



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

PowerLog Web. Plataforma web para la gestión de  
entrenamientos

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Eljishi Calzada, Manuel Emilio

Tutor/a: Valderas Aranda, Pedro José

CURSO ACADÉMICO: 2022/2023



# Resumen

El objetivo de este trabajo es proveer a los entrenadores de PowerLifting de una herramienta que les facilite la gestión de los entrenamientos. Unificando los distintos apartados de esta gestión tales como la creación de los ejercicios, la consulta de los resultados por parte de los atletas, la creación de rutinas entre otras. El trabajo se realizará manteniendo el contacto con los usuarios potenciales asegurando así que el trabajo sea acorde a la idea planteada y concluirá publicando una versión final de la aplicación web donde cada entrenador podrá realizar todas estas tareas siempre manteniendo la sencillez y minimalismo en la interfaz. Además se planteará la totalidad de la infraestructura de comunicaciones necesaria para persistir y sincronizar los datos creados por los usuarios

Palabras clave: aplicación, desarrollo web, diseño centrado en el usuario, powerlifting, deporte, salud, arquitectura de sistemas.

# Abstract

The objective of this work is to provide PowerLifting coaches with a tool that facilitates training management. Unifying the different sections of this management such as the creation of the exercises, the consultation of the results by the athletes, the creation of routines among others. The work will be done while maintaining contact with potential users to ensure that the work is in line with the idea proposed and will conclude by publishing a final version of the web application where each coach can perform all these tasks while maintaining the simplicity and minimalism in the interface. In addition, the entire communications infrastructure necessary to persist and synchronize the data created by the users will be proposed.

Keywords: application, web development, user-centered design, powerlifting, sport, health, system architecture.



# Tabla de contenidos

1.	Introducción .....	8
1.1.	Motivación .....	9
1.2.	Objetivos .....	10
1.3.	Impacto Esperado .....	10
1.4.	Estructura .....	11
1.5.	Colaboraciones .....	13
2.	Estado del arte .....	14
2.1.	Hevy .....	14
2.2.	Reactive training systems .....	15
2.3.	Hoja de Excel / Tabla en Word .....	18
2.4.	Crítica al estado del arte .....	19
2.4.1.	Crítica a Hevy .....	19
2.4.2.	Crítica a RTS .....	20
2.4.3.	Crítica a Excel / Word .....	21
2.4.4.	Propuesta .....	21
3.	Metodología .....	23
4.	Análisis de requisitos .....	27
4.1.	Requisitos funcionales .....	27
4.2.	Requisitos no funcionales .....	28
4.3.	Casos de uso .....	29
4.4.	Diagrama de clases .....	31
5.	Aspectos de planificación .....	33
5.1.	Seguridad .....	33
5.2.	Análisis energético .....	34
5.3.	Análisis del marco legal y ético .....	36
5.4.	Análisis de riesgos .....	37
5.5.	Identificación y análisis de soluciones disponibles .....	38
5.6.	Plan de trabajo .....	40
6.	Diseño de la aplicación .....	42
6.1.	Diseño de la base de datos .....	42

6.2.	Diseño de las interfaces.....	43
7.	Desarrollo de la aplicación.....	47
7.1.	Arquitectura del sistema.....	47
7.2.	Tecnologías utilizadas.....	50
7.3.	Ejemplos de código.....	51
8.	Implantación.....	55
9.	Producto desarrollado.....	57
10.	Conclusiones.....	63
11.	Trabajos futuros.....	66

# Índice de Ilustraciones

ILUSTRACIÓN 1 LOS 3 BÁSICOS DEL POWERLIFTING .....	9
ILUSTRACIÓN 2 INTERFAZ DE HEVY.....	15
ILUSTRACIÓN 3 INTERFAZ DE HEVY.....	15
ILUSTRACIÓN 4 FUNCIONALIDADES OFRECIDAS EN RTS .....	16
ILUSTRACIÓN 5 CREACIÓN DE UN ENTRENAMIENTO EN RTS, SELECCIÓN DE EJERCICIOS.....	17
ILUSTRACIÓN 6 PANTALLA DONDE SE MUESTRA EL ENTRENAMIENTO DEL DÍA.....	17
ILUSTRACIÓN 7 EJEMPLO DE UNA RUTINA CREADA USANDO MICROSOFT WORD .....	19
ILUSTRACIÓN 8 DIAGRAMA DE FLUJO DEL PROCESO INCREMENTAL .....	23
ILUSTRACIÓN 9 DIAGRAMA DE CASOS DE USO .....	29
ILUSTRACIÓN 10 DIAGRAMA DE CLASES DE LA APLICACIÓN POWERLOG .....	31
ILUSTRACIÓN 11 PLANIFICACIÓN DEL DESARROLLO DE POWERLOG. ....	40
ILUSTRACIÓN 12 CLASE DE EJERCICIO EN LA BASE DE DATOS .....	42
ILUSTRACIÓN 13 CLASE DAYLOG, UN DÍA DE ENTRENAMIENTO .....	43
ILUSTRACIÓN 14 CLASE QUE REPRESENTA LA LISTA DE EJERCICIOS DE CADA ENTRENADOR .....	43
ILUSTRACIÓN 15 PRIMER BOCETO EN PAPEL ESCANEADO DIGITALMENTE .....	44
ILUSTRACIÓN 16 PRIMER DISEÑO DE LA WEB .....	45
ILUSTRACIÓN 17 IMPLANTACIÓN ACTUAL DE LA WEB .....	46
ILUSTRACIÓN 18 ARQUITECTURA DEL SISTEMA POWERLOG .....	47
ILUSTRACIÓN 19 DIAGRAMA DE FLUJO EN LAS VISTAS DE LA PÁGINA WEB .....	50
ILUSTRACIÓN 20 ARQUITECTURA DE LA APLICACIÓN BACKEND .....	51
ILUSTRACIÓN 21 CÓDIGO DE CREACIÓN DE ENTRENAMIENTO .....	52
ILUSTRACIÓN 22 CÓDIGO DE INICIO SE SESIÓN Y REGISTRO .....	53
ILUSTRACIÓN 23 SERVICIO DE GESTIÓN DE EJERCICIOS PARA ENTRENADORES .....	54
ILUSTRACIÓN 24 PANTALLA INICIAL DE LA VERSIÓN DEL ENTRENADOR DE POWERLOG.....	57
ILUSTRACIÓN 25 VISTA DE GESTIÓN DE ATLETA CON UN ENTRENAMIENTO YA CREADO .....	58
ILUSTRACIÓN 26 VISTA DEL ENTRENADOR DE UN DÍA SIN DATOS .....	59
ILUSTRACIÓN 27 VISTA DE CREACIÓN DE UN EJERCICIO .....	59
ILUSTRACIÓN 28 LISTA DE EJERCICIOS DEL ENTRENADOR.....	60
ILUSTRACIÓN 29 VENTANA DE CREACIÓN DE EJERCICIO.....	61
ILUSTRACIÓN 30 BASE DE DATOS DE LA APLICACIÓN, TABLA DE ENTRENAMIENTOS .....	62
ILUSTRACIÓN 31 BASE DE DATOS DE LA APLICACIÓN, TABLA DE EJERCICIOS .....	62





# 1. Introducción

---

Desde hace años hemos visto como las nuevas tecnologías han revolucionado el mundo de los deportes. Nuevas aplicaciones que proporcionan soporte para la realización de entrenamientos utilizando tecnologías de realidad virtual o mixta, la optimización de técnicas y estrategias a través de simulaciones, la mejora de las capacidades de retransmisión, etc

Sin embargo, todas estas técnicas suelen quedar limitadas a aquellos deportes mayoritarios. Deportes con más aficionados y, por tanto, mucho capital destinado a mejorar su experiencia.

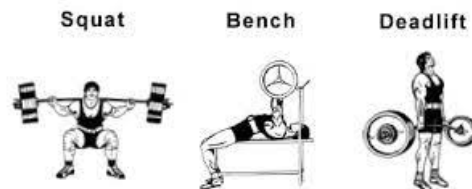
Esta circunstancia puede parecer extremadamente positiva desde el punto de vista tecnológico o de negocio, incluso desde el punto de vista de un aficionado de estos deportes mayoritarios donde estos cambios sí que se desarrollan, prueban e implementan constantemente, pero al mismo tiempo puede acabar representando una gran desventaja para los deportes minoritarios puesto que pueden acabar quedándose atrás e incluso perder aficionados al haber ofertas mucho más atractivas de cara al espectador.

El powerlifting es un deporte que cae en la segunda categoría. A pesar de llevar varios años en alza sigue siendo un deporte minoritario con unos pocos cientos de competidores por toda España, instalaciones limitadas y por consiguiente muy poca evolución tecnológica en lo que al ámbito del mismo se refiere.

En este deporte se crea un equipo de 2 personas (entrenador y atleta) donde el fin es levantar el mayor número de kilos en 3 movimientos distintos (squat, bench press y deadlift, fig. 1.1).

Durante meses o incluso años, el atleta y el entrenador trabajan e para optimizar sus esfuerzos de cara al objetivo de competición, analizan las capacidades y ventajas del atleta para cada ejercicio, optimizan la aplicación de fuerza, practican distintas posturas iniciales y finales junto con una gran variedad de patrones de movimiento distinto para así conseguir realizar los ejercicios maximizando la fuerza y limitando el riesgo de lesión.

Los entrenamientos se basan en un conjunto de ejercicios donde cada uno se realiza a un número determinado de series y un número determinado de repeticiones por serie. La intensidad que el atleta debe aplicar en cada serie viene determinada por un valor especificado por el entrenador (RIR o RPE) pues la intensidad no será la misma al realizar una sentadilla con 150kg o con 200kg.



*Ilustración 1 Los 3 básicos del powerlifting*

## 1.1. Motivación

---

Desde hace mucho tiempo mi objetivo principal se convirtió en encontrar un punto de sinergia entre el powerlifting y la informática. Es por eso que tras adquirir los conocimientos necesarios en el grado y convertirme en un atleta de powerlifting más experimentado decidí analizar el proceso de entrenamientos para buscar un punto donde poder aplicar mis conocimientos para optimizar o facilitar el mismo.

Tras mucho tiempo trabajando con mi entrenador me di cuenta de que la forma de planificar los entrenamientos seguía siendo muy arcaica, a través de una aplicación de escritorio con un diseño antiguo y poco intuitivo o directamente a través de una hoja de Excel que con el paso del tiempo se acaba perdiendo o acumulando en una carpeta de planificaciones antiguas sin ningún tipo de estructura, lo cual acaba complicando mucho el proceso de búsqueda de datos antiguos. Por esta razón decidí crear un sistema que facilitara la interacción entre el entrenador y el atleta, dando libertad al entrenador para poder esquematizar los entrenamientos acordes a sus necesidades, pero sin perder la capacidad de guardar los resultados de una forma estructurada y de fácil acceso para poder generar diagramas o gráficos de una manera fácil y rápida.

## 1.2. Objetivos

---

El objetivo principal de este trabajo es el de facilitar el trabajo de los entrenadores de powerlifting, por lo que el trabajo se ha realizado centrándose en los requisitos de mi entrenador. El foco del proyecto ha sido la creación de una aplicación web donde el entrenador pueda gestionar los entrenamientos de sus atletas de forma rápida y sencilla, crear y modificar rutinas sin sentirse cohibido por la aplicación, consultar resultados anteriores, etc. En concreto, en el contexto de este TFG se pretende desarrollar una aplicación web que de soporte a los siguientes objetivos específicos:

- Acceder a los datos del entrenador con un login convencional
- Consultar los datos de todos los atletas que estén asignados con el entrenador
- Crear y gestionar nuevos ejercicios separados por categorías
- Crear y modificar rutinas para cada uno de los atletas
- Poder consultar en tiempo real los resultados que los atletas reportan en su aplicación

## 1.3. Impacto Esperado

---

Tras realizar este proyecto queda clara la intención de convertir el Powerlifting en un deporte mucho más atractivo a los ojos del espectador, reducir la complejidad inicial del mismo y además facilitar la interacción de los entrenadores con los atletas.

De cara al entrenador se plantea una interfaz generalizada desde la cual poder realizar todas las consultas y modificaciones requeridas para ejercer su tarea.

Además de esto se espera que el resultado del proyecto promueva un estilo de vida más saludable pues se ha demostrado que la práctica de deporte intenso tiene un gran impacto positivo en una gran cantidad de ámbitos de la vida de las personas tales como mejorar la calidad del sueño, ayudar a controlar el peso, fomentar una mejor salud mental o reducir la posibilidad de ciertas enfermedades como por ejemplo el cáncer o aquellas de naturaleza cardiovascular.

## 1.4. Estructura

---

A continuación, se describe de manera rápida cuál será el contenido de cada uno de los apartados de esta memoria:

### 1. Introducción

El primer capítulo de esta memoria servirá para explicar los motivos por los que se decidió plantear este sistema. Se explica además cuáles son las motivaciones principales y cuales son los objetivos que se establecieron al comienzo del proyecto.

### 2. Estado del arte

En este capítulo se explican los principales sistemas que hay desarrollados actualmente con los cuales el producto de este trabajo, PowerLog, podría competir, además se explica el funcionamiento y las desventajas de cada uno de los sistemas expuestos.

### 3. Metodología

Explicación detallada de la metodología utilizada para el desarrollo del proyecto (desarrollo incremental). Además, en este apartado se explica cómo el equipo de desarrollo y los usuarios han trabajado en conjunto para alcanzar las metas propuestas

### 4. Análisis de requisitos

Durante este apartado se exponen los requisitos, tanto funcionales como no funcionales que presenta la aplicación, además de los diagramas desarrollados para explicar el dominio del problema y los casos de uso

### 5. Aspectos de planificación

Aquí se analizarán distintos apartados de la aplicación como la eficiencia energética, la seguridad, el marco legal donde se desarrolla el sistema

### 6. Diseño de la aplicación

Aquí se muestran los modelos creados con respecto a las bases de datos que usará la aplicación junto con los primeros bocetos creados durante el proceso de diseño

### 7. Desarrollo de la aplicación

Presentación de la arquitectura del programa junto con el stack tecnológico usado para llevar a cabo el proyecto, además de mostrar el desarrollo con algunos fragmentos del código

### 8. Implantación

Explicación en detalle del stack tecnológico usado para crear la aplicación y detalle de la arquitectura implementada, con todos los servicios y comunicaciones.

9. Producto desarrollado

Presentación del producto final desarrollado, explicación del funcionamiento completo de la aplicación web y resultados de los usuarios mediante capturas de pantalla

10. Conclusiones

Valoración final del proyecto y análisis de los resultados, tecnologías aprendidas, contratiempos superados etc

11. Trabajos futuros

Análisis de funcionalidades que se deseaba implementar pero por límites de tiempo y recursos acabaron quedando fuera del objetivo. Se podrían desarrollar en un futuro

## 1.5. Colaboraciones

---

Este trabajo se ha realizado junto al alumno Pablo Magraner Tamarit, que a su vez ha realizado el proyecto “PowerLog Móvil. Una aplicación móvil para la gestión de entrenamientos con componentes sociales para atletas y entrenadores”.

El alcance de este proyecto se ha limitado al diseño e implementación de la página web a través de la cual el entrenador podrá interactuar con el sistema. El trabajo de diseño con el entrenador para proponer soluciones cómodas e intuitivas, la creación en código de estas, así como la creación de la infraestructura de servidores necesaria para persistir los datos y sincronizarlos con la aplicación móvil.

El proyecto “PowerLog Móvil” ha trabajado paralelamente junto con los atletas puesto que estos serían los principales usuarios de la aplicación móvil. En este trabajo se han diseñado las diferentes vistas y funcionalidades centradas en los conocimientos de los atletas, posteriormente se ha creado una aplicación con todas las vistas planteadas y se han gestionado todas las interacciones con el servidor desde un punto de vista móvil.

## 2. Estado del arte

---

En esta sección vamos a explicar aquellos sistemas ya existentes contra los que nuestro producto podría llegar a competir, o bien por funcionalidades básicas o bien por el cliente potencial que esperan atraer.

### 2.1. Hevy

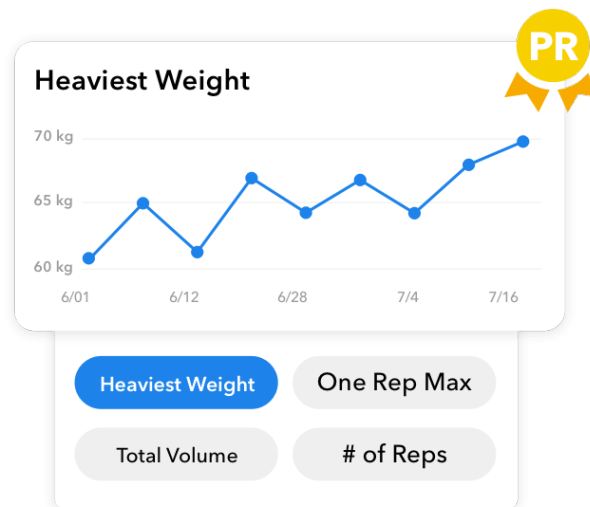
---

Hevy es una aplicación creada en 2019 por los ingenieros de software Guillem Ros y Desmond McNamee. Según narra la página oficial[1]:

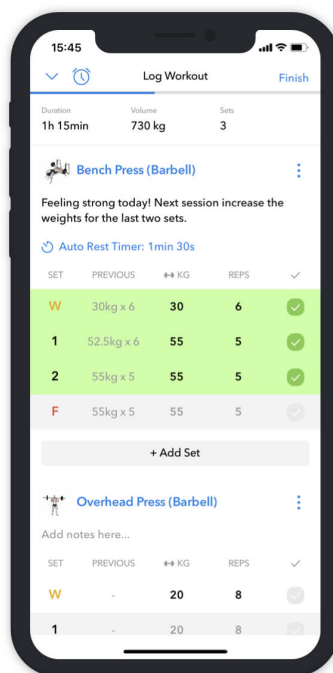
“Estábamos muy descontentos con las aplicaciones para registrar entrenamientos que había en el momento y decidimos crear un sistema más completo que permitiese a los usuarios interactuar con sus amigos... Nuestra misión es proveer a todos los atletas de las herramientas necesarias para conseguir sus metas, tanto estéticas como de fuerza”.

El sistema Hevy cubre una gran variedad de casos de uso, sin embargo, en este análisis nos centraremos en aquellos que más se asemejan a las funcionalidades que deseamos implementar en nuestro sistema. A la hora de registrar nuestros entrenamientos, al utilizar Hevy dispondremos de las siguientes funcionalidades:

- Creación de rutinas personalizadas, dentro de las que podremos determinar el tiempo e intensidad deseados, ejercicio, peso y repeticiones objetivo (tanto repeticiones de calentamiento o incluso series hasta el fallo) e incluso superseries entre dos o más ejercicios.
- Visualizador de progreso, en la aplicación tendremos una pantalla principal donde se nos mostrarán todos los entrenamientos que hayamos realizado hasta la fecha lo cual nos ayudará a “visualizar nuestro progreso y mantener la motivación”.
- Dentro de la aplicación tendremos la posibilidad de añadir amigos con los que podremos compartir nuestra rutina con otras personas y al mismo tiempo comparar e incluso copiar los planes de otros usuarios.
- Crear ejercicios customizados, si a la hora de crear una rutina los ejercicios introducidos por defecto en la aplicación no cumplen con todos nuestros requisitos, se nos ofrece la posibilidad de crear nuestros propios ejercicios para poder así cubrir la totalidad de nuestro entrenamiento desde una sola app.



*Ilustración 2 Interfaz de Hevy*



*Ilustración 3 Interfaz de Hevy*

## 2.2. Reactive training systems







---



Reactive Training Systems[2] es una empresa fundada por Mike Tuchserer , un famoso entrenador de Powerlifting americano. Mike comenzó a practicar powerlifting en 1997 y pronto se convirtió en una figura muy im

Reactive Training Systems (RTS) es una empresa que ofrece una gran cantidad de productos distintos, enfocados tanto a atletas como entrenadores por igual. Entre todos estos productos ofertados podemos destacar asesorías privadas con algunos de los mejores entrenadores del momento, libros de teoría centrados en el entrenamiento y la ganancia de fuerza, cursos online, material de entrenamiento para optimizar los entrenamientos.

De entre todos los productos anteriormente mencionados vamos a centrarnos en la aplicación de RTS, una aplicación web donde tanto entrenador como atleta comparten una misma interfaz desde la cual se les ofrece la posibilidad de plantear y crear nuevos entrenamientos además de consultar una gran cantidad de métricas

 Dashboard Dashboard Central Competition Progress Preferences	 Calendar Calendar Central Calendar Listing	 Reporting Reports	 TRAC Action Items Trend Analysis Short Term Long Term New Entry Edit Entries	 Training Log New Workout Manage Exercises Manage Modifiers Custom RPE Charts Edit Workouts	 Classroom Classroom Central
---	--	--	--	---	--

*Ilustración 4 Funcionalidades ofrecidas en RTS*

Dentro de la aplicación podremos utilizar todas estas opciones para monitorear nuestro progreso en el entrenamiento de fuerza, calcular índices tales como el stress index el 1RM estimado de cada uno de los ejercicios o incluso los puntos Wilks

Si seleccionamos las opciones de crear un entrenamiento, se nos redirigirá a una pantalla con un gran listado de posibles ejercicios ya creados por defecto en la aplicación. A medida que vamos seleccionando los ejercicios estos se van incluyendo en una lista para posteriormente poder definir la estructura de los mismos (tanto las series que vamos a realizar como las repeticiones y el índice de esfuerzo relativo o rpe) de cada uno de ellos. Además de estos valores principales podremos apuntar otros valores como por ejemplo el rom, tempo, el tipo de barra o incluso el tipo de stance.

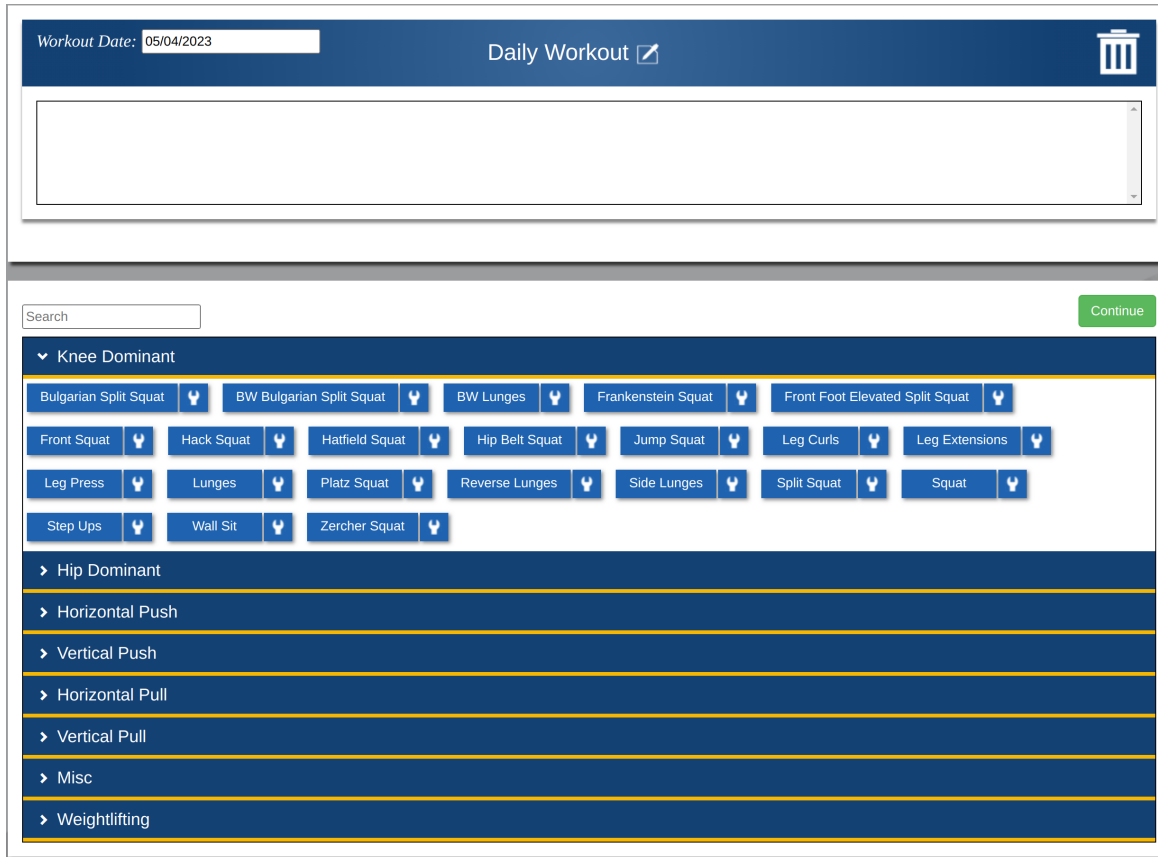


Ilustración 5 Creación de un entrenamiento en RTS, selección de ejercicios



Ilustración 6 Pantalla donde se muestra el entrenamiento del día

## 2.3. Hoja de Excel / Tabla en Word

---

Otra opción para gestionar los entrenamientos utilizada por una gran cantidad de entrenadores tanto en el ámbito nacional como internacional son las hojas de Excel o una tabla en un documento Word.

Estos programas han sido utilizados durante años multitud de ámbitos distintos, financieros, control de almacenes, matemáticas e incluso en el mundo del deporte. Tanto Word como Excel permiten al usuario guardar grandes cantidades de información de muchos tipos (texto, números, símbolos, imágenes, etc) la falta de una estructura predeterminada permite a los ofrece a los entrenadores una mayor libertad para adaptar las planificaciones por completo a cada uno de los atletas.

Tanto Word como Excel también ofrecen la posibilidad de generar cálculos automáticos, es decir, cuando el entrenador define una serie de ejercicios con sus respectivos parámetros, puede añadir ciertas configuraciones para obtener varios parámetros valiosos a la hora de generar un entrenamiento tales como el stress index , el TTP el 1RM estimado en cada uno de los ejercicios básicos etc.

Al emplear estas tecnologías se puede lograr una mejor comunicación con el atleta, a pesar de ser un intercambio de información asíncrona, la posibilidad de escribir mensajes o comentarios más largos, añadir imágenes que aclaren ciertos aspectos al atleta o incluso incluir enlaces a videos explicativos puede llegar a ser un gran añadido a la hora de trabajar, sobre todo, con atletas principiantes y menos experimentados.

EJERCICIO	SERIES	REPS	RPE	KG	REPES	RPE	NOTAS
COMP. BP	1 3	1 4	8 6	11 5 10 0	1 3	8 6	
Remo unilateral	5	10	9	50	10	9	Maquina, mancuerna o polea.
Bíceps a tu elección	4	10	9	50	10	9	
Tríceps a tu elección	4	10	9	60	10	9	

*Ilustración 7 Ejemplo de una rutina creada usando Microsoft Word*

Como se puede comprobar, la posibilidad de añadir o eliminar columnas a la tabla según el entrenador lo pueda considerar necesario es un gran añadido y una herramienta muy útil dada la diferencia tan pronunciada que puede haber entre los distintos atletas que un mismo entrenador pueda gestionar.

## 2.4. Crítica al estado del arte

### 2.4.1. Crítica a Hevy

Como hemos podido ver, Hevy puede ser una de las aplicaciones más completas que hemos analizado en esta memoria pues cuenta con una interfaz moderna que se hace muy sencilla de utilizar desde el primer día. El equipo de desarrollo cuenta con gente muy capacitada y con una amplia experiencia además de tener una grande y diversa comunidad de atletas por todo el mundo con los que podremos compartir conocimientos e interactuar.

Sin embargo, el enfoque de Hevy es radicalmente distinto al planteado en este proyecto. Hevy se centra únicamente en los atletas ofreciéndoles una muy buena aplicación móvil pero pasa por alto el punto de vista del entrenador. Cada atleta ha de tener la aplicación instalada en su smartphone para poder consultar los datos y no existe una manera de consultar

los resultados de los atletas desde un punto centralizado ni se puede crear una relación entrenador atleta dentro de la misma aplicación. Para realizar la utilidad que PowerLog plantea, el entrenador deberá instalar la aplicación en su dispositivo móvil y recordar las credenciales de cada uno de sus atletas (un entrenador puede tener de media entre 10 a 50 atletas que requieren revisiones semanales y actualizaciones de sus objetivos) y deberá iniciar sesión en cada una de las cuentas para poder consultar los datos de los atletas. Este flujo de trabajo supone un gran malgasto de tiempo además de una mayor incomodidad a la hora de realizar las tareas administrativas pues en una pantalla de mayor tamaño se podrían mostrar más datos en una sola interfaz.

## 2.4.2. Crítica a RTS

---

La aplicación web de Reactive Training Systems tiene un enfoque mucho más dedicado a los entrenamientos de fuerza personalizados, es por eso que la gran cantidad de métricas e índices que nos permite calcular de manera automática presenta una gran ventaja frente al resto de aplicaciones que se han analizado.

No obstante, parece que RTS no tiene muy en cuenta al usuario objetivo pues por lo general, los atletas y los entrenadores que la usan son usuarios básicos o medios de la informática. Por esto se considera que no es una aplicación nada fácil de usar pues como se han podido ver en las imágenes mostradas de la web (*REFERENCIAR IMÁGENES*), la interfaz no es del todo sencilla ni intuitiva. Si bien es cierto que la primera vez que entramos en la aplicación se nos hace un onboarding, hay muchos aspectos de la funcionalidad que quedan ocultos y que no son nada fáciles para un usuario menos experimentado de encontrar.

Una ventaja que el planteamiento de RTS presenta es el hecho de ser una aplicación Web. Esto permite poder acceder a consultar o rellenar datos desde cualquier dispositivo que tenga un navegador web pero al mismo tiempo presenta una desventaja, la adaptación a móvil o diseño responsive. Si la aplicación ya es compleja de manejar desde un ordenador con una pantalla generalmente de, como mínimo, 15 pulgadas cuando se accede desde el móvil todo se ve muy reducido e incluso la gran mayoría de funcionalidades se encuentran teniendo que deslizar horizontalmente, cosa que por lo general dificulta aún más el uso.

Otro aspecto negativo de la aplicación puede ser, tal y como se ha explicado en el análisis de Hevy es como se enfocan las cuentas en la aplicación. El entrenador no puede crear

ninguna relación con los atletas y por tanto ha de crear una cuenta para cada uno de los mismos, cosa que implica recordar una gran cantidad de datos de inicio de sesión lo que supone una pérdida de tiempo y un proceso muy incómodo que se repetirá con mucha frecuencia.

### 2.4.3. Crítica a Excel / Word

---

Como se ha explicado, ambos programas pertenecientes a la suite Office 365 de Microsoft ofrecen sistema muy versátil y flexible. Esto permite a los entrenadores gestionar de una forma muy personal los entrenamientos de cada uno de los atletas y adaptar los requerimientos y planificaciones prácticamente en su totalidad.

Entre los puntos negativos cabe destacar que a pesar de poder gestionar grandes cantidades de datos de manera automática, ha de ser el entrenador quien aplique todas las fórmulas esto junto a la posibilidad antes mencionada de adaptar los entrenamientos para cada persona significa que el mismo entrenador debe mantener una estructura parecida para todas las personas o se verá obligado a aplicar los cálculos necesarios individualmente, cosa que se convierte en un punto de fallo y un proceso muy tedioso.

Otro aspecto a tener en cuenta es que, de nuevo, si el entrenador no fuerza una estructura a lo largo del tiempo, la falta de esta se convierte en un problema si se necesita consultar datos históricos, una necesidad muy importante en los deportes de levantamiento de fuerza pues así se puede consultar el progreso, repetir protocolos de entrenamiento que hayan funcionado bien o incluso crear una estrategia en el momento de la competición.

### 2.4.4. Propuesta

---

Tras el análisis de las metodologías y sistemas más utilizados actualmente se puede observar como la ineficiencia e incomodidad se convierten en el día a día de las personas que las utilizan. En este trabajo se propone un sistema que combine lo mejor de todos estos métodos utilizados: la interfaz web, vista como un punto central de información donde el entrenador podrá consultar todos los entrenamientos con únicamente unos pocos clicks y sin tener que cerrar e iniciar sesión cada vez que quiera cambiar de atleta. La versatilidad y flexibilidad que Excel y Word proporcionan, facilitando así la personalización de los entrenamientos junto con el proyecto “PowerLog Móvil. Una aplicación móvil para la gestión de entrenamientos con componentes sociales para atletas y entrenadores” donde se creará la

aplicación móvil trabajando mano a mano con los atletas para poder utilizar todos los datos creados por el entrenador de una forma cómoda, eficiente y sin perder el componente social que aplicaciones como Hevy implementan y que pueden ayudar a mantener la motivación del usuario.

### 3. Metodología

Como se ha explicado anteriormente, la intención del proyecto no es desarrollar un sistema completamente nuevo y revolucionario, sino que el desarrollo de PowerLog se ha enfocado desde una perspectiva de evolución. El objetivo de la aplicación era juntar varias ideas ya desarrolladas entre sí (a través de la relación atleta-entrenador), cosa que permitiría sincronizar la información de la aplicación móvil y la aplicación web, pero al mismo aplicar un evolutivo que ofreciese una mejor experiencia de usuario, una interfaz más sencilla, algo que una persona con reducida experiencia en el uso de productos informáticos fuera capaz de utilizar prácticamente sin barrera de entrada. Teniendo todo esto en cuenta y tras analizar varias metodologías y enfoques distintos, el proyecto se ha creado mediante el desarrollo centrado en el usuario.

El modelo de desarrollo incremental es una forma de separar el ciclo de vida del software en diferentes etapas. Cada una de estas etapas, desde la etapa de especificación y diseño hasta la etapa de entrega al cliente se agrupan en lo que se suele conocer como un ciclo o un sprint, es decir, que durante la duración del proceso de desarrollo en el proceso incremental se realizan varias entregas al cliente, cada una considerándose como una evolución, o un incremento, de la anterior.

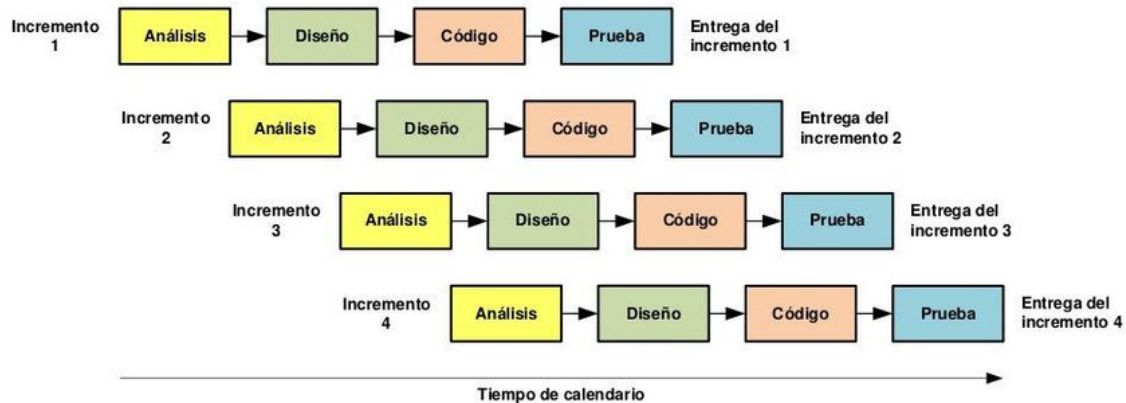


Ilustración 8 Diagrama de flujo del proceso incremental [3]

Como se puede ver en el diagrama anterior, la primera fase del proceso es analizar los requisitos del usuario, definir por qué se va a desarrollar la aplicación, cuáles son los problemas que resuelve y todas aquellas funcionalidades principales que son necesarias para



conseguir el objetivo final. La obtención de esta información se puede llevar a cabo de muchas maneras siendo entre ellas:

- Entrevistas
- Cuestionarios
- Observación

En el desarrollo de PowerLog se han utilizado las entrevistas y la observación:

Gracias a la cercanía con los usuarios potenciales de la aplicación, las entrevistas se han podido hacer personalmente. Mediante reuniones entre los desarrolladores y los entrenadores se expusieron todas aquellas funcionalidades importantes que ya estaban implementadas en las aplicaciones ya existentes. Posteriormente se analizó la lista para detectar posibles añadidos que completasen los casos de uso y finalmente se analizó la mejor forma de unión para todas estas funcionalidades, es decir, cómo hacer una aplicación que ofreciese todos aquellos requisitos que se consideraron necesarios sin sacrificar la experiencia del usuario, tiempos de carga, complejidad cognitiva de la aplicación etc.

La técnica de observación también fue muy útil durante esta primera etapa de desarrollo pues la propia experiencia usando las aplicaciones ya existentes ofrece un punto de vista muy sólido sobre todas las posibilidades que estas ofrecen. En el ámbito del gimnasio es fácil conocer a otros atletas que usen sistemas similares y, por tanto, únicamente con observar cómo usan la aplicación o documento, determinar cuál es el flujo que ellos siguen para cumplir el fin.

La siguiente fase del desarrollo incremental es el diseño, convertir todas esas funcionalidades en una interfaz de una forma intuitiva y sin sobrecargar al usuario. Esta es una de las fases más complicadas para el desarrollo pues se parte desde cero, con una gran cantidad de aspectos a tener en cuenta y a su vez con muchas limitaciones, es decir, el objetivo de la fase de diseño es conseguir juntar todos aquellos requisitos especificados anteriormente, pero al mismo tiempo mantener los principios de usabilidad, experiencia del usuario etc.

La primera fase del diseño comienza siendo un sketch, una primera idea creada de manera rápida y sin alcanzar mucho detalle, principalmente a mano y en papel. En este se describen los componentes principales de cada vista. La siguiente fase son los mockups, cuando los sketches se consideran lo suficientemente correctos, estas ideas se digitalizan y se utiliza software de diseño para conseguir un mayor grado de detalle, además, se conectan las vistas existentes para poder crear una simulación de lo que vería el usuario a la hora de usar la aplicación cuando todas estas pantallas estén implementadas. Durante esta fase, los diseños generados se ponen en común con los usuarios interesados para poder así perfilar detalles y

asegurar que durante el ciclo actual se realizará un trabajo que aportará al resultado final, siendo poco probable que este acabe descartado.

La siguiente fase del ciclo consiste en la implantación. Durante esta fase y ya con todos los diseños finales con gran cantidad de detalle, el equipo de desarrollo puede comenzar a implantar la solución. Por lo general, en un equipo de desarrollo completo, todos los integrantes son capaces de llevar a cabo cualquier tipo de tarea ya sea FrontEnd, BackEnd, Móvil, de gestión de servidores... El desarrollo de PowerLog no fue diferente, con ambas aplicaciones, tanto web como móvil comenzando simultáneamente para crear todas esas funcionalidades y con los desarrolladores trabajando conjuntamente en el BackEnd para tener un sistema completo, que pueda servir información a ambas plataformas sin causar caídas del servicio ni tiempos de espera muy elevados dado que el tiempo de respuesta es una de las métricas más importantes a la hora de desarrollar una REST API. Como se ha explicado anteriormente, esta fase también es un bucle pues es necesario validar que las funcionalidades cumplen con los requisitos esperados. En caso de pasar la validación de los usuarios, el proyecto puede publicarse, pero en caso contrario, los desarrolladores se verán obligados a reestructurar la aplicación o incluso rehacer partes de la misma desde cero para poder publicar el proyecto.

Una vez se publica un proyecto no significa que este está completo. Es imposible crear una aplicación completa desde cero y, por tanto, el proceso completo se irá repitiendo para añadir funcionalidades nuevas a la aplicación, corregir errores o bugs, cambiar diseños, etc. El desarrollo de una aplicación es un proceso complejo, con muchas variables a tener en cuenta y esto deriva en una tarea complicada que rara vez se puede dar por finalizada pues incluso al considerar que la aplicación no necesita más funcionalidades ni ningún añadido, seguirá siendo necesaria una actividad de mantenimiento para gestionar la seguridad, actualizar las dependencias de la aplicación, corregir cualquier pequeño error que pudiese aparecer...

Como se ha explicado, es muy fácil que, durante la fase de diseño o desarrollo, el equipo acabe implementando funcionalidades que o bien no son necesarias o directamente son inútiles. Muchas veces como desarrolladores damos por asumidos ciertos conocimientos y técnicas con ordenadores y por lo tanto tendemos a implementar diseños que, en muchos casos, pueden acabar siendo confusos para personas con menos experiencia en el ámbito. El desarrollo centrado en el usuario permite al equipo de desarrollo mantener al cliente final en el centro del ciclo productivo ya que cualquier nueva tarea, diseño o funcionalidad habría de ser validada por él. Esto ayuda a mantener las ideas alineadas durante todo el proceso de desarrollo, asegurándonos de que el trabajo que se realiza es válido y que no va a acabar siendo

descartado en la versión final del producto porque el cliente no ha podido probarlo hasta entonces.

## 4. Análisis de requisitos

---

En este apartado del trabajo se van a presentar por primera vez los requisitos del sistema que se propone para implementar. A continuación, se presentarán una lista de requisitos funcionales y no funcionales, los diagramas de clases propuestos y también se analizarán otros aspectos del sistema a desarrollar como por ejemplo la seguridad de los servidores, los objetivos de eficiencia energética, las legislaciones actuales y concernientes al objetivo y los desafíos y riesgos que se puede esperar encontrar a lo largo del desarrollo.

### 4.1. Requisitos funcionales

---

A continuación, se presenta una lista de todas las funcionalidades que se espera poder introducir en la aplicación:

- El entrenador debe tener un menú desde el que pueda consultar una lista con todos los atletas con los que se haya generado una relación atleta-entrenador.
- El entrenador debe poder acceder de manera fácil a un código que, una vez compartido al atleta servirá para crear la relación de entrenador y atleta.
- El entrenador tendrá una pantalla desde la que podrá consultar todos los ejercicios que haya ido creando a lo largo del tiempo. La pantalla mostrará las distintas categorías de ejercicios que el entrenador especifique y dentro de las mismas se mostrarán los ejercicios. Se ofrecerá la posibilidad de modificar ejercicios para cambiar el nombre o la categoría, de borrarlos y sobre todo de crear ejercicios nuevos.
- Cuando el entrenador seleccione un atleta para gestionar, será llevado a una vista donde podrá consultar los resultados de cada día a través de un calendario, es decir, los pesos e información que el atleta ha introducido al acabar un entrenamiento. Se debe poder cambiar de día de consulta fácilmente y sin la necesidad de salir de la vista en la que se encuentre.
- La vista de mostrar resultados deberá mostrar toda la información de una forma accesible, evitando la necesidad de usar muchos clics para consultar la información.
- Además de la consulta, se implementará otra vista desde la que el entrenador, tras seleccionar el atleta para gestionar podrá crear nuevos entrenamientos. Esta vista ha

de proporcionar la posibilidad de modificar todos los apartados de un entrenamiento, es decir: modificar el ejercicio, el número de series a realizar, las repeticiones por serie, se podrá especificar el tipo de esfuerzo a medir ofreciendo las siguientes cuatro opciones: RPE, RIR, Peso Fijo y % del RM y se pondrá un apartado de comentarios desde el que se podrá escribir un mensaje al atleta.

- Por último, dado que la estructura de un entrenamiento puede repetirse frecuentemente durante la duración de una planificación se añadirá un botón que sirva para copiar automáticamente el entrenamiento que se creó hace 7 días (el mismo día de la semana anterior). Con esta opción se copiará el esqueleto del entrenamiento al día actual pero se eliminarán los datos que haya introducido el atleta, quedando, así como si fuese un entrenamiento creado desde cero.

## 4.2. Requisitos no funcionales

---

En este apartado se explicarán las propiedades que se esperan del sistema a desarrollar, aquellos apartados que son generalmente transparentes al usuario tales como aspectos tecnológicos o de eficiencia pero que pueden resultar en una menor satisfacción del usuario objetivo o en un aumento de los costes de producción

- La capa de persistencia de la base de datos ha de ser independiente al resto de la infraestructura. Se implementará un sistema cloud de DBMS para ahorrar tiempo a la hora de montar la base de datos y facilitar la escalabilidad de la misma frente a un aumento en el tráfico o en el tamaño de la información a guardar.
- Las peticiones al servidor deben ser resueltas en un tiempo máximo de un segundo. Cualquier proceso que pueda tardar más de este tiempo o pueda llegar a bloquear la entrada de nuevas peticiones será delegado a un servicio externo.
- Todo el código desplegado en los servidores ha de ser testeado antes de entrar al servidor de producción. Se usarán tests unitarios y tests de integración para asegurar el correcto funcionamiento del sistema y de todas las nuevas funcionalidades.
- El servicio backend será flexible en el despliegue, siendo posible desplegarlo tanto en un entorno Docker como en una máquina virtual.
- Se habilitará una estrategia de rollback para permitir que cualquier fallo introducido en el entorno de producción sea fácilmente.
- Los despliegues serán automatizados y no podrán tardar más de 10 minutos en reflejarse en los entornos de producción.

- Será conveniente utilizar una estrategia blue-green de despliegue para hacer así que los despliegues sean transparentes al usuario.

### 4.3. Casos de uso

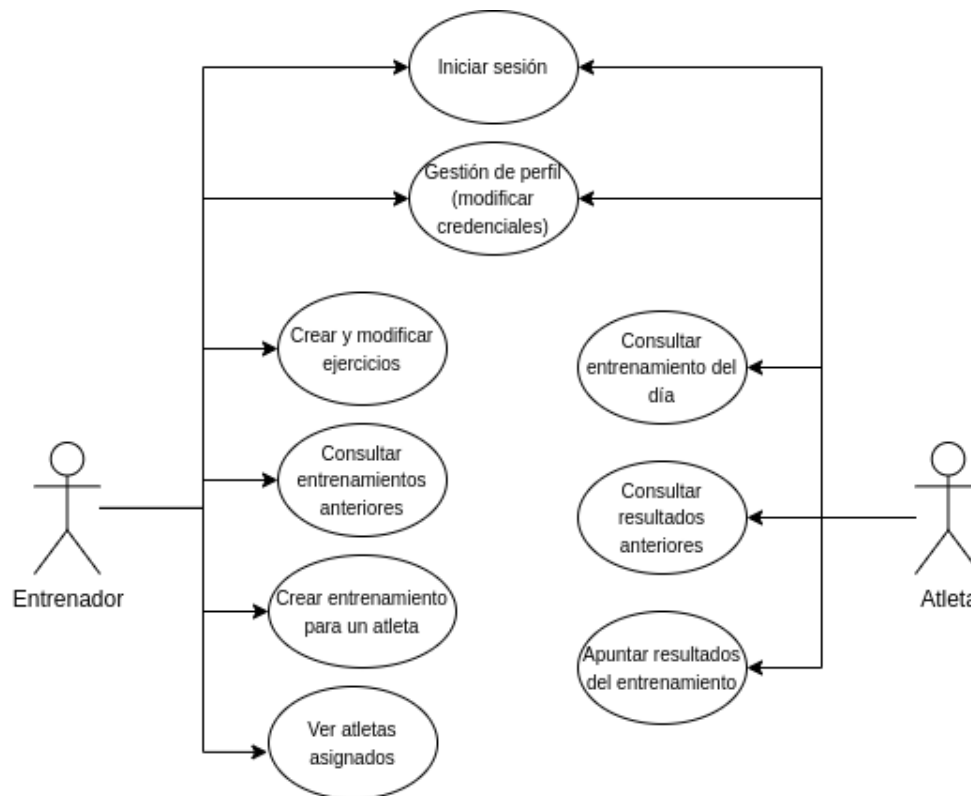


Ilustración 9 Diagrama de casos de uso

En el diagrama de casos de uso se exponen todas las funcionalidades que se contemplan para la aplicación propuesta. A pesar de que los casos de uso de los atletas se desarrollan en “PowerLog Móvil. Una aplicación móvil para la gestión de entrenamientos con componentes sociales para atletas y entrenadores”, es necesario representar todo el sistema como un conjunto para facilitar la comprensión de la aplicación. Los casos de uso planteados como objetivo en el alcance de este proyecto son los siguientes:

- Iniciar sesión, registro y modificación de perfil, funcionalidad genérica de cualquier aplicación donde se requiera que un usuario pueda crear una cuenta, identificarse para recuperar sus datos y modificar cualquiera de los campos de su perfil para mantener su información actualizada.

- Crear y modificar ejercicios, el entrenador debe tener la posibilidad de crear ejercicios para que estos sean posteriormente consultados por el atleta en la aplicación móvil. También ha de poder modificar la información de un ejercicio si se considerase necesario.
- Consultar entrenamientos anteriores, dado que los resultados anteriores de un atleta sirven para determinar los entrenamientos futuros y poner en perspectiva la evolución, una parte crucial de la aplicación es que el entrenador sea capaz de consultar estos datos de una forma sencilla y fiable.
- Crear entrenamiento para un atleta, se considera la creación de un entrenamiento el juntar varios ejercicios junto con otros parámetros tales como comentarios, inicio y fin de bloque y la asignación de estos datos a un atleta en concreto.
- Ver atletas asignados, el entrenador tiene que ser capaz de acceder de manera fácil a un listado de todos sus atletas además de poder cambiar el atleta al cual le va a crear los entrenamientos.

## 4.4. Diagrama de clases

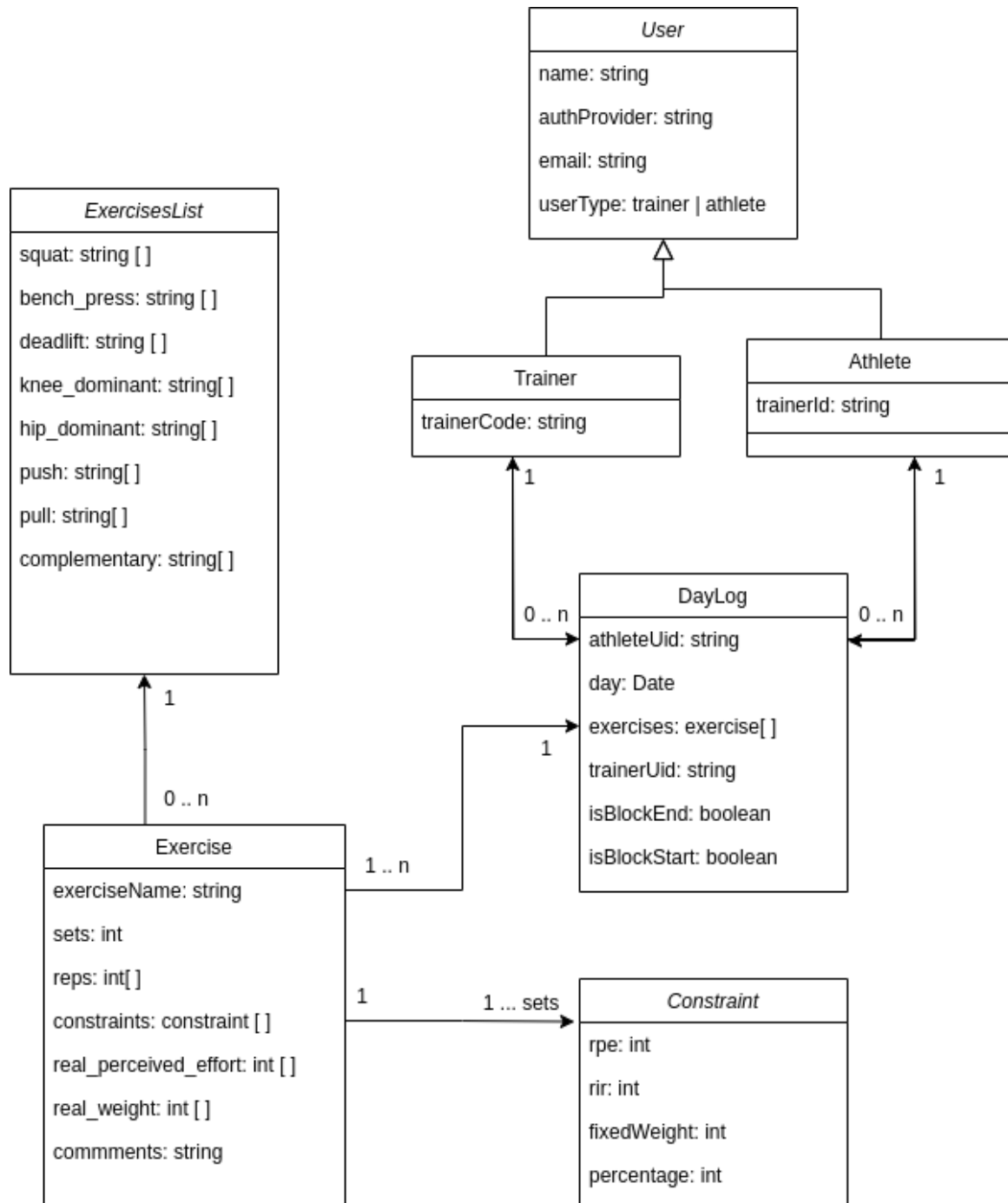


Ilustración 10 Diagrama de clases de la aplicación PowerLog



En el diagrama se muestra como se gestionará la aplicación desde dentro, el día de entrenamiento creado por el entrenador se relacionará con el atleta a través de la Id del mismo. Cada entrenamiento contendrá una lista de ejercicios, cada uno con sus respectivos parámetros tal y como se indica en la figura. Además de esto, cada entrenamiento se identificará por dos valores únicos en conjunto, día y atleta, es decir, cada atleta puede tener únicamente un entrenamiento por día. Cada entrenador podrá crear su propia lista de ejercicios para añadirlos a su gusto y preferencia.

Cada uno de los campos tiene especificado un tipo, por ejemplo, la palabra “string” significa que el campo será una cadena de texto, “int” es para números. Otro caso es el tipo “exercise[]” el cual indica que el campo será una lista de ejercicios.

## 5. Aspectos de planificación

---

Las aplicaciones desarrolladas en la actualidad son mucho más extensas de lo que parece a simple vista. Además de una interfaz bonita y tiempos de carga reducidos, los desarrolladores y diseñadores deben tener en cuenta otros aspectos tales como la seguridad de los datos y el tráfico, la eficiencia energética y algorítmica de la aplicación y otros aspectos fuera del marco de la ingeniería, como leyes o estándares a los que el desarrollo debe adherirse.

El conjunto de todos estos aspectos es muy importante pues puede condicionar en gran medida el desarrollo de una aplicación además de otros aspectos como su coste.

### 5.1. Seguridad

---

La seguridad del sistema a desarrollar es un criterio cada vez más importante pues no es extraño que nuestras aplicaciones guarden datos personales de los usuarios que han de ser tratados con discreción y que bajo ninguna circunstancia pueden ser publicados sin el consentimiento explícito del propietario.

En primer lugar las credenciales y datos personales del usuario: estos datos se van a usar para generar el perfil del usuario, ya sea atleta o entrenador. En este caso, para gestionar la información de inicio de sesión y mantener todos los datos seguros además de cumplir con la legislación vigente, se usará Google Firebase, un sistema cloud de Google que nos permite crear un flujo de registro, inicio de sesión y mantenimiento de perfiles de usuario de manera rápida, automática y segura, de este modo la aplicación podrá ser utilizada por usuarios reales sin la necesidad de gestionar los apartados de seguridad de datos en profundidad.

Los datos de entrenamientos estarán guardados en una base de datos MongoDB. Esta tecnología permite crear la base de datos en diferentes entornos, ya sea localmente para desarrollar, en los servidores propios de la aplicación o incluso utilizar algún servicio en la nube que facilite la creación y mantenimiento de la base de datos. En este caso y como ya se ha explicado en el apartado de requisitos no funcionales, además de tener una base de datos más fiable y escalable, MongoDB Atlas[5] nos ofrece bases de datos más seguras pudiendo configurar fácilmente la configuración de red de nuestra base de datos para limitar ciertas IPs

o direcciones desde las que nos podremos conectar, crear usuarios con distintos permisos para evitar escrituras o lecturas indeseadas en la base de datos etc.

En cuanto al servidor backend, se alijará en una máquina virtual en red en Digital Ocean[6]. En esta plataforma podemos crear “droplets”, máquinas virtuales con un sistema operativo y recursos limitados en las que podemos instalar nuestra aplicación para servirla a la red. El uso de una plataforma como digital ocean ofrece muchas medidas de seguridad como una correcta configuración en red, estableciendo un límite para no exceder el límite mensual de ancho de banda ni de CPU evitando así incurrir en mayores costes no inesperados. Para hostear el backend en el servidor será necesario configurar el firewall de manera que solo acepte tráfico en el puerto 443 y 80 (HTTPS y HTTP) respectivamente y además, utilizando NGINX[7] se configurará el *port forwarding* desde el puerto 443 al puerto en el que nuestro backend esté escuchando. Además, se implementarán librerías en el lado del servidor tales como *CORS* para bloquear peticiones desde *dominios* que no sean el nuestro y *HELMET*, una librería que se encarga de realizar pequeñas modificaciones en *las cabeceras HTTP* para conseguir así ocultar ciertas piezas de información relevantes.

En el *frontend* dado que es el lado del cliente, queda de parte del usuario protegerse del malware y páginas fraudulentas de la red, sin embargo el programador ha de mantener una serie de estándares de seguridad entre ellos *sanear y parsear* los datos introducidos por el usuario para protegerse de los ataques de XSS, evitar en la medida de lo posible la inyección de servicios de terceros como Google Analytics, Google Tag Manager, Intercom etc ya que si alguno de estos servicios se viese comprometido, nuestra aplicación también estaría en riesgo y siempre que se añada un enlace a una aplicación externa hay que asegurarse de mantener la política “Referrer-policy” con el valor “no-referrer” para evitar que las páginas a las que redireccionamos puedan obtener datos de la sesión del usuario.

## 5.2. Análisis energético

---

Los componentes y equipos electrónicos han sufrido una gran evolución durante las últimas décadas hasta el punto que son capaces de ejecutar y mantener en activo la mayoría de aplicaciones que se suelen utilizar hoy en día, incluso hasta el punto de permitirnos tener varias de ellas abiertas al mismo tiempo. La mejora en el rendimiento y potencia junto con una mayor eficiencia de la mayoría de programas y herramientas nos ofrece permite a los programadores la posibilidad de crear programas más complejos e incluso, en ocasiones, menos eficientes de lo que deberían ser ya que por lo general, la fase de análisis y optimización

en el desarrollo de un programa suele estar situada al final del ciclo y puede convertirse en un proceso extremadamente tedioso y muy caro.

No obstante, en la fase de programación y de diseño es conveniente pensar y diseñar las soluciones al problema de la manera que se considere más eficiente, entendiendo como eficiencia la carga sobre el equipo en el que se va a ejecutar, la *cambiabilidad* del sistema, la facilidad para testarlo y asegurarnos de que funciona correctamente. Para conseguir mantener la correctitud del sistema en estos aspectos se pueden seguir una serie de pasos para conseguir una aplicación más eficiente como por ejemplo:

- Utilizar un sistema de *caché*: el cliente sincroniza sus datos con el servidor a través de lo que se conoce como “peticiones en red”, procesos de intercambio de datos en red que son muy caros a nivel computacional además de lentos e ineficientes. Es por esto que usar una caché local, es decir, mantener datos de acceso frecuente dentro del cliente se convierte en una gran idea. Cuando el cliente necesita pedir una serie de datos al servidor lo primero que hace es comprobar si ya ha obtenido esos datos con anterioridad para evitar tener que lanzar una petición al servidor y ahorrando así tiempo y energía.
- Actualización de componentes: el diseño web moderno se basa en componentes, pequeños contenedores de datos con un cierto formato y estilos que se pueden reutilizar por toda la aplicación, facilitando así la modificación y testeabilidad de los mismos. Estos componentes a veces mantienen un estado (datos actuales que están mostrando) que necesita ser cambiado cuando un usuario realiza una acción o interacciona con la página web. Estas acciones de cambio son muy costosas para el cliente si el componente es muy grande y ha de ser *pintado* por completo de nuevo, por eso es conveniente mantener el estado lo más cerca posible de los apartados de datos que sean dependientes de ese dato, para poder reconstruir los componentes con las modificaciones más pequeñas posibles.
- En el lado del servidor se han de definir los procesos que recibirán las peticiones enviadas por el cliente. Estas peticiones esperan una serie de datos concretos para poder realizar la ejecución esperada (si se envía una petición de registro de un usuario se espera recibir un nombre, un email, una contraseña, etc) que, en caso de no existir cualquiera de estos datos o de tener un formato correcto (un email sin @) podrían causar un error en la ejecución del proceso. Tanto recuperarse de un error reiniciando el proceso, ejecutar una petición sin los datos que se esperaban o tener incluso que reiniciar el servidor por completo en algunos casos extremos donde la falta de datos puede causar incluso una corrupción completa del sistema son procedimientos altamente ineficientes y costosos, por esto es conveniente crear un sistema de

validación que se ejecute antes que las peticiones del cliente y donde podamos validar que los datos que se han recibido del cliente sean los correctos y los esperados y por tanto la petición pueda proceder con total normalidad.

Tras este análisis queda claro que crear sistemas eficientes y sostenibles es una tarea difícil pero no imposible hoy en día, sobre todo teniendo en cuenta la gran cantidad de herramientas que se le ofrece al programador como por ejemplo librerías de control de caché para manejar la caché en el cliente o funcionalidades para validar clases en el lado del cliente que automáticamente pueden rechazar una petición en caso de que esta no contenga los datos necesarios.

### 5.3. Análisis del marco legal y ético

---

Hoy en día son muchas las leyes que un nuevo servicio debe asegurarse de cumplir antes de presentarse al mercado además de ser muy claro sobre como se van a utilizar los datos de los usuarios en cada momento y asegurarse de obtener consentimiento o bien haciendo que el consentimiento del tratamiento de los datos conste como implícito al utilizar la aplicación o añadiendo un botón de confirmación de consentimiento cuando el usuario envía datos a nuestro sistema.

Como se ha explicado con anterioridad, los datos de los usuarios serán guardados y gestionados a través de Google Firebase[8] para asegurar que el servicio de bases de datos que se utiliza en la nube cumple con la legislación vigente además de poseer varias certificaciones por el correcto proceder de la empresa con respecto a los datos de los usuarios, tal y como se explica en su página de términos y condiciones[9]. Tras confirmar que los servicios que se usan en la aplicación cumplen con las leyes, es necesario garantizar una serie de condiciones en nuestro sistema para que aquellos componentes que sean desarrollados en este proyecto cumplan con los requisitos legales:

- Transferir los datos a través de un canal seguro y encriptado, en este caso utilizando el protocolo HTTPS
- Utilización de permisos para que un usuario únicamente pueda leer sus datos y aquellos datos que sean necesarios para el correcto funcionamiento de la aplicación (como un entrenador que ha de conocer el nombre de los atletas que entrena) controlando también que un usuario únicamente pueda modificar sus propios datos personales.

- También se han de garantizar al usuario la totalidad de sus derechos sobre sus datos personales como por ejemplo, en caso de desearlo, ha de poder eliminar por completo sus datos del sistema.

## 5.4. Análisis de riesgos

---

El desarrollo de aplicaciones centrado en el usuario no es una tarea fácil, incluso menos aún si no se cuenta con una amplia experiencia en el sector y con recursos muy limitados. Los contratiempos que se pueden encontrar a lo largo del proceso de desarrollo son muchos y muy variados, pero entre ellos cabe destacar los siguientes:

- Errores en la especificación inicial. Si no se tiene una visión muy clara del proyecto es fácil desviarse de la idea principal y acabar implementando funcionalidades complejas y que los usuarios no necesitan o no han pedido.
- Competencia. Cuando se desarrolla una aplicación en un mercado desde cero es fácil que los usuarios ya estén acostumbrados a una aplicación en concreto y por tanto no les convenga realizar el cambio puesto que este conllevaría una modificación casi por completo de su flujo de trabajo.
- Conflictos de interés. Al tener muchos usuarios utilizando la aplicación de una forma distinta, pero con recursos muy limitados existe la posibilidad de que cada persona solicite una funcionalidad por separado o incluso totalmente contraria a otra que haya solicitado otro usuario. Esto suele derivar en conflictos y supone un aumento de la carga de trabajo pues el equipo de desarrollo ha de comunicarse con ambas partes y establecer un punto común para el desarrollo o tener que priorizar ciertas tareas, retrasando la entrega de otras. Esto puede causar el descontento de muchas personas y por tanto la pérdida de usuarios de la aplicación.
- Falta de actualizaciones. Los recursos son incomparables a los que puede tener una empresa grande y con financiación y por tanto es posible que la aplicación no se actualice regularmente con nuevas características y mejoras, causando que los usuarios pierdan el interés y opten por utilizar otras aplicaciones que sean más actualizadas y relevantes.
- Falta de confianza. Al ser un proyecto nuevo y sin mucho bagaje de experiencia, ni posibilidad de auditorías ni sistemas de seguridad muy avanzados es normal que se

genere una cierta desconfianza o aversión a la hora de tener que dar datos personales para poder crear un perfil en la aplicación

## 5.5. Identificación y análisis de soluciones disponibles

---

Como ya se ha mostrado anteriormente, existe una gran variedad de maneras de enfocar una aplicación de entrenamientos y la implementación de una u otra de esas variantes, por pequeñas que puedan llegar a parecer, pueden acabar generando una aplicación totalmente distinta con un flujo de trabajo que no tendría nada que ver con la implementación de otra de las variantes, a continuación, se expondrán varios de los enfoques que se han visto hasta ahora:

- Una cuenta por atleta y sin cuenta de entrenador. En este caso, como se ha visto en Hevy y en RTS, el entrenador pierde por completo la posibilidad de tener los datos separados del atleta y por ende, en muchos casos la confidencialidad. Esto puede no ser un punto negativo muy importante, pero a veces es necesario que el entrenador pueda guardar ciertos cálculos para sí mismo y así evitar prejuicios y afectar al atleta en cuanto a los entrenamientos. Además, tal y como se ha explicado con anterioridad, esto acaba haciendo que el entrenador tenga que recordar un gran número de credenciales lo cual resulta muy difícil y tedioso.
- Generar únicamente una aplicación para ordenador (de escritorio o web). Tal y como se ha explicado en el caso de RTS, las aplicaciones de ordenador ofrecen la ventaja de ser muy versátiles, además de muy flexibles por el mayor tamaño de las pantallas para mostrar información. También se puede asumir que el trabajo del entrenador (parecido a la ofimática, de entrada y consulta de datos) es más cómodo en una pantalla más grande y más rápido con teclado y ratón pero al mismo tiempo perdemos la portabilidad del móvil, que ofrece al atleta la posibilidad de ir consultando y guardando los datos en tiempo real mientras realiza el entrenamiento. Existe la posibilidad de diseñar la aplicación de ordenador de manera que sea *responsive* pero esto deriva en una mayor carga de trabajo en el equipo de desarrollo además de ser mucho menos eficiente que una aplicación nativa[10].

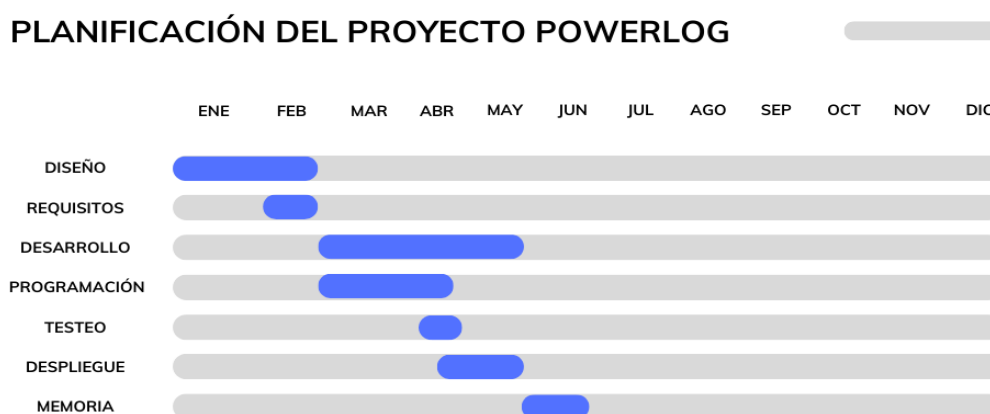
- Aplicación móvil nativa. La aplicación nativa de móvil parece ser una mejor alternativa para la vista de los atletas, una aplicación ligera, minimalista, que permita a los usuarios poder guardar y consultar sus resultados en todo momento sin tener que preocuparse por causar un gran consumo de batería o de datos móviles. Sin embargo, la vista del entrenador, a pesar de ser algo posible de implementar en móvil, acabaría resultando mucho más pesada ya que ha de realizar más interacciones con el servidor, causando así un mayor consumo en la batería. Además, con menos superficie de pantalla sobre la que mostrar las cosas, el número de clics para acceder a determinada información se haría cada vez mayor, aumentando así la carga cognitiva del entrenador al hacer un uso prolongado de la aplicación.

En este proyecto, se ha tomado la decisión de implementar cada uno de los aspectos conforme parecía más conveniente a cada caso de uso, por eso hemos trabajado de cerca con entrenadores y atletas muy variados para conseguir un resultado que cumpla los requisitos deseados pero sin dejar de lado la experiencia del usuario y la ergonomía de la aplicación.



## 5.6. Plan de trabajo

---



*Ilustración 11 Planificación del desarrollo de PowerLog.*

El alcance del proyecto de desarrollo de PowerLog es una iteración de un ciclo constante de desarrollo en el que cada vez se pretende implementar una funcionalidad nueva o refinar una aplicación ya existente. En este caso se parte desde cero y por tanto la primera etapa de este ciclo es el diseño, crear bocetos y sketches junto con los usuarios potenciales de la aplicación para entender cuál es la mejor forma de crear la aplicación.

Cuando se ha diseñado la aplicación, el equipo de desarrolladores pasa a elicitar los requisitos, tanto funcionales como no funcionales, es decir, todo aquello que entrenadores y atletas hayan considerado necesario mencionar a lo largo de las entrevistas o en los diseños, de esta manera se pretende obtener una lista extensa y detallada de todas las funcionalidades a implementar. Además de las funcionalidades es conveniente que los desarrolladores tengan en cuenta los requisitos no funcionales, aquellos conceptos no visibles al usuario pero que pueden afectar en gran manera a la experiencia del usuario tales como tiempos de carga, peso de la aplicación, consumo de energía etc.

Lo siguiente es la fase de desarrollo, donde el equipo programará tanto la aplicación web para los entrenadores, la aplicación móvil de los atletas y el servidor junto con las bases de datos necesarias para poder comunicar las aplicaciones entre ellas y persistir los datos. Además de esto, el sistema será completamente testeado para verificar la seguridad y estabilidad del mismo, garantizando así que la entrega se pueda hacer con normalidad y los usuarios puedan usar la aplicación y por último la fase de despliegue, donde las aplicaciones ya programadas y verificadas son publicadas y abiertas al público.

Como última fase del proyecto se ha considerado la creación de una memoria donde se refleja y explica todo el proceso.

Durante todo este proceso, se han invertido una media de dos horas de trabajo al día, sin incluir los días festivos, pero sí los fines de semana lo cual resulta en aproximadamente unos ciento cincuenta días de trabajo y en aproximadamente unas trescientas horas de trabajo.

## 6. Diseño de la aplicación

---

En este capítulo se presenta el resultado final del diseño de la aplicación, es decir, la implantación final de aspectos tales como las clases de la base de datos o la evolución de las interfaces que se han diseñado a lo largo del proyecto

### 6.1. Diseño de la base de datos

---

Como se ha explicado anteriormente, para el desarrollo de este proyecto se decidió utilizar como sistema de bases de datos la tecnología de MongoDB. MongoDB nos ofrece un sistema de bases de datos no relacional basado en documentos. Los documentos son entradas de datos compuestas por el sistema clave-valor y que además permiten tener diferentes entradas dentro de un mismo documento.

```
class Exercise {
  exercise_name: string;
  sets: number;
  reps: string[];
  constraints: IConstraint[];
  real_perceived_effort: number[];
  real_weight: number[];
  comments: string;
}
```

*Ilustración 12 Clase de ejercicio en la base de datos*

En la figura anterior se muestran los campos de un ejercicio en la base de datos. Como se puede ver, cada campo del ejercicio se guardará con el sistema clave-valor (por ejemplo “sets”:4).

```
export class DayLog {
  @Prop({ required: true })
  day: Date;

  @Prop({ required: true })
  exercises: Exercise[];

  @Prop({ required: true })
  athleteUid: string;

  @Prop({ required: false })
  isBlockStart: boolean;

  @Prop({ required: false })
  isBlockEnd: boolean;

  @Prop({ required: false })
```

```

athleteComments: string;

@Prop({ required: false })
athleteFeelings: number;
}

```

*Ilustración 13 Clase Daylog, un día de entrenamiento*

En esta figura se muestra la clase correspondiente a un entrenamiento, como se puede ver se identifica con una fecha, la id de un atleta y además contiene campos como un conjunto de ejercicios a realizar y otros campos extras para añadir información de sobre los entrenamientos.

```

export class ExerciseList {
  @Prop({ required: true })
  trainerId: string;
  @Prop({ required: true })
  squat: string[]
  @Prop({ required: true })
  bench_press: string[]
  @Prop({ required: true })
  deadlift: string[]
  @Prop({ required: true })
  knee_dominant: string[]
  @Prop({ required: true })
  hip_dominant: string[]
  @Prop({ required: true })
  push: string[]
  @Prop({ required: true })
  pull: string[]
  @Prop({ required: true })
  complementary: string[]
}

```

*Ilustración 14 Clase que representa la lista de ejercicios de cada entrenador*

La clase ExerciseList representa la estructura en la base de datos de la lista de ejercicios de cada entrenador que se ha explicado anteriormente en esta memoria. Como se puede ver, esta clase se identifica con la id del entrenador y el nombre de cada una de las categorías de ejercicios existentes como clave para un conjunto de nombres de ejercicios.

## 6.2. Diseño de las interfaces

---

A pesar de la visión de la aplicación siempre fue clara y con buena cantidad de detalle, plasmar la idea sobre un boceto se convirtió en una tarea realmente tediosa. Transmitir una idea mediante palabras es una tarea complicada y más aún cuando las partes implicadas provienen de dos campos radicalmente distintos. El primer paso fue buscar un punto de inicio común

que, tras varios intentos resultó en un boceto en papel donde se representaban los componentes principales de la aplicación.

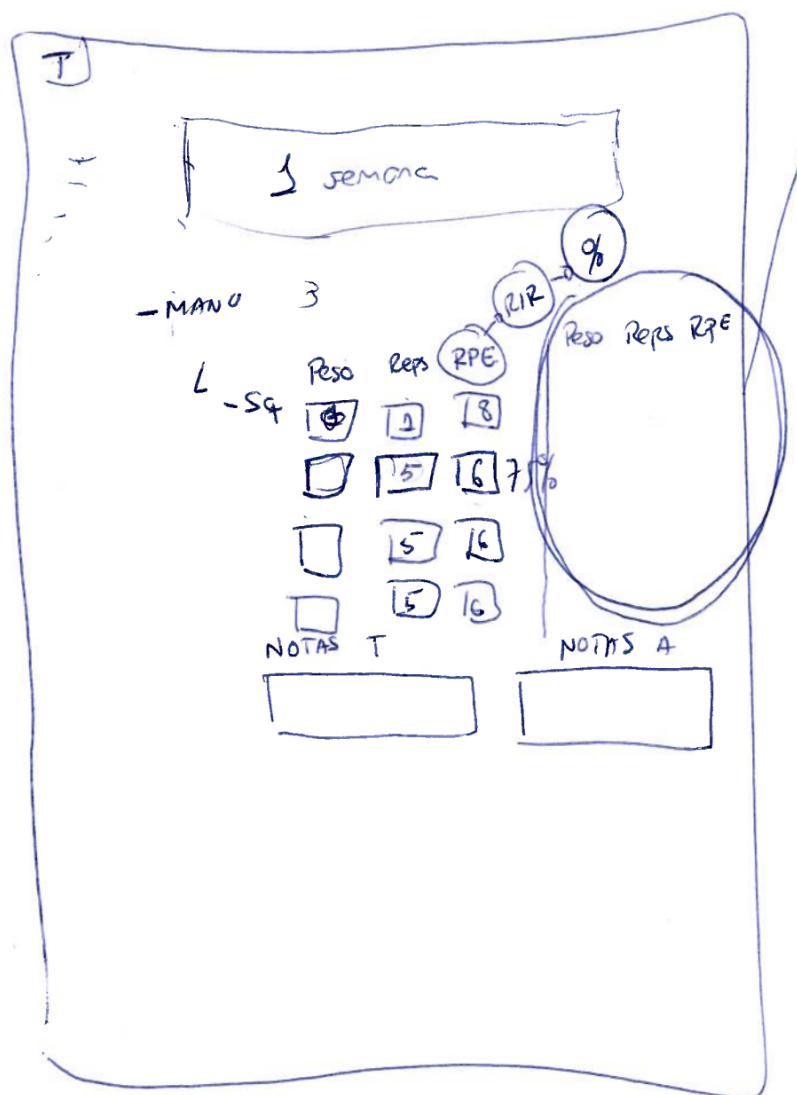


Ilustración 15 Primer boceto en papel escaneado digitalmente

Como se puede ver, la figura anterior representa una visión básica del producto objetivo pero puede ayudar a las partes implicadas a asentar una base común sobre la cual construir. Se representa un listado de atletas en la esquina superior izquierda, se enumeran varios campos a representar por cada uno de los ejercicios, como distribuir la información etc.

Tras desarrollar este primer diseño, se pasó a digitalizar y añadir detalle para poder pasar a la fase de desarrollo. Tras varias sesiones y correcciones por parte de los entrenadores, surgió el primer diseño de la aplicación web.



*Ilustración 16 Primer diseño de la web*

La figura anterior representa lo que acabaría convirtiéndose en la primera versión de la aplicación web de PowerLog, un diseño básico y minimalista, que implementaba las funcionalidades fundamentales de la aplicación para permitir que los entrenadores comenzasen a utilizarla con sus atletas.

Tras implementar esta primera versión, se comienza a buscar el desarrollo de una interfaz más bonita para el entrenador, además implementar nuevas funcionalidades como por ejemplo la lista de ejercicios o los comentarios del atleta y, por tanto, en la fase de diseño del nuevo sprint se desarrolla una nueva versión de los diseños.

Lista de ejercicios Cerrar Sesión

Manuel Eijshi  
Pepe Garcia  
Juaquin Perez  
Hola hollita

Calendario

### Resultados del atleta

**Nombre del ejercicio**

Series	Repeticiones	RPE	Peso Real	RPE Real
1	x	x	x	x
2	x	x	x	x

**Nombre del ejercicio**

Series	Repeticiones	RPE	Peso Real	RPE Real
1	x	x	x	x
2	x	x	x	x

**Nombre del ejercicio**

Series	Repeticiones	RPE	Peso Real	RPE Real
1	x	x	x	x
2	x	x	x	x

Comentarios: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

SENSACIONES Y COMENTARIOS DEL ATLETA

*Ilustración 17 Implantación actual de la web*

La figura anterior corresponde al último diseño desarrollado en el proyecto. Como se puede ver, el calendario se ha pasado al lazo izquierdo de la pantalla, de esta manera, el entrenador podría navegar por los detalles de un entrenamiento manteniendo la información relativa a la fecha y al atleta siempre a la vista. También se añadió una barra de navegación superior para permitir que el entrenador accediese a la lista de ejercicios de manera rápida y con pocos clicks

## 7. Desarrollo de la aplicación

En el siguiente apartado se explicará la arquitectura que se ha seleccionado para el desarrollo de la aplicación PowerLog. Se comenzará explicando la infraestructura que se ha construido desde o y poco a poco se irá entrando en detalle de cada uno de los componentes que conforman la aplicación.

### 7.1. Arquitectura del sistema

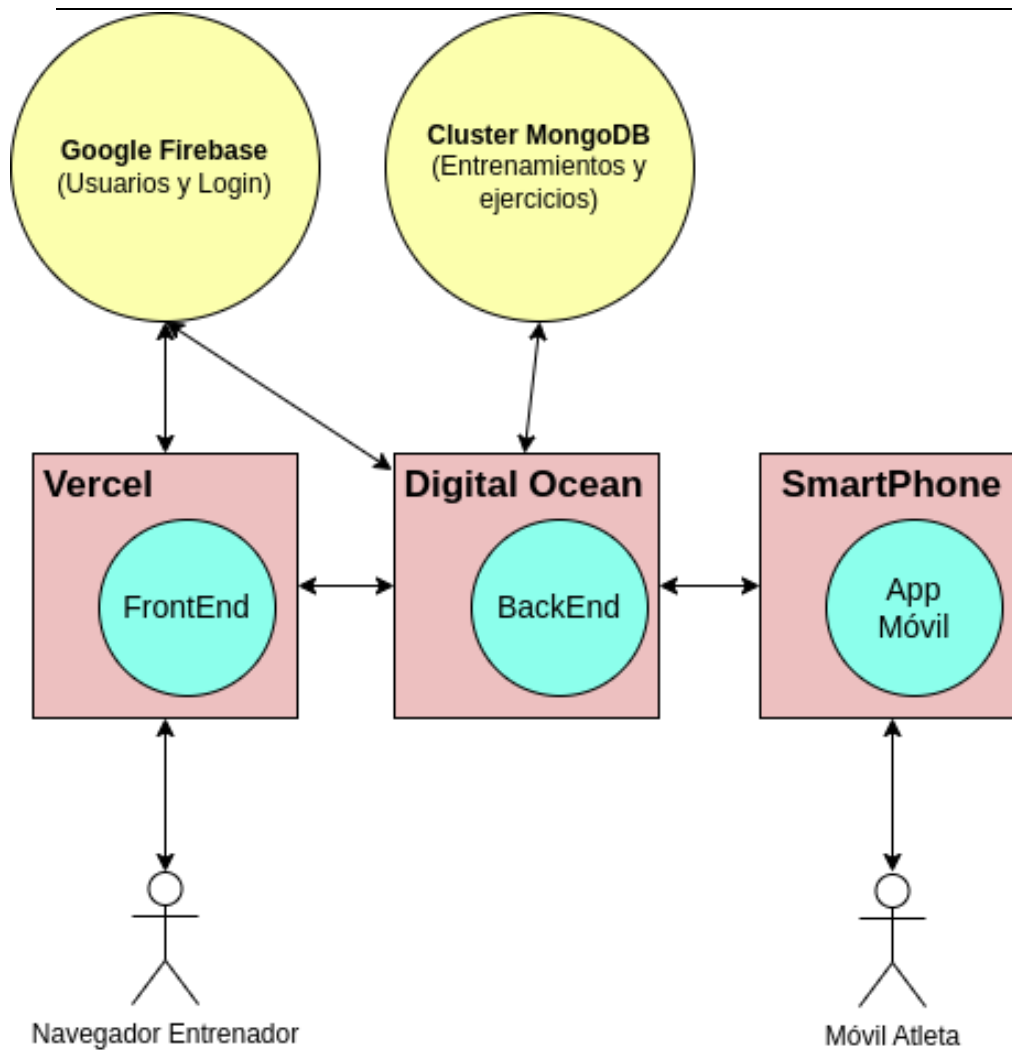


Ilustración 18 Arquitectura del sistema PowerLog



En el diagrama anterior se puede ver fácilmente cuales van a ser las 3 piezas principales de la aplicación. Para la vista del entrenador, se creará un FrontEnd, una página web que se alojará en un servicio llamado Vercel[11]. Vercel es una plataforma cloud donde podemos alojar nuestras aplicaciones web fácilmente. Viene con varios *frameworks* configurados por defecto, tales como ReactJs[12], NextJs[13], Vue[14], Svelte[15]. Además, Vercel ofrece también un “*tier gratuito*” para aplicaciones de Hobby, lo cual lo convierte en una gran alternativa para lanzar inicialmente una aplicación, crear una prueba de concepto o un proyecto personal del desarrollador. Es una aplicación moderna y eficiente, con un gran número de valoraciones sobre todo positivas por parte de la comunidad de desarrolladores y grandes empresas.

El servicio BackEnd, la capa intermedia de tratamiento de datos que se situará entre los clientes y las bases de datos estará instalado en un *Droplet* de Digital Ocean. El Droplet nos ofrece una máquina virtual de bajo coste y con recursos suficientes para mantener una aplicación con un tráfico limitado y no mucha carga de CPU. Estas máquinas están conectadas a internet para que podamos conectarnos desde nuestro ordenador de manera remota y son altamente personalizables en cuanto a *sistema operativo, región, recursos físicos*. Además de la opción de configurar la máquina desde cero para usuarios más avanzados también ofrece imágenes de máquinas ya preconfiguradas según las necesidades del cliente ya sea para alojar una página web, un backend, un almacenamiento de archivos etc.

En cuanto a la aplicación móvil, será desarrollada en Flutter[16], un framework del lenguaje de programación Dart[17] que permite crear aplicaciones multiplataforma, es decir, que una misma base de código se puede utilizar para diferentes dispositivos como por ejemplo una página web, una aplicación nativa de escritorio o para aplicaciones móviles para los sistemas operativos más vendidos de la industria, Android e IOS. La aplicación móvil se desarrollará principalmente para Android e IOS, obviando el resto de las plataformas soportadas por el lenguaje y se distribuirá a través de los siguientes medios:

- Para sistemas Android, se *buildeará* un archivo .apk que será enviado a los usuarios a sus móviles según el mismo usuario pueda preferir recibir el instalable a través de correo electrónico, aplicaciones de mensajería o alguna plataforma de almacenamiento de ficheros.
- Para sistemas *IOS (iPhone)*, dado que la opción de enviar manualmente el archivo instalable al usuario no es soportada en esta plataforma, se creará una cuenta de desarrollador de *Apple* y se distribuirá la aplicación en *TestFlight*, un formato de prueba de las aplicaciones de *Apple* donde se pueden instalar aplicaciones sin la necesidad de publicarlas en la *App Store* .

La arquitectura que se seguirá en la aplicación del servidor es una aplicación reducida de la arquitectura hexagonal. En esta arquitectura, los datos que llegan en las peticiones, tras ser validados, son recibidos por el controlador, donde se hacen las primeras validaciones y *parseos*, donde se convierten en un Data Transfer Object (*DTO*). Tras este primer paso, el controlador envía los datos al servicio, el cual se encarga de asegurar que los datos que han llegado sean inmutables (*VALUEOBJECT*) para asegurar la correctitud de las peticiones enviadas por el cliente y de enrutar, dentro del sistema BackEnd el recién creado ValueObject por el controlador a la función del repositorio que se encargará de generar la consulta de bases de datos adecuada para enviar los datos o la consulta a la base de datos que corresponda. Las consultas de bases de datos, en caso de ser una operación de escritura pueden no devolver nada más que un código de operación indicando el fin de la operación, pero en caso de ser una consulta de lectura, el repositorio devolverá una Entidad la cual será retornada al cliente.

Todos los servicios antes mencionados están desplegados en diferentes servidores de empresas totalmente distintas y que a su vez están usando redes independientes por tanto es necesario definir un protocolo de comunicaciones entre los mismos:

El protocolo utilizado es *HTTPS*, la versión cifrada del protocolo *HTTP* a través del cual, los servicios pueden intercambiar información por la red de forma segura sin la necesidad de estar conectados físicamente, únicamente es necesario definir un “lenguaje” (en este caso se ha usado *JSON*) que todos sean capaces de entender o adaptar a sus necesidades.

## 7.2. Tecnologías utilizadas

Dado el alcance de este proyecto, solo se van a explicar las arquitecturas empleadas en la vista del entrenador a través de la página web de PowerLog y la arquitectura utilizada en el BackEnd, el servicio del servidor que servirá para comunicar todas las distintas piezas de nuestro sistema.

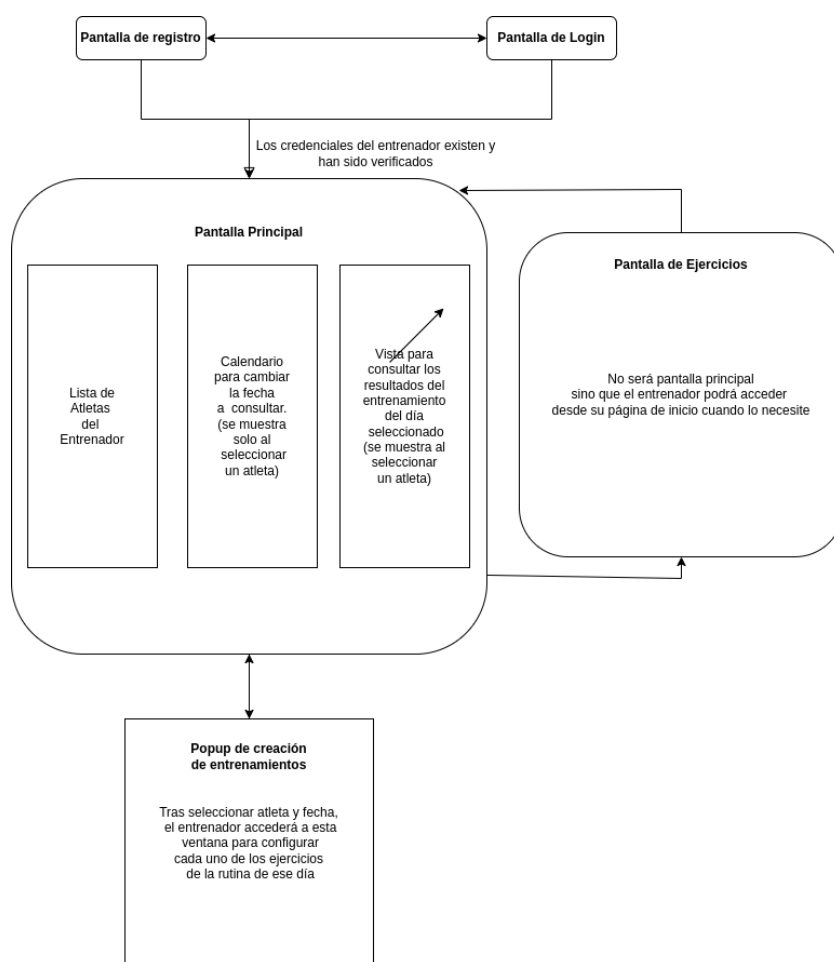


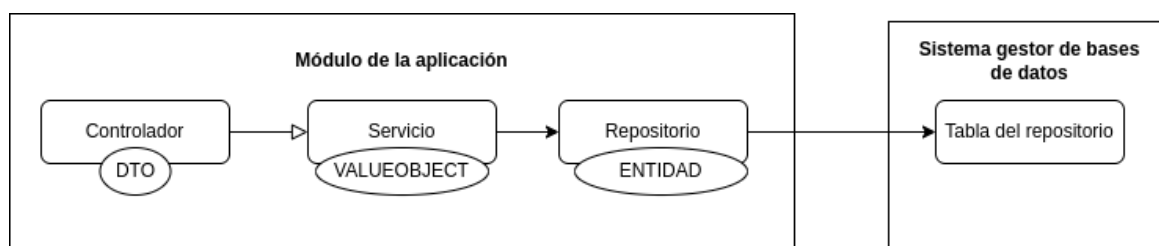
Ilustración 19 Diagrama de flujo en las vistas de la página web

Como se ha explicado anteriormente, la página web estará desarrollada en [ReactJs](#) con el lenguaje [Typescript](#).

React, es una librería que te permite construir interfaces mediante piezas llamadas componentes. React permite al desarrollador crear los componentes para luego combinarlos entre ellos y formar páginas enteras, pantallas e incluso aplicaciones

~ Página web de ReactJs en mayo de 2023

Como se puede apreciar en el diagrama anterior, la única forma de acceder a la aplicación que tiene un entrenador es utilizando su cuenta, ya sea creándola en el momento o iniciando sesión. Una vez las credenciales son validadas por el servicio de Firebase, el usuario es redirigido a la página principal donde podrá encontrar una lista con todos los usuarios asociados a su cuenta. Al seleccionar un atleta, aparecerán en la misma pantalla dos componentes nuevos, un calendario para cambiar de día y una pantalla donde el entrenador pueda consultar los entrenamientos. En el menú de navegación el entrenador tendrá siempre disponibles los botones para navegar a la pantalla principal y a la pantalla de gestión de ejercicios para poder gestionar los ejercicios en todo momento. La única forma que tendrá un entrenador de ver la pantalla de creación de ejercicios será seleccionando dicha opción en la pantalla principal, desde el apartado de consulta y creación.



*Ilustración 20 Arquitectura de la aplicación BackEnd*

El servicio BackEnd está creado con el framework de Typescript llamado NestJs[18].

NestJs es un framework para crear aplicaciones del lado del servidor eficientes y escalables en NodeJs. NestJs ofrece una capa de abstracción sobre otros famosos frameworks como Express o Fastify pero al mismo tiempo expone la API de cada uno de estos módulos para ofrecer al desarrollador la libertad de usar la miríada de otras librerías de terceros que hay para los módulos ya mencionados.

Además de esto, para coordinar el desarrollo se ha usado git[19] y Github[20] para el control de versiones y para la sincronización del código entre los dos proyectos que se han realizado paralelamente. Además, como editor de código se ha usado Visual Studio Code[21], un potente editor de código de Microsoft que ofrece una gran cantidad de plugins para customizarlo.

## 7.3. Ejemplos de código

---

En esta sección se van a mostrar algunos fragmentos del código desarrollado para mostrar la manera en que se han implementado las distintas funcionalidades.

```
@Post()
@HttpCode(HttpStatus.CREATED)
async createWeek(@Body() newDay: DayLogDto) {
  try {
    newDay.exercises.forEach((exercise) => {
      //check that all values arrays are the same length as sets
      if (
        exercise.sets !== exercise.constraints.length ||
        exercise.sets !== exercise.real_weight.length ||
        exercise.sets !== exercise.reps.length ||
        exercise.sets !== exercise.real_perceived_effort.length
      ) {
        throw new HttpException("Invalid exercise", HttpStatus.BAD_REQUEST);
      }
      exercise.real_weight = exercise.real_weight.fill(0);
      exercise.real_perceived_effort = exercise.real_perceived_effort.fill(0);
    });
    const resp = await this.routineService.createDay(newDay);
    return resp;
  } catch (err) {
    throw new HttpException(err.message, HttpStatus.INTERNAL_SERVER_ERROR);
  }
}
```

*Ilustración 21 Código de creación de entrenamiento*

En la ilustración anterior se muestra el código del Backend que recibe las peticiones de creación de entrenamiento. Como se puede ver, la función recibe un DayLogDTO, una clase que contiene los campos del DayLog mostrados en el diagrama de clases. El siguiente paso es, por si la creación viene de una copia de la semana anterior, borrar todos aquellos campos que el atleta haya podido rellenar, dejando solo el esquema de entrenamiento para que así el atleta pueda volver a realizar el entrenamiento y actualizar sus resultados. Como último paso se llama al servicio de creación de entrenamientos para persistir el mismo en la base de datos.

```
const loginWithEmailAndPassword = async (email: string, password: string) => {
  try {
    await signInWithEmailAndPassword(auth, email, password);
  } catch (err: any) {
    console.error(err);
  }
};

const registerWithEmailAndPassword = async (name: string, email: string, password: string) => {
  try {
    const res = await createUserWithEmailAndPassword(auth, email, password);
    const user = res.user;
    await addDoc(collection(db, 'users'), {
```

```

        uid: user.uid,
        name,
        authProvider: 'local',
        email,
        userType: 'trainer'
    });
} catch (err: any) {
    console.error(err);
    alert(err.message);
}
};

```

*Ilustración 22 Código de inicio de sesión y registro*

El código anterior pertenece a la aplicación del FrontEnd. Como se ha explicado, para la gestión de usuarios se ha usado Firebase para así asegurar la seguridad total de los datos. La primera función “loginWithEmailAndPassword” recibe el email y la contraseña por parámetros, hace la llamada de login a Firebase y se espera a recibir una respuesta para definir el acceso del usuario.

La función “registerWithEmailAndPassword”, además del email y la contraseña recibe el nombre del usuario. Como la aplicación web es únicamente para entrenadores, esta función crea un objeto usuario con los parámetros que recibe y añade dos campos más, userType, que sirve para definir si el usuario es atleta o entrenador y authProvider, un campo requerido por Firebase y posteriormente hace la llamada al servicio para guardar los datos del usuario en nuestra base de datos.

```

@Injectable()
export class TrainerDataService {
    constructor(
        @InjectModel(ExerciseList.name)
        private exerciseListModel: Model<ExerciseListDocument>,
    ) {}

    async createNewExercise(trainerId: string, exerciseCat: string, exerciseName:
string) {
        return await this.exerciseListModel.updateOne(
            { trainerId },
            { $push: { [exerciseCat]: exerciseName } },
            { upsert: true },
        );
    }

    async getExerciseList(trainerId: string) {
        return await this.exerciseListModel.find({ trainerId });
    }

    async updateExercise(trainerId: string, oldCategory: string, newCategory: string,
oldName: string, newName: string) {
        return await this.exerciseListModel.updateOne(
            { trainerId },

```

```
    { $pull: { [oldCategory]: oldName }, $addToSet: { [newCategory]: newName } }  
  )  
  }  
}
```

*Ilustración 23 Servicio de gestión de ejercicios para entrenadores*

Para ilustrar el funcionamiento de la capa de servicios se muestra la Ilustración 20, correspondiente a la clase que contiene los métodos de consulta a la base de datos.

En primer lugar, la clase instancia la conexión a la base de datos a través del constructor, donde se recibe gracias al proceso de inyección de dependencias. Tras habilitar la conexión con la base de datos, la clase contiene tres métodos principales:

- `createNewExercise`, este método recibe la id del entrenador, la categoría del ejercicio que va a guardar y el nombre del ejercicio para posteriormente incluir el ejercicio en la lista de la categoría indicada.
- `getExerciseList`, que recibe únicamente la id del entrenador y recupera la lista completa de ejercicios que ha creado dicho entrenador.

`updateExercise`, en caso de que se quiera cambiar el nombre o categoría de un ejercicio, el método borra el registro anterior a través de los campos “oldCategory” y “oldName” y guarda un nuevo registro gracias a los parámetros “newCategory” y “newName”.

## 8. Implantación

---

La fase de implantación corresponde al último paso del ciclo productivo explicado en el punto cinco. Tras validar los objetivos del proyecto con el cliente y realizar el desarrollo, el proyecto se despliega o entrega al cliente para que este lo pueda utilizar, o bien en un entorno cerrado de pruebas donde se tenga un control del sistema o en un entorno abierto donde cualquier usuario pueda usar el sistema. En este caso, para evitar los costes asociados al despliegue público, también llamado de producción, la aplicación PowerLog se ha entregado a los clientes en un entorno de prueba, el cual a su vez permitía un rápido despliegue para arreglar errores o incluso ser capaz de reiniciar el sistema a mano en caso de haber algún fallo más grave.

Como se ha explicado anteriormente, el FrontEnd, desarrollado con ReactJs, estará alojado en Vercel. Vercel ofrece la posibilidad de tener una aplicación de manera gratuita con el “Hobby Tier”[22]. Dentro de esta plataforma contamos con la posibilidad de conectar la aplicación con nuestro sistema de control de versiones (Github en este caso) para crear despliegues automáticos de nuestra aplicación. Vercel nos ofrece una previsualización de la página por cada nueva versión que publiquemos en nuestro sistema, pero una nueva versión no siempre implica querer desplegar al momento y por tanto se decidió configurar el despliegue de Vercel de manera que únicamente se desplegasen nuevas versiones de la aplicación cuando el desarrollador lo desee. De esta forma se evita publicar funcionalidades aún sin probar y también se reducen los costes pues Vercel, al elegir el servicio gratuito, nos impone un máximo de carga en sus servidores que, en caso de ser sobrepasado supondría un coste para los desarrolladores.

Vercel también nos asigna un *dominio* para poder acceder a la página a través del navegador, sin embargo, se decidió hacer uso de un subdominio propio para evitar hacer uso de sistemas desconocidos y así tener un mayor control sobre el enrutamiento del tráfico. El dominio del FrontEnd es <https://powerlog.manueljishi.dev> y el dominio del BackEnd es <https://powerlogapi.manueljish.dev>. Gracias al uso de dominios propios se puede asumir el control del mismo y crear nuevos subdominios según lo requieran las circunstancias.



El BackEnd de la aplicación, desarrollado en NestJs está alojado en un Droplet de Digital Ocean. Un droplet es una máquina virtual con recursos limitados, pero también con un precio reducido[23]. Ya que la aplicación PowerLog no suponía una carga muy grande y la aplicación en sí era reducida en tamaño, se escogió un Droplet básico, con un coste de 6 euros al mes. Esto permitía tener un servidor disponible las 24 horas del día con la confianza de que DigitalOcean ofrecía un eficiente mantenimiento y un *uptime*[24] del 99.99%. Esto es un punto muy a favor dado que los atletas y entrenadores deben poder acceder a las aplicaciones en cualquier momento del día y una caída del sistema imposibilitaría el acceso a las mismas durante la duración del mismo.

En cuanto a la configuración del servidor, el proyecto está directamente conectado con el repositorio de Github para traer los nuevos cambios. Al ser esta una máquina virtual que únicamente ejecuta el BackEnd de la aplicación, esta se puede ejecutar directamente sobre la máquina, sin la necesidad de virtualizarla. El BackEnd funciona en el puerto 3000 del servidor y para mantener los estándares del protocolo HTTP (puerto 80 para HTTP y puerto 443 para HTTPS) y reducir el número de puertos que se exponen al tráfico externo, se ha utilizado *NGINX*, un servicio que actúa como *reverse proxy*, para redirigir el tráfico del puerto 443 al puerto 3000 y así hacer transparente esta diferencia en los puertos de la aplicación. Para el certificado de seguridad HTTPS, que sirve para encriptar las comunicaciones en la red y verificar la autenticidad del servidor, evitando así posibles ataques como los famosos “Man in the middle” o “Packet sniffing”, se ha utilizado el servicio de Let’s Encrypt[25] que ofrece certificados de manera automática y gratuita.

## 9. Producto desarrollado

---

En este apartado se mostrarán y analizarán parte por parte distintas capturas de la aplicación en funcionamiento, para demostrar las funcionalidades que se han conseguido implementar a lo largo del proyecto.

En primer lugar, las pantallas de Login y Registro se van a obviar puesto que son interfaces triviales con una funcionalidad muy básica. En la pantalla de Login se le muestran al usuario dos campos, uno de email y uno de contraseña, que ha de rellenar con sus credenciales para ser redirigido a la aplicación y en la pantalla de Registro, se le pide al usuario introducir un email, un nombre y la misma contraseña dos veces para verificar que se ha escrito correctamente.

Tras pasar la autenticación, el entrenador es automáticamente redirigido a la pantalla principal, una pantalla básica donde únicamente se le presenta una lista de atletas para seleccionar aquel con el que quiera trabajar, la posibilidad de cerrar sesión o de ir a la pantalla de gestión de ejercicios:



*Ilustración 24 Pantalla inicial de la versión del entrenador de PowerLog*

A continuación, siguiendo un flujo de trabajo normal para un entrenador, se selecciona un atleta para consultar sus entrenamientos o introducir nuevos datos:

**Resultados de Vicente**

mayo 2023

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

**22 may 2023**

Inicio de bloque  Fin de bloque

**Block DL Tempo 505**

Serie	Repeticiones	Intensidad	Kg	Intensidad Real
1	8	rpe: 6	250	6
2	6	rpe: 6	240	6

Comentarios: Ten cuidado con la espalda, no te hagas daño

**Sentadilla de competición**

Serie	Repeticiones	Intensidad	Kg	Intensidad Real
1	8	rpe: 5	150	5
2	6	rpe: 8	200	8
3	4	rpe: 7	180	7

Comentarios: Haz las sentadillas con precaución

**Comentarios del atleta**

La verdad que me lo he pasado muy bien, aunque el entrenamiento ha sido un poco mas duro de lo que esperaba

😊 😊 😊 😊 😊 😊

[Copiar datos de la semana pasada](#) [Nuevo Ejercicio](#) [Guardar datos](#)

*Ilustración 25 Vista de gestión de atleta con un entrenamiento ya creado*

Como se puede observar, cuando el entrenador selecciona el atleta se le presenta una vista con un calendario y la sección de detalle de entrenamiento. Por defecto aparece el día actual y se muestran los detalles del entrenamiento. Desde aquí el entrenador puede consultar los resultados del atleta. Entre estos datos se encuentran: el número de ejercicios realizados junto con un desglose de cada serie donde se pueden ver los kilogramos levantados por el atleta junto con la sensación de esfuerzo percibida por el atleta. En la parte inferior de la pantalla se añadió, por petición tanto de entrenadores como de atletas, una sección donde el atleta podía añadir comentarios sobre los entrenamientos junto con una valoración de las sensaciones generales. De esta manera se conseguiría una mejor comunicación entre ambas partes junto con una mejor planificación de cara a entrenamientos posteriores. Como se puede ver, el entrenador tiene a su disposición un botón de edición, en caso de haber algún dato incorrecto y un botón de borrado para eliminar cualquiera de los ejercicios.

En caso de no haber ningún entrenamiento ya creado por parte del entrenador, la sección de detalle se muestra vacía, únicamente con dos botones:

- Botón de copiar datos. Al seleccionar esta opción se copian los datos de la semana anterior eliminando aquellos introducidos por el atleta de forma que se presenta como un entrenamiento vacío esperando a ser completado. De esta manera el entrenador puede mantener la misma estructura de entrenamientos durante un tiempo de manera rápida y automática
- Botón de nuevo ejercicio. En caso de que se desee crear un nuevo entrenamiento manualmente

Ambas opciones se pueden usar conjuntamente en caso de que se desee crear un entrenamiento parecido al de la semana anterior, pero se desee modificar algún detalle manualmente.

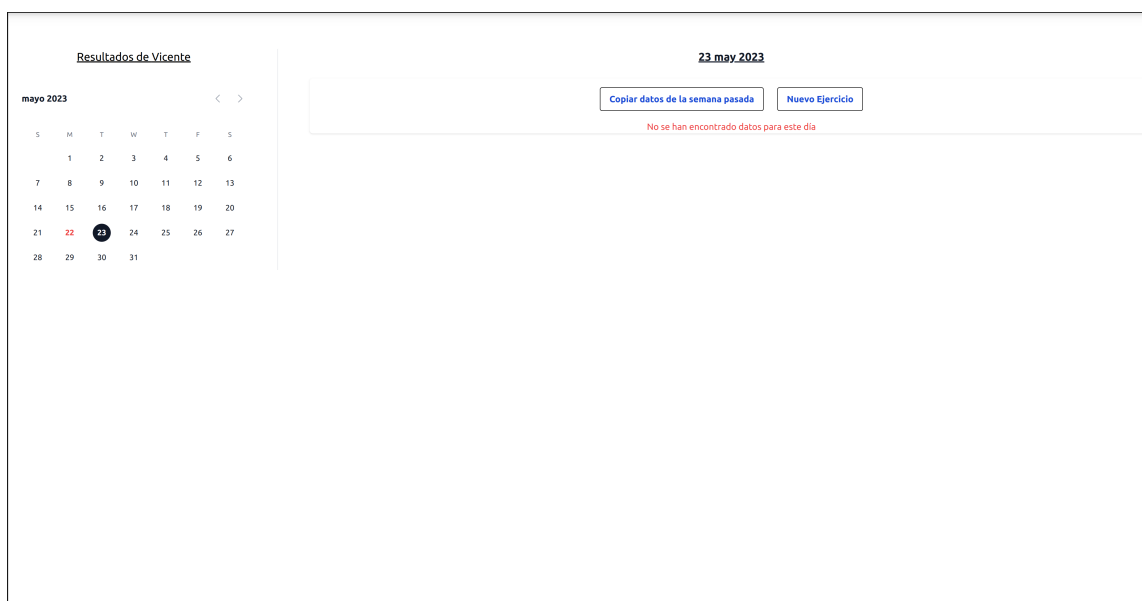


Ilustración 26 Vista del entrenador de un día sin datos

Al seleccionar la opción de nuevo ejercicio, al entrenador se le muestra una ventana donde podrá concretar todos los detalles del nuevo ejercicio.

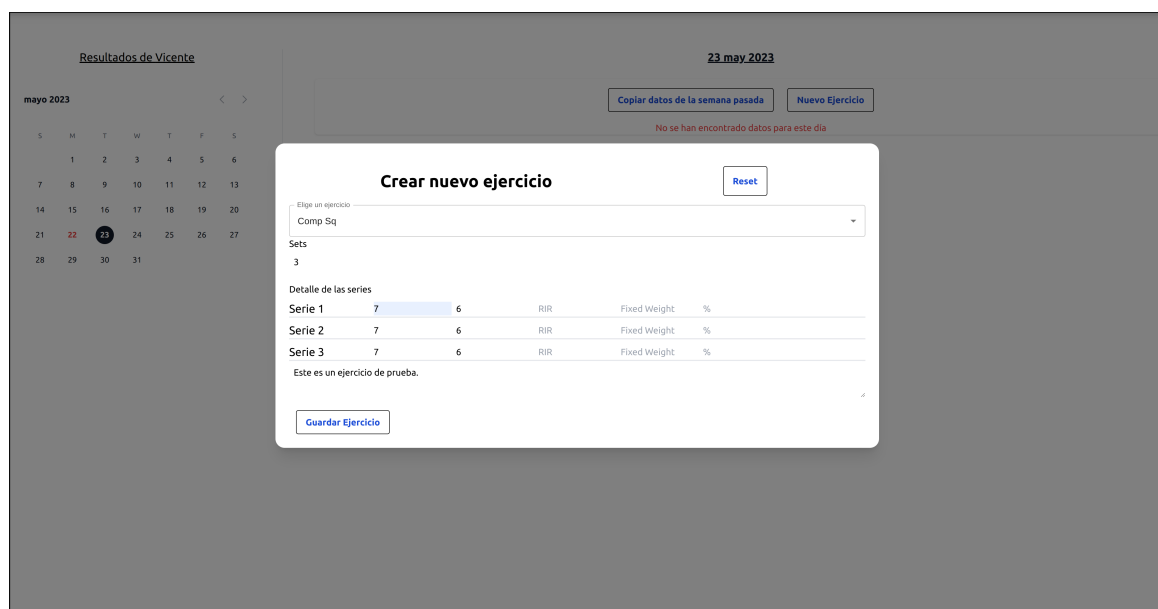
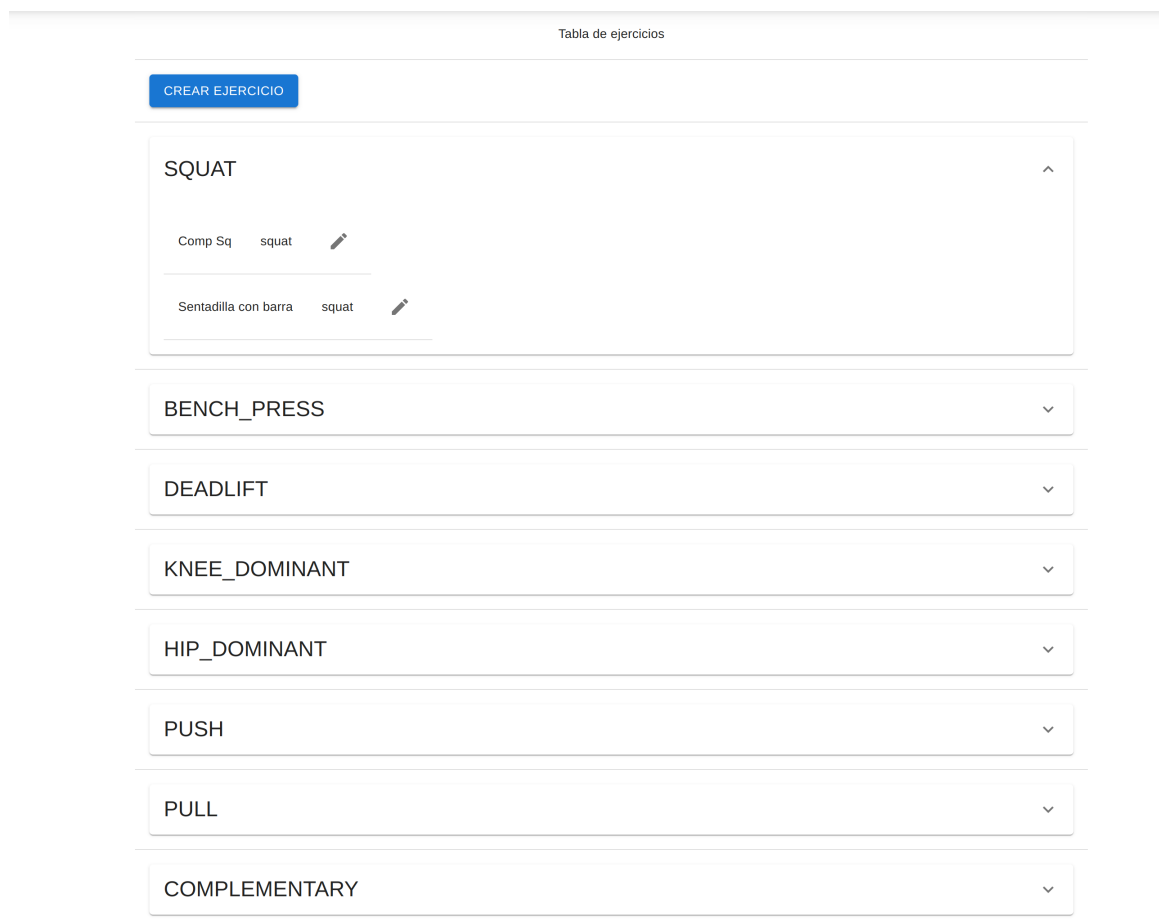


Ilustración 27 Vista de creación de un ejercicio

En esta pantalla al entrenador se le presenta una lista con todos los ejercicios que haya creado hasta el momento. Desde aquí puede seleccionar el número de series por ejercicio y especificar

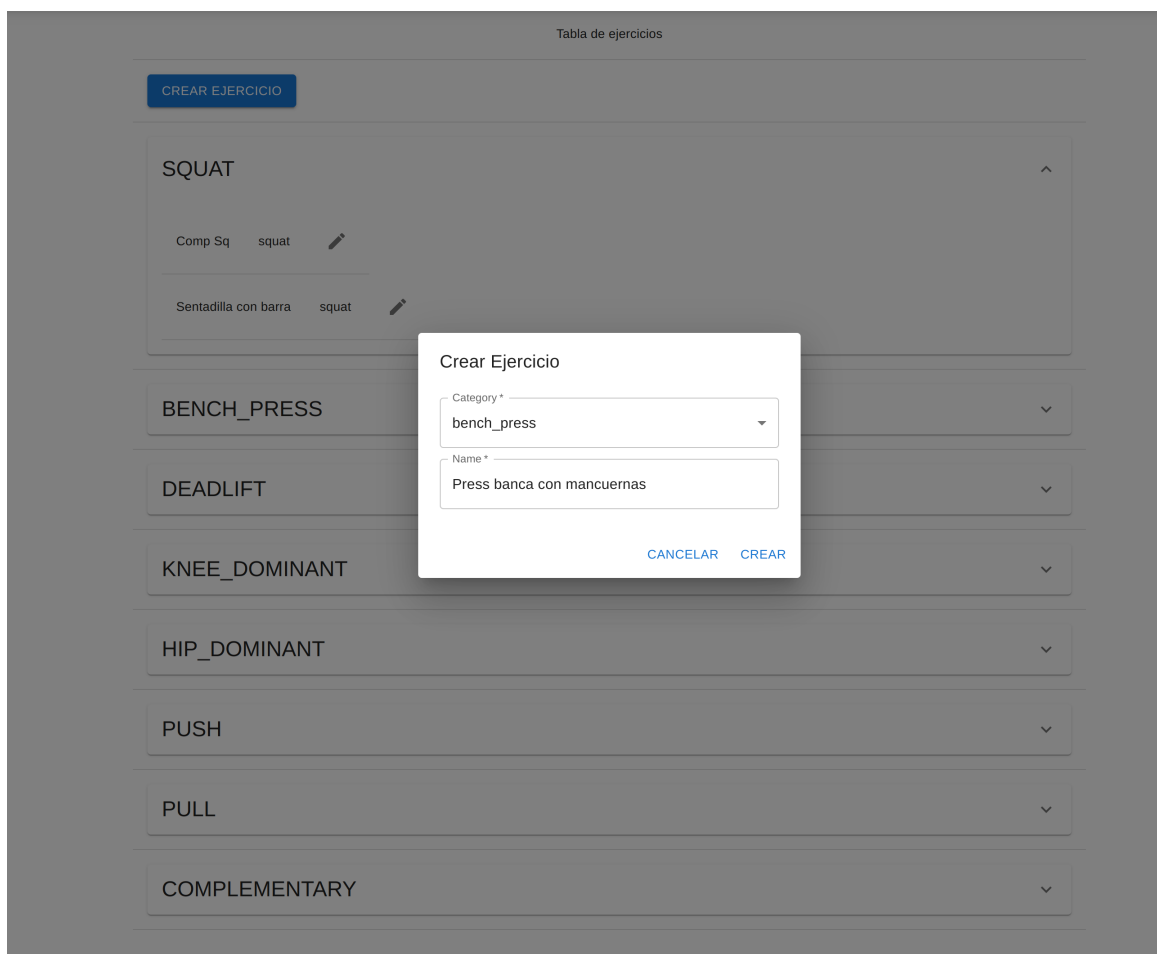
los valores de esfuerzo de cada una de las series junto con comentarios específicos para cada uno de los ejercicios.

Además de consultar y crear entrenamientos para los atletas, cada entrenador tiene una preferencia distinta por los tipos de ejercicios a realizar y por eso una lista global no serviría para todos. Por esta razón, el entrenador puede acceder al apartado de ejercicios donde podrá ver y crear nuevos ejercicios separados por categoría:



*Ilustración 28 Lista de ejercicios del entrenador*

En la figura anterior se muestra la pantalla de ejercicios, en ella aparece una lista de categorías que, al desplegar el apartado de interés permite consultar los ejercicios creados en esa categoría.



*Ilustración 29 Ventana de creación de ejercicio*

Dentro de la vista de ejercicios, el entrenador puede crear nuevas entradas. Para crear un nuevo ejercicio se muestra una ventana con dos campos, la categoría del ejercicio que se va a crear, siendo esta un desplegable donde seleccionar las categorías existentes y un campo de nombre que el entrenador debe rellenar con el nombre del ejercicio.

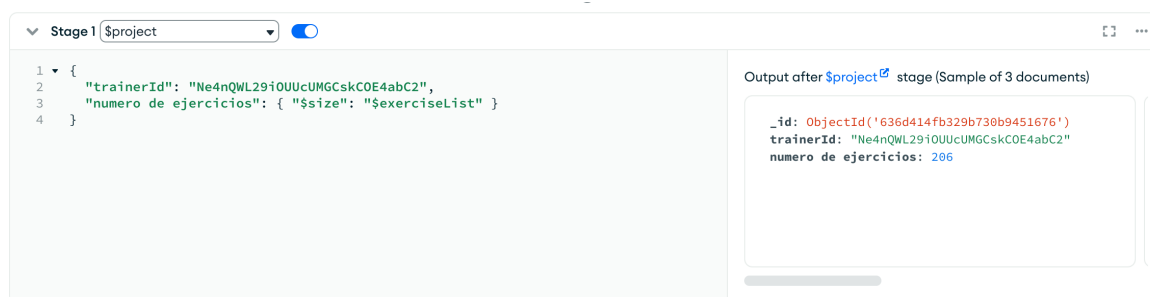
Como ya se ha explicado anteriormente, el proceso de desarrollo de la aplicación ha sido un proceso incremental manteniendo una comunicación constante con entrenadores y atletas para acordar las nuevas funcionalidades que se iban añadiendo a la aplicación. Gracias a este enfoque incremental la aplicación se ha podido ir probando desde la fase más temprana del desarrollo y por ende se ha conseguido un sistema completamente funcional que cumpliera su cometido.

## test.daylogs

STORAGE SIZE: 696KB LOGICAL DATA SIZE: 2.45MB TOTAL DOCUMENTS: 1002 INDEXES TOTAL SIZE: 60KB

*Ilustración 30 Base de datos de la aplicación, tabla de entrenamientos*

La figura anterior corresponde a la tabla de entrenamientos de la base de datos de MongoDB que se ha utilizado con los usuarios. Como se puede ver en el apartado de documentos, durante el proyecto se han registrado 1002 entrenamientos hasta la fecha. Cada entrenamiento corresponde a un atleta y día distinto.



*Ilustración 31 Base de datos de la aplicación, tabla de ejercicios*

En esta captura se muestra una consulta a Mongo para analizar la tabla de ejercicios de los entrenadores, como se puede apreciar en el lado derecho de la pantalla, uno de los entrenadores ha creado un total de 206 ejercicios distintos para añadir a los entrenamientos de sus atletas.

## 10. Conclusiones

---

El objetivo del proyecto siempre fue el de crear una aplicación que ayudase a los entrenadores de PowerLifting a gestionar los entrenamientos de sus atletas. Se concluyó que la forma más eficiente sería a través de una aplicación web ya que así se podría acceder a dicha aplicación desde cualquier dispositivo que contase con una conexión a internet y un navegador.

En el estado actual de la aplicación el entrenador puede realizar la mayoría de tareas que se especificaron al principio de la memoria tales como:

- Iniciar sesión para consultar sus datos y la lista de todos los atletas.
- Consultar, desde una misma página los resultados de cada uno de sus atletas, evitando tener que cambiar de sesión o tener que dar más de unos pocos clicks para poder acceder a la información desada.
- Crear, modificar y borrar nuevos entrenamientos según se considere necesario, además los entrenamientos son privados y únicamente pueden ser consultados por el entrenador y atleta en cuestión.
- Gestionar los ejercicios según las preferencias del entrenador, siendo cada lista de ejercicios independiente de las demás.
- Que cada uno de los ejercicios permitiese la flexibilidad suficiente como para que el entrenador expresase sus ideas de la manera más precisa posible, pero sin perder la estructuración y orden.

Sin embargo, este no ha resultado un proyecto difícil pues, aunque el Grado en Ingeniería Informática nos ayudó a sentar las bases de los sistemas de computación en aspectos como redes de ordenadores, gestión de proyectos, eficiencia algorítmica la realidad es que la aplicación de estas teorías es mucho más compleja en el mundo real que en un aula.

Para empezar, la comunicación con los entrenadores y atletas, durante la carrera se explica cómo afrontar una entrevista, cómo definir los casos de uso e incluso el proceso de elicitación de requisitos, pero en la realidad se convierte en un proceso más complejo pues es muy probable que los usuarios no tengan totalmente claro lo que buscan, dando descripciones abstractas de las funcionalidades. La comunicación también se convierte en un aspecto complicado pues muchas veces el tiempo es limitado y los desarrolladores deben conseguir la mayor cantidad de información posible en un tiempo muy reducido. Los requisitos son cambiantes e incluso muchas veces, al no entender la complejidad de ciertos aspectos, los



usuarios solicitan funcionalidades que con los recursos del proyecto eran inalcanzables en su totalidad

La etapa más complicada del proyecto fue sin duda la de desarrollo, en gran parte por la falta de experiencia desarrollando aplicaciones desde cero. Personalmente ya contaba con experiencia trabajando con ReactJs pero en este caso decidí usar Typescript para añadir una capa más de seguridad al código lo cual hacía que fuese ligeramente más complejo y que todas las clases debían estar especificadas y diseñadas de antemano. Además de esto, la aplicación requería estar bien estructurada de cara a ser capaces de mantenerla, cambiar funcionalidades o añadir nuevas. El código debía de estar estructurado de una forma que permitiese aplicar los cambios rápidamente, sin influir a la funcionalidad del resto de la aplicación.

En cuanto al aspecto de servidores, en el trabajo del BackEnd tenía más experiencia, tanto por la carrera como por proyectos personales. Durante varias asignaturas ya se había enseñado a trabajar con el protocolo HTTP y en varios proyectos de la carrera usamos NestJs para crear el servicio del BackEnd con lo cual aspectos como crear las funcionalidades básicas, comunicar las peticiones con las bases de datos, generar una organización de ficheros sostenible eran menos pesadas que en el FrontEnd. Sin embargo hubo otros aspectos que tuve que aprender a implementar en NestJs como por ejemplo la validación de los datos de las peticiones, la propagación y manejo de errores etc.

El aspecto más interesante de este proyecto fue sin duda el de la creación y mantenimiento de los servicios. Durante la carrera rara vez se explicó como desplegar un proyecto de este estilo, es decir, cómo montar un servicio web y configurar los despliegues automáticos o crear distintos entornos de desarrollo. Otro aspecto interesante del despliegue es que los mencionados Droplets de DigitalOcean son máquinas virtuales, es decir, ordenadores que tienes que configurar desde cero para ajustar al caso de uso. Esto no se ha estudiado durante la carrera pero es realmente interesante aprender como abrir o cerrar puertos de red desde el propio sistema operativo, configurar NGINX para transferir las peticiones HTTP entre los distintos procesos del ordenador, crear y configurar certificados HTTPS, configurar los registros DNS para poder acceder a los servicios a través del dominio que queramos etc.

Personalmente creo que el resultado del proyecto ha sido muy positivo, la aplicación creada ha dado muy buenos resultados, tal y como se ha mostrado en el punto anterior, con la cantidad de entrenamientos que se han creado. También me ha ayudado en gran cantidad a descubrir cuales son los aspectos del proceso de desarrollo que más disfruto pues mientras que con la aplicación web el trabajo resultaba realmente arduo y agotador, con la parte de mantenimiento de sistemas, despliegues automáticos, solución de problemas y programación

BackEnd me parecían realmente entretenidos y me mantenían buscando más información sobre cómo optimizar los procesos, reducir los fallos etc.

En general la experiencia de desarrollar una aplicación propia desde cero resulta muy gratificante a la vez que en muchos casos se puede volver muy estresante, pero como aspecto general ayuda a mejorar las habilidades como programador mucho más de lo que cualquier curso o asignatura podrían hacer porque te expone a escenarios propios, donde la documentación puede ser escasa y ha de ser el programador quien aprenda los fundamentos de lo que está haciendo para resolver el problema.

## 11. Trabajos futuros

---

Si bien la aplicación desarrollada cumple con su cometido pues se ha llegado a usar con atletas y entrenadores reales, no es más que una primera versión de lo que podría ser tanto por funcionalidades que han solicitado los propios usuarios como por optimizaciones de código y arquitectura que se podrían implementar por parte de los desarrolladores. Algunas de estas funcionalidades se han tratado de desarrollar, pero se ha acabado viendo que no era factible la implantación con los recursos disponibles tanto de personal como de tiempo y económicos. Entre las funcionalidades que los propios usuarios solicitaron se encuentran las siguientes:

- Implantación de un componente social en la aplicación, donde los usuarios pudiesen agregar amigos con los que compartir los entrenamientos que habían realizado e incluso compartir fotos, comentarios, dar me gusta a los entrenamientos de otros atletas etc.
- La posibilidad de que un entrenador, además de indicar el ejercicio con todos los parámetros, pudiese subir fotos o vídeos explicando la técnica del ejercicio.
- La posibilidad de que un atleta pudiese, además de rellenar los datos de un entrenamiento, subir vídeos de los ejercicios para que el entrenador pudiese evaluarlos tanto a nivel de técnica como de esfuerzo.

Ambas funcionalidades relacionadas con los vídeos se trataron de implementar en una fase temprana del proyecto, pero se vio que la máquina virtual que se había alquilado para alojar el BackEnd no era capaz de procesar los vídeos que los SmartPhones modernos graban (60FPS 1080p) y por tanto resultaría en una sobrecarga del servidor lo cual podría derivar en una caída del sistema. Además, los vídeos requerían alquilar un servidor de almacenamiento externo pues el servidor del BackEnd únicamente tenía 5GB de almacenamiento e iba a alcanzar el límite de capacidad con mucha rapidez.

Otra funcionalidad que el equipo de desarrollo planteó diseñar fue la posibilidad de generar gráficos con los datos del atleta. Esta funcionalidad también se llegó a desarrollar, pero generaba mucha carga en el servidor y por tanto se decidió retirar pues no era crucial para el desarrollo del sistema.

En cuanto a mejoras no funcionales, se planteó separar la funcionalidad de separar gráficos en un microservicio independiente el cual fuese llamado por el BackEnd.

## 12. Bibliografía

---

- 1 Heavy. A free workout tracker & planner for iOS and Android. <https://www.hevyapp.com/about-us/> [Última consulta, marzo de 2023]
- 2 RTS Home. <https://www.reactivetrainingsystems.com/Home/Main> [Última consulta, marzo de 2023]
- 3 Alejandro Aldás Alarcón. 2013. Figura del proceso incremental. [https://www.researchgate.net/figure/Figura-10-Modelo-de-proceso-incremental-Fuente\\_fig6\\_326571456](https://www.researchgate.net/figure/Figura-10-Modelo-de-proceso-incremental-Fuente_fig6_326571456) [Última consulta, mayo de 2023]
- 4 Docker: Accelerated, Containerized Application Development <https://www.docker.com/> [Última consulta, mayo de 2023]
- 5 MongoDB Atlas Database. <https://www.mongodb.com/cloud/atlas/> [Última consulta, mayo de 2023]
- 6 DigitalOcean the developer cloud. <https://www.digitalocean.com> [Última consulta, mayo de 2023]
- 7 Advanced Load Balancer, Web Server, & Reverse Proxy-NGINX. <https://www.nginx.com/> [Última consulta, abril de 2023]
- 8 Firebase. <https://firebase.google.com/> [Última consulta, marzo de 2023]
- 9 Terms of service for Firebase Services. <https://firebase.google.com/terms/> [Última consulta, abril de 2023]
- 10 William Jobe. (2014). Native Apps Vs. Mobile Web Apps. International Journal of Interactive Mobile Technologies (iJIM) 7(4):pp. 7. DOI: 10.3991/ijim.v7i4.3226
- 11 Vercel: Develop. Preview. Ship. For the best frontend teams <https://vercel.com/> [Última consulta, junio de 2023]
- 12 React. <https://react.dev/> [Última consulta, junio de 2023]
- 13 Next.js by Vercel - The React Framework. <https://nextjs.org/> [Última consulta, marzo de 2023]
- 14 Vue.js - The Progressive JavaScript Framework | Vue.js. <https://vuejs.org/> [Última consulta, marzo de 2023]
- 15 Svelte. Cybernetically enhanced apps. <https://svelte.dev/> [Última consulta, marzo de 2023]
- 16 Flutter - Build apps for any screen. <https://flutter.dev/> [Última consulta, febrero de 2023]

- 17 Dart programming language. <https://dart.dev/> [Última consulta, febrero de 2023]
- 18 NestJs – A progressive Node.Js framework. <https://nestjs.com/> [Última consulta, junio de 2023]
- 19 Git. <https://git-scm.com/> [Última consulta, febrero de 2023]
- 20 Github. <https://github.com/> [Última consulta, junio de 2023]
- 21 Visual Studio Code - Code Editing. Redefined <https://code.visualstudio.com/> [Última consulta, marzo de 2023]
- 22 Pricing – Vercel. <https://vercel.com/pricing> [Última consulta, abril de 2023]
- 23 Pricing Overview | DigitalOcean. <https://www.digitalocean.com/pricing/droplets> [Última consulta, febrero de 2023]
- 24 High Availability Web Hosting (99.99% Uptime) | DigitalOcean <https://www.digitalocean.com/products/networking> [Última consulta, febrero de 2023]
- 25 A nonprofit Certificate Authority providing TLS certificates to 300 million websites. <https://letsencrypt.org/> [Última consulta, marzo de 2023]

## ANEXO

### OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

<b>Objetivos de Desarrollo Sostenibles</b>	<b>Alto</b>	<b>Medio</b>	<b>Bajo</b>	<b>No Procede</b>
ODS 1. <b>Fin de la pobreza.</b>				<b>X</b>
ODS 2. <b>Hambre cero.</b>			<b>X</b>	
ODS 3. <b>Salud y bienestar.</b>	<b>X</b>			
ODS 4. <b>Educación de calidad.</b>			<b>X</b>	
ODS 5. <b>Igualdad de género.</b>		<b>X</b>		
ODS 6. <b>Agua limpia y saneamiento.</b>				<b>X</b>
ODS 7. <b>Energía asequible y no contaminante.</b>				<b>X</b>
ODS 8. <b>Trabajo decente y crecimiento económico.</b>				<b>X</b>
ODS 9. <b>Industria, innovación e infraestructuras.</b>				<b>X</b>
ODS 10. <b>Reducción de las desigualdades.</b>				<b>X</b>
ODS 11. <b>Ciudades y comunidades sostenibles.</b>				<b>X</b>
ODS 12. <b>Producción y consumo responsables.</b>		<b>X</b>		
ODS 13. <b>Acción por el clima.</b>		<b>X</b>		
ODS 14. <b>Vida submarina.</b>				<b>X</b>
ODS 15. <b>Vida de ecosistemas terrestres.</b>				<b>X</b>
ODS 16. <b>Paz, justicia e instituciones sólidas.</b>				<b>X</b>
ODS 17. <b>Alianzas para lograr objetivos.</b>				<b>X</b>

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

En cuanto a la relación de este trabajo con los Objetivos de Desarrollo Sostenible, salta a la vista cuáles son aquellos que más relación tienen con el trabajo reflejado en la memoria. Se ha creado una aplicación de entrenamientos para fomentar y optimizar los entrenamientos de los atletas. Un mejor y más riguroso entrenamiento tiene muchos beneficios entre los cuales cabría destacar una mejor salud cardiovascular, reduciendo así el riesgo a sufrir enfermedades como diabetes o problemas cardíacos. El entrenamiento revisado por un profesional también es mucho más seguro ya que nos garantiza un correcto descanso y recuperación del cuerpo. Además, la práctica habitual de ejercicio mejora el humor y ayuda al descanso, aportando considerablemente al bienestar del atleta. Pero el entrenamiento no solo aporta en los aspectos de salud ya que durante años los entrenamientos de gimnasio se han considerado un ejercicio únicamente para hombres pero gracias a PowerLog esperamos acercar la facilidad y ganas a todas aquellas mujeres que deseen practicar este maravilloso deporte, donde todo el mundo, da igual sus características, tiene un lugar. Además, el PowerLifting no solo es entrenar pesas en un gimnasio, una práctica muy habitual en este deporte es el cardio de baja intensidad como por ejemplo andar o dar un paseo en bicicleta. Los atletas se ponen objetivos diarios de pasos o de actividad lo cual les fuerza a desplazarse a pie, en bicicleta u otros medios que requieran potencia humana para poder así quemar esas calorías extra además de evitar medios de transporte privados como podría ser el coche. Con respecto a los otros ODS indicados, aunque menos relevantes, es cierto que se puede considerar que mantienen una relación con la aplicación como por ejemplo el objetivo de reducir el hambre. Una correcta alimentación garantiza que cada persona ingiere únicamente los alimentos indicados para su situación, (generalmente recomendado por un nutricionista), esto reduce el desperdicio de los atletas y fomenta el consumo responsable de bienes, permitiendo que más personas puedan beneficiarse de ese excedente, por lo tanto, fomentar la práctica del deporte e incluir a más atletas derivará en un consumo de recursos responsable. En cuanto al objetivo de la educación, la aplicación PowerLog ayudaría a los atletas a adquirir un mejor conocimiento sobre su propio cuerpo y sus capacidades. Gracias al consejo de los entrenadores se puede llevar al cuerpo cerca de sus propios límites sin prácticamente riesgo, esto genera una mayor confianza en los atletas a la hora de realizar actividades por su cuenta pues permite conocer el límite de cada uno dando margen al propio atleta a gestionar su intensidad sabiendo en qué momento el cuerpo le está advirtiendo de que se está sobrepasando. En cuanto al resto de objetivos de desarrollo sostenible, si bien se podría considerar que el fomento del deporte podría tener un efecto beneficioso sobre ellos, tal vez sería una influencia ínfima, no necesaria de mención o por el contrario, que la creación de más atletas y de una población más activa tuviese un efecto negativo como por ejemplo, se podría argumentar, que un atleta al ser más activo consume

más alimentos que una persona sedentaria y por ende una población más activa generaría un mayor consumo de los recursos producidos. Este punto de vista contrapuesto a lo explicado anteriormente con respecto al consumo responsable puede parecer totalmente válido si bien también se podría contraargumentar que una población más sana reduciría el uso de hospitales y por consiguiente el consumo de recursos de los mismos, quedando prácticamente en el punto inicial.