

---

This is the **published version** of the bachelor thesis:

Mota Paredes, Jean Aleksei; Oropesa Fisica, Ana, dir. Desarrollo de aplicación web front-end para un cliente con la finalidad de fomentar la actividad física. 2024. (Grau en Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/298915>

under the terms of the  license

# Desarrollo de aplicación web front-end para un cliente con la finalidad de fomentar la actividad física

Jean Aleksei Mota Paredes - 1599525

**Resumen**—El trabajo presenta el desarrollo de una aplicación web frontend para un cliente, centrada en promover la actividad física a través de una plataforma amigable y personalizada para el usuario. Utilizando tecnologías modernas como React, JavaScript y Jest para pruebas, el proyecto destaca la importancia del control de errores, la refactorización del código y la realización de pruebas para garantizar la calidad del software. A pesar de algunos desafíos en el diseño y la implementación, el proyecto demuestra cómo la tecnología puede fomentar un estilo de vida activo.

**Palabras clave**— Desarrollo web, Frontend, Amplify, AWS, Gantt, React, JavaScript, CSS3, Jest.

**Abstract**—This project involves the development of a frontend web application aimed at promoting physical activity. The application is designed to be user-friendly and personalized, utilizing modern technologies such as React and JavaScript, with Jest for testing. The project emphasizes error handling, code refactoring, and thorough testing to ensure software quality. Despite some challenges in design and implementation, the project demonstrates how technology can support an active lifestyle.

**Index Terms**— Web development, Frontend, Amplify, AWS, Gantt, React, JavaScript, CSS3, Jest.



## 1 INTRODUCCIÓN - CONTEXTO DEL TRABAJO

HOY en día la tecnología la encontramos por todos lados, incluso en el mundo del *fitness* [1]. Desde la comodidad de nuestros dispositivos podemos llegar a acceder a cualquier tipo de información o servicio para poder alcanzar nuestros objetivos, por lo tanto, se pretende crear una aplicación web concretamente enfocada en este sector de la salud y bienestar para los usuarios, que sea funcional y cómoda para cualquiera que pretenda utilizarla.

## 2 MOTIVACIÓN

La motivación detrás de este proyecto surge de ofrecer una plataforma que inspire a las personas a llevar un estilo de vida más activo y saludable, además de brindar las herramientas para poder alcanzar sus objetivos sin barreras tecnológicas.

Se aspira a proporcionar un espacio donde se pueda acceder a entrenamientos personalizados, registrar su progreso, conectarse con una comunidad activa y recibir el apoyo necesario para mantenerse comprometidos con su bienestar a largo plazo.

Como motivación personal, al ser un desarrollo tan cercano a la realidad, se ve como una oportunidad única para maximizar la experiencia en el sector de desarrollo y

acercarse el máximo a la realidad, interactuando con los clientes, trabajando en grupo y teniendo un plazo determinado para finalizar el proyecto.

Con estos propósitos en mente, se espera hacer una contribución significativa al mundo del fitness digital y, en última instancia, mejorar la calidad de vida de quienes la utilicen.

## 3 OBJETIVOS

### 3.1 Objetivo general

Para poder llegar a completar el trabajo se pretenden completar los siguientes objetivos, los cuales serán separados en un objetivo general y diferentes objetivos específicos para poder llegar a completar el general.

- Desarrollar una aplicación integral de ejercicio que brinde a los usuarios una experiencia completa y personalizada para mejorar su salud y bienestar a través del ejercicio físico.

### 3.2 Objetivos específicos

Como este trabajo se compone de dos partes, pero este trabajo se centra en la parte del front-end, los objetivos específicos se verán enfocados en ello.

1. Se creará una interfaz capaz de registrar a dos tipos de usuarios: entrenadores y deportistas, permitiendo diferentes funcionalidades a ambos y comprobando la creación de los usuarios

- 
- E-mail de contacto: [JeanAleksei.Mota@autonoma.cat](mailto:JeanAleksei.Mota@autonoma.cat)
  - Mención realizada: *Tecnologías de la Información*
  - Trabajo tutorizado per: *Ana Oropesa Física (Departamento de Ingeniería de la Información y de las Comunicaciones)*
  - Curso 2023/24

- en la base de datos a través de un inicio de sesión. Este objetivo se completará en menos de una semana y se la completará para poder llevar a cabo las demás funcionalidades.
2. Se implementará una interfaz de usuario que permita a los usuarios modificar, visualizar e interactuar con sus datos personales de manera efectiva. Esta interfaz deberá ser capaz de mostrar todos los campos relevantes de datos personales y permitir su edición de manera intuitiva. Se separarán en mis entrenos, mi perfil y mis reservas en un plazo de 1 semana para poder cumplir con la planificación.
  3. Se desarrollará un sistema de planificación de entrenamientos que sea fácil de entender y utilizar, mejorando así la experiencia del usuario. Se evaluará la eficacia de este sistema con retroalimentación de los usuarios, asegurando que se adapte a diferentes niveles de experiencia y objetivos de fitness en un plazo de 1 semana.
  4. Crearemos un sistema de registro de actividades y progreso que incluya gráficas de contraste para permitir a los usuarios comparar medidas actuales con pasadas de manera rápida y visual. Nos comprometemos a cumplir con esta funcionalidad antes de la fecha de entrega.
  5. Integramos contenido educativo sobre fitness, nutrición y bienestar en la aplicación web, proporcionando a los usuarios acceso a videos, artículos y consejos relevantes. Colaboraremos con expertos en el campo para obtener contenido preciso y útil, con el objetivo de mejorar la autonomía del usuario. Con la confirmación del cliente para ver si el contenido se adecua a la página web antes de la fecha de despliegue del producto.
  6. Se diseñará una interfaz limpia y atractiva para realizar reservas, simplificando el proceso para los usuarios. Utilizaremos principios de diseño UX/UI para garantizar una experiencia fluida de reserva, con el objetivo de mejorar la satisfacción del usuario. Nos comprometemos a completar el diseño de reserva dentro de un plazo definido de antes de 1/05/2024.
  7. Se desarrollará e integrará una interfaz de usuario interactiva que facilite la comunicación bidireccional entre usuarios y entrenadores. Se implementarán herramientas de mensajería y notificación para garantizar una interacción fluida y efectiva. Confirmando su finalización con pruebas entre entrenadores y usuarios utilizando diferentes pruebas antes del 1/05/2024.
  8. Se diseñarán desafíos, logros, tablas de clasificación e insignias para integrar en la aplicación con el objetivo de mantener a los usuarios motivados. Evaluando la participación y el compromiso de los usuarios con estos elementos de

gamificación, asegurando el desarrollo de sistemas que incentiven la participación y la superación de metas. Con un compromiso a implementar estos elementos antes de la fecha de entrega.

9. Se desarrollará una interfaz que permita a los usuarios compartir sus logros y progresos en redes sociales de manera intuitiva y efectiva. Integrando botones de compartir y funcionalidades que conecten con las redes sociales más populares, con el objetivo de fomentar la participación de la comunidad y la difusión de la aplicación. Se dará por finalizada la funcionalidad cuando se hagan las pruebas pertinentes para comprobar que funciona correctamente.

## 4 PLANIFICACIÓN

Para la planificación se utilizará un diagrama Gantt [2] para planificar y programar las tareas a lo largo del tiempo. Se verán las diferentes fases del proyecto debajo de sus correspondientes fechas de inicio y las tareas que se llevarán a cabo dentro de cada fase y sus fechas de inicio y fin para ellas.



Fig. 1. Diagrama de Gantt planificación.

Como se ve en la Fig 1, el trabajo se divide en 4 fases principales con un tiempo aproximado de un mes por cada una de ellas. Teniendo en cuenta posibles imprevistos, se dejará también un mes de margen adicional para poder finalizar el proyecto, todo y que se presenten problemas. Concretando con los tiempos de las tareas, se trabaja con tiempos específicos de una semana y si se necesita más supervisión de 10 días, como es el caso del testing.

Especificando sobre las fases del proyecto, la elección es de 4 donde primero se definen los requisitos de la aplicación a llevar a cabo, junto a los casos de uso y a una reunión con el cliente para poder refinar sobre estos anteriores. Posteriormente se pasa al desarrollo. Una vez que están las tecnologías escogidas y está todo preparado (servidor, dominio, base de datos) para poder empezar a programar, las tareas de desarrollo se llevarán a cabo de esta manera, donde podemos tener dos tareas por semana, porque tienen una relación muy cercana entre ellas. Cuando se finalice el desarrollo, se empezará el desarrollo del código de test para comprobar el correcto funcionamiento de la aplicación y se finalizará haciendo las correcciones pertinentes.

Para acabar, se redactará la documentación sobre el trabajo realizado y se desplegará la aplicación de manera definitiva para el cliente.

## 5. METODOLOGÍA

Para llevar a cabo el trabajo y la realización de la aplicación se usarán metodologías ágiles y prácticas de desarrollo modernas para garantizar un desarrollo eficiente, organizado y de calidad. En específico, Scrum y DevOps.

El enfoque Scrum [3] facilitará la división del proyecto en sprints. En el inicio de cada sprint, se realiza una sesión de planificación para definir las actividades a realizar. A lo largo del sprint, se realizan reuniones periódicas para monitorear el progreso. Al finalizar el sprint, se efectúa una revisión del trabajo completado.

Mientras que DevOps [4] representa un conjunto de metodologías de desarrollo de software y operaciones, con la finalidad de reducir los tiempos de trabajo, fomentar la colaboración y asegurar las entregas.

Para conseguir el despliegue de la aplicación se utilizarán los servicios de AWS [5], concretamente Amplify [6] que permite desplegar la aplicación, dominio para la web, configuración del servidor y configuración de la base de datos, de esta manera se conseguirá un desarrollo automatizado y escalable.

## 6. ESTADO DEL ARTE

### 6.1 Contexto

La transformación digital también abarca muchos ámbitos, cambiando la forma en que las personas abordan su salud y bienestar, facilitando el acceso a la información, entrenamientos, seguimiento del progreso físico a través de dispositivos inteligentes, aplicaciones móviles y plataformas en línea.

Las tendencias actuales en el desarrollo de estas aplicaciones web dedicadas al fitness incluyen la integración de tecnologías emergentes como la realidad virtual y aumentada [7], el enfoque en la experiencia del usuario y la gamificación para mantener motivados a los usuarios. Las principales contribuciones recientes en este campo son aplicaciones de entrenamiento virtual, el uso de inteligencia artificial [8] para análisis de datos y la integración con dispositivos *wearables*.

A pesar del progreso, aún existen áreas de mejora en accesibilidad, diversidad, salud mental y personalización avanzada. Por ello mismo se realiza este trabajo que, en diferencia al resto de webs ya disponibles en el mercado, ofrece una personalización al usuario más concreta para realizar reservas en diferentes sesiones que se ofrezcan, selección de diferentes entrenos o una integración concreta de un chat con los entrenadores para mantener siempre un contacto cercano con el usuario, proporcionando una plataforma inspiradora y funcional para promover un estilo de vida activo y saludable.

### 6.2 Tecnologías empleadas

El desarrollo de aplicaciones web se ha mantenido

durante más de 11 años liderado por el lenguaje de JavaScript [9], además React ha emergido como uno de los marcos de trabajo front-end más populares y poderosos en la actualidad [10], promoviendo el desarrollo modular a través de componentes reutilizables y la implementación del DOM virtual [11] para una actualización eficiente de la interfaz de usuario. La gestión del estado global como Context API [12] simplifica la gestión de datos en aplicaciones complejas.

CSS [14] y HTML [15], las cuales son fundamentales para el diseño y la estructura de las aplicaciones web front-end, donde la escritura de estilos en JavaScript ha ganado popularidad y el diseño web receptivo se ha convertido en una práctica estándar. El uso adecuado de etiquetas semánticas en HTML mejora la accesibilidad y la indexación de los motores de búsqueda.

También prácticas de desarrollo guiadas por pruebas y herramientas de automatización como Jest [16] para garantizar la calidad y estabilidad del código.

Por todas estas razones y la cantidad de información que existe para poder trabajar con estas tecnologías aplicando buenas prácticas y ofrecer escalabilidad, mejora de la experiencia del usuario, eficiencia y una web atractiva visualmente, se decide utilizar para el proyecto propuesto en este trabajo.

Concretando un poco, se ha decidido usar navegación creando una SinglePage [17] dentro de la web en vez de una OnePage [18] para ofrecer un entorno más separado donde el usuario se puede guiar entre diferentes secciones de manera fluida y eficiente; para conseguir esto se integra el BrowserRouter de React [19].

Para gestionar el estado global de la aplicación y compartir datos entre componentes de forma eficiente, se hace uso de Context API. Esta característica de React permite centralizar y compartir el estado entre diferentes partes de la aplicación sin necesidad de pasar props manualmente a través de múltiples niveles de componentes. El uso de Context API simplifica la gestión de datos en aplicaciones complejas y mejora la organización y mantenimiento del código.

## 7. DESARROLLO

### 7.1 Casos de uso

Para comenzar el desarrollo, primero de todo se procede a preparar los casos de uso que describirán las funcionalidades, diferentes de la aplicación y cómo interactuará el usuario con la misma.

Como esta aplicación se compone de diversas funcionalidades se decide separar en varios casos de uso la aplicación para poder modular el trabajo, acabando así con siete casos de uso que explican las funcionalidades como el chat con los entrenadores, los entrenamientos, la gamificación, el inicio de sesión y registro, los perfiles de usuario, el progreso y las reservas.

Una vez realizados los casos de uso, se proceden a revisar junto al cliente para poder acabar de aclarar dudas o corregir funcionalidades y se crean los nuevos casos de uso

refinados. Se pueden ver en el apéndice A1-A7.

Estos casos de uso han sido elegidos porque gracias a ellos se pueden alcanzar los objetivos propuestos del trabajo y con ello cumplir con las funcionalidades que ofrecen a los usuarios una aplicación web única.

### 7.1.1. Caso de uso chat

El primero de todos los casos de uso viene relacionado con un chat bidireccional para permitir la comunicación entre entrenadores y deportistas. Se pretende crear una comunicación en tiempo real de manera privada entre ambos tipos de usuario, añadiendo las notificaciones de mensajes y la posibilidad de ver todos los entrenadores y tener un registro de los mensajes anteriores.

### 7.1.2. Caso de uso de entrenamientos

Este caso de uso está centrado en ofrecer la funcionalidad única a entrenadores de crear entrenos, eliminarlos y modificarlos. Permitiendo a cualquier usuario posteriormente visualizar estos entrenos ya creados y añadirlos al apartado de “mis entrenos” ofreciendo junto a ellos imágenes descriptivas sobre los ejercicios y contenido educativo, como pueden ser videotutoriales para autonomizar al usuario y que no necesite depender del entrenador en la gran mayoría de casos.

### 7.1.3. Caso de uso de gamificación

Se quiere desarrollar un ranking y ofrecer la creación y eliminación de desafíos y logros los cuales se puedan compartir por redes sociales para así mantener la motivación de los usuarios y promover el uso de la web.

### 7.1.4. Caso de uso inicio de sesión y registro de usuario

Para poder llevar a cabo la aplicación se pretende registrar a dos tipos de usuario: entrenadores y deportistas. Cada uno con sus propias funcionalidades, junto a sus datos personales para poder posteriormente utilizarlos a lo largo de la aplicación y tenerlos registrados en la base de datos.

### 7.1.5. Caso de uso de perfiles de usuario

Una vez registrados los usuarios, se quiere permitir la modificación de los datos personales y guardar tanto las reservas que tenga hechas, como los entrenos que el usuario decida seleccionar como favoritos.

### 7.1.6. Caso de uso de progreso

Se quiere llevar un seguimiento del progreso del usuario registrando diferentes metas que proponga. Estas pueden ser a largo plazo y a corto plazo y pueden ser actualizadas y compartidas en redes sociales para mantener actividad del usuario constante en la aplicación.

### 7.1.7. Caso de uso de reservas

Se quiere ofrecer la funcionalidad de hacer reservas a diferentes tipos de sesiones que creen los entrenadores, llevando un conteo del aforo máximo y mostrando la diversa información sobre la sesión, permitiendo también cancelar las reservas y darles el privilegio de eliminarlas a los entrenadores. Todo esto con el fin de no tener que utilizar una aplicación externa para realizar reservas y mantener la

comodidad del usuario.

## 7.2 Base de datos

Con los casos de uso se procede a crear la base de datos teniendo en cuenta las diferentes funcionalidades y cómo van a estar relacionadas entre ellas todas y que tipo de información se necesita en cada lugar.

Como elección se usa una base de datos relacional en vez de una SQL, ya que en la página web se hacen más lecturas que escrituras al estar pidiendo tantos datos para mostrar por pantalla al usuario.

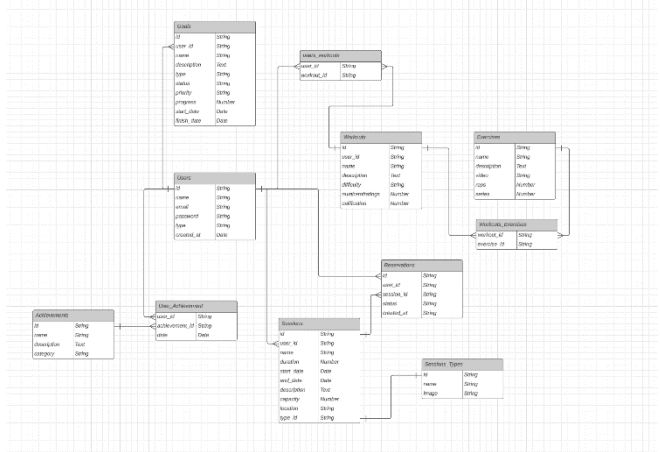


Fig. 2. Base de datos utilizada en la aplicación web.

## 7.3. Diseño de la página

Antes de comenzar a programar directamente, se ha de mostrar un ejemplo de cómo quedaría la página web al cliente y con esto que nos sugiera cambios o ideas de cómo le gustaría que fueran las vistas.

Primero, el cliente ofrece diferentes ejemplos de páginas web para tomar ideas de ellas y seguir una estructura parecida al diseñar la página. Con estas webs se comienza a utilizar Figma[20]. En ella se desarrollan diferentes vistas de la página y el flujo del usuario a través de ellas.

Una vez se tienen estas vistas, se muestran al cliente para poder tener un diseño final después de la reunión y se indican pequeños cambios y una vez realizados estos cambios, se pasa este diseño a código.

## 7.4 Contextos

Como se explica anteriormente en el documento, se utilizan contextos a lo largo de la aplicación para poder mantener variables o información que se use de manera global y de esta manera evitar peticiones innecesarias. Estos contextos se encuentran proveyendo información desde el momento que se monta el *main* para poder acceder desde cualquier lugar y porque cuando se utilizan los *WebSockets*[21] para el chat se necesitan activar desde un principio para poder escuchar todos los eventos, por ejemplo, por si recibe un mensaje, saberlo desde cualquier lugar de la aplicación.

### 7.4.1 Contexto usuario

Mientras se va navegando por la aplicación, se pueden ir mostrando datos como el nombre o hacer comprobaciones sobre el tipo de cuenta de usuario que es para mostrar algunas rutas específicas que contienen funcionalidades exclusivas para entrenadores o para usuarios.

Por lo tanto, este contexto se guarda, el nombre, el correo, el id y el tipo de usuario de la cuenta que se encuentre en ese momento activa, haciendo una petición sobre estos datos cada vez que se refresca la página porque estos datos se pierden en cada actualización de esta. Todo esto se almacena en un objeto llamado *user* y se proporciona su *setter* para ser usado en caso de registrar o iniciar la sesión y una función *logout* que permite eliminar el objeto *user* y eliminar el token existente relacionado con ese usuario del navegador.

### 7.4.2 Contexto de conversaciones

En el chat nos encontramos que queremos almacenar primero de todo en un contexto solamente con qué usuarios tenemos una conversación activa y cuál es su último mensaje de usuario para poder mostrarlo en una lista lateral que más adelante se explicará.

### 7.4.3 Contexto de mensajes de las conversaciones

También se quiere almacenar los mensajes de las conversaciones que tenemos para poder mostrarlo en tiempo real en el chat, en caso de que se pulse una de las conversaciones que tenemos. En el apartado de la funcionalidad del chat se explicará más en profundidad el uso de cada uno de los contextos y cómo se ve reflejada esta información en la interfaz y cómo se actualiza cada uno junto a los *WebSockets*.

## 7.5 Funcionalidades

Una vez todo se encuentra preparado para la codificación se procede a realizar el código de las funcionalidades descritas anteriormente.

### 7.5.1 Funcionalidad de registro de usuario

Para la página web las primeras funcionalidades a desarrollar serán el registro e inicio de sesión para así poder tener un token de identificación y acceder a las demás funcionalidades de la página autenticando el usuario ya que exceptuando las peticiones de inicio de sesión y de registro en todas las demás hará falta proporcionar el token para realizar la petición o si no se devolverá un error por no tener permisos.

Entonces para poder registrar un usuario se hace una petición POST hacia el back-end pasándole un formulario con 4 campos: nombre, email, contraseña y tipo de usuario (deportista o entrenador). Se pasa a formato JSON y se espera el resultado de la promesa para así poder asignar el token cifrado que devuelve en el *localStorage* y se redirige al usuario a la página principal. Para evitar posibles errores en el back-end por una contraseña que no sea suficientemente segura se utiliza un componente, el cual muestra por pantalla al usuario los requisitos que tiene que cumplir su contraseña y se va actualizando en tiempo real a medida que se escribe, además se deshabilita el botón para hacer la

petición en caso de que no se cumplan estos requisitos o que alguno de los campos del formulario se encuentre vacío. Si se realiza de manera correcta se guarda el contenido del usuario en el contexto de usuario.

### 7.5.2 Funcionalidad de inicio de sesión

Una vez registrado el usuario en la base de datos se procede a crear la funcionalidad de iniciar sesión para ello se hace una petición POST hacia el back-end donde a partir de otro formulario de 2 campos: email y contraseña se devuelve el token al usuario para poder ser utilizado durante un tiempo definido en el back-end. Y se guarda en el contexto de usuario la información.

En caso de que devuelva un error la petición se le indica a través de un componente el error al usuario ya sea si no existe el correo o si la contraseña es incorrecta.

### 7.5.3 Funcionalidad de reservas

La funcionalidad de las reservas es la siguiente que se procede a crear separando está en diversos apartados, una se enfoca en los deportistas y la otra en los entrenadores.

Primero se hace una petición al cargar la página, de las sesiones que existan en la base de datos junto a todos sus datos, para hacer la petición GET se pide el token del usuario para poder acceder y se guardan las posibles reservas en un objeto que se recorre después en el HTML para mostrar todos los datos.

También se hace una petición GET para obtener los IDs de las sesiones que tenga ya reservadas el usuario para así en el HTML hacer una comprobación para mostrar como deshabilitado el botón de reservar o de cancelar dependiendo de si la sesión ya se encuentra reservada o no.

Cuando el usuario decide reservar en una de las sesiones existentes se hace una petición POST, enviando como datos el ID de la sesión seleccionada y dejando de mostrar por último la pestaña del aviso emergente que sale al reservar, mostrando en la página de reservas que ya ha reservado actualizando los botones y el estado de las sesiones. Para evitar errores solo se deja reservar si las reservas actuales son menores a la capacidad máxima de la sesión.

En el caso de querer cancelar una reserva, se envía el ID de la sesión que se ha seleccionado junto al token para comprobar que es un deportista quien hace la petición y se vuelven a actualizar los botones y el estado de las sesiones que ayudan a mostrar si se puede realizar la reserva o no.

Como punto adicional se comprueba que el usuario que accede a visualizar las sesiones sea un deportista para dejarle utilizar estas funcionalidades, en caso contrario se muestra un mensaje avisando de que no se pueden usar las funcionalidades por ser un entrenador y además salen como deshabilitadas.

Pasando con las funcionalidades de los entrenadores se implementa la eliminación de sesiones ya creadas enviando una petición DELETE con el id de la sesión que se quiere eliminar junto al token para comprobar que sea un entrenador el tipo de cuenta, sino se recibe un error y se cancela la petición. Para ver que se ha actualizado correctamente esta sesión se elimina de las existentes junto a su información y se muestra un mensaje de éxito.

Otra de las funcionalidades destinada a los entrenadores exclusivamente es poder crear nuevas sesiones y que los clientes se registren en ellas, esto se consigue a través de un formulario con 8 campos para así indicar: tipo de sesión, el nombre, la descripción, la duración, la capacidad de personas, en qué lugar se hace y su fecha de inicio y de finalización con hora incluida.

Posteriormente se hace la petición POST enviando todos los datos para que sean registrados.

Hay que remarcar que los tipos de sesiones que hay se encuentran en la base de datos entonces al cargar la página se hace otra petición GET para enseñarlo en el formulario, esta información se traduce a través de una función para poder mostrar en la etiqueta *select* los nombres en español al usuario.

#### 7.5.4 Funcionalidad de crear ejercicios

Esta es otra funcionalidad destinada a los entrenadores la cual a través de un formulario de 4 campos: Nombre del ejercicio, descripción, video explicativo e imagen relacionada, se crea un entreno, hay que matizar que se indica a través del *placeholder* que la imagen y el vídeo no se pueden subir como contenido ya que no se cuenta con almacenamiento de este tipo y que el formato ha de ser una URL de internet.

Toda esta información es enviada a través de una petición POST mirando si el usuario es del tipo entrenador para permitir la realización de esta acción.

#### 7.5.5 Funcionalidad de crear entrenos

Una vez existen los ejercicios se pueden crear los entrenos, para esto primero hay que rellenar los campos de: Nombre del entreno, descripción y dificultad, como en un caso anterior las dificultades se encuentran registradas en el back-end por lo tanto se hace una petición GET para poder obtener los valores y mostrarlos al usuario una vez traducidos por una función, para poder terminar de rellenar los datos se ofrece al usuario una selección de cartas donde salen los ejercicios registrados en la base de datos enseñando el nombre del ejercicio y su imagen, a partir de aquí se deja añadir a la lista de ejercicios del formulario los ejercicios que el entrenador seleccione pero solamente en caso de rellenar los campos de series y repeticiones, en caso contrario sale el botón como deshabilitado para poder añadirlo y una vez añadido a la lista se muestra un marco rojo junto a un mensaje informativo de que este ejercicio se encuentra añadido y por lo tanto no puede volver a añadirlo y se deshabilita toda interacción con esta carta de ejercicio, además de mostrar en el formulario los ejercicios que se van seleccionados. Una vez el usuario añade todos los ejercicios que quiera se muestra un mensaje por pantalla de aviso por si quiere asignar este entreno a un usuario concreto o si lo quiere añadir como público, en caso de seleccionar que no quiere asignarlo se realiza una petición POST pasando todos los datos anteriores del formulario e indicando con los ID los ejercicios que se han seleccionado, para poder realizar esta acción se tiene que añadir como mínimo un ejercicio porque si no el botón para hacer la petición se encontrara oculto hasta el momento en que se añada un ejercicio y solo se podrá hacer si

todos los campos del formulario están rellenos.

#### 7.5.6 Funcionalidad de asignar entrenos

La funcionalidad anterior ocurre solamente si se indica que no se quiere asignar, pero en caso de querer asignarlo seleccionando la opción de "Si" se muestra un mensaje el cual contiene un buscador que muestra en un desplegable todos los posibles usuarios que existen en la base de datos para poder seleccionar a quien asignar el entreno creado, estos usuarios provienen de una petición GET y al seleccionar uno se acaba confirmando la petición a través de un botón, esta se lleva a cabo de la misma manera que la funcionalidad anterior de crear entrenos pero se añade el campo de *sporter* indicando el ID del usuario seleccionado el cual es opcional y en caso de pasarlo vacío se realiza la petición de crear entrenos públicos, un punto importante de esto es que solamente puede visualizar el entreno el usuario al que se le asigna y se trata como un entreno privado el cual se accede a partir del apartado mis entrenos.

#### 7.5.7 Funcionalidad de filtrar ejercicios

Como se describe anteriormente al crear un entreno se seleccionan uno o diversos ejercicios para completar el formulario, pero para poder acceder de una manera más fácil se añade una barra de búsqueda que permite al usuario introducir el nombre del ejercicio para encontrarlo, esto se hace a través del front-end con la función *filter* y este filtro solo se ejecuta al pulsar *Enter* en el teclado buscando coincidencias con lo que haya escrito el usuario y en caso de borrar el texto se vuelven a enseñar de manera automática todos los ejercicios existentes.

#### 7.5.8 Funcionalidad de ver entrenos

Los entrenos que se crean a través de la funcionalidad del apartado 7.4.5 son mostrados en un apartado de la web, estos son accesibles para todos los usuarios donde al cargar la página se hace una petición GET para visualizarlos todos. Estos entrenos al principio salen en una lista indicando el nombre y la dificultad, pero al ser seleccionados se muestra una carta con la descripción, los ejercicios junto a sus series y repeticiones, una imagen descriptiva del ejercicio y por último un enlace hacia el video que se haya indicado en el momento de crear el ejercicio para ofrecer contenido educativo.

Al existir una calificación para los entrenos se ha de permitir que los usuarios pueden añadir calificaciones, entonces se crea una petición que primero comprueba que sea un deportista quien realiza la calificación para poder enviarla a través de una petición POST escogiendo el valor del *rating* dado a través de un sistema de estrellas, en caso de que sea un entrenador quien lo haga se muestra un mensaje por pantalla indicando que al ser un entrenador no se puede realizar la calificación.

También se cuenta con una funcionalidad para añadir a el apartado de "mis entrenos" un entreno público que elija el usuario, se hace con una petición POST con el ID del entreno seleccionado.

#### 7.5.9 Funcionalidad filtrar entrenos

Como pueden existir muchos entrenos existen dos tipos de filtros para la búsqueda de entrenos, uno depende de la

dificultad que se seleccione y el otro de la calificación del entreno, se utilizan dos funciones *filter* para conseguir mostrar solamente los entrenos que cumplan todas las condiciones especificadas por el usuario.

#### 7.5.10 Funcionalidad de mi perfil

En este apartado primero cuando carga la página se pide a través de una petición GET los datos específicos del usuario que esta con la sesión iniciada y se muestra: el nombre, correo, las últimas reservas.

Además, se define la funcionalidad para actualizar los datos del usuario con una petición PATCH donde se envían los campos que el usuario quiera modificar, en caso de estar vacíos estos no se actualizarán y se mantendrán. Además, se añade un componente para indicar al usuario si la contraseña que introduce es válida o no en caso de que la quiera modificar y que la confirme repitiéndola.

#### 7.5.11. Funcionalidad mis entrenos

En esta vista como se ha comentado con anterioridad se podrán visualizar todos los entrenos que se hayan añadido a favoritos y sean entrenos públicos o que un entrenador nos haya asignado.

Para conseguir esto al cargar la página se llama a una petición GET que nos devuelve los datos comentados.

#### 7.5.12. Funcionalidad del chat

Por último, contamos con la funcionalidad del chat en tiempo real, la cual se encuentra subdividida en varios componentes donde en cada uno se hacen peticiones diferentes.

Primero contamos con una petición que nos devuelve cada una de las conversaciones que tengamos en ese momento junto al último mensaje y el nombre del contacto, estos datos se almacenan en el contexto de conversaciones.

Por otro lado al seleccionar una de las conversaciones descritas anteriormente se realiza una petición GET para recoger los últimos cien mensajes de aquella conversación y mostrarlos diferenciando si son recibidos o enviados con ayuda del contexto del usuario, esta información se guarda en el contexto de mensajes de las conversaciones el cual cuenta con un método para poder enviar y recibir mensajes, en este punto es donde entran en acción los WebSockets que se activan al escuchar los eventos desde el momento en el que se crea la aplicación, de esta manera se consiguen recibir mensajes todo y que el usuario se encuentre en cualquier otro apartado de la aplicación. Al recibir o enviar se indica el evento que está sucediendo a través de los sockets y se guardan los datos en el contexto, gracias a los sockets se añadirá el mensaje o la conversación en la base de datos para cuando se cargue o recargue la página se utilicen estos datos, en cambio si no se refresca la página se mostraran con los datos del contexto que son exactamente los mismos que los de la base de datos pero no hace falta que el usuario recargue para ver estos cambios y se pueda mantener una buena experiencia del usuario.

Y la última funcionalidad relacionada con el chat es poder crear conversaciones, para ello se ofrece una barra de búsqueda al usuario que al pulsar en ella muestra los datos recibidos a través de una petición GET que devuelve a

todos los entrenadores con los que no se tiene una conversación iniciada indicando en un desplegable el nombre de todos y permitiendo filtrar al escribir en la barra de búsqueda los nombres coincidentes, la pulsar en uno de los entrenadores se muestra un mensaje de confirmación para el usuario sobre si quiere iniciar una conversación con este entrenador y en caso de ser positiva la respuesta del usuario se crea y se muestra en la barra lateral de las conversaciones.

#### 7.6 Control de errores

En la gran mayoría de peticiones se hace un control de errores mirando cual es el código de error que se devuelve con la petición, en caso de ser un 401 se redirige al usuario a la página de inicio de sesión para poder volver a tener un token válido, si es un 403 se indica con una mensaje por pantalla que no se tienen permisos para poder realizar aquella funcionalidad o acción, normalmente se encuentra ya deshabilitado en el front-end para no poder ni siquiera intentar la petición pero se hace la comprobación de la misma manera y en caso de surgir otro error se indica que no se han podido cargar o enviar los datos y se indica al usuario que lo intente más tarde.

Hay casos más específicos como por ejemplo el inicio de sesión donde un 401 indica que la contraseña es incorrecta o un 404 que indica que el correo no está registrado en la base de datos.

Pero siempre se mantiene un *feedback* constante con el usuario para poder indicarle que están cargando los datos o la petición y en caso de que falle mostrarle un mensaje de error para indicarle que está pasando en todo momento.

#### 7.7. Refactorización del código

Para conseguir que este código pueda llegar a ser utilizado por otras u otros desarrolladores se tomó la decisión de refactorizar el código para aplicar algún principio de diseño y estructurar más el código para hacerlo así más escalable, entendible y legible.

Para ello se separan en la medida de lo posible la lógica de peticiones y control de estados en un `customHook[22]` y se deja solamente en las vistas la tarea de renderizar los datos que le llegan vía estas peticiones y que solamente utilice los datos que quiera utilizar de este *hook* ofreciendo escalabilidad al poder añadir más funciones, separando responsabilidades entre componentes donde cada uno se ocupa de una o dos tareas como máximo y ofreciendo una legibilidad de código mayor ya que se sabe qué hace una función por su nombre y en que vista se utiliza pero sin ver su lógica y si quieres pasar a ver su lógica solamente tienes que irte a mirar al archivo donde esta encapsulada todo esta información y realizar los cambios allí sin tener que estar pendiente que los cambios afecten a la vista.

#### 7.8. Testing

Para finalizar con la parte del desarrollo y comprobar su correcto funcionamiento se realizaron pruebas sobre el código ya finalizado con la ayuda de Jest, como ya se había explicado con anterioridad.

En total se hicieron cuarenta y dos tests en todas las vistas que utilizaban los componentes, exceptuando solamente el chat en el que se hizo una prueba manual con las



interfaces.

En las pruebas escritas se hicieron comprobaciones para asegurarse que se cargaban correctamente los componentes, en cada uno de los casos posibles como podían ser tener datos y renderizarlos, no tener datos y mostrar un mensaje por pantalla de que no existen y, por último, en caso de estar cargando los datos, informar al usuario con un mensaje.

Por otro lado, también se comprobó que en el momento de realizar una petición se llamaran a las funciones de los *hooks* correspondientes y por lo tanto el funcionamiento fuera el esperado para el usuario. En algunos componentes más sencillos se pudieron hacer comprobaciones sobre mensajes de error por pantalla en caso de realizar peticiones incorrectas.

Por último, el chat fue el único sin código de prueba escrito, pero se descargó el repositorio en dos ordenadores diferentes y se inició sesión con una cuenta de deportista y otra de entrenador para comprobar que se podían comunicar en tiempo real los usuarios.

## 8. ANÁLISIS DE RESULTADOS

Una vez finalizado el desarrollo de la aplicación se comentarán los resultados obtenidos en el trabajo; para ello se irán numerando los diferentes objetivos específicos propuestos en un principio, gracias a los cuales se podrá decidir si el objetivo general ha sido cumplido. Además, como la aplicación está destinada hacia un cliente, él mismo ofrecerá su opinión sobre cada parte de la web, valorando si es válida o no y sus respectivas opiniones o cambios que quería haber podido realizar.

### 8.1 Objetivo tipos de usuario

Se planteaba poder crear dos tipos de usuario en la página web y que cada uno contara con diferentes funcionalidades permitidas y comprobar que estos usuarios se crean y se puede acceder a sus datos a través de un inicio de sesión.

El objetivo se ha conseguido gracias a la funcionalidad de registrar usuarios e iniciar sesión con estos, la cual permite registrar a los dos tipos de usuarios especificados, donde cada uno cuenta con sus funcionalidades individuales, porque tanto en el front-end como en el back-end se comprueba que usuario es para mostrar o no las funcionalidades y/o dejar realizarlas gracias a su token identificativo y el tipo de usuario creado.

Para comprobar que se podían utilizar estos usuarios, se comprobaba su creación y su uso de datos a través de un inicio de sesión y el uso de estos datos a través del contexto de usuario a lo largo de toda la aplicación.

El cliente comentó que la funcionalidad era correcta y el diseño era de su agrado al ser simple y minimalista, pero contando con sus colores identificativos. A parte, puntuó de manera muy positiva el registro donde se podía ver en todo momento si la contraseña introducida era correcta o no y los mensajes informativos ante posibles errores que surgieran. El único punto negativo que remarcar fue que cualquier persona podría hacer una cuenta de entrenador y tener todos los permisos y funcionalidades disponibles sin ninguna comprobación; por lo tanto, puntuó esta parte

como válida, pero con ese pequeño cambio para realizar un control.

### 8.2 Objetivo mi perfil

En un principio se especificaba permitir al usuario poder modificar sus datos personales y poder visualizar en 3 apartados diferentes los datos personales, las reservas y los entrenos.

Para llevar a cabo este objetivo se han creado dos apartados, el de “mi perfil” y “mis entrenos”, y aunque se planteaban tres, en un principio se ha decidido juntar “mi perfil” y mis “reservas” en uno solo.

Se han desarrollado las peticiones para permitir al usuario modificar sus datos de manera intuitiva a través de una interfaz gracias a las funcionalidades de “mi perfil” y mostrar y guardar los entrenos que decida el usuario. Por lo tanto, se ha cumplido el objetivo satisfactoriamente.

Por parte del cliente, puntuó de manera positiva el hecho de poder ver las reservas dentro del propio perfil, ya que no ocupaba demasiado espacio y era rápido de acceder; otro punto positivo fue poder ver los entrenos de la misma manera que se ven en los entrenos generales, y por último, en la modificación de datos, ofrecer otra vez un sistema para indicar al usuario si la contraseña es válida y si la ha repetido correctamente o no, además de poder visualizarla o esconderla, validando finalmente esta parte.

### 8.3 Objetivo planificación de entrenamientos

En este objetivo, se planteaba ofrecer al usuario diferentes entrenos que se pudieran adaptar a todas las dificultades o niveles y ofrecer retroalimentación para tener *feedback* sobre estos entrenos.

Este objetivo se ha completado al desarrollar la funcionalidad de crear entrenos, que permite crear ejercicios eligiendo una dificultad para asignarlo al entreno, esta dificultad se utiliza posteriormente para poder filtrar los entrenos que se encuentren disponibles y ofrecer al usuario un entreno adaptado a sus necesidades e incluso ofrecer entrenos privados específicos para cada uno de los usuarios. Para cubrir la parte del *feedback* se ofrece un sistema de puntuación vía *rating* en la interfaz del usuario para que estos puedan calificar los entrenos y ofrecer a los entrenadores una visión general.

El cliente valoró positivamente el hecho de tener un buscador en los ejercicios, el cuál fue añadido de manera opcional, contar con una visualización de imágenes de los ejercicios y el diseño de las cartas de los entrenamientos junto al *rating* y poder añadir los entrenos a favoritos.

Como puntos negativos remarcó que quizá se podía haber realizado un filtro por nombre de entrenos también, la estructura de los filtros y el listado para formato móvil lo encontraba correcto, pero para ordenador le hubiera gustado un poco más separado y como último punto remarcó que se hubiera añadido un campo de tiempo a parte de los de series y repeticiones, validando de manera correcta esta parte, pero con unos cuantos puntos a ser modificados o añadidos.

### 8.4 Objetivo progreso del usuario

Se planteaban realizar una interfaz para permitir al usuario añadir metas u objetivos y poder ver el progreso propio a

través de gráficas.

No se ha conseguido completar este objetivo por una falta de consideración de tareas a completar y un tiempo de desarrollo superior en otras funcionalidades. En un apartado posterior se explica con exactitud cuales han sido los problemas que han surgido para no poder completar este objetivo.

No existe *feedback* del cliente en este caso al no estar realizado.

### 8.5 Objetivo contenido educativo

En este objetivo, se pretende ofrecer al usuario autonomía en el momento de utilizar la aplicación y ofrecer datos relacionados con el fitness.

Este objetivo se ha conseguido completar ofreciendo en la funcionalidad de crear entrenos un apartado específico para ofrecer un video explicativo sobre los ejercicios que se realizan, además una imagen representativa y por último una pequeña descripción de cómo se debería realizar el ejercicio y que parte del cuerpo trabaja, de esta manera en el momento que el usuario acceda a los entrenos se podrá visualizar toda esta información.

El cliente validó esta parte, ya que las imágenes, los videos explicativos y las descripciones podían ser propias siempre y cuando estuvieran en internet, todo y que se comentó una posible mejora añadiendo un apartado de consejos o preguntas frecuentes a donde se podrían redirigir a los usuarios a foros de opinión o estudios de interés realizados por expertos.

### 8.6. Objetivo interfaz reservas

Se había propuesto crear una interfaz sencilla y amigable para el usuario que dejara realizar reservas en la propia página web.

Este objetivo está completo gracias a las funcionalidades que ofrecen visualizar todas las reservas creadas junto a su información; permite reservar y cancelar las reservas e incluso se ofrece eliminar reservas a los entrenadores. La interfaz es sencilla, ofreciendo botones para realizar acciones y mensajes de confirmación para realizar todo esto.

Esta parte fue validada correctamente y se comentó que tanto la visualización, la información añadida y la facilidad para crear, cancelar y eliminar estas reservas eran muy intuitivas y sencillas para cualquiera. Destacó que los mensajes de confirmación al realizar una acción también eran un plus y que el eliminado automático de las reservas ya caducadas era perfecto para no tener que hacer todo manualmente.

### 8.7 Objetivo chat

Se planteaba ofrecer un chat bidireccional privado entre entrenadores y deportistas, ofreciendo mensajería y creación de chats junto a notificaciones.

Este objetivo se ha cumplido en casi su totalidad gracias a las funcionalidades que ofrecen mensajería en tiempo real y un chat privado entre entrenadores y deportistas, contando también con la posibilidad de crear conversaciones.

El único punto que no se ha acabado consiguiendo ha sido el de notificaciones dado que el desarrollo de esta funcionalidad ha sido superior al esperado.

El cliente puntuó de manera positiva el diseño, que fuera en tiempo real el chat y el buscador que permitía crear conversaciones; el único punto negativo fue el hecho de no poder ser notificado desde otra parte de la aplicación, todo y que el mensaje llegara, validando esta parte, pero con ganas de poder llegar a tener las notificaciones en un futuro.

### 8.8 Objetivo gamificación

Este objetivo especificaba ofrecer la creación de logros para que los usuarios lo pudieran completar y ofrecer una tabla de clasificación para fomentar la competitividad entre usuarios y por lo tanto el uso de la aplicación.

Este objetivo ha sido substituido por otra funcionalidad, ya que al realizarse una reunión con el cliente solicitó poder crear entrenos privados, lo cual en un principio no estaba planteado y solamente se iban a crear entrenos públicos.

Este nuevo objetivo ha sido completado con éxito gracias a la funcionalidad que permite crear un entreno tanto público como privado y asignarlo directamente a un usuario.

El cliente validó este apartado, ya que se creaba de la misma manera el entreno privado como el público, y la única diferencia era el selector de un usuario que funcionaba correctamente, pero el estilo que tenía no seguía del todo la temática de la página y fue puntuado negativamente y le hubiera gustado añadir una notificación para cuando los usuarios al añadirle un entreno.

### 8.9. Objetivo compartir

Se había propuesto poder compartir entrenos, logros u objetivos completados en la aplicación a través de redes sociales.

Este objetivo, al no haberse realizado ni los logros ni los objetivos de progreso, se intentó implementar en la función de entrenos para poder compartirse estos últimos. Pero surgió un inconveniente no conocido en la parte del back-end que no dejó proseguir con esta funcionalidad y, por lo tanto, no existe *feedback* del cliente.

### 8.10 Problemas surgidos

Hubo un par de problemáticas las cuales retrasaron el desarrollo de algunas funcionalidades y por lo tanto no se pudieron cumplir algunos de los objetivos como se había explicado anteriormente.

Estos problemas fueron no tener en cuenta el tiempo de diseño de la aplicación previa al desarrollo, ya que se tuvo en cuenta solamente la finalización de los casos de uso refinado y el comienzo con el desarrollo; por otra parte, el tiempo que llevó poder conseguir la funcionalidad del chat fue superior al esperado y acaparo más tiempo del planificado. Además, al ir integrando algunas subfuncionalidades extra como poder encontrar a todos los usuarios, a todos los entrenadores, algunos filtros añadidos que no se pedían y el tiempo de realizar el diseño responsive han provocado no poder cumplir con todos los objetivos específicos.

### 8.11 Objetivo general

El objetivo de este trabajo era promover la actividad física

a través de una aplicación web que en concreto fuera amigable para el usuario y estuviera personalizada.

Este objetivo se ha cumplido todo y que no todos los objetivos específicos se hayan cumplido ya que igualmente se acaba ofreciendo una web para un cliente con el que se ha estado en contacto para poder recibir *feedback* sobre la página y se ha moldeado en base a sus opiniones y funcionalidades de entrenos, reservas, perfiles, chat... conformando una aplicación sólida y funcional para los usuarios.

Se considera que, pese a ser un trabajo relativamente grande y que no se haya completado del todo, se han conseguido resultados muy buenos en el tiempo estipulado por el trabajo de solamente dos personas.

## 9. CONCLUSIONES

El proyecto ha aportado significativamente al ámbito del *fitness* digital al proporcionar una plataforma que promueve un estilo de vida activo y saludable. Los elementos más importantes del proyecto fueron completados con éxito, ofreciendo una experiencia de usuario sólida y funcionalidades esenciales para la mejora del bienestar físico. No obstante, algunos elementos no fueron tratados y algunos objetivos no se completaron. En particular, ciertas funcionalidades avanzadas que se consideraron como de menor importancia a las otras desarrolladas, como son las de gamificación y elementos de la interfaz de progreso de usuario.

Los problemas surgidos durante el desarrollo permitieron identificar y priorizar las funcionalidades más importantes para cumplir con el objetivo general, incluso sin completar todas las funcionalidades planeadas. Gracias a esto se vio la importancia de la planificación detallada y la consideración de todos los aspectos necesarios para la completitud del proyecto. De cara a un futuro, se pretende solventar estas problemáticas cambiando la planificación para poder realizar un *feature* por semana en vez de dos para así tener un tiempo suficiente para cumplir con todos los objetivos, todo y que el abasto del trabajo sea menor y, por supuesto, tener en cuenta el tiempo de preparación previo de diseño y de preparación con el lenguaje en caso de ser la primera vez programando con él.

En términos de posibles extensiones del trabajo, se podrían completar las funcionalidades pendientes de gamificación y progreso, mejorar la interfaz de usuario con más detalles interactivos, añadiendo las notificaciones de mensajes o confirmaciones de reserva. También sería beneficioso integrar tecnologías emergentes como la realidad virtual y aumentada para enriquecer la experiencia del usuario. Ampliar la personalización mediante análisis de datos y algoritmos de inteligencia artificial, así como mejorar la accesibilidad y la diversidad de contenidos educativos.

En cuanto a la elección de tecnologías, se consideran varios aciertos y algunas áreas de mejora. React fue una elección acertada debido a la abundante información disponible, la extensa documentación y la arquitectura de componentes reutilizables que aceleran el trabajo. HTML, trabajando juntamente con React, también se considera un acierto. Sin embargo, se podría haber utilizado TypeScript en lugar de JavaScript para un mayor control de errores y

mejorar la robustez del código, especialmente en previsión de futuras ampliaciones de la aplicación. Además, substituir CSS3 por Tailwind CSS podría haber acelerado el proceso de codificación de estilos, manteniendo la personalización de la página y acelerando el proceso de crear el diseño *responsive*; por último, se considera un acierto el uso de Jest para la codificación de pruebas unitarias y de integración de manera eficiente, asegurando que cada componente de la aplicación funcione correctamente antes de proceder con su integración total.

Se considera que este trabajo demuestra como el ámbito tecnológico puede también evitar el sedentarismo en vez de promoverlo con aplicaciones como esta y que la tecnología no tiene por qué ser antónimo de una vida enérgica.

## AGRADECIMIENTOS

Me gustaría agradecer la realización de este trabajo primero de todo a Ana Oropesa Física por tutorizarme a lo largo del trabajo, responder de manera rápida a las dudas que surgían y estar pendiente sobre cómo iba progresando todo. Y por otro lado también me gustaría agradecer a mi compañero Samuel Vitales Ponce que ha trabajado en la parte del back-end de la aplicación y he tenido una muy buena comunicación con él y hemos realizado un muy buen trabajo en equipo para poder completar la web.

## BIBLIOGRAFÍA

- [1] Pérez Púa, Y. (2023). Propuesta metodológica para mejorar la competitividad a través de la digitalización del capital humano de los centros fitness. Recuperado de <http://hdl.handle.net/10609/148680>
- [2] "Diagrama de Gantt". [en línea] (2024). Wikipedia. Disponible en: [https://es.wikipedia.org/wiki/Diagrama\\_de\\_Gantt](https://es.wikipedia.org/wiki/Diagrama_de_Gantt)
- [3] "Metodología Scrum". [en línea] (s.f.). MentorDay. Disponible en: <https://mentorday.es/wikityps/metodologia-scrum/>
- [4] "¿Qué es DevOps?". [en línea] (s.f.). Microsoft Azure. Disponible en: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops>
- [5] "¿Qué es AWS?". [en línea] (s.f.). Amazon Web Services. Disponible en: <https://aws.amazon.com/es/what-is-aws/>
- [6] "Amazon Amplify". [en línea] (s.f.). Amazon Web Services. Disponible en: <https://aws.amazon.com/es/amplify/>
- [7] "Realidad virtual, 'gaming' y fitness 3.0: así es el metaverso que sacude las bases del entrenamiento". [en línea] (s.f.). 2Playbook. Disponible en: [https://www.2playbook.com/fitness/realidad-virtual-gaming-fitness-30-es-metaverso-sacude-bases-entrenamiento\\_5701\\_102.html](https://www.2playbook.com/fitness/realidad-virtual-gaming-fitness-30-es-metaverso-sacude-bases-entrenamiento_5701_102.html)
- [8] "¿Cómo está transformando la IA el mundo del fitness?". [en línea] (s.f.) HSNBlog. Disponible en: <https://www.hsnstore.com/blog/deportes/fitness/uso-de-inteligencia-artificial/>
- [9] "Las preferencias de los programadores para 2023, según Stack Overflow" [en línea] (2023), campusMVP. Disponible en: <https://www.campusmvp.es/recursos/post/resultados-encuesta-stack-overflow-2023.aspx>
- [10] "Top 12 React UI Frameworks You Should Know in 2023". [en línea] (2023) iFour. Disponible en: <https://www.ifourtechnology.com/blog/top-12-react-ui-frameworks-you-should-know-in-2023>
- [11] "Virtual DOM and Internals" [en línea] (2024). React. Disponible en: <https://legacy.reactjs.org/docs/faq-internals.html>
- [12] "createContext" [en línea] (2024). React. Disponible en: <https://react.dev/reference/react/createContext>
- [13] "Framework" [en línea] (2024). Wikipedia. Disponible en:

- <https://es.wikipedia.org/wiki/Framework>
- [14] "HTML: HyperText Markup Language" [en línea]. (2024). Mdn web docs. Disponible en: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [15] "CSS: Cascading Style Sheets" [en línea] (2024). Mdn web docs. Disponible en: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [16] "Getting started" [en línea] (2024). Jest. <https://archive.jestjs.io/docs/en/22.x/getting-started>
- [17] "Qué es una single page application, cómo funciona y ejemplo" [en línea] (2023) Disponible en: <https://blog.hubspot.es/web-site/que-es-single-page-application>
- [18] "¿Qué es una one-page? Estructura y ejemplos" [en línea] (2022) NeoAttack. Disponible en: <https://neoattack.com/blog/one-page/>
- [19] "createBrowserRouter" [en línea]. (s.f.) React Router. Disponible en: <https://reactrouter.com/en/main/routers/create-browser-router>
- [20] "Figma: qué es y cómo funciona" [en línea]. (2023) Pixarprinting. Disponible en: <https://www.pixartprinting.es/blog/figma-que-es/>
- [21] "Introduction" [en línea]. (2024) Socket.io Disponible en. <https://socket.io/docs/v4/>
- [22] "Reusing Logic with Custom Hooks" [en línea] (2024) React Disponible en: <https://react.dev/learn/reusing-logic-with-custom-hooks>



### A6. CASO DE USO PROGRESO



### A7. CASO DE USO RESERVAS

