es una librería desarrollada por *Google* en el lenguaje de programación Python capaz de facilitar este proceso, ya que ofrece la funcionalidad de detectar puntos clave de la cara, de las manos o del esqueleto completo, lo cual es fundamental para este proyecto.

Además, para un análisis más detallado de los datos, podríamos considerar elementos como el uso de mancuernas, bancos de entrenamiento o máquinas de ejercicio. Sin embargo, estos aspectos están fuera de los objetivos y requisitos del proyecto.

6.3 Aprendizaje Profundo

El aprendizaje profundo se basa en el uso de las denominadas redes neuronales artificiales que son un conjunto de técnicas de inteligencia artificial inspiradas en la estructura del cerebro humano, instrumentalizadas para aprender patrones intrincados en los datos.

Se basan en aprender y mejorar su eficacia a través del aprendizaje, es decir, con el tiempo y con datos. Permiten clasificar y agrupar los datos de entradas en datos de salida.

Las redes neuronales están compuestas por neuronas que reciben una o varias entradas ponderadas, realiza una serie de operaciones matemáticas en ellas y produce una salida. Estas neuronas se organizan en capas, la entrada de la red se encuentra en la capa de entrada y la salida en la capa de salida. Además, puede haber una o varias capas intermedias, conocidas como capas ocultas. (Figura 16)

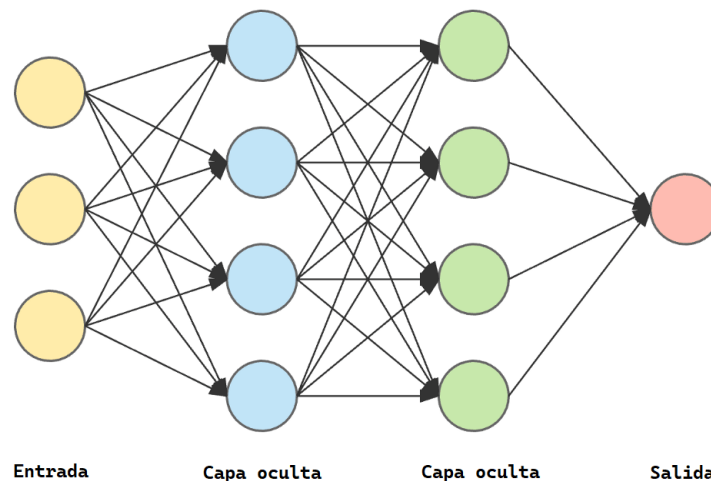


Figura 16. Funcionamiento de redes neuronales [15]

Dentro del amplio catálogo de técnicas de aprendizaje profundo, una de las más populares se centra en las denominadas redes neuronales convolucionales (CNN). Estas fueron diseñadas originalmente para procesar datos de imágenes, pero su arquitectura ha demostrado ser efectiva en una variedad de aplicaciones más allá de la visión por

computadora. Las CNN destacan en la extracción de características espaciales y temporales de datos, esta cualidad las hace muy útiles para este proyecto.

En lugar de conectar cada entrada a cada neurona, utilizan capas convolucionales que aplican filtros a regiones locales de entrada, esta convolución permite que la red aprenda representaciones y comparta parámetros.

Para este proyecto, se decidió usar una red neuronal para cada fuente de datos, es decir, una para los sensores y otra distinta para la cámara, ya que no se pueden obtener datos de los dos a la vez.

6.4 Representación de los datos

Una vez seleccionadas las fuentes de datos y el método que se va a utilizar es importante representar esos datos de modo que la red neuronal sea capaz de encontrar las diferencias y similitudes entre los ejercicios y sea capaz de clasificarlos.

La representación de los datos no puede variar de un entrenamiento a otro, se debe usar siempre la misma representación tanto para el entrenamiento de la red neuronal como para posteriormente la detección.

6.4.1 Representación de los sensores

En el contexto de los sensores, se han tomado dos decisiones cruciales. Primero, determinar la frecuencia de captura de datos, es decir, cuántas veces por segundo se deben almacenar los datos. Segundo, establecer el método para almacenar estos datos.

Es imperativo registrar la información sobre el instante en el que se toma cada dato, lo que permite a la red neuronal comprender en qué intervalo se lleva a cabo cada ejercicio durante el entrenamiento.

Por lo tanto, la representación adoptada es una lista y cada posición corresponde a un objeto de dimensión 4, con la siguiente estructura: (Marca de tiempo, X, Y, Z). Se requiere una lista para cada sensor, es decir, uno para los datos del acelerómetro y otro para los datos del giroscopio. En la Figura 17 se puede apreciar una captura real de los datos.

```

[[1703927158207, -0.4919891357421875, -0.7085113525390625, -0.4944305419921875],
[1703927158307, -0.490692138671875, -0.70904541015625, -0.492919921875],
[1703927158408, -0.4846038818359375, -0.70703125, -0.4868011474609375],
[1703927158508, -0.4776611328125, -0.706634521484375, -0.48870849609375],
[1703927158609, -0.500762939453125, -0.721649169921875, -0.4878997802734375],
[1703927158710, -0.4900665283203125, -0.7422027587890625, -0.4803619384765625],
[1703927158812, -0.52288818359375, -0.726776123046875, -0.4874114990234375],
[1703927158911, -0.4725494384765625, -0.691497802734375, -0.474334716796875],
[1703927159010, -0.455108642578125, -0.713043212890625, -0.46478271484375],
[1703927159111, -0.5312042236328125, -0.7432403564453125, -0.4702606201171875],
[1703927159212, -0.5099334716796875, -0.7311859130859375, -0.4759979248046875],
[1703927159312, -0.6323699951171875, -0.763153076171875, -0.531524658203125],
[1703927159412, -0.4569549560546875, -0.701171875, -0.369171142578125],
[1703927159509, -0.4604949951171875, -0.7098236083984375, -0.4504852294921875],
[1703927159613, -0.4774169921875, -0.6906280517578125, -0.5130615234375],
[1703927159715, -0.3040771484375, -0.7271728515625, -0.5474853515625],
[1703927159811, -0.296783447265625, -0.719879150390625, -0.5657806396484375],
[1703927159913, -0.309967041015625, -0.704071044921875, -0.5465240478515625],
[1703927160014, -0.415802001953125, -0.7598876953125, -0.5586395263671875],
[1703927160112, -0.4193115234375, -0.7651824951171875, -0.5492706298828125],
[1703927160214, -0.418182373046875, -0.7865447998046875, -0.5267486572265625],
[1703927160314, -0.382049560546875, -0.78204345703125, -0.5214691162109375],
[1703927160414, -0.4116363525390625, -0.734344482421875, -0.52288818359375],
[1703927160516, -0.4148712158203125, -0.7706756591796875, -0.522430419921875],
[1703927160613, -0.4423980712890625, -0.7652740478515625, -0.5008697509765625],
[1703927160714, -0.2653045654296875, -0.6903076171875, -0.5424346923828125],
[1703927160815, -0.4298248291015625, -0.831268310546875, -0.544952392578125],
[1703927160916, -0.42755126953125, -0.6400909423828125, -0.511077880859375],
[1703927161014, -0.295501708984375, -0.5658111572265625, -0.2993621826171875],
[1703927161116, -0.358612060546875, -0.6180267333984375, -0.4952239990234375],
[1703927161215, -0.37704467734375, -0.7215423583984375, -0.5213623046875],

```

Figura 17. Representación de los datos provenientes de los sensores

Se ha optado por una frecuencia de muestreo de 10 Hz para la recopilación de datos, lo que implica la captura de información diez veces por segundo. Esta elección se basa en los requisitos específicos del caso; seleccionar una frecuencia inferior habría dificultado la capacidad de la red neuronal para detectar cambios de posición, mientras que una frecuencia más alta podría haber generado una sobreabundancia de datos, lo que ralentizaría así el proceso de aprendizaje.

6.4.2 Representación de la cámara

Para la correcta representación de la cámara, es esencial comprender la información proporcionada por *Mediapipe*. (Figura 18)

```
0 - nariz
1 - ojo izquierdo (interno)
2 - ojo izquierdo
3 - ojo izquierdo (externo)
4 - ojo derecho (interno)
5 - ojo derecho
6 - ojo derecho (externo)
7 - oreja izquierda
8 - oreja derecha
9 - boca (izquierda)
10 - boca (derecha)
11 - hombro izquierdo
12 - hombro derecho
13 - codo izquierdo
14 - codo derecho
15 - muñeca izquierda
16 - muñeca derecha
17 - dedo meñique izquierdo
18 - dedo meñique derecho
19 - dedo índice izquierdo
20 - dedo índice derecho
21 - pulgar izquierdo
22 - pulgar derecho
23 - cadera izquierda
24 - cadera derecha
25 - rodilla izquierda
26 - rodilla derecha
27 - tobillo izquierdo
28 - tobillo derecho
29 - talón izquierdo
30 - talón derecho
31 - índice del pie izquierdo
32 - índice del pie
```

Figura 18. Puntos clave anatómicos que proporciona *Mediapipe* [16]

Cada uno de los puntos anatómicos viene representado por tres valores, que indican la posición en 3 dimensiones. En la Figura 19 se muestran los puntos sobre fotogramas de uno de los vídeos usados para el entrenamiento.



Figura 19. Puntos clave anatómicos marcados en el cuerpo

En este proyecto, se ha optado por utilizar los siguientes puntos clave anatómicos para representar el cuerpo: nariz, hombro izquierdo, hombro derecho, codo izquierdo, codo derecho, muñeca izquierda, muñeca derecha, cadera izquierda, cadera derecha, rodilla izquierda, rodilla derecha, tobillo izquierdo y tobillo derecho.

Esta selección se ha considerado la más relevante para llevar a cabo el reconocimiento. Optar por menos información podría resultar en la falta de datos útiles, mientras que utilizar más información podría sobrecargar los datos, lo que ralentizaría los procesos de entrenamiento y reconocimiento.

Al igual que en el caso de los sensores, es crucial registrar cuándo se ha hecho la toma de cada dato. En el contexto de los vídeos, no es necesario conocer la marca de tiempo; simplemente, los fotogramas del vídeo permiten identificar los intervalos en los que se realizan los ejercicios.

Los vídeos se toman con una tasa de refresco de 30 fotogramas por segundo, por tanto, habrá 30 líneas de datos por cada segundo de vídeo. Los datos se representan en tres listas, una con los datos del eje X, otra con los datos del eje Y, y una última con los datos del eje Z. En la Figura 20, se muestra una captura real de los datos.

```

[[[13, 0.400785356760025, 0.47093522548675537, 0.35484281182289124, 0.25884267687797546, 0, 0.09675716608762741,
0.2960115969181061, 0.28517842292785645, 0.20771069824695587, 0.20246627926826477, 0, 0.16448815166950226, 0],
[14, 0.4302472472190857, 0.47378507256507874, 0.39396169781684875, 0.29815956950187683, 0, 0.10517837107181549,
0.12621678411960602, 0.29015201330184937, 0.25037139654159546, 0.29744642972946167, 0, 0.2521190941333771, 0],
[26, 0.39386266469955444, 0.3732565641403198, 0.4127660393714905, 0.34740644693374634, 0.43696266412734985,
0.47418245673179626, 0.5352944135665894, 0.22743453085422516, 0.2403533160686493, 0.255370557308197, 0,
0.18929587304592133, 0],
[31, 0.475637823343277, 0.5661917924880981, 0.17374040186405182, 0.7856388688087463, 0.10911506414413452,
0.6105222105979919, 0.4533124268054962, 0.4919006824493408, 0.2681736350059509, 0.6193408966064453, 0,
0.5813257694244385, 0],
[32, 0.5091666579246521, 0.5669606328010559, 0.18757189810276031, 0.7741718888282776, 0.1358262151479721,
0.6247594356536865, 0.46125727891921997, 0.512561559677124, 0.277877539396286, 0.6127005219459534, 0,
0.5804126858711243, 0],
[33, 0.544246256351471, 0.6153167486190796, 0.21921683847904205, 0.774280309677124, 0.20801638066768646,
0.5479021072387695, 0.5059425830841064, 0.5395776033401489, 0.2945716083049774, 0.6428496241569519, 0,
0.5814812779426575, 0],
[34, 0.5792452096939087, 0.6440033316612244, 0.23240390419960022, 0.7716072201728821, 0.21349447965621948,
0.44331681728363037, 0.5207555890083313, 0.5833505988121033, 0.32269638776779175, 0.6208582520484924, 0,
0.5797309279441833, 0],
[35, 0.6063922047615051, 0.6831297874450684, 0.2585008442401886, 0.772102952003479, 0.26015806198120117,
0.46338236319397, 0.5704867839813232, 0.6084789633750916, 0.3449704647064209, 0.612748920917511, 0.29294559359550476,
0.5677376985549927, 0],
[36, 0.6208884716033936, 0.7216584086418152, 0.28180253505706787, 0.7696759700775146, 0.29442134499549866,
0.41507095098495483, 0.6126036047935486, 0.6293689012527466, 0.36482957005500793, 0.61160808801651,
0.2923297882080078, 0.5585064888000488, 0],
[37, 0.6470237970352173, 0.728941798210144, 0.29611483216285706, 0.7683280110359192, 0.3271004557609558,
0.3950175344944, 0.637667715549469, 0.6334884762763977, 0.37154853343963623, 0.6115840077400208, 0.29335007071495056,
0.560029923915863, 0],
[38, 0.6984149217605591, 0.7356483340263367, 0.3084600269794464, 0.7878513336181641, 0.36367690563201904,
0.3977803587913513, 0.6882851123809814, 0.6422892808914185, 0.37669837474823, 0.6116246581077576, 0.3010184168815613,
0.5610389709472656, 0],
[39, 0.708340048789978, 0.7705859541893005, 0.3358975350856781, 0.8140242099761963, 0.3681447505950928,
0.4174153506755829, 0.692219614982605, 0.6532801389694214, 0.4038277864456177, 0.6104304790496826, 0.3055781424045563,
0.5611532926559448, 0],
[40, 0.7217447757720947, 0.7689263820648193, 0.3457660675048828, 0.8143239617347717, 0.38773199915885925,
0.4330562651157379, 0.7189682126045227, 0.6564909219741821, 0.39938971400260925, 0.6085919141769409,
0.31002387404441833, 0.5564181804656982, 0],
]]]]

```

Figura 20. Representación de los datos provenientes de la cámara

Nótese que en la Figura 20 hay fotogramas que no aparecen en los datos, esto se debe a que cuando hay un fotograma donde no se reconoce ningún punto (por ejemplo, que el usuario no esté en el plano), directamente no se añade a la información.

Además, *Mediapipe* proporciona información sobre la visibilidad de cada punto con un número entre 0 y 1. Para este proyecto solo los puntos que tienen una visibilidad superior a 0,5 son añadidos a los datos, en caso opuesto se añade un 0 para esos datos.

6.5 Toma de datos

Para la toma de datos de los sensores se ha recogido la información mientras se realizaban los ejercicios utilizando el soporte para el dispositivo móvil, además se han apuntado los instantes de tiempo en los que se han empezado y finalizado cada ejercicio, además de las repeticiones realizadas para guardarlo en un archivo en formato csv y poder analizarlos.

En la Figura 21 se observa un ejemplo de fichero «*labels.csv*» para los sensores. El primer y segundo valor son las marcas de tiempo en las que se empieza y termina el ejercicio respectivamente, el tercer valor corresponde con el número de repeticiones realizadas y por último el nombre del ejercicio.

```

1 1703930079928,1703930093858,5,squat
2 1703930114500,1703930128525,5,squat
3 1703930143056,1703930157184,5,push_up
4 1703930180334,1703930191857,5,push_up
5 1703930209494,1703930225828,5,bench_press
6 1703930253485,1703930269100,5,bench_press
7 1703930283146,1703930300084,5,pull_up
8 1703930315714,1703930331447,5,pull_up

```

Figura 21. Fichero «*labels.csv*» de los sensores

Para la toma de datos de la cámara, se han grabado vídeos realizando los ejercicios y posteriormente se han calculado los puntos clave anatómicos y se han guardado en un fichero para su posterior procesamiento.

En la Figura 22 se observa un ejemplo de fichero «*labels.csv*» para la cámara. El primer y segundo valor son los fotogramas en los que empieza y termina el ejercicio respectivamente, el tercer valor corresponde con el número de repeticiones realizadas y por último el nombre del ejercicio.

```

1 952,1270,6,squat
2 1800,2137,6,squat
3 2942,3284,6,push_up
4 3805,4132,6,push_up
5 4350,4689,6,push_up
6 5851,6230,6,pull_up
7 6750,7176,6,pull_up
8 7381,7830,6,lateral_raise
9 8150,8530,6,lateral_raise
10 8910,9289,6,lateral_raise

```

Figura 22. Fichero «*labels.csv*» de la cámara

Para los sensores, se han recopilado datos de un total de 15 sesiones de entrenamiento. De estos, 13 provienen de un hombre con una estatura de 178 centímetros y un peso corporal de 75 kilogramos, mientras que los 2 restantes proceden de una mujer con una estatura de 154 centímetros y un peso de 48 kilogramos.

Para la cámara se han recopilado datos de un total de 10 sesiones de entrenamiento, 9 de estos provienen igualmente de un varón con una estatura de 178 centímetros y un peso corporal de 75 kilogramos, mientras que el entrenamiento restante procede de una mujer con una estatura de 154 centímetros y un peso de 48 kilogramos.

Posteriormente se estudiará la eficacia del modelo en función de las características corporales del usuario.

6.6 Entrenamiento y reconocimiento

Para empezar el entrenamiento y reconocimiento también hay que seleccionar los ejercicios que se van a recoger, y por tanto, intentar reconocer. Se han elegido: sentadillas, flexiones, dominadas y levantamientos laterales. No se han elegido más ejercicios para no complicar el proceso de toma de datos. Los ejercicios han sido escogidos porque trabajan distintas partes del cuerpo.

Existen varias decisiones de diseño de la red neuronal que hay que realizar: el número de periodos (*epochs*), la tasa de aprendizaje (*learning rate*) y las capas (*layers*). Para el proyecto, se han realizado pruebas sobre distintos valores para obtener el modelo con los mejores resultados.

En cada periodo la red neuronal realiza una fase de evaluación y actualiza los valores de las neuronas, posteriormente, utiliza otro grupo distinto al de entrenamiento para validar esos resultados para a partir de esa validación cambiar la tasa de aprendizaje en caso de que sea necesario.

Cuando la fase de entrenamiento ha finalizado al completo, se realiza la fase de evaluación, en la que se utiliza otro grupo de datos distinto al de evaluación y validación para evaluar el resultado final de la red neuronal. Es importante que los grupos de datos sean independientes unos de los otros.

Siempre que finaliza una fase de entrenamiento, validación o evaluación se calcula la precisión y el error del entrenamiento y se muestra por la terminal para ver la evolución de la red neuronal.

En cuanto a los sensores, los resultados obtenidos después de 25 periodos son los de la Figura 23.

```
Validation: Accuracy: 1.0000, Loss: 0.3074: 100%|
Epoch [11/25], Accuracy: 0.6786, Loss: 0.7737: 100%|
Validation: Accuracy: 1.0000, Loss: 0.3064: 100%|
Epoch [12/25], Accuracy: 0.9048, Loss: 0.2801: 100%|
Validation: Accuracy: 0.6667, Loss: 0.7055: 100%|
Epoch 12: reducing learning rate of group 0 to 1.0000e-04.
Epoch [13/25], Accuracy: 0.8095, Loss: 0.3508: 100%|
Validation: Accuracy: 1.0000, Loss: 0.1126: 100%|
Epoch [14/25], Accuracy: 0.9881, Loss: 0.0849: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0975: 100%|
Epoch [15/25], Accuracy: 1.0000, Loss: 0.0544: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0609: 100%|
Epoch [16/25], Accuracy: 1.0000, Loss: 0.0363: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0415: 100%|
Epoch [17/25], Accuracy: 1.0000, Loss: 0.0269: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0309: 100%|
Epoch [18/25], Accuracy: 1.0000, Loss: 0.0211: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0261: 100%|
Epoch [19/25], Accuracy: 1.0000, Loss: 0.0173: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0216: 100%|
Epoch [20/25], Accuracy: 1.0000, Loss: 0.0143: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0206: 100%|
Epoch [21/25], Accuracy: 1.0000, Loss: 0.0115: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0161: 100%|
Epoch [22/25], Accuracy: 1.0000, Loss: 0.0099: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0126: 100%|
Epoch [23/25], Accuracy: 1.0000, Loss: 0.0083: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0105: 100%|
Epoch 23: reducing learning rate of group 0 to 1.0000e-05.
Epoch [24/25], Accuracy: 1.0000, Loss: 0.0075: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0106: 100%|
Epoch [25/25], Accuracy: 1.0000, Loss: 0.0074: 100%|
Validation: Accuracy: 1.0000, Loss: 0.0106: 100%|
```

Figura 23. Entrenamiento de red neuronal de sensores

Se consigue un 100% de precisión desde el periodo 15 tanto en el entrenamiento como en la validación. Una vez finalizado este proceso, se hace la evaluación, cuyos resultados

se puede observar en la Figura 24. También se consigue un 100% de precisión en la evaluación, por lo que podríamos decir que la red neuronal de los sensores consigue un buen rendimiento y es posible de reconocer el ejercicio en la mayoría de los casos.

```

for label: tensor([1]) predicted: tensor([2])
Test: Accuracy: 1.0000, Loss: 0.0408: 83%|
for label: tensor([0]) predicted: tensor([0])
Test: Accuracy: 1.0000, Loss: 0.0434: 88%|
for label: tensor([1]) predicted: tensor([1])
Test: Accuracy: 1.0000, Loss: 0.0477: 92%|
for label: tensor([1]) predicted: tensor([1])
Test: Accuracy: 1.0000, Loss: 0.0509: 96%|
for label: tensor([2]) predicted: tensor([2])
Test: Accuracy: 1.0000, Loss: 0.0487: 100%|

```

Figura 24. Evaluación de red neuronal de sensores

En cuanto a la cámara, los resultados obtenidos después de 25 periodos son los de la Figura 25.

```

Validation: Accuracy: 0.7500, Loss: 0.3800: 100%|
Epoch [17/30], Accuracy: 0.7927, Loss: 0.4613: 100%|
Validation: Accuracy: 0.7500, Loss: 0.3638: 100%|
Epoch [18/30], Accuracy: 0.7317, Loss: 0.4481: 100%|
Validation: Accuracy: 0.7500, Loss: 0.3598: 100%|
Epoch [19/30], Accuracy: 0.8293, Loss: 0.4360: 100%|
Validation: Accuracy: 0.7500, Loss: 0.3490: 100%|
Epoch [20/30], Accuracy: 0.7927, Loss: 0.4333: 100%|
Validation: Accuracy: 1.0000, Loss: 0.3296: 100%|
Epoch [21/30], Accuracy: 0.8659, Loss: 0.4079: 100%|
Validation: Accuracy: 1.0000, Loss: 0.3110: 100%|
Epoch [22/30], Accuracy: 0.8171, Loss: 0.4098: 100%|
Validation: Accuracy: 0.7500, Loss: 0.3274: 100%|
Epoch [23/30], Accuracy: 0.8049, Loss: 0.3914: 100%|
Validation: Accuracy: 1.0000, Loss: 0.2803: 100%|
Epoch 23: reducing learning rate of group 0 to 1.0000e-05.
Epoch [24/30], Accuracy: 0.9024, Loss: 0.3537: 100%|
Validation: Accuracy: 1.0000, Loss: 0.2755: 100%|
Epoch [25/30], Accuracy: 0.9390, Loss: 0.3472: 100%|
Validation: Accuracy: 1.0000, Loss: 0.2708: 100%|

```

Figura 25. Entrenamiento de red neuronal de cámara

En cuanto a la validación se consigue el 100% de precisión, pero en el conjunto del entrenamiento se consigue un 93,90%, sigue siendo una muy buena precisión capaz de reconocer la gran mayoría de ejercicios.

La evaluación, como se puede observar en la Figura 26, consigue un 100% de precisión, esto puede hacer pensar también que en el conjunto de datos de la evaluación existe la posibilidad de que hubiese datos erróneos.

```

Test: Accuracy: 1.0000, Loss: 0.2593:
tensor([1]) =? tensor([1])
Test: Accuracy: 1.0000, Loss: 0.3290:
tensor([4]) =? tensor([4])
Test: Accuracy: 1.0000, Loss: 0.3549:
tensor([4]) =? tensor([4])
Test: Accuracy: 1.0000, Loss: 0.3754:
tensor([4]) =? tensor([4])
Test: Accuracy: 1.0000, Loss: 0.3904:
tensor([2]) =? tensor([2])
Test: Accuracy: 1.0000, Loss: 0.3577:
tensor([3]) =? tensor([3])
Test: Accuracy: 1.0000, Loss: 0.3296:

```

Figura 26. Evaluación de red neuronal de cámara

6.7 Integración en la aplicación

En esta sección se expone cómo, una vez finalizado el estudio y análisis basado en técnicas de ciencia de datos y tras obtener unos resultados positivos, en la aplicación se

han integrado estas técnicas, ofreciendo así al usuario la nueva funcionalidad de reconocimiento directamente en la aplicación móvil en una nueva iteración.

Para el reconocimiento, se toman los datos en la aplicación móvil y se envían al servidor, donde son procesados y se usa el mejor modelo obtenido para detectar el ejercicio y devolverlo como respuesta a la aplicación móvil.

6.7.1 Cambios en el servidor

En el servidor se ha implementado un nuevo *endpoint* que recibe los datos de los sensores y realiza el reconocimiento. Para implementar este *endpoint* se ha creado un directorio en la carpeta «utils» con el modelo de los sensores, además se ha creado un fichero «predict_sensors.py» con la función encargada de reconocer el ejercicio (Figura 27).

```
def predict_data(input_acc, input_gyr):
    input_acc = torch.from_numpy(input_acc).to(device, non_blocking=True)
    input_gyr = torch.from_numpy(input_gyr).to(device, non_blocking=True)
    acc_rows = input_acc.size(0)
    gyr_rows = input_gyr.size(0)

    sensor_transform = transforms.Compose([Resample(target_length=c.SENSOR_WINDOW_LENGTH)])

    final_acc_data = torch.as_tensor(sensor_transform(
        input_acc[0:acc_rows-1, 1:].T), dtype=torch.float).to(device, non_blocking=True)

    final_gyr_data = torch.as_tensor(sensor_transform(
        input_gyr[0:gyr_rows-1, 1:].T), dtype=torch.float).to(device, non_blocking=True)

    final_acc_data = final_acc_data.unsqueeze(0)
    final_gyr_data = final_gyr_data.unsqueeze(0)

    # load and prepare the model
    accelerometer_model = ConvAutoencoder(return_embeddings=True).to(device, non_blocking=True)
    gyroscope_model = ConvAutoencoder(return_embeddings=True).to(device, non_blocking=True)

    multimodal_ae_f_in = 240
    multimodal_ae_model = MyMultimodalAutoencoder(f_in=multimodal_ae_f_in, accelerometer=accelerometer_model, gyroscope=gyroscope_model,
        layers=c.AE_LAYERS, hidden_units=c.AE_HIDDEN_UNITS, f_embedding=c.EMBEDDING_UNITS, dropout=c.AE_DROPOUT,
        return_embeddings=True).to(device, non_blocking=True)

    f_in = c.EMBEDDING_UNITS
    model = MultimodalFcClassifier(f_in=f_in, num_classes=c.NUM_CLASSES, multimodal_ae_model=multimodal_ae_model,
        layers=c.LAYERS, hidden_units=c.HIDDEN_UNITS,
        dropout=c.DROPOUT).to(device, non_blocking=True)

    checkpoint = torch.load("models/model.pth", map_location=device)
    model.load_state_dict(checkpoint['model_state_dict'])

    # make the prediction
    output = model(final_acc_data, final_gyr_data)
    predicted_class = torch.argmax(output, dim=1)
    return reverse_mapping.get(predicted_class.item())
```

Figura 27. Función que recibe los datos de los sensores y predice el ejercicio realizado

Además, se ha implementado el *endpoint* para el reconocimiento de ejercicio a través de un vídeo enviado desde la aplicación móvil. El proceso para el reconocimiento con un vídeo es diferente al de los sensores, ya que antes de usar el modelo es necesario procesar el vídeo.

Por esto, primero se utiliza una función auxiliar que devuelve los puntos clave anatómicos que se indicaron en el apartado del estudio de la ciencia de datos (Figura 28).

Y posteriormente se puede utilizar el modelo para reconocer los datos que recibimos de la función anterior (Figura 29Figura 28).

```

def get_points_from_video(video):
    mp_pose = mp.solutions.pose
    cap = cv2.VideoCapture(video)
    selected_landmark_indices = {0, 11, 12, 13, 14, 15, 16, 23, 24, 25, 26, 27, 28}
    pose_landmarks_list = []

    with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
        while cap.isOpened():
            ret, frame = cap.read()

            if not ret:
                break

            image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            image.flags.writeable = False

            results = pose.process(image)

            image.flags.writeable = True
            image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

            video_frame_number = int(cap.get(cv2.CAP_PROP_POS_FRAMES))
            if results.pose_landmarks:
                landmarks_without_visibility = []
                landmarks_without_visibility.append(video_frame_number)
                for i, landmark in enumerate(results.pose_landmarks.landmark):
                    if i in selected_landmark_indices:
                        if landmark.visibility > 0.5:
                            landmarks_without_visibility.append({
                                'x': landmark.x,
                                'y': landmark.y,
                                'z': landmark.z,
                            })
                        else:
                            landmarks_without_visibility.append({
                                'x': 0,
                                'y': 0,
                                'z': 0,
                            })
                pose_landmarks_list.append(landmarks_without_visibility)

    pose_landmarks_array = np.array(pose_landmarks_list)
    final_array = reshape_data(pose_landmarks_array)

```

Figura 28. Función para obtener los puntos clave a partir de un vídeo

```

def predict_camera_data(data):
    data_transforms_skeleton = transforms.Compose([Unit()])
    data = torch.as_tensor(data_transforms_skeleton(
        data[:, 0:300, 1:]), dtype=torch.float)

    data = data.unsqueeze(0)

    pose_model = ConvAutoencoder(return_embeddings=True).to(device, non_blocking=True)
    multimodal_ae_f_in = 336
    multimodal_ae_model = MyMultimodalAutoencoder(f_in=multimodal_ae_f_in, pose=pose_model,
        layers=c.AE_LAYERS, hidden_units=c.AE_HIDDEN_UNITS, f_embedding=c.EMBEDDING_UNITS, dropout=c.AE_DROPOUT,
        return_embeddings=True).to(device, non_blocking=True)

    f_in = c.EMBEDDING_UNITS
    model = MultimodalFcClassifier(f_in=f_in, num_classes=c.NUM_CLASSES, multimodal_ae_model=multimodal_ae_model,
        layers=c.LAYERS, hidden_units=c.HIDDEN_UNITS,
        dropout=c.DROPOUT).to(device, non_blocking=True)

    checkpoint = torch.load("models/model_camera.pth", map_location=device)
    model.load_state_dict(checkpoint['model_state_dict'])

    output = model(data)
    predicted_class = torch.argmax(output, dim=1)
    return reverse_mapping.get(predicted_class.item())

```

Figura 29. Función que recibe los datos de la cámara y reconoce el ejercicio realizado

6.7.2 Cambios en la aplicación móvil

Para la aplicación móvil solo ha sido necesario cambiar la pantalla para añadir o modificar un nuevo entrenamiento, permitiendo añadir un ejercicio a través de reconocimiento.

Cuando se presiona sobre el botón de reconocimiento se abre una pantalla para elegir la fuente de los datos, sensores o cámara. Una vez seleccionado se procederá a la recogida a los mismos.

Cuando el usuario presione el botón para finalizar el reconocimiento, se enviará directamente los datos al servidor y se esperarán los resultados (Figura 30).

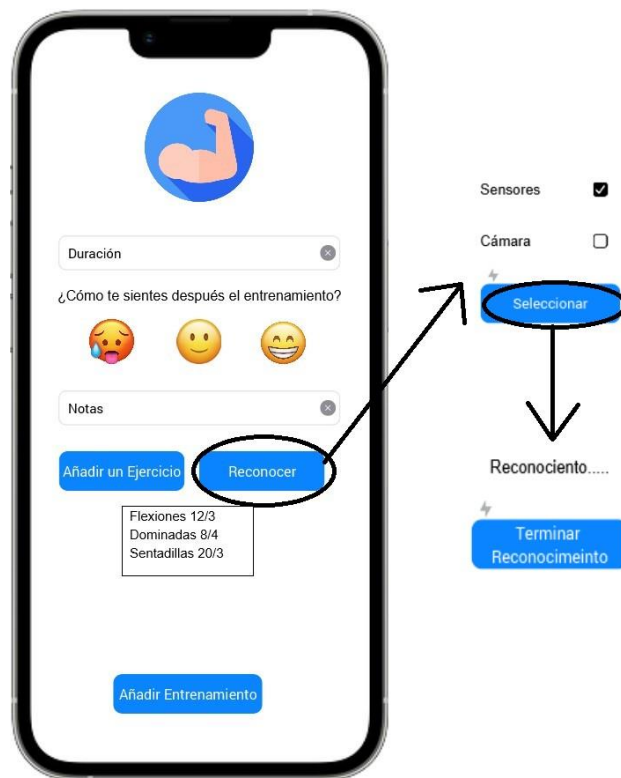


Figura 30. Nueva pantalla para añadir un ejercicio con reconocimiento

6.8 Pruebas en la aplicación

Se han realizado pruebas de detección una vez se ha implementado el reconocimiento en la aplicación y se han recopilado los resultados. Para ello se ha probado para cada fuente de información cada ejercicio en 4 ocasiones y se ha observado si el reconocimiento era exitoso o no.

Ejercicios	Sentadillas	Flexiones	Dominadas	Laterales
Sentadillas	100%	0%	0%	0%
Flexiones	0%	100%	0%	0%
Dominadas	0%	0%	100%	0%
Laterales	0%	0%	0%	100%

Tabla 4. Matriz de reconocimiento de los sensores

Ejercicios	Sentadillas	Flexiones	Dominadas	Laterales
Sentadillas	75%	0%	0%	0%
Flexiones	0%	100%	0%	0%
Dominadas	25%	0%	100%	0%
Laterales	0%	0%	0%	100%

Tabla 5. Matriz de reconocimiento de la cámara

Los resultados revelan un buen desempeño, ya que los modelos lograron identificar correctamente el 100% de los ejercicios, a excepción de las sentadillas, al emplear el

reconocimiento por cámara. Estas pruebas se llevaron a cabo en condiciones ideales, con la misma persona realizando la mayoría de las capturas de datos desde un ángulo constante. Para evaluar la versatilidad del modelo, se presentarán más pruebas en los próximos apartados.

6.8.1 Tiempo de detección

En este apartado se compara el tiempo que tarda el proceso de reconocimiento completo para distintos tamaños de entrada con las distintas fuentes de datos. En la Tabla 6 se puede observar la gran diferencia que existe entre reconocer un ejercicio a partir de los sensores o hacerlo a través de la cámara.

Tiempo/Fuente	Sensores	Cámara
15 segundos	0,8 segundos	19 segundos
30 segundos	1,2 segundos	37,3 segundos
1 minuto	1,5 segundos	1 minuto y 15 segundos
1 minuto y 30 segundos	1,9 segundos	1 minuto y 50 segundos
2 minutos	2,1 segundos	2 minutos y 21 segundos

Tabla 6. Tiempo para reconocer ejercicio según la fuente de datos

El tiempo que tarda la detección con los sensores es apenas insignificante, en cambio, para la cámara, tarda más en detectar el ejercicio que la duración del propio vídeo, esto hace que la opción de la cámara poco usable.

Por tanto, en el apartado del tiempo, los sensores son claramente una mejor opción. Esto se debe al tiempo que tarda el vídeo en enviarse al servidor y posteriormente ser procesado, ya que el tiempo de detección después de obtener los puntos es comparable al de los sensores.

Para mejorar el tiempo de detección de la cámara se podría implementar en la aplicación móvil la detección de puntos clave anatómicos en tiempo real y directamente mandarlos al servidor. Esto además mejoraría el rendimiento del servidor, que no tiene que realizar ninguna transformación a los datos recibidos.

6.8.2 Reconocimiento desde diversos ángulos

Hasta ahora todos los datos se han recogido de manera que el cuerpo estaba posicionado mirando hacia el este, y en el caso de la cámara estaba siempre frontal al cuerpo.

En esta sección se va a probar la detección de ejercicios teniendo en cuenta rotaciones de 0º, 90º, 180º y 270º para comprobar si el reconocimiento es posible o si es necesario la toma de más datos en distintos ángulos para los distintos ejercicios.

Ángulo/Fuente	Sensores	Cámara
0º	100%	100%
90º	100%	50%
180º	100%	0%
270º	100%	0%

Tabla 7. Porcentaje de detección exitosa al cambiar el ángulo para sentadillas

Ángulo/Fuente	Sensores	Cámara
0º	100%	100%
90º	100%	100%
180º	100%	0%
270º	100%	75%

Tabla 8. Porcentaje de detección exitosa al cambiar el ángulo para flexiones

Ángulo/Fuente	Sensores	Cámara
0º	100%	100%
90º	100%	25%
180º	75%	0%
270º	100%	25%

Tabla 9. Porcentaje de detección exitosa al cambiar el ángulo para dominadas

Ángulo/Fuente	Sensores	Cámara
0º	100%	100%
90º	100%	50%
180º	100%	25%
270º	100%	100%

Tabla 10. Porcentaje de detección exitosa al cambiar el ángulo para laterales

Los resultados son favorables para la detección con sensores, pero en cuanto a la cámara los resultados no son los deseados, para la mayoría de los ejercicios solo consigue un buen rendimiento si se realizan las tomas desde el mismo punto.

6.8.3 Reconocimiento en cuerpos diferentes

En esta sección, se ha realizado el reconocimiento en todos los ejercicios en distintas personas con características distintas, esto permitirá observar si el modelo es flexible para cualquier persona con distintas características físicas.

Las personas que han participado junto con sus características quedan plasmadas en la Tabla 11.

Persona/Característica	Altura (cm)	Peso (kg)	Edad
Persona 1	178	75	21
Persona 2	174	80	52
Persona 3	158	56	52
Persona 4	154	54	21

Tabla 11. Características de las personas que han probado el reconocimiento

Cada una de las personas ha realizado 2 veces cada ejercicio, por tanto, los valores de las tablas resultantes para cada ejercicio podrán tomar los siguientes valores: 0%, 50%, 100%.

Ejercicio/Persona	Persona 1	Persona 2	Persona 3	Persona 4
Sentadillas	100%	100%	100%	100%
Flexiones	100%	100%	75%	75%
Dominadas	100%	100%	100%	75%
Laterales	100%	100%	100%	100%

Tabla 12. Resultados del reconocimiento con sensores en cuerpos con características distintas

Ejercicio/Persona	Persona 1	Persona 2	Persona 3	Persona 4
Sentadillas	100%	100%	75%	75%
Flexiones	100%	100%	75%	50%
Dominadas	100%	100%	75%	75%
Laterales	100%	100%	100%	100%

Tabla 13. Resultados del reconocimiento con la cámara en cuerpos con características distintas

Los resultados son prometedores para las personas 1 y 2, ya que comparten similitudes con la persona de referencia en la recopilación de datos. Sin embargo, las personas 3 y 4, que difieren en características físicas respecto a la persona de referencia, muestran resultados menos favorables. Una posible solución a este problema sería enriquecer el conjunto de datos de entrenamiento con información de personas que presenten una variedad más amplia de características físicas.

7

Conclusiones

En esta sección, se presentan las conclusiones derivadas tanto del proceso de implementación de la aplicación como del desarrollo y entrenamiento de la red neuronal convolucional, junto con las dificultades encontradas durante estas fases.

Todos los requisitos planteados para la aplicación han sido satisfechos, incluyendo la capacidad de reconocimiento de ejercicios mediante la cámara o los sensores. La aplicación no solo registra las sesiones de entrenamiento de los usuarios y los ejercicios realizados, sino que también ofrece la visualización de gráficas para monitorizar el progreso de los entrenamientos.

Inicialmente, se abordaron minuciosamente los requisitos esenciales, los casos de uso y el diseño de la aplicación, permitiendo así la adaptación de la estructura organizativa a las necesidades reales del proyecto. La aplicación de esta metodología resultó fundamental para la planificación efectiva de las iteraciones; su omisión habría complicado significativamente dicho proceso.

En una etapa inicial, se consideró la opción de incorporar algún dispositivo inteligente que facilitara la adquisición de datos. Sin embargo, se constató que *React Native* aún carece de bibliotecas que simplifiquen este proceso, posiblemente debido a su condición de herramienta relativamente reciente (lanzada en 2015). Además, aunque existen bibliotecas utilizables en *Android* o *iOS*, no se encuentran disponibles en ambas plataformas, lo que implica una elección cuidadosa para garantizar el funcionamiento adecuado de la aplicación en ambos sistemas operativos.

En relación con la transmisión de vídeos de entrenamiento para su posterior detección, se observó un aumento considerable en el tamaño de los archivos conforme aumentaba la duración. Se contempló la opción de realizar la extracción de puntos clave directamente en la aplicación de forma local utilizando *Python*. No obstante, no se

identificó un método sencillo y funcional para llevar a cabo este proceso, por lo que se optó por enviar el vídeo directamente al servidor. Este enfoque, sin embargo, se traduce en un tiempo de detección considerablemente mayor cuando se trata de vídeos.

En cuanto al reconocimiento, se obtuvieron resultados positivos en términos de la información obtenida, logrando un porcentaje significativo de precisión en el reconocimiento. A pesar de estos logros, aún existe margen para mejoras, especialmente en la obtención de datos suficientes para que la cámara pueda detectar el ejercicio sin importar su ubicación o la persona que lo realiza. Aunque el reconocimiento actual no alcanza el nivel necesario para convertirse en una herramienta esencial en la rutina diaria de entrenamiento, proporciona una base sólida sobre la cual establecer un plan de mejora.

7.1 Posibles líneas futuras

A lo largo del desarrollo del proyecto, han surgido posibles líneas de desarrollo futuro que podrían ser implementadas en caso de que el proyecto avance. A continuación, se presentan las más relevantes:

- **Ampliación de gráficas de visualización de resultados:** Aunque se han creado diversos tipos de gráficas, existen otros aspectos interesantes que el sistema podría ofrecer al usuario. Se podría explorar la posibilidad de proporcionar recomendaciones sobre su progreso, indicando si es demasiado rápido o lento, y sugiriendo posibles mejoras en el entrenamiento. Esta función adicional podría proporcionar una visión más completa y personalizada del progreso del usuario.
- **Reconocimiento de repeticiones y series:** Además del reconocimiento de ejercicios, sería beneficioso implementar el reconocimiento de series y repeticiones. Esto permitiría al usuario evitar detener el reconocimiento en cada serie, sino únicamente al finalizar el ejercicio completo. En un futuro, se podría incluso intentar reconocer todo el entrenamiento con una única toma, proporcionando una experiencia más fluida y eficiente.
- **Contador de pasos:** No solo los ejercicios de fuerza son cruciales para mantenerse en forma, se podría considerar la incorporación de un contador de pasos en la aplicación. Este contador gestionaría la cantidad de pasos que el usuario realiza diariamente, semanalmente, etc. Además, podría ofrecer recomendaciones relacionadas con la actividad física basadas en el recuento de pasos, proporcionando así una perspectiva más completa del estilo de vida activo del usuario.
- **Funcionalidad de comunidad:** Introducir características sociales que fomenten la interacción con el usuario también podría ser interesante, esto podría incluir la capacidad de compartir logros, participar en desafíos y competir de manera amistosa, lo que podría aumentar la motivación y el compromiso de los usuarios.

- **Mejorar la interfaz de usuario:** Refinar la experiencia de usuario, especialmente en la sección de historial de entrenamientos, donde la interfaz podría resultar poco intuitiva para algunos usuarios. Una posible mejora sería la implementación de un buscador por calendario, facilitando la navegación hacia días anteriores de manera más eficiente.
- **Asesoramiento nutricional:** Reconociendo que la salud no se limita al ejercicio, sino que también está intrínsecamente ligada a la alimentación, sería beneficioso agregar una sección dedicada a asesoramiento nutricional. Esto permitiría que la aplicación evolucione más allá de ser simplemente una plataforma de ejercicio, transformándola en una herramienta integral para la salud y el bienestar.

Apéndice A

Manual de instalación

1.Requisitos

Para una instalación exitosa se necesitarán las siguientes aplicaciones previamente instaladas en su ordenador con sistema operativo Windows:

- **Docker:** Es una plataforma de software que permite a los desarrolladores empaquetar, distribuir y ejecutar aplicaciones dentro de contenedores seguros.
- **Node:** Se trata de un entorno de ejecución de *JavaScript* del lado del servidor, es necesario a la hora de ejecutar la aplicación móvil.
- **Python:** Es un lenguaje de programación de alto nivel, interpretado y multipropósito y se necesita para ejecutar el servidor.
- **Conda:** Se trata de un sistema de gestión de paquetes, entornos y dependencias multiplataforma, en este proyecto será muy útil para instalar todas las dependencias que necesita el servidor para funcionar.
- **Dispositivo Android o iOS:** Necesario para ejecutar la aplicación, también se puede usar un emulador de Android para ordenador, pero no se podrán usar las funciones de reconocimiento.

2.Despliegue de la base de datos

Para desplegar la base de datos se hará uso de *Docker*, para asegurarnos que se ejecuta en un entorno seguro. Para desplegarlo hay que tener iniciada la aplicación de *Docker* y posteriormente situarse en la carpeta «docker», desde aquí necesitaremos abrir una terminal y ejecutar el siguiente comando: «docker compose up --build».

Con este comando se descargará la imagen de *mysql* y además se añadirá la base de datos necesaria para la aplicación (Figura 31).

```

C:\Users\arion\OneDrive\Escritorio\docker>docker-compose up --build
[+] Running 11/11
 ✓ mysql 10 layers [#####] 0B/0B Pulled 28.1s
   ✓ 558b7d69a2e5 Pull complete 12.4s
   ✓ 2cb5a921059e Pull complete 0.5s
   ✓ b85878fb9bb2 Pull complete 0.7s
   ✓ d16f3fd26a82 Pull complete 2.1s
   ✓ afd51b5329cb Pull complete 1.4s
   ✓ 374d2f7f3267 Pull complete 2.3s
   ✓ 4ea1bb2c9574 Pull complete 13.9s
   ✓ 1c9054053605 Pull complete 2.9s
   ✓ d79cd2da03be Pull complete 11.4s
   ✓ e3a1aa788d17 Pull complete 11.9s
[+] Building 0.0s (0/0)
                                docker:default
[+] Running 2/2
 ✓ Network docker_default Created 0.0s
 ✓ Container tfg_database Created 2.0s
Attaching to tfg_database
tfg_database | 2024-02-03 11:00:43+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.3.0-1.el8 started.

```

Figura 31. Despliegue base de datos en Docker

3. Despliegue del servidor

Para el despliegue del servidor hay que situarse en el directorio «*server*» y lo primero que hay que realizar es la instalación de las dependencias a través de *Conda* para ello habrá ejecutar el siguiente comando: «*conda env create -f environment.yml*». Este proceso puede tardar unos minutos (Figura 32).

```

C:\Users\arion\OneDrive\Documentos\TFG\FlaskServer>conda env create -f enviroment.y
Channels:
 - defaults
 - anaconda
 - conda-forge
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Installing pip dependencies: / Ran pip subprocess with arguments:
['C:\Users\arion\anaconda3\envs\serverr\python.exe', '-m', 'pip', 'install',
Collecting bcrypt==4.0.1 (from -r C:\Users\arion\OneDrive\Documentos\TFG\FlaskServe
Collecting blinker==1.6.3 (from -r C:\Users\arion\OneDrive\Documentos\TFG\FlaskServ
Collecting cachelib==0.10.2 (from -r C:\Users\arion\OneDrive\Documentos\TFG\FlaskSe
Collecting click==8.1.7 (from -r C:\Users\arion\OneDrive\Documentos\TFG\FlaskServer
Collecting colorama==0.4.6 (from -r C:\Users\arion\OneDrive\Documentos\TFG\FlaskSer
Collecting flask==3.0.0 (from -r C:\Users\arion\OneDrive\Documentos\TFG\FlaskServer
Collecting flask-bcrypt==1.0.1 (from -r C:\Users\arion\OneDrive\Documentos\TFG\Flas
Collecting flask-cors==4.0.0 (from -r C:\Users\arion\OneDrive\Documentos\TFG\FlaskS
Collecting flask-session==0.5.0 (from -r C:\Users\arion\OneDrive\Documentos\TFG\Fla
Collecting flask-sqlalchemy==3.1.1 (from -r C:\Users\arion\OneDrive\Documentos\TFG\
Collecting greenlet==3.0.1 (from -r C:\Users\arion\OneDrive\Documentos\TFG\FlaskSer

```

Figura 32. Creación del ambiente Conda

Una vez finalizado podemos activar el *environment* (ambiente), en una terminal con permisos de administrador con el comando: «*conda activate server*» y podremos iniciar el servidor con el comando «*python main.py*». Es importante haber realizado previamente el despliegue de la base de datos y tener el contenedor en funcionamiento (Figura 33).

```
C:\Users\arion\OneDrive\Documentos\TFG\FlaskServer>conda activate server

(server) C:\Users\arion\OneDrive\Documentos\TFG\FlaskServer>python main.py
2024-02-03 12:47:26,008 INFO sqlalchemy.engine.Engine SELECT DATABASE()
2024-02-03 12:47:26,010 INFO sqlalchemy.engine.Engine [raw sql] ()
2024-02-03 12:47:26,014 INFO sqlalchemy.engine.Engine SELECT @@sql_mode
2024-02-03 12:47:26,014 INFO sqlalchemy.engine.Engine [raw sql] ()
2024-02-03 12:47:26,016 INFO sqlalchemy.engine.Engine SELECT @@lower_case_table_names
2024-02-03 12:47:26,016 INFO sqlalchemy.engine.Engine [raw sql] ()
2024-02-03 12:47:26,019 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2024-02-03 12:47:26,020 INFO sqlalchemy.engine.Engine DESCRIBE `tf`.`user`
2024-02-03 12:47:26,020 INFO sqlalchemy.engine.Engine [raw sql] ()
2024-02-03 12:47:26,082 INFO sqlalchemy.engine.Engine DESCRIBE `tf`.`user_details`
2024-02-03 12:47:26,083 INFO sqlalchemy.engine.Engine [raw sql] ()
2024-02-03 12:47:26,087 INFO sqlalchemy.engine.Engine DESCRIBE `tf`.`day`
2024-02-03 12:47:26,088 INFO sqlalchemy.engine.Engine [raw sql] ()
2024-02-03 12:47:26,090 INFO sqlalchemy.engine.Engine DESCRIBE `tf`.`exercise`
2024-02-03 12:47:26,090 INFO sqlalchemy.engine.Engine [raw sql] ()
2024-02-03 12:47:26,094 INFO sqlalchemy.engine.Engine DESCRIBE `tf`.`training_exercises`
2024-02-03 12:47:26,094 INFO sqlalchemy.engine.Engine [raw sql] ()
2024-02-03 12:47:26,097 INFO sqlalchemy.engine.Engine COMMIT
```

Figura 33. Despliegue del servidor

3. Despliegue de la aplicación móvil

Para configurar la aplicación móvil, primero ubíquese en el directorio «SportApp» y acceda al archivo `.env`. Este archivo contiene la dirección IP del servidor web, que en este caso es la misma que la de su ordenador. Debe actualizar esta dirección con la suya. Para encontrar su dirección IP en Windows, abra una terminal y ejecute el comando «`ipconfig`». Busque la dirección `IPv4` asociada con su adaptador de `WiFi`.

Una vez hecho esto, ejecute el comando «`npm install --force`». Este comando instalará todas las dependencias necesarias para que el proyecto funcione correctamente. Una vez completada la instalación, puede iniciar la aplicación utilizando el comando «`npm start`». La primera vez que la aplicación se despliega, deberá crear una cuenta en Expo e iniciar sesión durante ese proceso.

```
C:\Users\arion\OneDrive\Documentos\TFG\SportApp>npm start
> awesomeproject@1.0.0 start
> expo start

Starting project at C:\Users\arion\OneDrive\Documentos\TFG\SportApp
Starting Metro Bundler

<img alt="QR code for Expo Go" data-bbox="371 698 484 784"/>

> Metro waiting on expo://192.168.1.143:8081
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Using Expo Go
> Press s | switch to development build

> Press a | open Android
> Press w | open web

> Press j | open debugger
> Press r | reload app
> Press m | toggle menu
> Press o | open project code in your editor

> Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
```

Figura 34. Despliegue de la aplicación móvil

4. Iniciar la aplicación en nuestro dispositivo

Para poder ejecutar la aplicación en un dispositivo *Android* o *iOS* hay que instalar la aplicación *Expo Go*, habrá que iniciar sesión con la misma cuenta con la que se inició sesión en el ordenador para poder ver la opción de nuestra aplicación. En la pantalla inicial de la aplicación aparecerá la aplicación si se han realizado todos los pasos correctamente (Figura 35).

Existe un usuario creado previamente con datos de prueba con usuario «*tfg_database*» y contraseña «*tfg123123*», también se puede crear un nuevo como se explica en el manual de usuario.

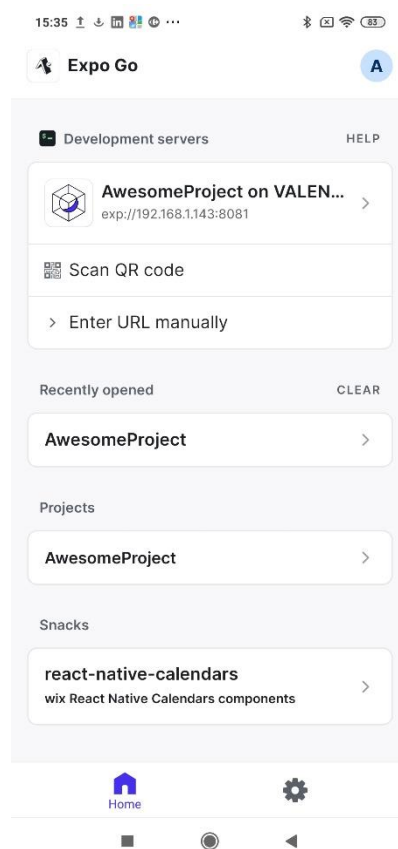


Figura 35. Selección de la aplicación en la aplicación móvil

Apéndice B

Manual de usuario

Este es el manual de usuario donde se detallan las acciones que se pueden realizar en la aplicación de gestión de entrenamientos. Las imágenes provienen de un dispositivo móvil con sistema operativo *Android*.

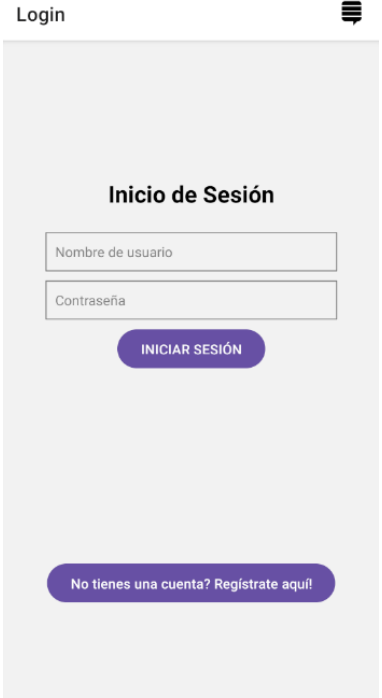
Antes de empezar con el manual de usuario, cabe destacar que las acciones relacionadas con el reconocimiento en emuladores virtuales no funcionan por razones obvias: los sensores son estáticos y no son capaces de usar la cámara. El resto de las funcionalidades sí están disponibles.

1. Identificación de usuario

1.1 Inicio de sesión

Para acceder a la plataforma, si aún no dispone de una cuenta, deberá crear una a través del proceso de registro de usuario. En caso de ya contar con una cuenta, puede proceder al inicio de sesión.

Una vez en la aplicación, para identificarse, complete los campos «Nombre de Usuario» y «Contraseña» como se indica en la Figura 36. Si los datos son ingresados correctamente y la sesión es válida, accederá a la pantalla inicial, tal como se muestra en la Figura 38.



The screenshot shows a mobile application interface for logging in. At the top left, the word 'Login' is displayed. At the top right, there is a hamburger menu icon. The main heading is 'Inicio de Sesión'. Below this, there are two text input fields: the first is labeled 'Nombre de usuario' and the second is labeled 'Contraseña'. Underneath these fields is a purple button with the text 'INICIAR SESIÓN'. At the bottom of the screen, there is another purple button with the text 'No tienes una cuenta? Regístrate aquí!'.

Figura 36. Pantalla de inicio de sesión

1.2 Registro de nuevo usuario

En caso de no poseer una cuenta, será necesario dirigirse a la pantalla de registro. Desde la pantalla de inicio de sesión, seleccione la opción «¿No tienes una cuenta? ¡Regístrate aquí!» para ser redirigido a la pantalla de registro, como se ilustra en la Figura 37.

Para completar el registro con éxito, es obligatorio llenar todos los campos marcados con un asterisco. En caso contrario, aparecerá un mensaje de error en la parte superior.

Además, para garantizar la seguridad del usuario, se aplican las siguientes medidas en la contraseña: debe tener al menos 8 caracteres, incluir al menos un dígito, una mayúscula, una minúscula y al menos un símbolo. También se verifica que el email y el número de teléfono tengan el formato correcto.

Una vez que el formulario se completa correctamente, se mostrará un mensaje de registro exitoso, como se indica en la Figura 37, y podrá iniciar sesión desde la pantalla correspondiente.

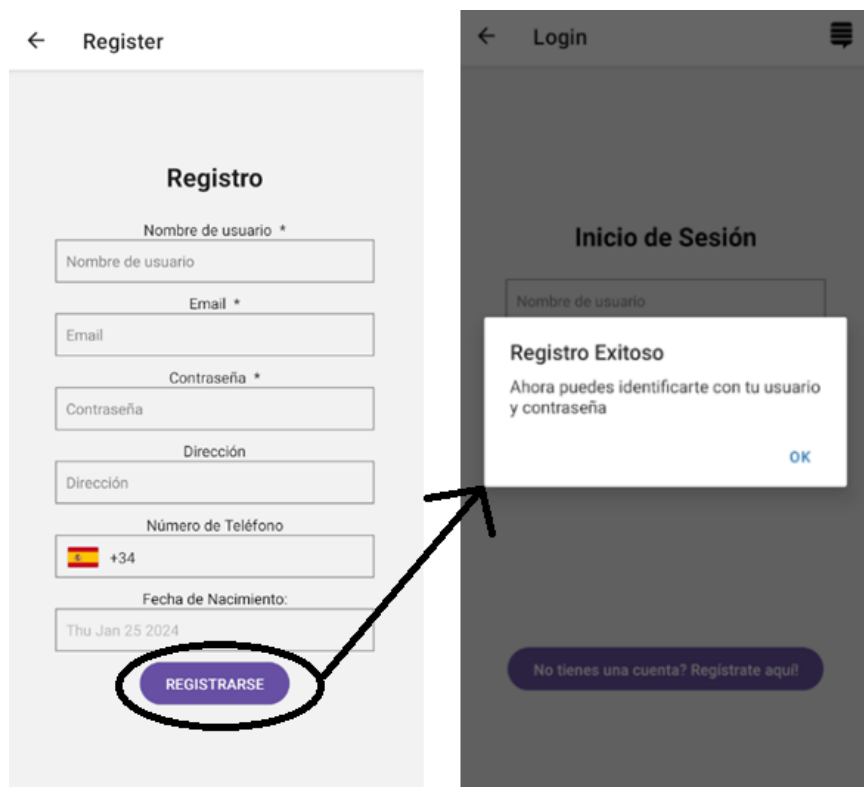


Figura 37. Pantalla de registro

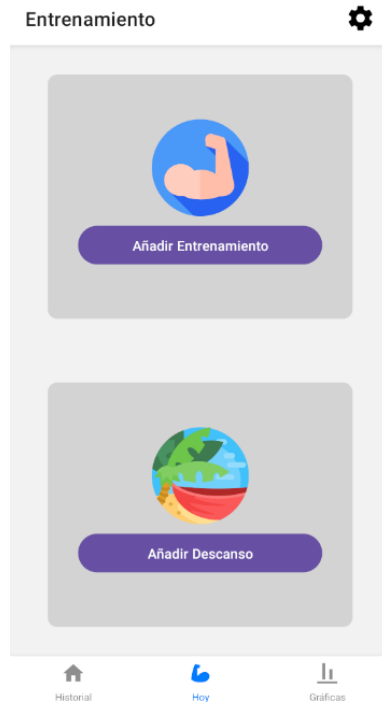


Figura 38. Pantalla principal

1.3 Cerrar sesión

Si está ya identificado y desea cerrar sesión, desde la pantalla principal presione sobre el botón superior derecho para entrar a la pantalla de ajustes que se puede observar en la Figura 40.

Después, seleccione la opción «Cerrar sesión» y se mostrará el mensaje de la Figura 39, aquí presione sobre «Cerrar sesión» para confirmar la acción, en caso de que presione sobre «Cancelar» volverá a la pantalla de ajustes sin ningún cambio.

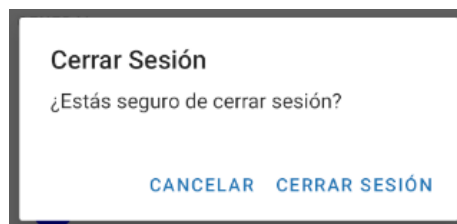


Figura 39. Pantalla para cerrar sesión

2. Editar perfil

Para modificar su perfil debe estar identificado en la aplicación. Entonces, desde la pantalla principal presione el botón ubicado en la esquina superior derecha.

Este botón le llevará a la pantalla de Ajustes, como se muestra en la Figura 40. Desde allí, seleccione «Editar Perfil» para acceder a la pantalla correspondiente.

Una vez en la pantalla de Edición de Perfil, que se visualiza en la Figura 40, podrá ver toda la información de su perfil. Para modificarla, presione sobre el botón «Editar».

Esto activará la edición de los campos, permitiéndole cambiar la información que desee. Una vez realizadas las modificaciones, presione «Guardar» para confirmar los cambios, si presiona sobre el botón cancelar ninguna de las modificaciones se guardarán.

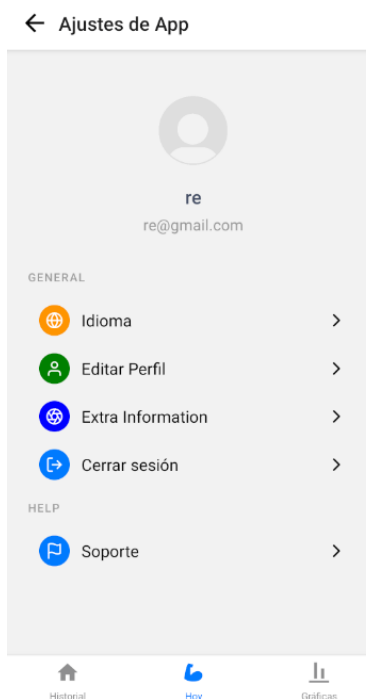


Figura 40. Pantalla de ajustes

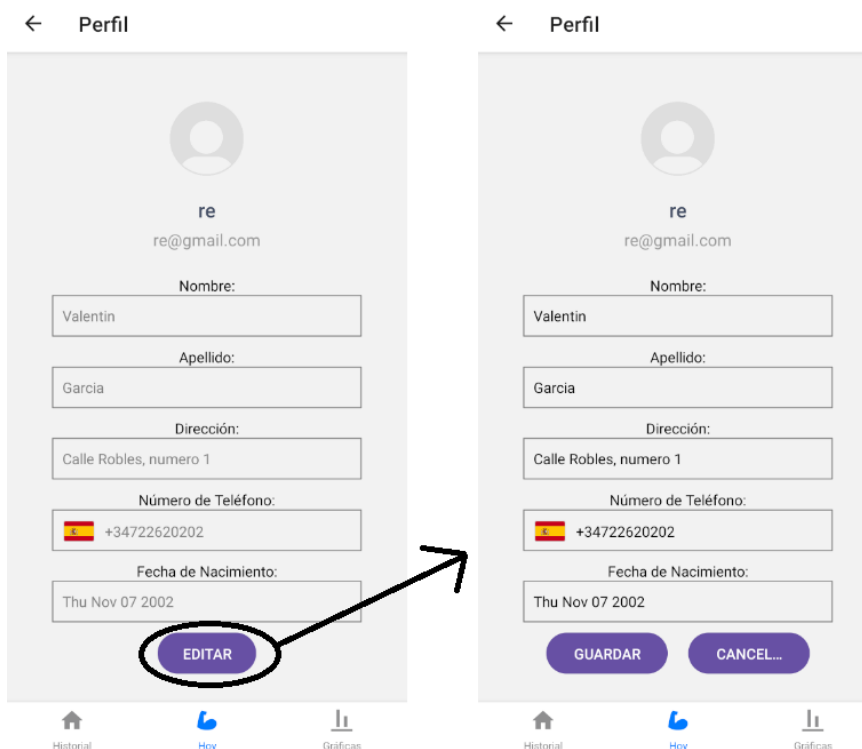


Figura 41. Pantalla para editar Perfil

3. Cambiar el idioma

Para cambiar el idioma, dependerá de si está identificado o no, en caso de no estar identificado, desde la pantalla de inicio de sesión, presione sobre el botón superior derecho y aparecerá la pantalla de la Figura 42.

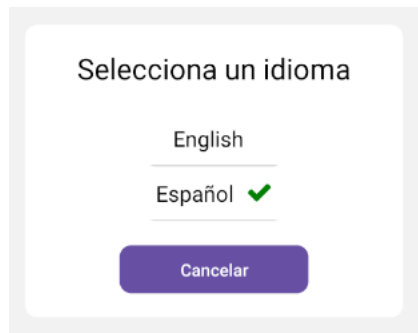


Figura 42. Pantalla para seleccionar idioma

En esta pantalla de selección de idioma, se muestran todos los idiomas disponibles. El idioma actualmente seleccionado se indica con una marca de verificación a la derecha. Al elegir un idioma, volverá a la pantalla anterior con el idioma cambiado. Si decide «Cancelar», regresará a la pantalla anterior sin realizar cambios.

Si está identificado, desde la pantalla principal de la Figura 38, presione el icono de engranaje que se encuentra en la esquina superior derecha de engranaje para acceder a los ajustes. Posteriormente, seleccione la opción «Idioma» y aparecerá la pantalla de la Figura 42 para seleccionar el nuevo idioma.

4. Crear entrenamiento o descanso

Si desea añadir un entrenamiento, existen dos maneras de hacerlo: si quiere añadirlo para el día actual, desde la pantalla principal se da la opción de añadir un entrenamiento o añadir un día de descanso.

En caso de que quisiera añadir un día de entrenamiento a un día específico, seleccione la opción "Historial" de la barra inferior, para posteriormente seleccionar el día deseado como en la Figura 43.

Si el día es anterior a los días que aparecen en el Historial, presione sobre el botón «Cargar más...» para acceder a días anteriores.

Si intenta añadir un entrenamiento a un día posterior al día actual se mostrará un mensaje de error, ya que no se debería poder apuntar entrenamientos todavía no realizados.



Figura 43. Historial de entrenamientos

Cuando se ha seleccionado el día se mostrará la pantalla de la Figura 44 muy parecida a la pantalla principal, aquí deberá seleccionar si desea añadir un entrenamiento o un día de descanso.

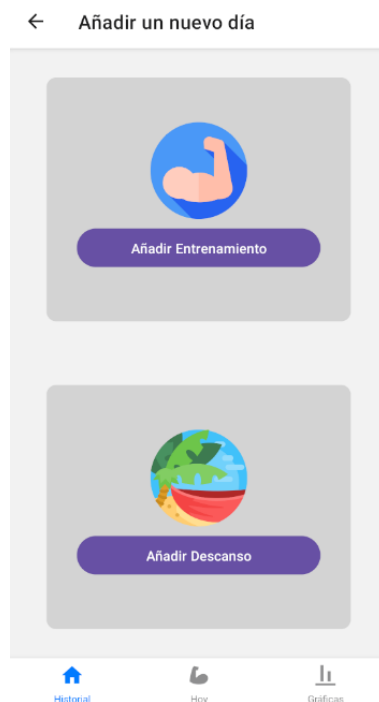


Figura 44. Pantalla para añadir un entrenamiento/descanso

4.1. Añadir entrenamiento

Cuando elija la primera opción para agregar un entrenamiento, aparecerá la pantalla que se muestra en la Figura 45. Aquí tendrá la posibilidad de especificar:

- **La duración del entrenamiento**, seleccionando las horas y los minutos que requirió.
- **Las sensaciones experimentadas durante el entrenamiento**, para evaluarlas en cada sesión.
- **Notas o descripciones adicionales**, para añadir información relevante sobre ese entrenamiento.

Además, más adelante en este manual de usuario, se detalla cómo agregar ejercicios al entrenamiento.



Figura 45. Pantalla para añadir un nuevo entrenamiento

4.2 Añadir un día de descanso

Para añadir un día de descanso, debe presionar el botón «Añadir Descanso» y se navegará a la pantalla «Añadir un nuevo Descanso» que se muestra en la Figura 46, donde podrá indicar las sensaciones de ese día, para saber en cada día de descanso si se

encontraba cansado, bien, etc. Además, podrá indicar información adicional en el apartado de «Notas». Al presionar el botón «Añadir Descanso» se guardará el día de descanso y la aplicación le redirigirá a la pantalla anterior.

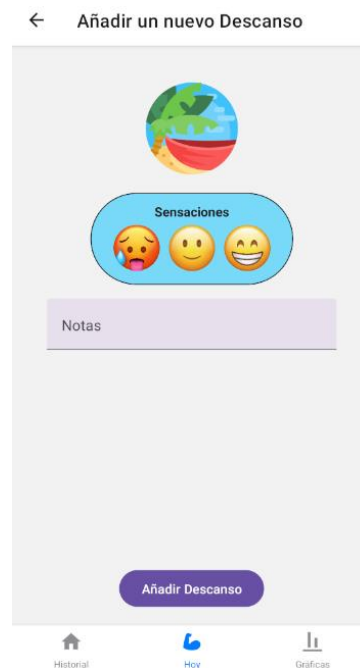


Figura 46. Pantalla para añadir un día de descanso

5. Modificar de entrenamiento o día de descanso

5.1 Modificar entrenamiento

Para modificar un entrenamiento, seleccione en la pantalla del Historial el entrenamiento que desea modificar y le aparecerá la pantalla que se muestra en la Figura 47 a la izquierda.



Figura 47. Modificar entrenamiento

Seleccione la opción «Modificar Entrenamiento» y la pantalla se mostrará como en la Figura 47 a la derecha con los datos rellenos del entrenamiento seleccionado. Cambie los datos que desea modificar y presione sobre el botón «Modificar».

5.2 Modificar descanso

Para modificar un descanso, seleccione en la pantalla del Historial el descanso que desea modificar y le aparecerá la pantalla que se muestra en la Figura 48 en la parte izquierda.

Seleccione la opción «Modificar Descanso» y la pantalla se mostrará como en la Figura 48 en la parte derecha con los datos cumplimentados sobre el descanso seleccionado. Cambie los datos que desea modificar y presione sobre el botón «Modificar».

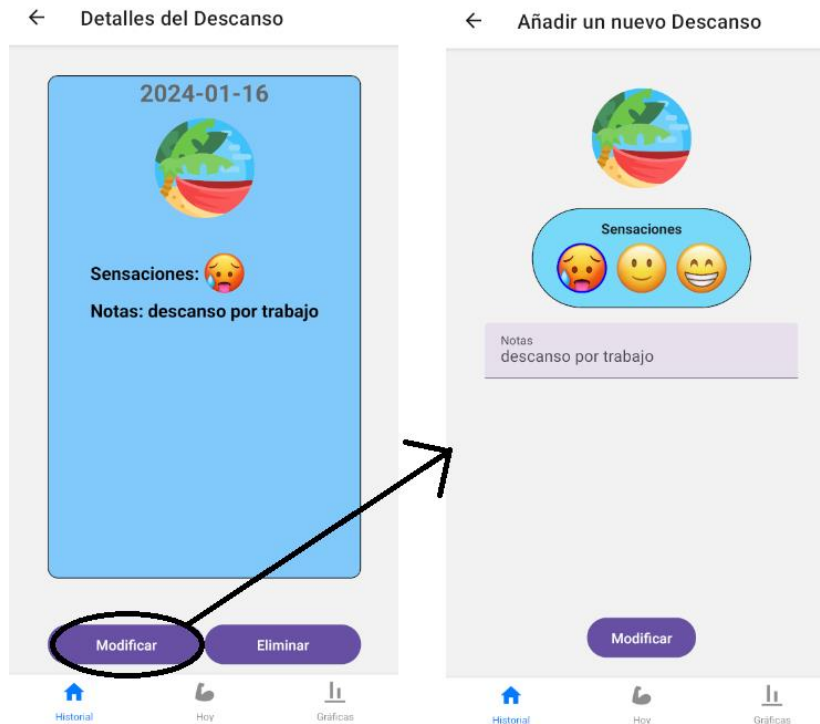


Figura 48. Modificar descanso

6. Eliminar entrenamiento o descanso

6.1 Eliminar entrenamiento

Para eliminar un entrenamiento, seleccione en la pantalla del Historial el entrenamiento que desea eliminar y le aparecerá la pantalla que se muestra en la parte izquierda de la Figura 49.

Seleccione la opción «Eliminar» y se mostrará un mensaje de confirmación, una vez aceptado el entrenamiento se habrá eliminado.

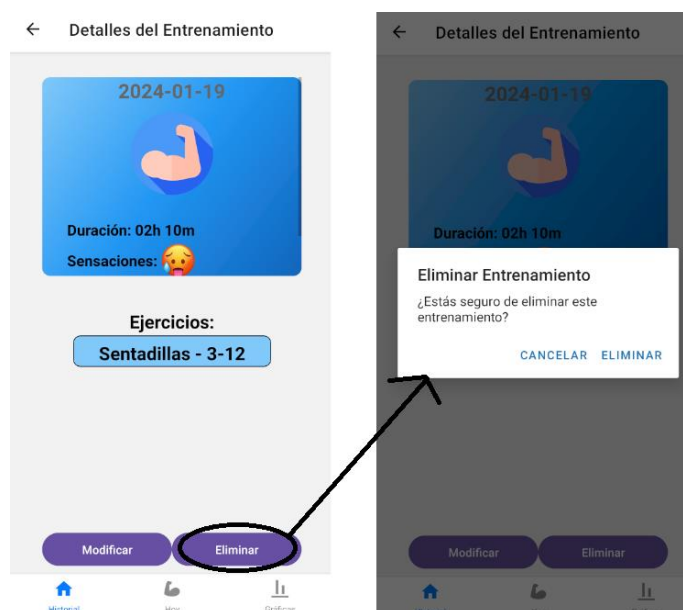


Figura 49. Eliminar un entrenamiento

6.2 Eliminar descanso

Para eliminar un descanso, seleccione en la pantalla del Historial el descanso que desea eliminar y le aparecerá la pantalla que se muestra en la Figura 50.

Seleccione la opción «Eliminar Descanso» y el descanso se eliminará y se le redirigirá a la pantalla anterior.

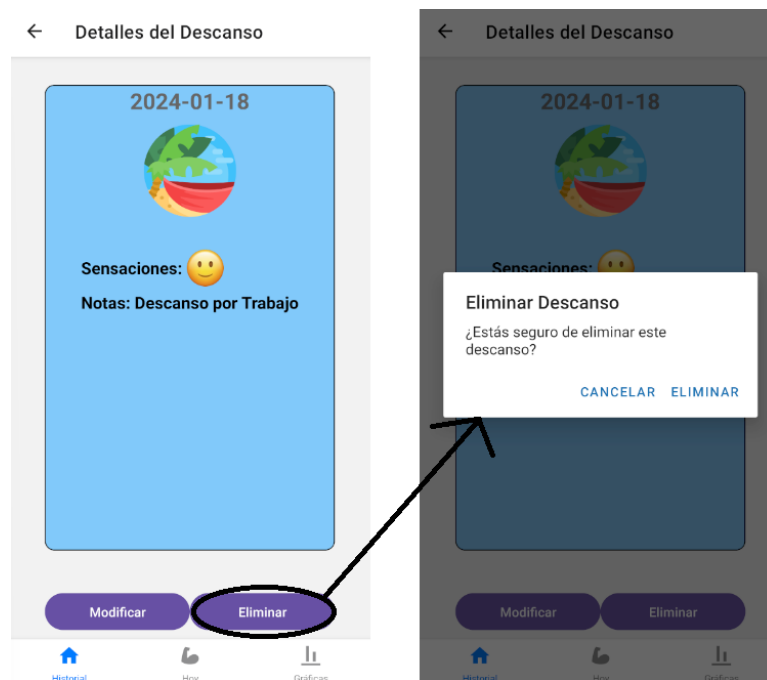


Figura 50. Eliminar descanso

7. Añadir ejercicio

Si desea añadir un ejercicio a un entrenamiento, lo primero es acceder al entrenamiento en el que queremos añadir el ejercicio y estar en la pantalla de modificación o creación.

Aquí presionaremos sobre el botón para Añadir un Ejercicio y se mostrará la pantalla que se puede observar en la Figura 51. En esta pantalla será capaz de seleccionar el ejercicio realizado, indicar las series, repeticiones y peso. Al presionar «Aceptar» y si los datos están correctos, se añadirá el ejercicio y se podrá ver como se muestra en la Figura 51.

Al seleccionar un ejercicio solo tendrá la opción de elegir entre aquellos ejercicios que todavía no ha realizado en ese entrenamiento, es decir, si ha realizado ya sentadillas en ese entrenamiento, no podrá añadirlo otra vez.

No se permiten números de series, repeticiones o peso negativos, en caso contrario la aplicación mostrará un mensaje indicando el error.

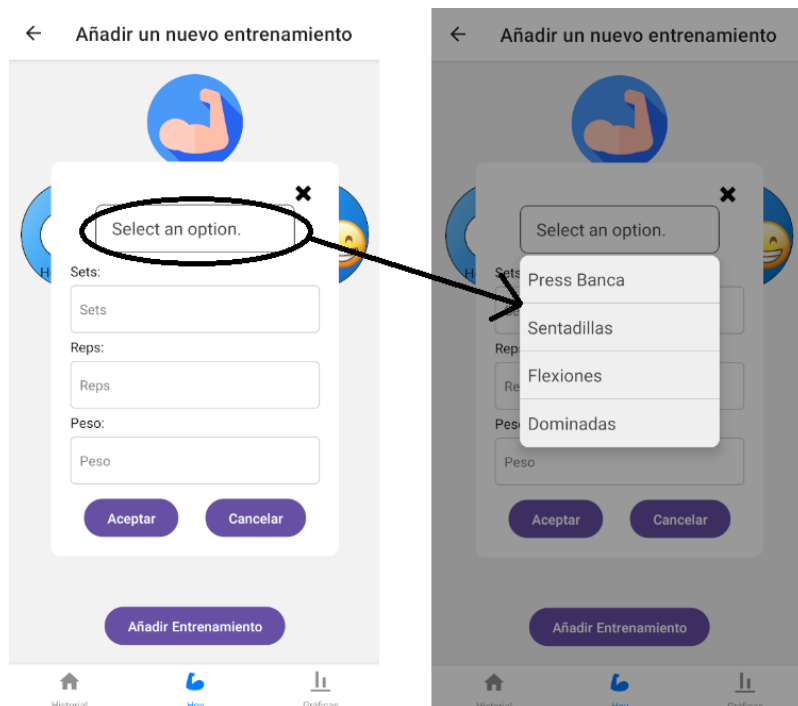


Figura 51. Pantalla para añadir un ejercicio

8. Modificar un ejercicio

Para modificar un ejercicio, debe encontrarse en la página de modificación o creación del entrenamiento en el que se encuentra el ejercicio a modificar.

En esta pantalla hay una lista de ejercicios como se ve en la Figura 52, posteriormente, seleccione la imagen de lápiz del ejercicio que quiere modificar, le saldrá una pantalla como la de creación de ejercicios, pero con los campos rellenos como se observa en la Figura 52.

Modifique los datos que desee, y posteriormente presione el botón de «Modificar», si no ha habido ningún error, se habrán modificado los campos.

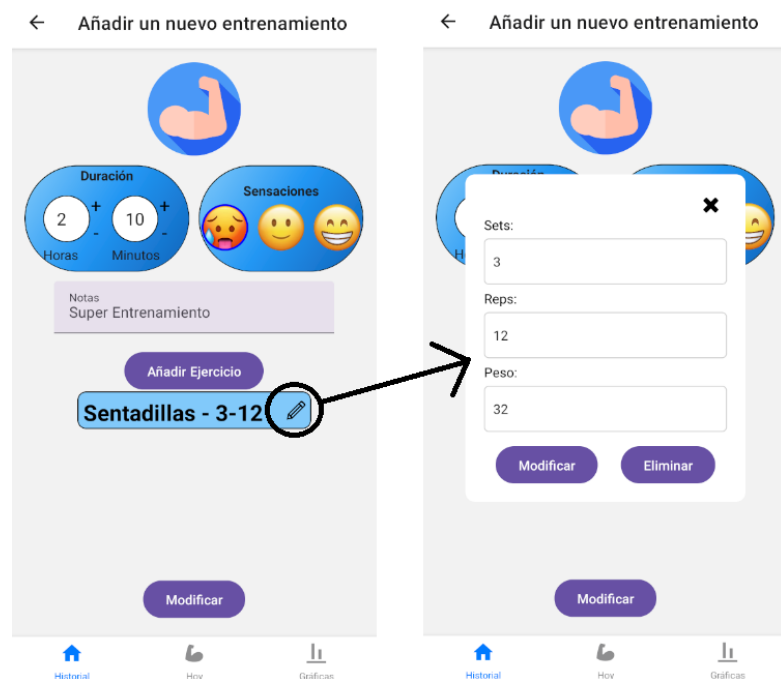


Figura 52. Modificar un ejercicio

Nota: Hasta que no presione en modificar/crear el entrenamiento los cambios no surgirán efecto

9. Eliminar un ejercicio

Para eliminar un ejercicio, debe encontrarse en la página de modificación o creación del entrenamiento en el que se encuentra el ejercicio a eliminar.

En esta pantalla, verás una lista de ejercicios, como se muestra en la Figura 53. Luego, selecciona el icono de lápiz junto al ejercicio que deseas eliminar. Esto te llevará a una pantalla similar a la de creación de ejercicios, pero con los campos ya completados, como se ilustra en la Figura 53.

Presione sobre el botón «Eliminar» y el ejercicio será eliminado y será redirigido a la pantalla anterior.

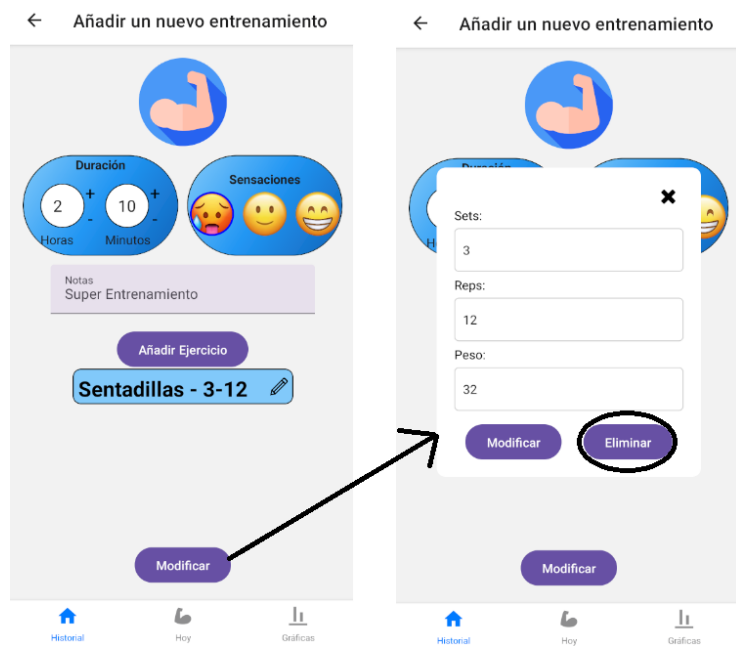


Figura 53. Eliminar un ejercicio

Nota: Hasta que no presione en modificar/crear el entrenamiento los cambios no surgirán efecto.

10. Gráficas de progresión

Para acceder a las gráficas de progresión debe presionar sobre el botón «Gráficas» que se encuentra en la navegación, en la zona inferior derecha. Esto le redirigirá a la pantalla de la Figura 54 para indicar las características de la gráfica que quiere ver.

Cuando se encuentre en la pantalla de opciones de la gráfica, puede usar las flechas de la derecha o de la izquierda para intercambiar entre las distintas opciones de la gráfica.

Y una vez elegido la opción preferida, debe indicar las opciones específicas para ese tipo de gráficas y posteriormente presionar sobre el botón «Aceptar».

Si hay algún campo sin rellenar, se le mostrará un mensaje de error indicando la opción que le falta por rellenar.

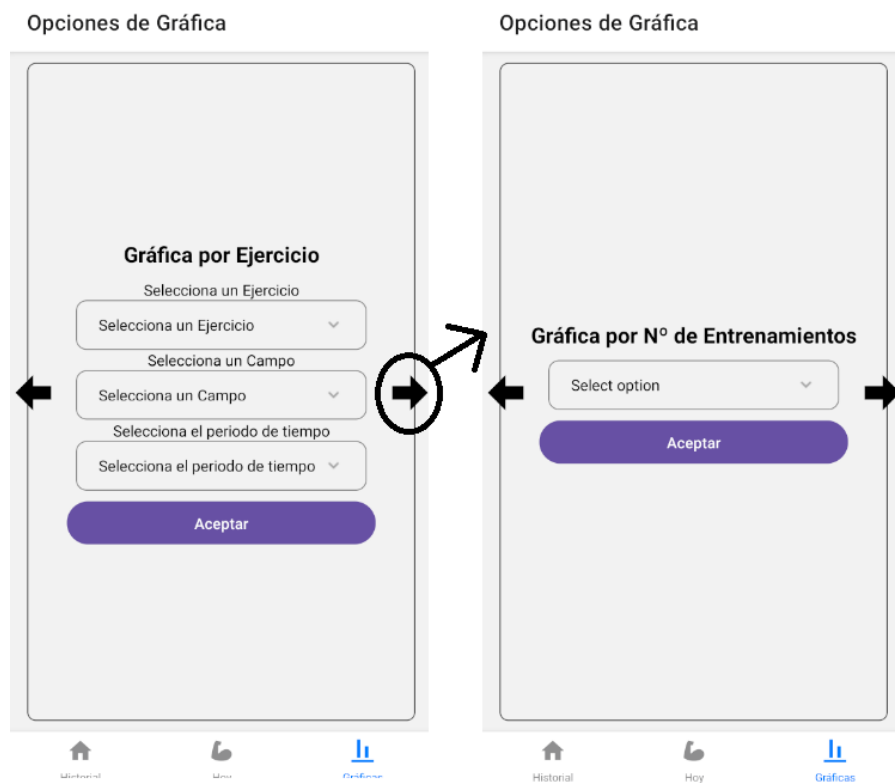


Figura 54. Pantalla de gráficas

10.1 Gráfica por ejercicio

Si selecciona la gráfica por ejercicio, deberá seleccionar los distintos campos obligatorios:

- **Ejercicio:** Entre los ejercicios que actualmente están disponibles para añadir.
- **Campo:** Para indicar que aspecto del ejercicio se quiere comparar, puede ser: Series, Repeticiones o Peso.
- **Periodo de tiempo:** Para obtener los datos para un intervalo de tiempo específico, puede ser: «Todos los días» o «Últimos 6 meses».

Y al presionar “aceptar” verá una gráfica similar a la de la Figura 55 (la gráfica puede variar dependiendo de los campos seleccionados en la pantalla anterior).

Además, una vez dentro, se le permitirá cambiar el campo a comparar sin necesidad de volver a la pantalla anterior.

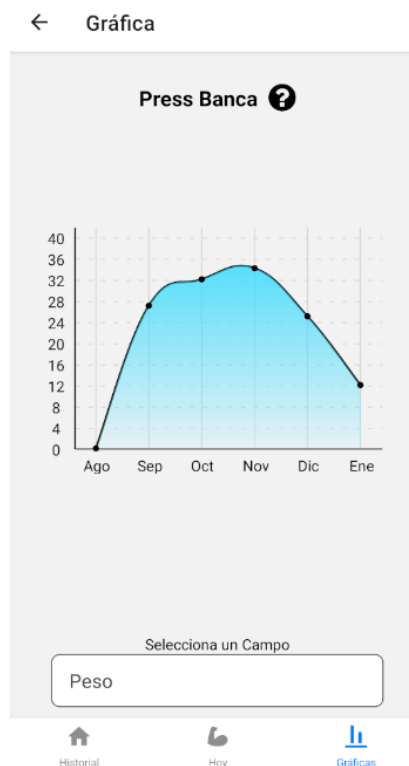


Figura 55. Gráfica por ejercicio

10.2 Gráfica por número de entrenamientos

Si selecciona la gráfica por número de entrenamientos, deberá seleccionar solo un campo, el periodo de tiempo para obtener los entrenamientos comprendidos dentro de ese intervalo o agrupados en ese intervalo, las opciones son: «Días de la semana» y «Últimos 6 meses».

Al presionar aceptar verá una gráfica similar a la de la Figura 56 (la gráfica puede variar dependiendo del campo seleccionado en la pantalla anterior).

Si desea cambiar el intervalo de tiempo, deberá volver a la pantalla anterior e indicar el nuevo valor deseado.

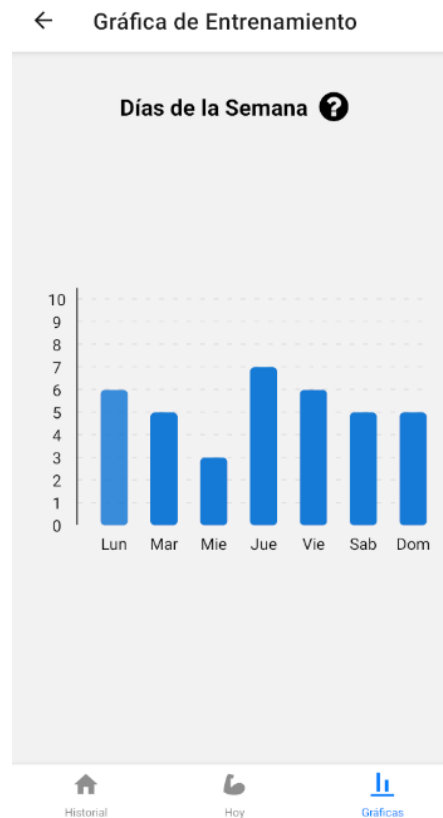


Figura 56. Gráfica por número de entrenamientos

11. Realizar detección de ejercicio

Para la detección de ejercicio tendrá que situarse en la pantalla para añadir o modificar un entrenamiento (Figura 45) y presionar el botón para reconocer. Se abrirá una pantalla donde puede seleccionar uno de los dos modos de reconocimiento: Sensores o cámara.

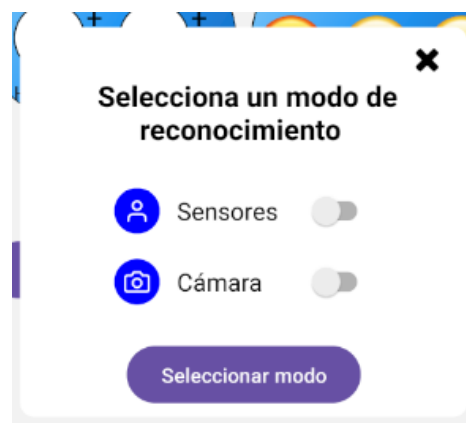


Figura 57. Selección del modo de reconocimiento

11.1 Reconocimiento con los sensores

Si selecciona el reconocimiento por sensores se abrirá la pantalla de confirmación de la Figura 58.

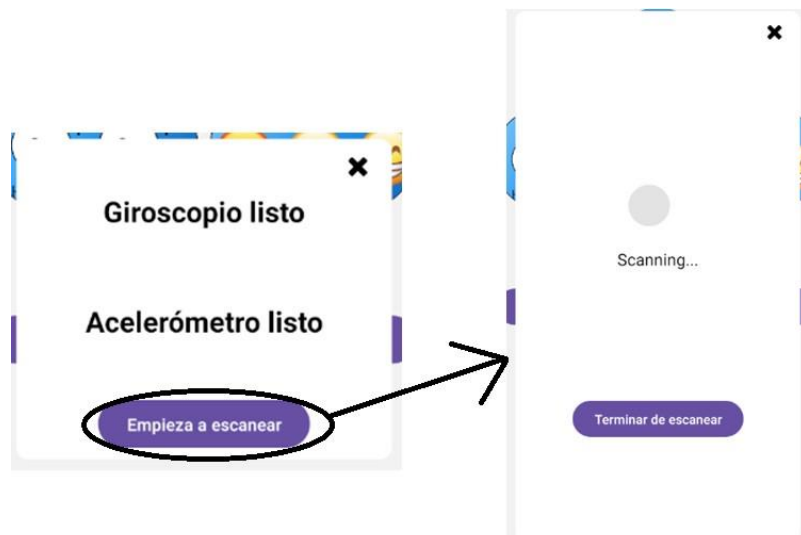


Figura 58. Proceso de reconocimiento con sensores

Una vez todo preparado (recuerde usar el soporte para el brazo correspondiente), presione el botón para empezar a escanear. Mientras se encuentre en la pantalla de la segunda pantalla de la Figura 58 el escaneo se está realizando y puede hacer cualquiera de los 4 ejercicios que ofrece la aplicación para detectar.

Una vez finalizado seleccione la opción para terminar el escaneo y la aplicación mostrará una alerta y se abrirá la pantalla para añadir el ejercicio con el ejercicio ya seleccionado.

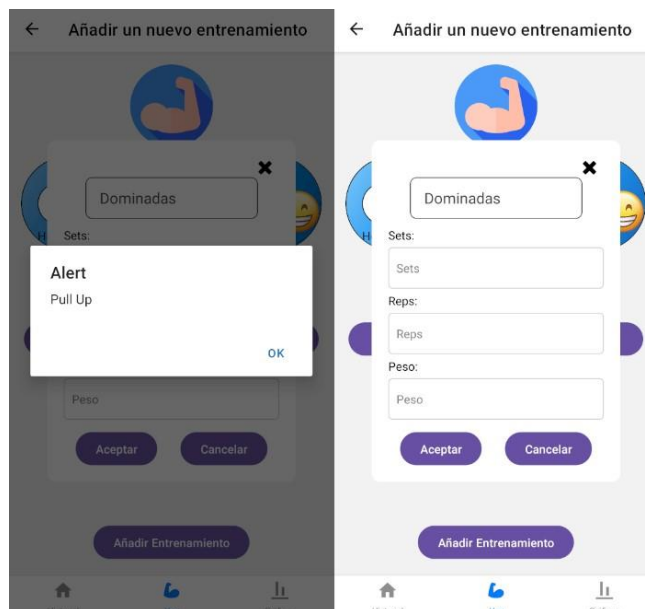


Figura 59. Finalización reconocimiento con sensores

11.2 Reconocimiento con la cámara

Si selecciona el reconocimiento con la cámara se abrirá una pantalla mostrando la imagen de la cámara frontal de su dispositivo (Figura 60), si es la primera vez que abre esta función, la aplicación le mostrará un mensaje solicitando permiso para acceder a la cámara.

Cuando presione el botón para empezar a escanear, el tiempo que aparece justo debajo de la cámara empezará a cambiar. Para asegurarse que la información es suficiente el tiempo mínimo de vídeo es de 15 segundos.

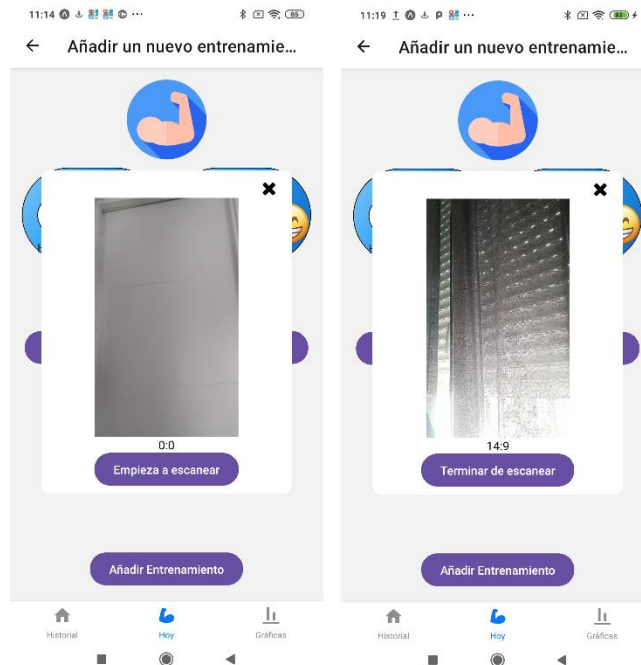


Figura 60. Proceso de reconocimiento de la cámara

Cuando presione sobre el botón para terminar de escanear, se mostrará una pantalla de espera hasta que el vídeo sea procesado y se reciba la respuesta del servidor con el ejercicio realizado (Figura 61).

Finalmente, al igual que con los sensores, se mostrará una alerta con el ejercicio realizado y se abrirá la pantalla para añadir el ejercicio con el ejercicio ya seleccionado.

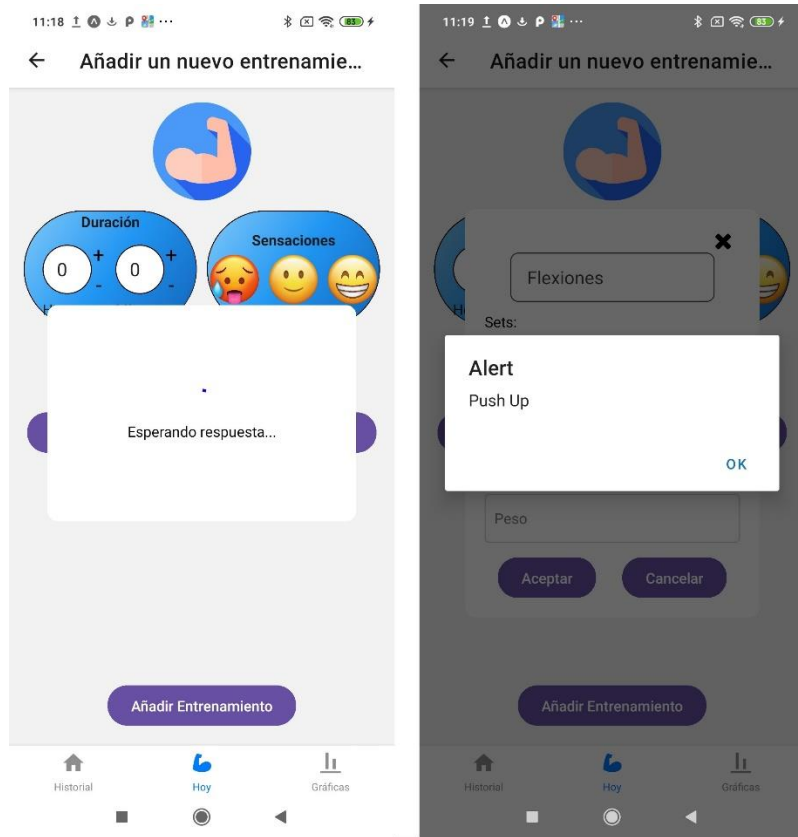


Figura 61. Finalización reconocimiento con la cámara

Referencias

- [1] Valgo, «Valgo,» 16 Diciembre 2022. [En línea]. Available: <https://www.valgo.es/blog/publicada-la-encuesta-de-habitos-deportivos-en-espana-2022?elem=293845>. [Último acceso: 02 Octubre 2023].
- [2] Flask, «Flask,» [En línea]. Available: <https://flask.palletsprojects.com/en/3.0.x/>. [Último acceso: 02 Febrero 2024].
- [3] Axios, «Axios-http,» [En línea]. Available: <https://axios-http.com/>. [Último acceso: 02 Febrero 2024].
- [4] «Expo,» Expo-Camera, [En línea]. Available: <https://docs.expo.dev/versions/latest/sdk/camera/>. [Último acceso: 02 Febrero 2024].
- [5] Expo-Sensores, «Expo,» [En línea]. Available: <https://docs.expo.dev/versions/latest/sdk/sensors/>. [Último acceso: 02 Febrero 2024].
- [6] i18next, «i18next,» [En línea]. Available: <https://www.i18next.com/>. [Último acceso: 02 Febrero 2024].
- [7] G. Charts, «Gifted Charts,» [En línea]. Available: <https://gifted-charts.web.app/>. [Último acceso: 02 Febrero 2024].
- [8] NumPy, «NumPy,» [En línea]. Available: <https://numpy.org/>. [Último acceso: 02 Febrero 2024].
- [9] PyTorch, «PyTorch,» [En línea]. Available: <https://pytorch.org/docs/stable/torch.html>. [Último acceso: 02 Febrero 2024].
- [10] SQLAlchemy, «Sqlalchemy,» [En línea]. Available: <https://www.sqlalchemy.org/>. [Último acceso: 02 Febrero 2024].
- [11] J. Cotzo, «Medium,» 4 Septiembre 2020. [En línea]. Available: <https://juliocotzo.medium.com/patrones-de-arquitectura-de-software-6cffda7dd39e>. [Último acceso: 1 Enero 2024].
- [12] T. Naeem, «Astera,» 1 Diciembre 2023. [En línea]. Available: <https://www.astera.com/es/type/blog/rest-api-definition/>. [Último acceso: 25 Enero 2024].
- [13] br.thptnganamst.edu.vn, «br.thptnganamst.edu.vn,» [En línea]. Available: <https://br.thptnganamst.edu.vn/introduzir-80-imagem-modelo-iterativo/>. [Último acceso: 25 Enero 2024].
- [14] RICHDMI, «Amazon,» [En línea]. Available: https://www.amazon.es/dp/B0C9ZGBR8C?starsLeft=1&ref_=cm_sw_r_cso_wa_ap_in_dp_G9MZRC97S0C7BDQPRGT. [Último acceso: 2 Febrero 2024].

- [15] P. Huet, «OpenWebinars,» 13 Abril 2023. [En línea]. Available: <https://openwebinars.net/blog/que-son-las-redes-neuronales-y-sus-aplicaciones/>. [Último acceso: 25 Enero 2024].
- [16] Google, «Google,» [En línea]. Available: https://developers.google.com/mediapipe/solutions/vision/pose_landmarker. [Último acceso: 2 Febrero 2024].
- [17] Wikipedia, «React Native,» 4 Noviembre 2023. [En línea]. Available: https://es.wikipedia.org/wiki/React_Native. [Último acceso: 25 Enero 2024].
- [18] P. Store, «Expo,» [En línea]. Available: https://play.google.com/store/apps/details?id=host.exp.exponent&hl=es_NI. [Último acceso: 25 Enero 2024].
- [19] Wikipedia, «Visual Studio Code,» [En línea]. Available: https://es.m.wikipedia.org/wiki/Archivo:Visual_Studio_Code_1.35_icon.svg. [Último acceso: 25 Enero 2024].
- [20] Aptoide, «Justinmind,» [En línea]. Available: <https://justinmind.es.aptoide.com/app>. [Último acceso: 25 Enero 2024].
- [21] Git, «Logos,» [En línea]. Available: <https://git-scm.com/downloads/logos>. [Último acceso: 25 Enero 2024].
- [22] Wikipedia, «Python,» 13 Enero 2024. [En línea]. Available: <https://es.m.wikipedia.org/wiki/Python>. [Último acceso: 25 Enero 2024].
- [23] 1000logos, «MySQL Logos,» [En línea]. Available: <https://1000logos.net/mysql-logo/>. [Último acceso: 25 Enero 2024].
- [24] React-Native-Gifted-Charts, «React-Native-Gifted-Charts,» [En línea]. Available: <https://gifted-charts.web.app/>. [Último acceso: 02 Febrero 2024].
- [25] Reboot, «Reboot,» [En línea]. Available: <https://reboot.studio/blog/es/estructuras-organizar-proyecto-react>. [Último acceso: 11 Febrero 2024].
- [26] S. a. R. V. David and Huang, «Github,» [En línea]. Available: <https://mmfit.github.io/>. [Último acceso: 11 Febrero 2024].
- [27] D. Mesquita, «Real Python,» [En línea]. Available: <https://realpython.com/python-ai-neural-network/>. [Último acceso: 11 Febrero 2024].
- [28] nam157, «GitHub,» [En línea]. Available: https://github.com/nam157/human_activity_recognition-. [Último acceso: 11 Febrero 2024].
- [29] MySQL, «Docker,» [En línea]. Available: https://hub.docker.com/_/mysql. [Último acceso: 11 Febrero 2024].
- [30] J. Molina, «j2logo,» [En línea]. Available: <https://j2logo.com/tutorial-flask-espanol/>. [Último acceso: 11 Febrero 2024].
- [31] M. García, «Real Python⁹,» [En línea]. Available: <https://realpython.com/flask-blueprint/>. [Último acceso: 11 Febrero 2024].
- [32] IBM, «IBM,» [En línea]. Available: <https://www.ibm.com/es-es/topics/convolutional-neural-networks>. [Último acceso: 11 Febrero 2024].
- [33] J. M. Llama, «UPM,» [En línea]. Available: <https://oa.upm.es/53050/>. [Último acceso: 11 Febrero 2024].

[34] R. Native, «React Native,» [En línea]. Available:
<https://reactnative.dev/docs/tutorial>. [Último acceso: 11 Febrero 2024].

Índice de figuras

Figura 1. Logo React Native [17]	5
Figura 2. Logo de Expo [18].....	5
Figura 3. Logo de <i>Visual Studio Code</i> [19]	6
Figura 4. Logo de <i>Justinmind</i> [20]	6
Figura 5. Logo de <i>Git</i> [21].....	6
Figura 6. Logo de <i>Python</i> [22]	6
Figura 7. Logo de <i>MySQL</i> [23]	6
Figura 8. Diseño de la base de datos	14
Figura 9. Arquitectura Cliente-Servidor [11]	15
Figura 10. Funcionamiento general arquitectura REST [12].....	15
Figura 11. Uso de <i>Blueprints</i> en el servidor	17
Figura 12. Organización de directorios del servidor	17
Figura 13. Organización de aplicación <i>React Native</i> por carpetas	18
Figura 14. Metodología iterativa e incremental [13]	19
Figura 15. Soporte para sujetar el móvil en el brazo [14]	47
Figura 16. Funcionamiento de redes neuronales [15].....	48
Figura 17. Representación de los datos provenientes de los sensores.....	50
Figura 18. Puntos clave anatómicos que proporciona <i>Mediapipe</i> [16].....	51
Figura 19. Puntos clave anatómicos marcados en el cuerpo	52
Figura 20. Representación de los datos provenientes de la cámara.....	53
Figura 21. Fichero « <i>labels.csv</i> » de los sensores	54
Figura 22. Fichero « <i>labels.csv</i> » de la cámara	54
Figura 23. Entrenamiento de red neuronal de sensores	55
Figura 24. Evaluación de red neuronal de sensores	56
Figura 25. Entrenamiento de red neuronal de cámara	56
Figura 26. Evaluación de red neuronal de cámara	56
Figura 27. Función que recibe los datos de los sensores y predice el ejercicio realizado	57
Figura 28. Función para obtener los puntos clave a partir de un vídeo.....	58
Figura 29. Función que recibe los datos de la cámara y reconoce el ejercicio realizado	58
Figura 30. Nueva pantalla para añadir un ejercicio con reconocimiento.....	59
Figura 31. Despliegue base de datos en <i>Docker</i>	68
Figura 32. Creación del ambiente <i>Conda</i>	68
Figura 33. Despliegue del servidor	69
Figura 34. Despliegue de la aplicación móvil.....	69
Figura 35. Selección de la aplicación en la aplicación móvil.....	70
Figura 36. Pantalla de inicio de sesión.....	71
Figura 37. Pantalla de registro	72

Figura 38. Pantalla principal	73
Figura 39. Pantalla para cerrar sesión	73
Figura 40. Pantalla de ajustes.....	74
Figura 41. Pantalla para editar Perfil.....	74
Figura 42. Pantalla para seleccionar idioma	75
Figura 43. Historial de entrenamientos	75
Figura 44. Pantalla para añadir un entrenamiento/descanso	76
Figura 45. Pantalla para añadir un nuevo entrenamiento	76
Figura 46. Pantalla para añadir un día de descanso	77
Figura 47. Modificar entrenamiento	78
Figura 48. Modificar descanso	79
Figura 49. Eliminar un entrenamiento	79
Figura 50. Eliminar descanso.....	80
Figura 51. Pantalla para añadir un ejercicio	81
Figura 52. Modificar un ejercicio.....	81
Figura 53. Eliminar un ejercicio	82
Figura 54. Pantalla de gráficas	83
Figura 55. Gráfica por ejercicio	83
Figura 56. Gráfica por número de entrenamientos	84
Figura 57. Selección del modo de reconocimiento.....	84
Figura 58. Proceso de reconocimiento con sensores	85
Figura 59. Finalización reconocimiento con sensores	85
Figura 60. Proceso de reconocimiento de la cámara	86
Figura 61. Finalización reconocimiento con la cámara	87