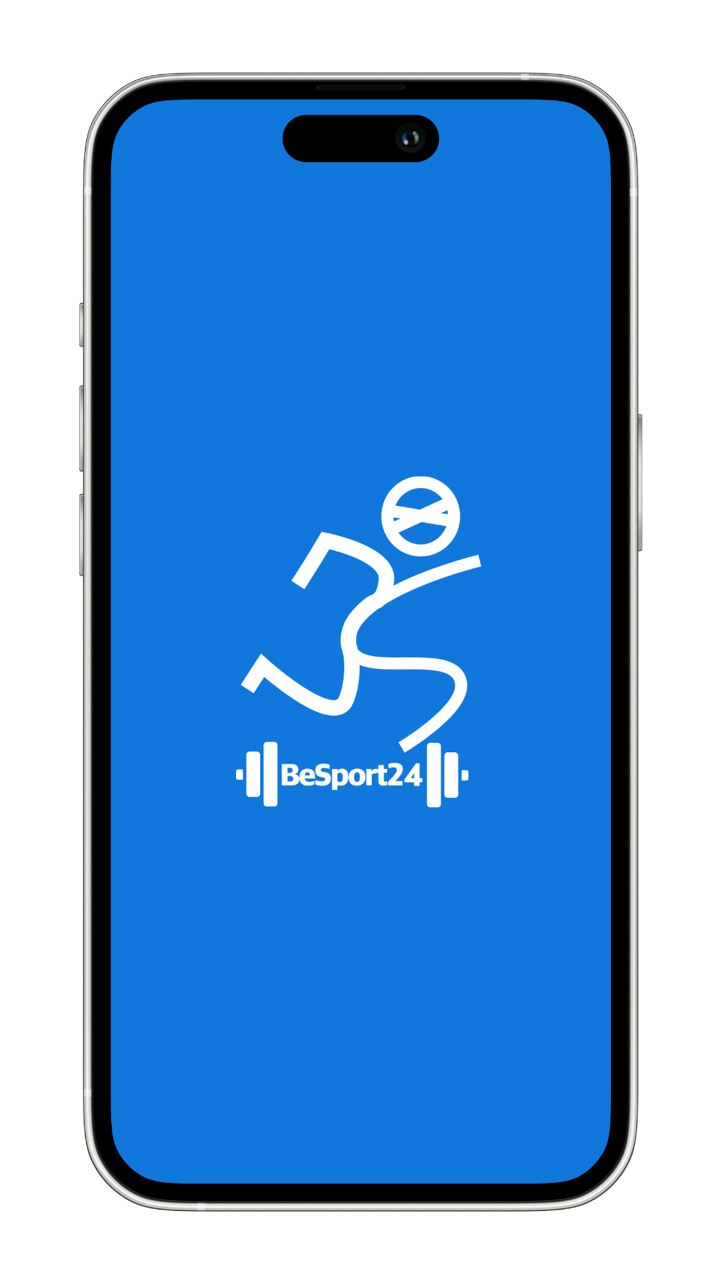


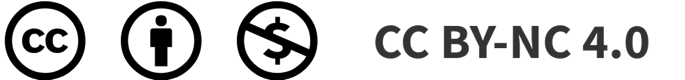
****

**BeSport2*4***Training

Alumno: David Peñalver Navarro

Curso: 2º DAW

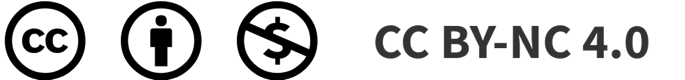
Promoción: 2023/2025



Trabajo bajo licencia:

**Atribución-No Comercial-Compartir Igual CC BY-NC-SA**

Usted puede compartir, adaptar y construir sobre este trabajo, siempre y cuando se acredite adecuadamente y no se utilice con fines comerciales y cualquier trabajo derivado se acredite bajo la misma licencia.



*Gracias a todas las personas que me han acompañado en este viaje de más de 29 años.*

*En especial, a mi familia por darme soporte y ayuda siempre que lo he necesitado. A mis amigos por esos ratos de confesiones, de risas y de momentos mágicos. A mi pareja por convertirse en un pilar fundamental en mi vida.*

*A mis profesores por todo lo que me han enseñado en estos dos años de aventura. A Flit2Go por todos estos aprendizajes y nuevos conocimientos.*

Índice

[Introducción 7](#_Toc194785375)

[Temporalización 9](#_Toc194785376)

[Herramientas y tecnologías 14](#_Toc194785377)

[Herramientas 14](#_Toc194785378)

[Macintosh 14](#_Toc194785379)

[iPhone 14](#_Toc194785380)

[PC Linux 15](#_Toc194785381)

[IntelliJ IDEA 15](#_Toc194785382)

[Visual Studio Code 16](#_Toc194785383)

[Xcode 16](#_Toc194785384)

[Simulador de Xcode 17](#_Toc194785385)

[GitHub 18](#_Toc194785386)

[Canva 19](#_Toc194785387)

[Draw.io 19](#_Toc194785388)

[Warp 19](#_Toc194785389)

[Konsole 20](#_Toc194785390)

[Tecnologías 20](#_Toc194785391)

[Git 20](#_Toc194785392)

[MariaDB 21](#_Toc194785393)

[Java 21](#_Toc194785394)

[Spring 21](#_Toc194785395)

[Maven 22](#_Toc194785396)

[JPA 22](#_Toc194785397)

[Lombok 23](#_Toc194785398)

[MapStruct 23](#_Toc194785399)

[CORS 23](#_Toc194785400)

[Postman 24](#_Toc194785401)

[Swift 24](#_Toc194785402)

[SwiftUI 24](#_Toc194785403)

[SFSymbols 25](#_Toc194785404)

[Implementación 26](#_Toc194785405)

[Análisis 26](#_Toc194785406)

[Diseño 26](#_Toc194785407)

[Manual de usuario 26](#_Toc194785408)

[Propuestas de mejora 27](#_Toc194785409)

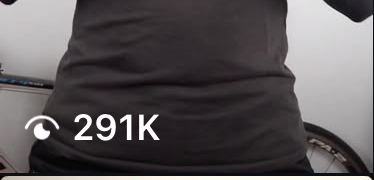
[Valoración personal 30](#_Toc194785410)

[Puntos a destacar del proyecto 31](#_Toc194785411)

[Referencias 32](#_Toc194785412)

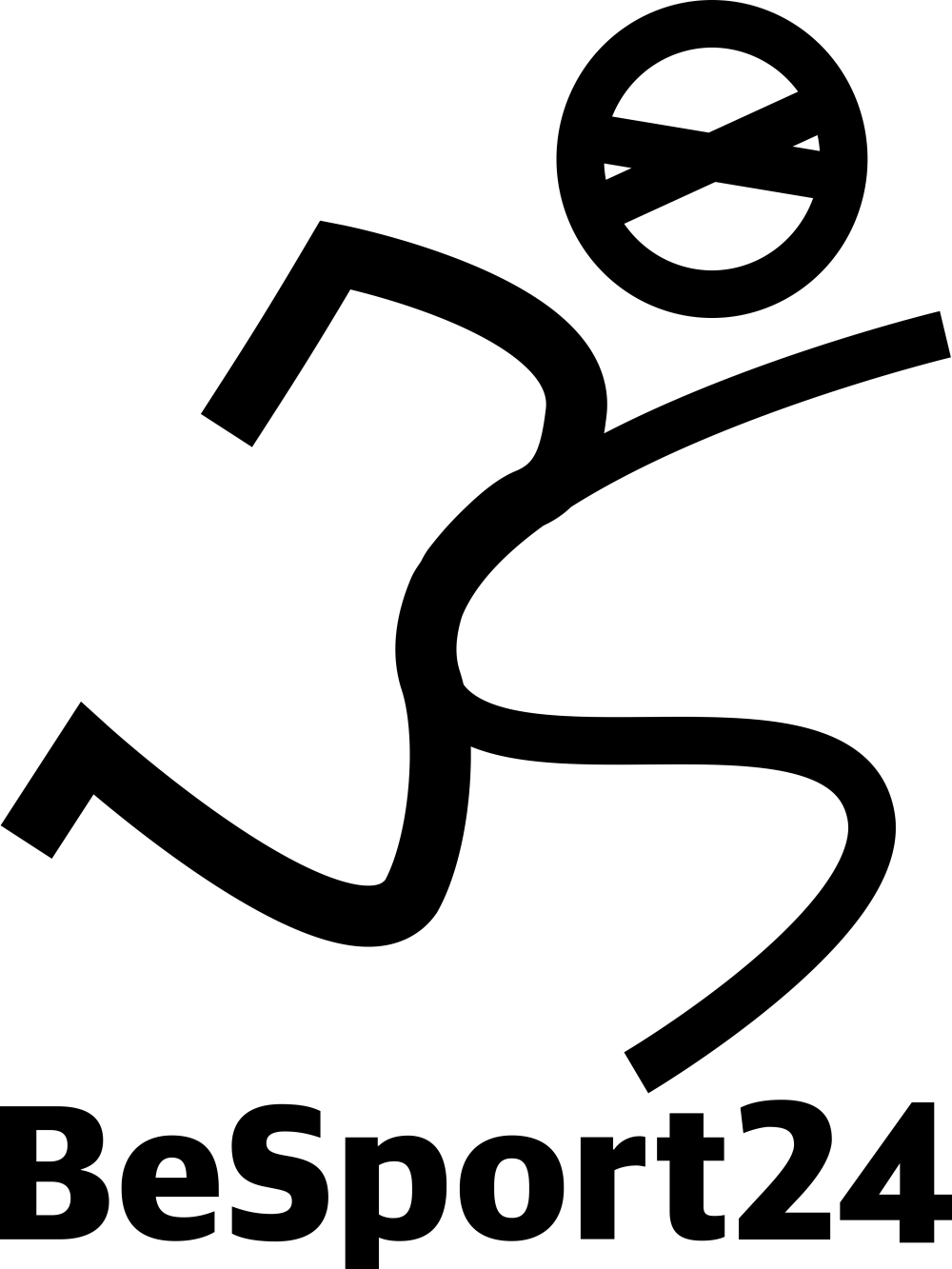
# Introducción

BeSport24 fue un proyecto de prácticamente cinco años de duración que combinó los entrenamientos presenciales con los entrenamientos online y la divulgación científica en plataformas como Instagram y YouTube. En dicha era de mi vida, el alcance del proyecto fue masivo, llegando a cientos de miles de visualizaciones y ayudando personalmente a decenas de personas a conseguir sus ansiados objetivos. Dichos objetivos fueron muy variados: desde futbolistas semiprofesionales que aspiraban a un puesto en Estados Unidos a pacientes oncológicos.



*Figura 1. Visualizaciones en uno de los Reels de Instagram.*

Tal y como dice la primera palabra de esta introducción, dicho proyecto se llamó BeSport24, ya que reflejaba la filosofía de que un estilo de vida saludable se logra a lo largo del día y no solamente en una hora de entrenamiento, por lo que se buscaba ese enfoque holístico. Asimismo, todo ello se llevaba a la práctica en base a la evidencia científica más actualizada, pudiéndose encontrar en la cuenta de Instagram centenares de vídeo-resúmenes de dichos *papers* o artículos científicos de cara a que la ciencia fuera más accesibles a la población general.



*Figura 2. Logo de BeSport24.*

Sin embargo, siempre había *algo* que se echaba en falta en ese proyecto: una aplicación de seguimiento de los entrenamientos y de valoración de cada uno de los pacientes. De esta forma, el seguimiento del entrenamiento sería eficiente y eficaz tanto para mis deportistas presenciales como para aquellos que entrenaban en México, Uruguay, Italia o Estados Unidos.

Tuve la ocasión de probar diferentes alternativas: un Excel de elaboración propia con macros, documentos PDF con las programaciones o aplicaciones web como Trainiks. Pero nada de eso me convencía. Necesitaba algo mucho más completo e intuitivo para el cliente y, al mismo tiempo, que me ahorrara faena en mis interminables jornadas laborales que duraban no de sol a sol, sino desde antes de la salida hasta la puesta de nuestro astro rey.

Es por ello que nace este proyecto como idea de TFG: la aplicación con la que tanto soñé para poder planificar, valorar y evaluar todo el proceso de entrenamiento.

En cuanto al nombre de esta aplicación, combina el nombre del proyecto de cara a una mayor visibilidad junto a la palabra entrenamiento en inglés (“training”) debido a su enfoque internacional. Asimismo, el nombre del proyecto está en negrita para diferenciarlo de “Training”, con el “4” en negrita y cursiva jugando como nexo de unión debido a su fonética similar a “for” (para, en inglés).

# Temporalización

De cara a la monitorización del avance del proyecto, se ha empleado una plantilla del diagrama de Gantt desarrollada por Vertex42. Dicha plantilla facilita la elaboración del tiempo de cada proyecto, conocer el progreso y las fechas concretas.

A continuación, se presenta el diagrama completo que, posteriormente, será desgranado en cada apartado para poder profundizar en mayor medida en la evolución del proyecto:

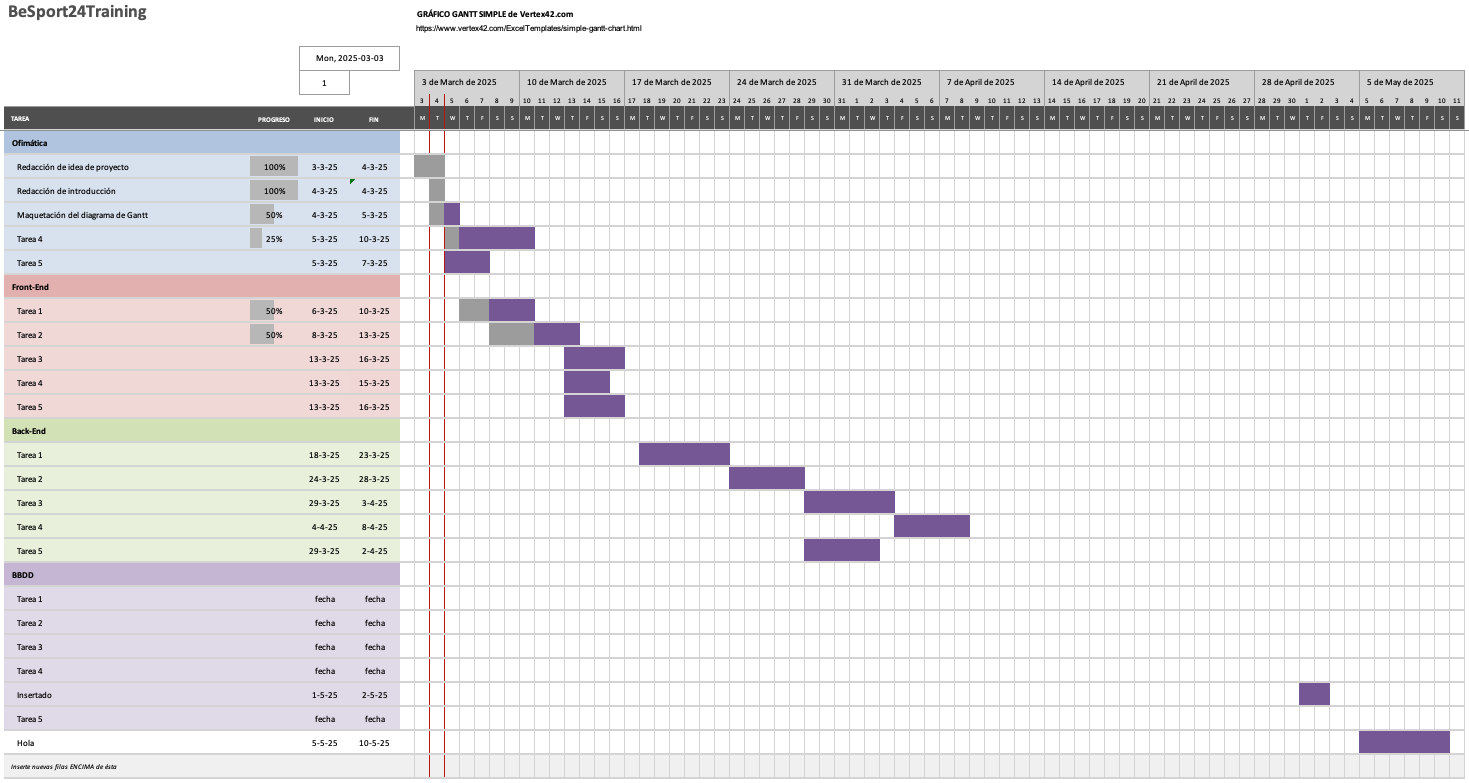
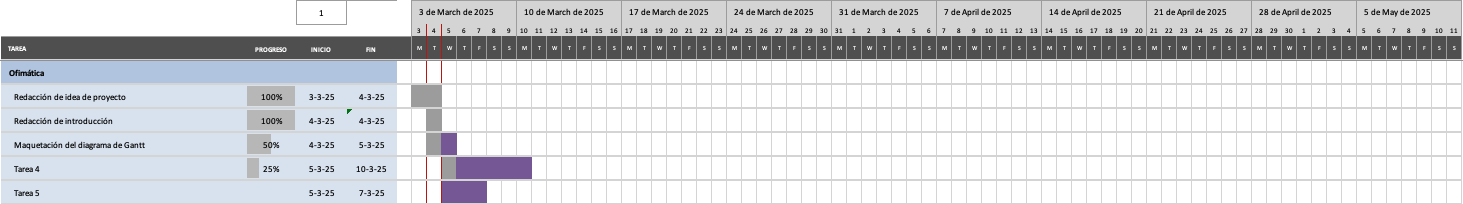


Figura 3. Diagrama de Gantt de **BeSport2*4***Training. CAMBIAR POR DEFINITIVAS.

En la primera fase, la de ofimática, tenemos las siguientes tareas: CAMBIAR POR DEFINITIVAS.



*Figura 4. Diagrama de Gantt: Ofimática.*

Tal y como se puede apreciar, el primer paso fue el de la elaboración de la idea del proyecto a realizar y la de la maquetación del trabajo escrito, de cara a ver qué tiempo tenía disponible, qué opciones se podían implantar, qué era necesario hacer y qué calendario debía estructurarme y cómo organizar el trabajo escrito de cara a plasmar todo aquello que habría desarrollado desde el inicio hasta la entrega.

Una vez maquetado cómo sería el trabajo y viendo qué tenía que hacer y, aproximadamente, cuándo, llegaba el turno de realizar el diagrama de Gantt de cara a organizar todo el trabajo de manera clara, precisa y concisa.

En cuanto a la maquetación, esta no es estricta, sino que cada semana se va analizando la viabilidad y ajustando.

Comenzando ya con la parte visible de este proyecto, entre el cuatro y cinco de marzo se realizó la redacción de la introducción.

Tal y como se puede observar, en los días siguientes se produjo una pausa, debida al viaje a Colonia para desconectar unos días del periodo final de exámenes, el comienzo de esta tarea y el trabajo por las tardes en el centro de entrenamiento personal. Posteriormente, se redactaron los ítems a realizar en cada fase del proyecto, se realizó el diagrama de casos de uso y se diseñó el logo de la aplicación, el cual se puede ver en la portada del trabajo y en el apartado de diseño, al igual que el diagrama también comentado en este párrafo.

SEGUIR TRAS LOGO

En la segunda fase, la de Base de Datos (BBDD), se realizaron las siguientes tareas:



*Figura 5. Diagrama de Gantt: BBDD.*

En este apartado, el primer paso consistió en la elaboración del diagrama entidad-relación de cara a cumplimentar los casos del diagrama de casos de uso satisfactoriamente. El diagrama en sí no fue excesivamente laborioso, si bien se alargó varios días de cara a asegurarme de que reunía todos los requisitos necesarios, sobre todo en el ámbito de poder plasmar correctamente las necesidades de cada sesión: el número de series de cada ejercicio, las repeticiones, el tipo del ejercicio o el subtipo para poder filtrarlos y registrarlos correctamente en cada entrenamiento.

Tras ello, se procedió a realizar la base de datos propiamente dicha. En primer lugar, se crearon las tablas basándome en el mapa entidad-relación para, posteriormente, insertar los diferentes registros.

En la tercera fase, la de *Back-End*, se llevaron a cabo las siguientes tareas:



*Figura 6. Diagrama de Gantt: Back-End.*

REDACTAR LO QUE SE VAYA HACIENDO

En la última fase, la de Front-End, disponemos de las siguientes tareas:



*Figura 7. Diagrama de Gantt: Front-En**d.*

Finalmente, se realizó el front-end de la aplicación. Debido a las tecnologías vistas durante el periodo de prácticas y, al mismo tiempo, a la naturaleza de las sesiones de entrenamiento, escogí realizarlo con el lenguaje de programación nativo de Apple y con su correspondiente *framework*: Swift y SwiftUI, respectivamente.

Por ello, las primeras fases de este periodo fueron las de aprendizaje tanto de las peculiaridades del lenguaje de programación como de su correspondiente framework. En este periodo de aprendizaje, destaco las diferencias en la sintaxis del lenguaje respecto a lo que estaba acostumbrado a hacer, que era Java y JavaScript. Asimismo, al ser un lenguaje orientado al dispositivo móvil (específicamente el iPhone), SwiftUI cuenta con muchas peculiaridades que no había visto previamente. Por estos motivos, esta fase se prolongó tanto en el tiempo.

REDACTAR LO QUE SE VAYA HACIENDO

# Herramientas y tecnologías

De cara a poder hacer este posible este proyecto y cumplir con la temporalización previa, han sido necesarias numerosas herramientas y tecnologías que se describen a continuación.

## Herramientas

El listado de herramientas empleadas es el siguiente:

### Macintosh

Macintosh o Mac es la línea de ordenadores personales diseñada por el gigante americano Apple.

El equipo Macintosh se ha utilizado como entorno principal de desarrollo para la parte del proyecto orientada a iOS, dado que el sistema operativo macOS es requisito para ejecutar herramientas como Xcode y el simulador de iPhone.

Para ser más concretos, el equipo empleado ha sido un Mac Mini con procesador M1 de 256GB de almacenamiento y 8GB de memoria RAM unificada, el cual emplea la arquitectura ARM en su CPU, permitiéndole una gran eficiencia a la par que potencia por watio.

El sistema operativo sobre el que ha corrido esta computadora es macOS Sequia 15.0.1.

### iPhone

El iPhone ha sido el dispositivo móvil utilizado para realizar pruebas reales de la aplicación, permitiendo validar su funcionamiento fuera del entorno simulado.

Para ser más concretos, se ha empleado un iPhone 11 Pro con 256GB de almacenamiento interno.

### PC Linux

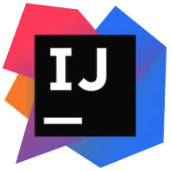
  

Un ordenador portátil con sistema operativo Linux se ha utilizado para el desarrollo del backend en Java. Asimismo, ha sido el equipo empleado como servidor a la hora de realizar las pruebas en un entorno real.

El equipo concreto es un Lenovo Ideapad Gaming 3 Gen 6 con CPU AMD Ryzen de la gama 5000H, 16GB de memoria RAM DDR5, 512GB de almacenamiento interno y GPU Nvidia GeForce RTX 3050.

El sistema operativo sobre el que corre es KDE Plasma 6.

### IntelliJ IDEA



IntelliJ IDEA es un entorno de desarrollo integrado (IDE) desarrollado por JetBrains, ampliamente reconocido por su potencia, velocidad y herramientas avanzadas de asistencia al programador.

Una de las características más destacadas de IntelliJ IDEA es su sistema de autocompletado inteligente, el análisis estático del código y la detección temprana de errores, lo que ha permitido un desarrollo más ágil y con menor margen de fallos.

En el contexto de este proyecto, se ha utilizado principalmente para el desarrollo del backend en Java. Se ha empleado la versión IntelliJ Idea Ultimate gracias a la cuenta de estudiante de CIPFP Mislata.

### Visual Studio Code



Visual Studio Code, desarrollado por Microsoft, es un editor de código fuente ligero, multiplataforma y de código abierto, que se ha convertido en una de las herramientas más populares entre desarrolladores de todo el mundo debido a que su instalación y uso son gratuitos.

Una de las grandes ventajas de VS Code es su sistema de extensiones, que permite ampliar sus funcionalidades de manera sencilla. Gracias a ello, se ha empleado para la elaboración de la base de datos y de las inserciones iniciales en la misma.

### Xcode



Xcode es el entorno de desarrollo integrado (IDE) oficial de Apple para macOS, utilizado para desarrollar aplicaciones para iOS, macOS, watchOS y tvOS. En el contexto de este proyecto, Xcode ha sido fundamental para construir y probar la aplicación destinada a dispositivos iOS, ya que proporciona todas las herramientas necesarias para compilar, depurar y lanza la aplicación.

Uno de los componentes más valiosos de Xcode ha sido su diseñador visual, especialmente útil para trabajar con **SwiftUI**, el framework de diseño de interfaces moderno de Apple. Gracias a su vista previa en tiempo real (*canvas*), ha sido posible ajustar de forma visual cada pantalla de la aplicación, comprobando cómo se adaptaba a distintos tamaños de dispositivo.

### Simulador de Xcode



El simulador de iOS es una herramienta clave para probar la aplicación sin necesidad de disponer físicamente de un iPhone o iPad en todo momento. Esto ha sido especialmente útil para validar comportamientos, realizar pruebas de diseño y verificar la navegación entre pantallas.

Una de las ventajas que tiene es que permite simular cualquier dispositivo iOS/iPad OS, ya sea iPhone o iPad, para poder comprobar que el aspecto visual y la funcionalidad es la adecuada en toda la gama de productos Apple en los que puede funcionar la aplicación.



*Figura 8. Ejemplo de visualización del simulador de un iPhone 16 Pro.*



*Figura 8. Ejemplo de visualización del simulador de un iPad 10ª generación.*

### GitHub



GitHub es una plataforma de desarrollo colaborativo basada en la web que utiliza el sistema de control de versiones Git. Ha sido una herramienta central en este proyecto, ya que ha permitido llevar un control detallado y organizado del código fuente y la documentación del mismo.

Gracias a su sistema de repositorios y ramas, ha sido posible dividir el trabajo en funcionalidades concretas sin afectar el desarrollo general, lo que ha favorecido un flujo de trabajo limpio y estructurado.

Por último, GitHub también ha servido como plataforma para alojar de forma segura el proyecto, permitiendo el acceso desde cualquier dispositivo con conexión a Internet, ya que recordemos que dos equipos distintos se han empleado para llevar a cabo esta tarea.

### Canva



Canva es una plataforma de diseño gráfico en línea que permite crear contenido visual de manera rápida, intuitiva y profesional, sin necesidad de conocimientos avanzados en diseño. Su interfaz basada en el sistema de arrastrar y soltar, junto con una amplia biblioteca de plantillas, iconos, imágenes y tipografías, la convierte en una herramienta ideal para crear elementos gráficos que complementen un proyecto, como presentaciones, carteles, logotipos o infografías.

En el presente proyecto, Canva se ha empleado para la realización del logo y materiales gráficos de este proyecto tales como la portada.

### Draw.io



Draw.io es una herramienta en línea gratuita y de código abierto que permite crear diagramas de manera sencilla y visual.

En este proyecto ha resultado útil para diseñar el diagrama de casos de uso y el diagrama de entidad-relación.

### Warp



Warp es una terminal moderna para macOS que permite autocompletado inteligente, historial navegable, bloques de comandos interactivos y compatibilidad con atajos modernos.

Durante el desarrollo de este proyecto, Warp ha sido una herramienta muy útil para interactuar con el sistemas de control de versiones en macOS.

### Konsole



Konsole es el emulador de terminal predeterminado del entorno de escritorio KDE Plasma en sistemas Linux.

Ofrece funcionalidades como pestañas múltiples, la cual ha sido ampliamente usada de cara a abrir en una pestaña la base de datos, en otra el servidor local, en otra el explorador de archivos Ranger y en otra situarme en el proyecto Java para ejecutar los comandos de Maven.

## Tecnologías

Por lo que respecta a las tecnologías, se han empleado las que se muestran a continuación:

### Git



Git es un sistema de control de versiones de código abierto desarrollado por Linus Torvald que permite gestionar el código fuente de manera eficiente.

Permite realizar seguimientos de cambios, crear ramas para nuevas funcionalidades y fusionar diferentes versiones de código sin temor a perder información. En el proyecto, Git se ha utilizado para gestionar el repositorio de código en sus distintas versiones: base de datos, backend, frontend y documentación.

### MariaDB



MariaDB es un sistema de gestión de bases de datos relacional de código abierto, desarrollado como un fork de MySQL. Actualmente, mantiene las mismas funcionales y características que este último, el cual está mantenido por Oracle.

En el contexto de este proyecto, MariaDB ha sido utilizada como la base de datos para almacenar la información de los usuarios, sesiones de entrenamiento y ejercicios.

### Java



Java es un lenguaje de programación orientado a objetos que se utiliza ampliamente en el desarrollo de aplicaciones empresariales, móviles y de servidor. Es conocido por su portabilidad, ya que las aplicaciones escritas en Java pueden ejecutarse en cualquier dispositivo que tenga instalada una Máquina Virtual Java (JVM).

En este proyecto, Java ha sido empleado para desarrollar la parte del backend, donde se gestionan las solicitudes, la lógica de negocio y la conexión con la base de datos mediante MariaDB.

### Spring



Spring Boot es un framework de Java que simplifica el proceso de creación de aplicaciones empresariales. Su principal objetivo es aumentar la productividad de los desarrolladores al proporcionar configuraciones predeterminadas sensatas y un conjunto de herramientas que permiten comenzar rápidamente cualquier proyecto.

Ofrece herramientas como inyección de dependencias, control de transacciones y seguridad, entre otras.

En este proyecto, se ha utilizado Spring Boot para agilizar el desarrollo del backend, simplificando la configuración y el despliegue, lo que ha permitido centrarse en la lógica de negocio y la integración con otras tecnologías de forma eficiente.

### Maven



Maven es una herramienta de gestión y construcción de proyectos para aplicaciones Java, que facilita la automatización de tareas como la compilación, el empaquetado y la ejecución de pruebas.

En el proyecto, se ha empleado para hacer operaciones como el Clean y el Install para poder probar el código.

### JPA



JPA es una especificación de Java para el mapeo de objetos Java a bases de datos relacionales.

JPA se ha usado en este proyecto para gestionar la persistencia de los objetos en la base de datos, simplificando las operaciones CRUD y optimizando la interacción entre el código de la aplicación y MariaDB.

### Lombok



Lombok es una biblioteca Java que elimina el código repetitivo mediante anotaciones, lo que mejora la legibilidad, acelera el flujo de trabajo y aumenta la productividad.

Al utilizar Lombok en este proyecto, se ha simplificado el proceso de creación de clases Java al eliminar la necesidad de escribir constructores, *getters*, *setters*, *toString* y *equals*.

### MapStruct



MapStruct es una herramienta de mapeo de objetos Java que facilita la conversión entre objetos de diferentes clases, generando automáticamente el código necesario para las conversiones de tipo.

En este proyecto, MapStruct se ha utilizado para transformar los objetos de JPA al modelo de negocio y del modelo de negocio al modelo de la capa de presentación.

### CORS

CORS (**Cross-Origin Resource Sharing**) es un mecanismo que permite a los recursos restringidos en una página web ser solicitados desde otro dominio. Este proceso es importante cuando una aplicación frontend interactúa con un backend alojado en un servidor diferente. En este proyecto, se ha implementado CORS para permitir que la aplicación iOS (frontend) realice peticiones seguras a la API backend (Java), garantizando que las solicitudes entre dominios sean correctamente gestionadas sin comprometer la seguridad de la aplicación.

### Postman



Postman es una herramienta de desarrollo que facilita la prueba de los diferentes métodos de una API REST. Gracias a Postman, se ha podido comprobar que la API devolvía aquello que se le solicitaba en cada método (GET) y que se podían realizar los cambios pertinentes (POST, PUT y DELETE).

### Swift



Swift es un lenguaje de programación creado por Apple para el desarrollo de aplicaciones en sus plataformas, como iOS, macOS, watchOS y tvOS. Swift es conocido por su velocidad y seguridad, ofreciendo características avanzadas como la inferencia de tipos y la programación funcional.

En este proyecto, Swift se ha utilizado para desarrollar el frontend de la aplicación iOS, permitiendo crear una interfaz de usuario atractiva y eficiente para gestionar sesiones de entrenamiento y monitorear el progreso de los usuarios.

### SwiftUI



SwiftUI es un framework declarativo de interfaz de usuario para desarrollar aplicaciones en plataformas Apple. Facilita la creación de interfaces de usuario complejas mediante el uso de una sintaxis simple y concisa.

En este proyecto, SwiftUI ha sido la herramienta principal para desarrollar la interfaz de usuario, permitiendo una integración fluida con el backend, mientras que su capacidad de previsualización instantánea ha agilizado el proceso de diseño y desarrollo de las pantallas.

### SFSymbols



SFSymbols es un conjunto de iconos vectoriales diseñados por Apple que se integran nativamente en las aplicaciones de iOS. Estos iconos están optimizados para adaptarse a cualquier tamaño de pantalla y mantener una apariencia coherente en todas las aplicaciones.

En este proyecto, se ha utilizado SFSymbols de cara a enriquecer la interfaz de usuario de la aplicación, incorporando iconos que mejoran la usabilidad y la estética de la aplicación sin tener que crear iconos personalizados.

Asimismo, al ser iconos desarrollados por la propia Apple, son coherentes con el diseño del resto del sistema operativo.

# Implementación

Texto.

## Análisis

Texto.

## Diseño

Texto.

* Logo
* Casos de uso
* Diagrama entidad-relación
* Base de datos
* Back End
* Front End

## Manual de usuario

Texto.

* Login
* Explicar cada caso de uso haciendo capturas con todo el ciclo

# Propuestas de mejora

La lista de futuras ampliaciones del proyecto es la siguiente:

* Sesiones de entrenamiento privadas y públicas.
* Equipamiento.
* Historial de progreso.
* Programas semanales.
* Multi idioma.
* Entrenador-deportista.
* Foco en privacidad.

A continuación, se muestra cada apartado explicado detalladamente para poder seguir más fácilmente el hilo de este apartado del trabajo:

* Sesiones de entrenamiento privadas y públicas.

Actualmente, las sesiones creadas en la aplicación son únicamente visibles por el propio usuario. Como mejora, se propone añadir una opción para configurar la visibilidad de cada sesión.

De esta manera, los usuarios podrán decidir si desean mantener sus entrenamientos en privado o compartirlos con otros miembros de la comunidad. Esta funcionalidad fomentaría tanto la privacidad como la interacción entre usuarios, permitiendo descubrir nuevas rutinas o inspirarse en los entrenamientos de otros.

Además, también permitiría que la propia aplicación contara con sesiones de entrenamiento con validez científica de cara a los usuarios que no cuenten con los conocimientos para elaborar sus propias sesiones de trabajo.

* Equipamiento.

En este apartado, la idea sería introducir en la base de datos una relación N-N en la cual se estipula el material que se emplea en cada ejercicio. Así pues, cada usuario o entrenador de la aplicación podría seleccionar los ejercicios que desea hacer en base al material del que dispone, ya sea entrenando en su casa, al aire libre o en un gimnasio.

* Historial de progreso.

La idea de este apartado es contar con gráficas en las cuales visualizar el progreso en los distintos ejercicios. Para que se entienda mejor, pondré dos ejemplos:

En el press banca, llegar a los 100kg es un objetivo que tienen muchos hombres. Poder contar con un gráfico en el cual se ve el progreso del ejercicio puede ayudarnos a comprobar el progreso que vamos realizando, evaluar que el programa de entrenamiento es exitoso o a mantener la motivación, entre muchos otros análisis.

En el caso del entrenamiento cardiovascular, el parámetro del VO2max nos puede prestar claves importantes de cara a saber el estado de nuestra salud general, ya que es un gran medidor de salud cardiovascular. Poder ver cómo evoluciona puede ayudarnos también de cara a prepararnos carreras como la Media Maratón de Valencia, pudiendo ver si el tiempo en el que queremos correrla es factible.

* Programas semanales.

Actualmente podemos crear sesiones, pero no establecer un calendario en el cual estipular qué sesión debemos realizar cada día. Esto es de gran ayuda a la hora de la programación del entrenamiento, ya que podremos centralizar todo el proceso en una misma aplicación y no tener que depender de terceros.

* Multi idioma.

Otra opción interesante es la de admitir más idiomas. Este es un apartado que he tenido que tener en cuenta en la empresa, ya que los idiomas han aumentado considerablemente, por lo que conozco como poder llevarlo a cabo. No obstante, con el objetivo que tengo en mente, conllevaría la modificación de la base de datos.

* Entrenador-deportista

Este es el apartado más ambicioso, ya que permitiría la creación de distintos roles: el entrenador y el deportista. Esto permitiría que expertos en ejercicio contaran con una aplicación integral en la cual desarrollar los planes de entrenamiento y que los deportistas solo tuvieran que seguir el planning. Debido a la envergadura de este proyecto, es una idea remota que por ahora no planteo llevar a la realidad.

* Foco en privacidad

Finalmente, mi mayor deseo es que la aplicación corra en local sin tener que enviar o recibir datos del exterior. Esto permitirá que la aplicación pueda funcionar sin una conexión a internet o sin depender de tener activo un servidor. Además, le da un plus de privacidad ya que los datos estarían encriptados en el propio dispositivo. Asimismo, se habilitaría una opción para exportar los datos por si se cambia de dispositivo.

Debido a que prácticamente desde el primer día estoy metido en el desarrollo del proyecto de la empresa en la cual estoy cursando las prácticas, no me es posible realizar todas estas funciones por la falta de tiempo. No obstante, y debido a que es una aplicación que quiero desarrollar para poder llevar el registro de mis entrenamientos, planeo ir desarrollando estas funciones poco a poco.

# Valoración personal

Texto.

# Puntos a destacar del proyecto

Texto.

# Referencias

Swift:

* Swift Fundamentals: Mastering the Basics with Swift & iOS – Stephan Dowless:

<https://www.udemy.com/course/swift-fundamentals/?couponCode=LETSLEARNNOW>

* Aprendiendo Swift 5.5 – Julio César Fernández Muños (Apple Coding Academy):

<https://www.udemy.com/course/comenzando-con-swift/>

SwiftUI:

* Swift desde Cero – Brais Moure:

<https://youtube.com/playlist?list=PLNdFk2_brsRetB7LiUfpnIclBe_1iOS4M&si=vCzHW54baML3w6Sz>

* Curso de programación SwiftUI y Xcode – SwiftBeta:

<https://youtube.com/watch?v=H0kihMlApn4&list=PLeTOFRUxkMcrJIZf4NnJQsL_ZNIq9JWQy>

* SwiftUI Crypto App – Swiftful Thinking:

<https://www.youtube.com/watch?v=TTYKL6CfbSs&list=PLwvDm4Vfkdphbc3bgy_LpLRQ9DDfFGcFu>

Java:

* Apuntes César: <http://cesguiro.es/doku.php>
* Spring Initializr: <https://start.spring.io>
* CORS: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/CORS>

MariaDB:

* Documentación oficial: <https://mariadb.org>