

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Planificador de entrenamientos de CrossFit en React JS



Trabajo Fin de Grado

Autor: Iván Arteta Triguero

Director: Francisco Javier Falcone Lanas

Pamplona, 10 de junio de 2022

Índice

1	Introducción.....	5
1.1	Resumen.....	5
1.2	Abstract	5
1.3	Introducción	5
1.4	Motivación	7
1.5	Objetivos	7
2	Estado del arte en el mundo del crossfit	9
2.1	¿Qué es el CrossFit?	9
2.2	Mercado del Crossfit	12
2.3	Análisis de mercado en aplicaciones de gestión de CrossFit	15
3	Aspectos técnicos del desarrollo.....	22
3.1	Metodología de trabajo	22
3.1.1	Scrum.....	22
3.1.2	Tablero Ágil Jira	24
3.2	Control de versiones	26
3.3	Lenguajes de programación	29
3.3.1	JavaScript.....	30
3.3.2	React.....	30
3.3.3	TypeScript.....	31
3.3.4	Otros lenguajes.....	31
3.4	Herramientas de desarrollo	32
3.5	Despliegue de aplicación.....	33
4	Implementación y desarrollo del módulo	34
4.1	Instalación y preparación del entorno	34
4.2	Planteamiento del desarrollo.....	35
4.3	Desarrollo del módulo.....	36
4.3.1	Estructura básica	36
4.3.2	Submódulo de programas	38
4.3.3	Submódulos restantes.....	41
4.3.4	Modals y pop-ups de metronic	44
4.3.5	Componentes reutilizables para todo el panel de administración	44
4.4	Despliegue.....	47

5	Conclusiones	49
6	Trabajos futuros.....	50
7	Bibliografía	51

Índice de figuras

Figura 1. Pirámide de CrossFit (Glassman, 2002) [16].....	11
Figura 2. Ejemplo de ropa deportiva Reebok [31].....	13
Figura 3. Prueba de CrossFit Games. (CrossFit Games, 2022) [10]	14
Figura 4. Esquema de las fases de CrossFit Games en la modalidad individual (CrossFit Games, 2022) [10]	15
Figura 5. Ejemplo de ejercicio. (Queiron CrossFit, 2022) [28].....	16
Figura 6. Aplicación Resawod (Nubapp S.L, 2022) [25]	17
Figura 7. Aplicación Wodzilla (StudioIP, LLC, 2022) [36]	18
Figura 8 . Aplicación Velites Timer App. (Velites, 2022) [38]	19
Figura 9 . Imagen de ejemplo de tablero ágil de Jira Software. [4].....	25
Figura 10 . Flujo de trabajo de ramas de GIT (Aula de software libre, 2021) [6]	29
Figura 11 Estructura básica del planificador.	38
Figura 12. Vista principal del módulo	39
Figura 13 . Vista de detalle de programas.....	39
Figura 14 . Vista de detalles de programa con el formulario lateral abierto	40
Figura 15. Ejemplo de vista de detalles de un ejercicio	42
Figura 16. Formulario de edición de ejercicios	43
Figura 17 . Vista de editar métricas con su componente adicional	43
Figura 18 . Ejemplo de un modal. Asignar programa a actividad o usuario.....	44
Figura 19 . Código del componente Card.....	45
Figura 20 . Código del componente Botón.....	46
Figura 21. Código de componente Input.....	46
Figura 22 Código de componente TextInputGroup.....	47

Índice de tablas

Tabla 1. Tabla comparativa de funcionalidades por cada aplicación.....	20
Tabla 2. Tabla resumen de los comandos más importantes de GIT.....	28

1 Introducción

1.1 Resumen

La empresa Nubapp Applications S.L, encargada de la realización y mantenimiento de software de centros deportivos tiene planeado introducir un nuevo módulo para su panel de administración. Este módulo consiste en un planificador de entrenamientos, actividades y más funcionalidades de centros de CrossFit. Para ello se utilizarán nuevas tecnologías de programación con nuevos lenguajes y una metodología ágil para la planificación de su desarrollo.

1.2 Abstract

The company Nubapp Applications S.L, responsible for the creation and maintenance of software for sports centers, plans to introduce a new module for its administration panel. This module consists of a training planner, activities and more features of CrossFit centers. To this end, new programming technologies will be used with new languages and an agile methodology for planning its development.

1.3 Introducción

El deporte se está volviendo muy importante dentro de la sociedad en los últimos años, ayudado por diversas campañas publicitarias de grandes marcas en las que se posicionan en contra del sedentarismo y nos animan a mover nuestro cuerpo para evitar la obesidad.

Unido a ello la cultura del fitness está en pleno auge, el canon de belleza actual ha cogido un rumbo fijo en el que los cuerpos humanos tienen que ser perfectos para ser socialmente aceptados.

Uno de los deportes con mayor crecimiento en los últimos años es el Crossfit, un deporte que consiste en hacer diversos ejercicios de fuerza intercalando con ejercicios de resistencia en un corto periodo de tiempo, para así quemar las máximas calorías posibles en el menor tiempo posible.

El CrossFit, es un deporte con mucha demanda en países europeos, por ello, la creación de un buen planificador de entrenamientos de CrossFit, que tenga diversas funcionalidades como el poder crear entrenamientos, asignar esos entrenamientos a un plan semanal, poder relacionar el material necesario para la realización de cada ejercicio con una tienda por si hiciese falta comprarlo, y también la posibilidad de crear rutinas alimentarias y menús saludables para las personas que requieran este servicio.

Para la realización de este planificador, centrándonos en la parte del frontend, se va a utilizar el lenguaje React JS con Typescript para poder tipar los componentes, siendo React un framework de JavaScript que permite la realización de componentes reutilizables para una mejor programación y una mayor velocidad de ejecución ya que el estado de cada componente es almacenado en el DOM del navegador, y si ese estado no se actualiza, no se vuelve a renderizar el componente. Como estilo para este frontend, se va a utilizar una librería de estilos css llamada Metronic, la licencia de esta librería será aportada por la empresa Nubapp S.L. Otras herramientas que se van a utilizar serán, PhpStorm como IDE de desarrollo, Docker como tecnología de organización en contenedores y GIT como sistema de control de versiones.

Finalmente, el frontend se conectará con una base de datos relacional SQL y un backend que será desarrollado en php por parte de la empresa dueña del producto.

1.4 Motivación

Estando el crossfit en lo más alto de popularidad de los deportes en los últimos años, y mirando el posible negocio internacional en países como Francia en el que este deporte es uno de los más populares, es una gran oportunidad de innovar en el sector y sacar al mercado un producto único.

Por ello, la empresa Nubapp S.L, empresa que se dedica a la creación de softwares de gestión para centros deportivos, ha visto una oportunidad de negocio en el sector del crossfit, y por ello ha decidido sacar un producto específico para este deporte y actualizar el software que tenían dedicado al crossfit.

Finalmente, se espera sacar al mercado un producto único e inigualable, con las diversas funcionalidades mencionadas anteriormente, que se expanda en el mercado internacional y tenga buena cabida en países en el que el crossfit sea un deporte mayoritario.

1.5 Objetivos

El objetivo final de este trabajo consiste en la creación del frontend para el planificador especializado para centros deportivos de crossfit. Para ello, una vez planteado el objetivo principal, se descompone el trabajo en objetivos más simples para dar forma final al proyecto y de esta forma dividir la complejidad del objetivo general.

El primer paso será aprender a utilizar correctamente las herramientas tecnológicas que se van a utilizar. Es decir, conseguir un cierto conocimiento en: el lenguaje de programación React Js con Typescript, aprender todas las ventajas de desarrollo que nos proporciona el IDE PhpStorm, aprender a gestionar las versiones del

proyecto con GIT, entendiendo los ciclos de vida de GIT y sus comandos, y por último mirando ejemplos de estilos de nuestro tema a utilizar, Metronic.

Seguido a esto, se va a realizar la estructura básica que van a compartir todas las vistas del planificador, con el fin de familiarizarse con el lenguaje de programación. Esta estructura consistirá de un sidebar lateral que contendrá el acceso al planificador y a sus diferentes submódulos, un topbar que contendrá información sobre el centro deportivo e información del usuario que está utilizando el producto, y por último una toolbar que nos proporcionará información de en qué submódulo nos encontramos.

Para la realización del proyecto, se ha dividido el planificador en diferentes submódulos para poder centrarse en la realización de cada uno por separado. En cada submódulo habrá diferentes vistas a realizar, en la que se insertarán componentes de React que se construirán lo más reutilizables posible para poder facilitar la realización de las posteriores vistas.

Por último, tras la realización de los diferentes submódulos y sus correspondientes vistas, se unirán todos los submódulos para tener el producto frontend final y así poder conectarlo al backend y tener el planificador totalmente funcional.

2 Estado del arte en el mundo del crossfit

2.1 ¿Qué es el CrossFit?

En la sociedad actual, la actividad física se ha convertido en una prioridad. Tal como nos asegura la Organización Mundial de la Salud (OMS, 2020), realizar una actividad física con regularidad nos produce grandes beneficios para la salud. En personas adultas, por ejemplo, la realización de actividad física puede reducir la mortalidad, mejorar la salud mental y la calidad del sueño de la persona que realiza el ejercicio con regularidad. Por otro lado, la OMS (2020) nos advierte que, en los últimos años, ligado a esta tendencia de actividad física, también hay una predisposición a los modos de vida cada vez son más sedentarios, debido al uso de transportes motorizados o trabajos con menos carga física, teniendo efectos muy negativos sobre nuestra salud. Con esto se indica que la realización de una actividad física es muy importante para tener una buena salud y evitar mayores problemas al tener una edad avanzada.

En consecuencia, en los últimos años han surgido diferentes e innovadoras modalidades de entrenamiento, todas destinadas a mantener o mejorar el estado de forma físico. Algunas de las modalidades que han ido surgiendo en los últimos años son el ciclismo, yoga, entrenamientos personales relacionados con el fitness, actividades al aire libre, etc. Con todas estas nuevas tendencias, el CrossFit sigue siendo una de las tendencias de entrenamiento que más éxito tiene.

Existen distintas definiciones en relación al CrossFit explicadas a continuación. Según Ormazábal y López (2018), el CrossFit es un deporte que contiene una variedad muy amplia de ejercicios, la mayoría de ellos funcionales, que se ejecutan con una intensidad elevada y en un corto periodo de tiempo. Añadido a esto, Beck et al. (2016) indican que con la realización de estos ejercicios se consigue un aumento de la capacidad física y psíquica, uniéndose así al desarrollo de la capacidad cardiopulmonar y vascular que van ligadas a la realización de ejercicios físicos. Asimismo, se considera al CrossFit

como un entrenamiento que combina ejercicios de sobrecarga muscular para el desarrollo de fuerza y ejercicios aeróbicos en una misma sesión, con el objetivo de combinar las dos modalidades y tener un entrenamiento completo y totalmente funcional, mejorando así la capacidad cardiovascular y la fuerza física de la persona que realiza el entrenamiento (Wilson, Marin, Rhea, Wilson, Loenneke y Anderson, 2012). En palabras de CrossFit, Inc. (citada por Salvatierra, 2014) la fórmula de CrossFit es: *“Es el deporte del fitness y es caracterizado por ser un sistema de acondicionamiento físico basado en movimientos funcionales, creatividad y variedad de ejercicios, alta intensidad y corta duración”* (Salvatierra, 2014, p.8). Para la empresa CrossFit, LLC, el Crossfit es: *“Un estilo de vida caracterizado por ejercicio seguro y efectivo y una nutrición sana.”* (CrossFit, LLC., 2022).

Aunque el CrossFit es conocido como un deporte, en realidad es una marca registrada. El CrossFit, con origen en Estados Unidos, nace en el año 1990 de la mano de Greg Glassmann, quien desarrolla y diseña esta nueva tendencia deportiva y sus programas de entrenamiento. En el año 2000, lo registra como marca propia, teniendo así todos los derechos sobre estas técnicas de entrenamiento diseñadas. Desde el año 2005 aumenta de manera muy continua y exponencial su popularidad (Beck y Tang, 2016).

El CrossFit se puede utilizar para diversos objetivos finales como perder peso, mejorar el estado de salud, o mejorar el rendimiento en un ámbito deportivo. Desde sus orígenes, el objetivo del CrossFit ha sido generar un fitness amplio, general e inclusivo, buscando preparar a todas las personas que lo practiquen independientemente de su estado físico. CrossFit no es un programa de fitness especializado, sino un intento de optimizar la competencia física en la resistencia cardiovascular y respiratoria, fuerza, flexibilidad, potencia, velocidad, fortaleza, coordinación, agilidad, equilibrio y precisión. (Salvatierra, 2014, p.8).

El CrossFit se basa en entrenamientos con movimientos funcionales que tienen como objetivo ser beneficiosos para la salud de la persona que está realizándolo, independientemente del objetivo final del deportista. Son movimientos naturales, efectivos y eficientes. Pero lo más destacable de los movimientos funcionales es la

capacidad de mover grandes cargas en largas distancias y hacerlo de forma rápida. (Frías et al.,2020).

El CrossFit no se centra solamente en el ámbito deportivo, tal y como se ha indicado antes, el CrossFit se define también como un estilo de vida, por lo tanto, se tiene muy en cuenta la nutrición y el acondicionamiento metabólico. Según nos indica la clínica Miguel Medina (2019), una alimentación correcta y variada conlleva a que el rendimiento del deportista sea óptimo, a su vez, llevar a cabo una buena nutrición ayuda al deportista a recuperarse más rápido después de haber realizado el esfuerzo físico durante el entrenamiento.

Tal y como se aprecia en la *figura 1*, el deporte y el levantamiento de peso es lo menos importante para tener un estilo de vida saludable, dando más importancia a la nutrición, que es fundamental y al acondicionamiento del cuerpo para que el tiempo de realización de ejercicio sea de provecho.

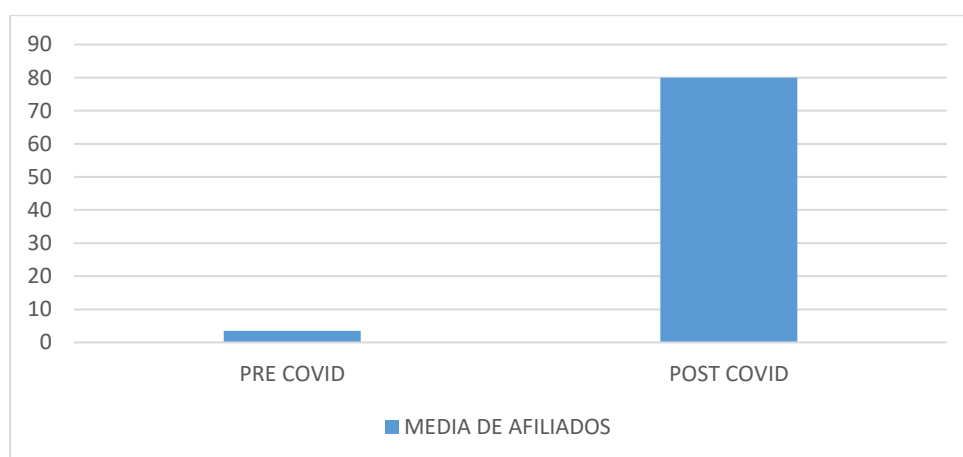


Figura 1. Pirámide de CrossFit (Glassman, 2002) [16]

2.2 Mercado del Crossfit

Según Nat Díez, country manager de CrossFit en España, a nivel económico, la afiliación a la marca CrossFit tiene un coste de 3000 dólares al año, unos 2680 euros anuales, por ello en España están creciendo los centros afiliados a la marca CrossFit.

Díez, nos comenta también que España es uno de los países que más está creciendo a nivel de afiliados. Históricamente, el crecimiento anual medio de España estaba en torno a 100 afiliados anuales, en ésta época post covid, la afiliación ha aumentado considerablemente, llegando a tener 100 afiliados mensuales en los meses de mayor crecimiento, cuando en épocas anteriores al covid se estaba creciendo en una media de 3 o 4 afiliados mensuales.



Gráfica comparativa de la media de afiliados en época anterior al covid y posterior. Fuente: elaboración propia. Datos: [2]

Por otro lado, en un ámbito más global, una de las marcas referencia en el mundo del Crossfit es Reebok. La marca estadounidense de zapatillas y ropa deportiva es uno de los mayores exponentes de este deporte a nivel mundial.

Como nos indica Wikipedia, en 2004 la marca comercial Adidas adquirió Reebok como una subsidiaria. Uniendo así dos de las compañías más grandes de material deportivo, pero manteniendo las marcas por separado, lo que produjo un aumento del patrimonio y de las posibilidades comerciales de Reebok. En 2010, Reebok anunció una asociación comercial con CrossFit, en el que se incluye un patrocinio de los CrossFit

Games, apertura de estudios de CrossFit y la introducción al mercado de una línea de calzado y vestimenta específicas para este deporte. Tras este acuerdo, Reebok empezó a perder contratos comerciales para la fabricación de vestimenta deportiva en ligas deportivas nacionales o para clubes privados, por ello, decidió orientar la marca y enfocarla principalmente hacia el fitness, por ello, Reebok es la marca referencia en el mercado de ropa y material deportivo para la realización del CrossFit.



Figura 2. Ejemplo de ropa deportiva Reebok [31]

Reebok introdujo su patrocinio en los CrossFit Games. Los CrossFit Games es la competición organizada por la empresa CrossFit en el que se corona al ser humano “*más apto de la tierra*” según recoge la propia página de los CrossFit Games. En las últimas ediciones de los juegos, el patrocinio de Reebok desapareció llegando un nuevo patrocinio por parte de una nueva marca del mundo del CrossFit, Nobull. Nobull es una marca creada en 2015 por Marcus Wilson y Michael Schaeffer, ambos veteranos de la empresa Reebok. Lanzaron esta empresa centrándose en su pasión, el CrossFit. En sus productos combinaron tecnología militar, para ofrecer durabilidad a sus productos, y versatilidad para poder ofrecer productos que no sean exclusivos del ámbito del gimnasio, ya que el perfil de deportista que ejerce el deporte de CrossFit también les gusta hacer otros deportes como bicicleta, correr al aire libre y demás. Por ello, esta versatilidad del producto le hizo saltar a la cima de las marcas de material de entrenamiento, convirtiéndose así en principal patrocinador de los CrossFit Games (Newcomb, 2020).



Figura 3. Prueba de CrossFit Games. (CrossFit Games, 2022) [10]

Los juegos consisten en varias fases, empezando por un abierto mundial en el que puede participar cualquier persona mayor de 14 años. En esta fase se seguirá una tabla de puntuaciones que depende de la realización de varios ejercicios del ámbito del fitness. Al final de la etapa del abierto mundial, los atletas mejor clasificados de cada continente, normalmente el 10% mejor clasificado, pasan a la etapa de las eliminatorias. Las eliminatorias consisten en cuartos de final, semifinales y clasificador de última oportunidad. Los atletas que consigan clasificarse de semifinales a la fase final, serán invitados a uno de los 10 eventos presenciales en los 6 continentes, lo que garantiza la presencia de al menos un atleta por continente manteniendo la diversidad buscada. El número de atletas clasificados para la fase final varía por continente, por ejemplo, por parte de Norte América se clasifican los 5 atletas mejor posicionados, y de por parte de Sur América solamente se clasifican los 2 atletas mejor posicionados en la clasificación. Antes de la fase final de los CrossFit Games, se realiza un clasificatorio de última oportunidad para los atletas que no hayan conseguido superar por poco el corte en las semifinales. Por último, la temporada culmina con la fase final de los CrossFit Games, ésta se define como *“la prueba de condición física definitiva”*. Los atletas que hayan logrado clasificar, se involucran en una serie de desafíos desconocidos para ellos hasta justo antes de comenzar el evento, así se evita una posible preparación y se es totalmente dependiente de la condición física. Se proclama ganador quien mayor puntuación consiga en la

realización de estos desafíos y se determina quien es “el más apto del mundo” tal y como define CrossFit, LLC en la web oficial de los CrossFit Games.

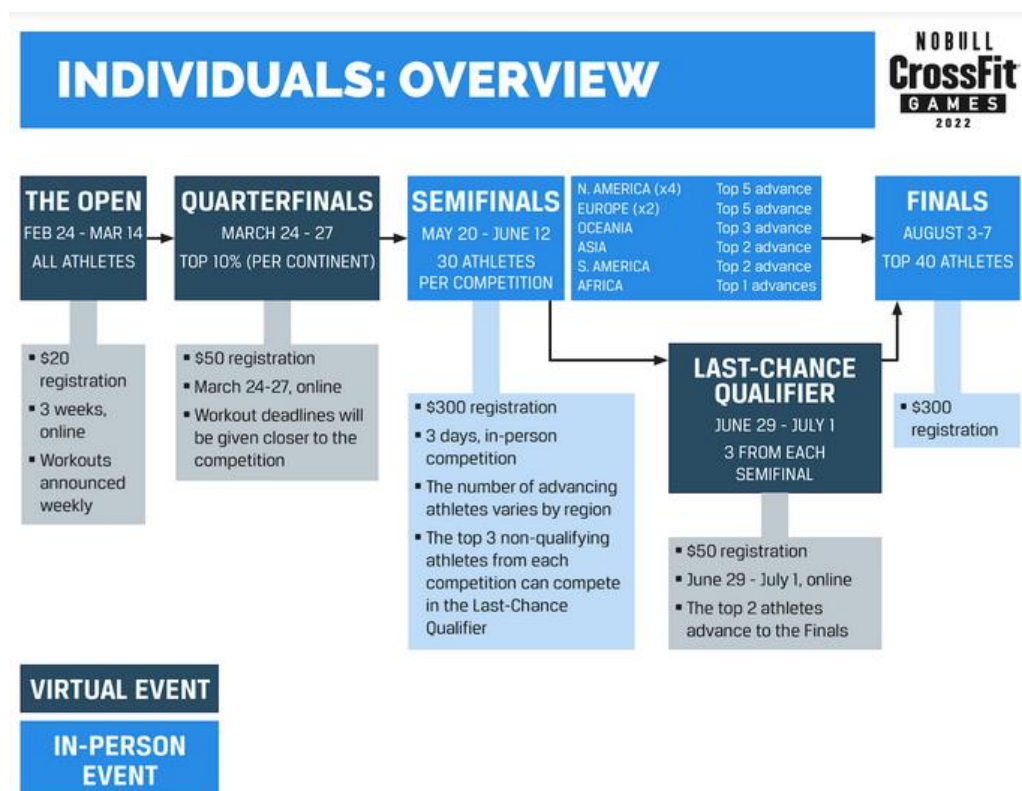


Figura 4. Esquema de las fases de CrossFit Games en la modalidad individual (CrossFit Games, 2022) [10]

2.3 Análisis de mercado en aplicaciones de gestión de CrossFit

Analizando soluciones existentes en el mercado que puedan satisfacer la necesidad de gestionar o planificar cualquier centro CrossFit, se encuentran diversas soluciones a cada cual diferente que son capaces de gestionar cualquier centro.

En cuanto aplicaciones web, diversos centros de CrossFit tienen su web propia en la que gestión se centra simplemente en la reserva horaria de la zona de entrenamiento, y la posibilidad de apuntarse al centro para disfrutar de sus instalaciones. Otros centros, como mucho, tienen en sus páginas web ciertos ejercicios de ejemplo que se puede utilizar en algún entrenamiento personal, ya sea dentro del centro o fuera.

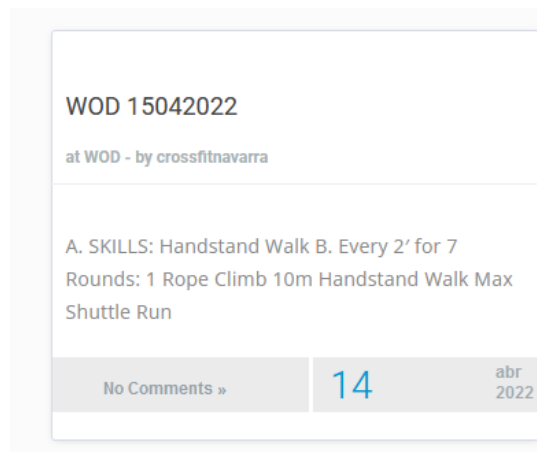


Figura 5. Ejemplo de ejercicio. (Queiron CrossFit, 2022) [28]

En el ámbito de las aplicaciones móviles, se encuentran muchas más soluciones que en las páginas web.

Las aplicaciones móviles para gestión de centros de CrossFit o para gestión de entrenamientos de CrossFit ofrecen muchas funcionalidades, se tratan de aplicaciones que están creadas como ayuda extra en tus entrenamientos, ya sea para guardar marcas y ver tu progreso en el tiempo, planificar mejor tus entrenamientos, seleccionar una amplia gama de ejercicios disponibles y posibilidad de poder reservar plaza en alguna actividad que pueda ofrecer el centro.

Este tipo de solución, la de aplicaciones móviles, es la que más se asemeja a la solución propuesta por Nubapp S.L en su planificador para centros de CrossFit. La funcionalidad ofrecida va desde poder planificar entrenamientos, tener una amplia gama de ejercicios, WODs, calentamientos y enfriamientos, la posibilidad de poder crear una planificación basada en programas, que se dividen en semanas, pudiendo planificar cada día de la semana y asignar ese programa a un cliente o una actividad, y la posibilidad de poder asignarle un plan nutricional a la persona que así lo desee, pudiendo ser asignado a un programa y llevar distribuciones semanales.

Investigando en el mercado de las aplicaciones móviles, se encuentran muchos ejemplos que son aplicables a la gestión de centros deportivos o a la planificación de CrossFit, como, por ejemplo:

- **Resawod:**

La aplicación Resawod es una aplicación creada por Nubapp S.L para la gestión de centros deportivos o para la gestión de boxes de CrossFit. El software pretende automatizar la gestión del box, conectar y fidelizar clientes y es totalmente funcional desde el momento de la contratación, tal y como explica Nubapp S.L en su propia página web. Ofrece diferentes funcionalidades más generales para todos los centros deportivos que contraten estos servicios y más adelante cuando el planificador de boxes de CrossFit que se está desarrollando, se introducirá esa funcionalidad más específica para los boxes de CrossFit. Algunas de las funcionalidades que ofrece son, por ejemplo, poder crear y asignar abonos y bonos personalizados, un módulo de control de accesos y aforo del centro, un calendario personalizado de actividades y reserva de sesiones, reservas y pagos online, etc. (Nubapp, S.L, 2022)

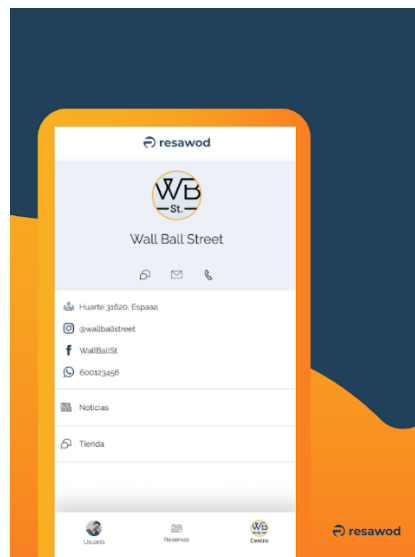


Figura 6. Aplicación Resawod (Nubapp S.L, 2022) [25]

- **Wodzilla:**

Wodzilla es una aplicación creada por StudioIP, LLC. Dedicada a la planificación y seguimiento de los entrenamientos para una mejor experiencia. Tal y como nos indica la empresa StudioIP, LLC (2022), la

aplicación recoge datos de cada entrenamiento, así como tiempos y marcas personales que se utilizan para llevar un seguimiento del progreso, también, se ofrecen entrenamientos y ejercicios que se puedan asemejar a los gustos de cada atleta, teniendo en cuenta los entrenamientos que han sido planificados anteriormente. La aplicación, según la empresa en su página web oficial, es ideal para atletas que quieran llevar su experiencia a un mayor nivel de exigencia.



Figura 7. Aplicación Wodzilla (StudioIP, LLC, 2022) [36]

- **Velites Timer App:**

Velites Timer App es una aplicación creada por la propia empresa Velites. Velites (2022) con su documentación sobre la aplicación se observa que es una aplicación con menor funcionalidad que las otras dos aplicaciones

mencionadas anteriormente. Esta aplicación, está pensada para ejercer de reloj, cronometro, cuenta atrás e intervalos en cualquier ejercicio que se realiza en un box o centro. La aplicación tiene una interfaz con números legibles desde una distancia lejana para poder colocar el dispositivo móvil en un lugar adecuado y así poder ejercitar con total libertad. Como nos indica la propia empresa, es una aplicación con una funcionalidad bastante simple pero muy útil en el mundo del CrossFit, ya que, en este deporte, cronometrar los tiempos o tener claro el tiempo que se ha consumido en la realización del ejercicio, para tener claras las marcas personales.



Figura 8 . Aplicación Velites Timer App. (Velites, 2022) [38]

Por lo tanto, estudiando las posibles soluciones de mercado, y con la funcionalidad propuesta en el nuevo planificador de centros de CrossFit por parte de Nubapp S.L, el nuevo planificador que se va a crear será el que mayor número de funcionalidades ofrezca y a su vez será el más completo.

	Resawod con nuevo WOD Planner	Resawod sin nuevo WOD Planner	Wodzilla	Velites
Abonos y bonos	✓	✓	X	X
Pagos y reservas	✓	✓	X	X
Control de accesos y aforo	✓	✓	X	X
Calendario de actividades	✓	✓	X	X
Planificador de entrenamientos	✓	X	✓	X
Cronómetro	X	X	X	✓
Marcas personales	✓	X	✓	X

Tabla 1. Tabla comparativa de funcionalidades por cada aplicación

Fuente: elaboración propia

Como podemos observar en la tabla 1, la aplicación que más se le acerca es Resawod, la aplicación de la misma empresa que va a realizar el nuevo planificador. Este nuevo planificador se añadirá como nueva funcionalidad en las aplicaciones de Resawod y las demás que dispone la empresa Nubapp S.L, haciendo así la aplicación mucho más potente. Con este añadido, por lo tanto, podemos afirmar que Nubapp S.L no tiene ahora mismo una competencia muy feroz en el software de gestión de centros deportivos. Los centros que no contratan los servicios de Nubapp S.L optan por tener una aplicación propia y personalizada o por el uso de aplicaciones con menor número de funcionalidades para que la experiencia de usuario sea simple y eficaz. Esto no implica que el nuevo planificador propuesto sea difícil de utilizar, al contrario, tendrá una experiencia de usuario simple e intuitiva a la vez que una funcionalidad muy completa.

La aplicación que más funcionalidad tiene de los dos restantes es la de Wodzilla. Al ser una aplicación simple y muy centrada solamente en el deporte del CrossFit, tiene únicamente la funcionalidad de planificar entrenamientos y poder guardar las marcas personales para ver el progreso.

Por último, la aplicación de Velites, aunque es muy útil para el desarrollo de un entrenamiento de CrossFit ya que en este deporte el tiempo invertido en cada ejercicio es muy importante para poder valorar la intensidad que se le aplica, simplemente se ajusta a ser un cronómetro y no ofrecer más funcionalidad que la que podría ofrecerte la aplicación de reloj de cualquier teléfono móvil. Solamente coge los estilos de un reloj digital grande y con colores vistosos para poder ser vistos fácilmente desde la lejanía si fuese necesario.

3 Aspectos técnicos del desarrollo

En este apartado se van a enumerar y explicar los diferentes aspectos técnicos del proyecto. En primer lugar, se explica la metodología de trabajo utilizada para el desarrollo del mismo. Añadido a la explicación de la metodología, se explica la herramienta que se usa para el seguimiento de tal método. En segundo lugar, se explicará la herramienta de control de versiones y la estructura que seguiremos para ello. A continuación, se enumerarán los diferentes lenguajes de programación utilizados para el desarrollo de este módulo y por qué se han elegido. Seguido a esto, se explicarán las diferentes herramientas utilizadas para el desarrollo y la escritura del código, así como diferentes plugings. Por último, se comentará el despliegue de este módulo conforme se va desarrollando.

3.1 Metodología de trabajo

3.1.1 Scrum

Para el desarrollo de este proyecto, la metodología utilizada es una metodología ágil llamada Scrum. Como nos indican Schwaber y Sutherland (2020), creadores de la guía Scrum para ayudar a las personas a entender Scrum, esta metodología se puede definir como un marco ligero que ayuda a las personas, equipos, organizaciones o empresas a generar valor a través de soluciones que se adaptan a problemas complejos. En otras palabras, en el ámbito del desarrollo, es una metodología que, durante el proceso de desarrollo, se antepone a cualquier problema poniendo soluciones de valor para obtener un producto mínimo viable al final de cada etapa. Igualmente, esta guía nos indica que la parte más importante en la metodología Scrum es el equipo o Scrum team. El Scrum team debe de ser pequeño para poder ser ágil y lo suficientemente grande para completar una cantidad de trabajo significativo dentro de cada etapa en la que se divide el proceso de Scrum. Si el Scrum team se hace demasiado grande, se debe considerar la idea de

organizar el Scrum team en equipos más pequeños cohesionados con el fin de mantener la agilidad.

Unido a esto, el mencionado Scrum team se compone de diferentes roles. En primer lugar, un dueño de producto o Product Owner, es la persona que conoce, define y marca los objetivos y prioridades del proyecto que se está desarrollando. Es la persona que sabe que producto exactamente quiere el cliente. En segundo lugar, se encuentra el rol de Líder de equipo o Scrum Master, es la persona que garantiza el funcionamiento y el cumplimiento de la metodología Scrum dentro del equipo. Ayuda al equipo ante cualquier problema que pueda aparecer y actúa de filtro con el equipo con posibles problemas externos al equipo que puedan afectar a la estructura o ritmo de trabajo. Por último, se encuentra el equipo de desarrollo, se trata de un equipo organizado que trabaja de forma conjunta realizando las diferentes actividades o tareas que haya planificado el Scrum Master o Product Owner para ese periodo de tiempo. Son responsables del éxito o del fracaso de la metodología ya que, si no se desarrolla un producto viable al final de cada etapa del método, se considera un fracaso (Ortiz, 2020).

La metodología ágil Scrum se divide en fases llamadas sprints. Estos sprints pueden ser de una longitud desde una semana hasta 4 semanas, pero lo más habitual es realizar sprints de dos semanas. Durante el sprint se realizan diversas reuniones con el fin de conocer el estado del proyecto y la evolución del mismo. Como indica la guía Scrum, la primera reunión que se realiza en el sprint es la reunión de planificación, en esta reunión se planifican los objetivos y las tareas a realizar en el sprint y se reparten los objetivos entre el scrum team con el objetivo de dividir la carga de trabajo y que todo siga la agilidad. Durante el sprint, se realiza la reunión daily o diaria. Esta reunión de aproximadamente quince minutos tiene como objetivo controlar el progreso del sprint hacia el objetivo final, y, si es necesario, modificar el objetivo final del sprint para que se pueda cumplir. Por último, tiene lugar la reunión de demostración o reunión de revisión. En este evento se inspecciona el resultado del sprint y se permiten corregir diferentes errores que puedan surgir, también se pueden definir futuros objetivos que se puedan introducir en el objetivo final del proyecto. Existe una última reunión que no se realiza con tanta frecuencia como las anteriores, se trata de la reunión retrospectiva. Esta

reunión trata de planificar entre todo el equipo scrum formas para aumentar la eficacia y la calidad del equipo con el fin de conseguir los objetivos con mayor antelación y mayor calidad (Schwaber y Sutherland, 2020).

Existen otras herramientas dentro de la metodología que sirven para mejorar el objetivo y la organización. Una de ellas es el llamado Backlog Product, en el que se colocan todas las tareas u objetivos del proyecto, ordenadas por prioridad y con una estimación inicial de tiempo. Por otro lado, existe el Sprint Backlog, que tiene un funcionamiento similar al Backlog Product pero solamente con las tareas asignadas al sprint. Por último, existen gráficos como, por un lado, el burn-down en el que te permite controlar el tiempo que falta para completar el sprint, también midiendo el esfuerzo restante y el esfuerzo ideal, y por otro lado el burn-up que permite al product owner ver la evolución del producto conforme la velocidad del equipo (Ortiz, 2020).

Por último, cabe destacar tal y como indica la guía scrum que, si durante un sprint o durante el desarrollo del proyecto hubiese algún problema que no se pudiese resolver, solamente el product owner tiene la capacidad de poder cancelar el proyecto o el sprint si fuese necesario, sin tener en cuenta la opinión del resto del scrum team.

3.1.2 Tablero Ágil Jira

Para poder seguir la metodología de trabajo y contabilizar el esfuerzo y el progreso se utiliza una herramienta de Jira Software, en concreto el tablero ágil de Jira Software. Jira Software es una web de la empresa Atlassian, que se puede vincular a un repositorio git de bitbucket, la web de repositorios de Atlassian, para poder tener vinculado el proyecto con el código y la metodología ágil. La empresa Atlassian nos indica que Jira Software tiene diferentes funcionalidades como construir roadmaps del progreso del proyecto, customizar el tablero de progreso pudiendo cambiar los tiempos y los campos que definen cada tarea, capacidad de poder buscar tareas que ya hayan sido realizadas y cerradas, una aplicación móvil con soporte continuo y un Marketplace con más de mil plugings disponibles para poder añadir al software y poder mejorar la experiencia.

En este caso, el tablero de Jira Software se compone de una serie de columnas que indican el estado en el que se encuentra la tarea. La columna “to do” indica que la tarea aún no se ha comenzado y está pendiente de ello, la columna “paused” indica que la tarea se ha comenzado a realizar, pero actualmente no se está trabajando en ella, por otro lado, la columna “in progress” indica que la tarea se está realizando en ese momento y todo el tiempo que se encuentre en esa columna contabiliza para el progreso y el esfuerzo del sprint, añadida a ellas se encuentra la columna “in review”, en ella se encuentran las tareas que se han dado por finalizadas pero están a la espera de revisión por parte de algún compañero, y por último, se encuentra la columna “done” en la que se ubican las tareas que se dan por finalizadas y ya se ha unido su contenido al proyecto principal.

La utilización de Jira Software permite una mejor experiencia y un mejor seguimiento de la metodología ágil Scrum, y es compatible con el repositorio de control de versiones que se utiliza, que se explicará en el siguiente apartado.

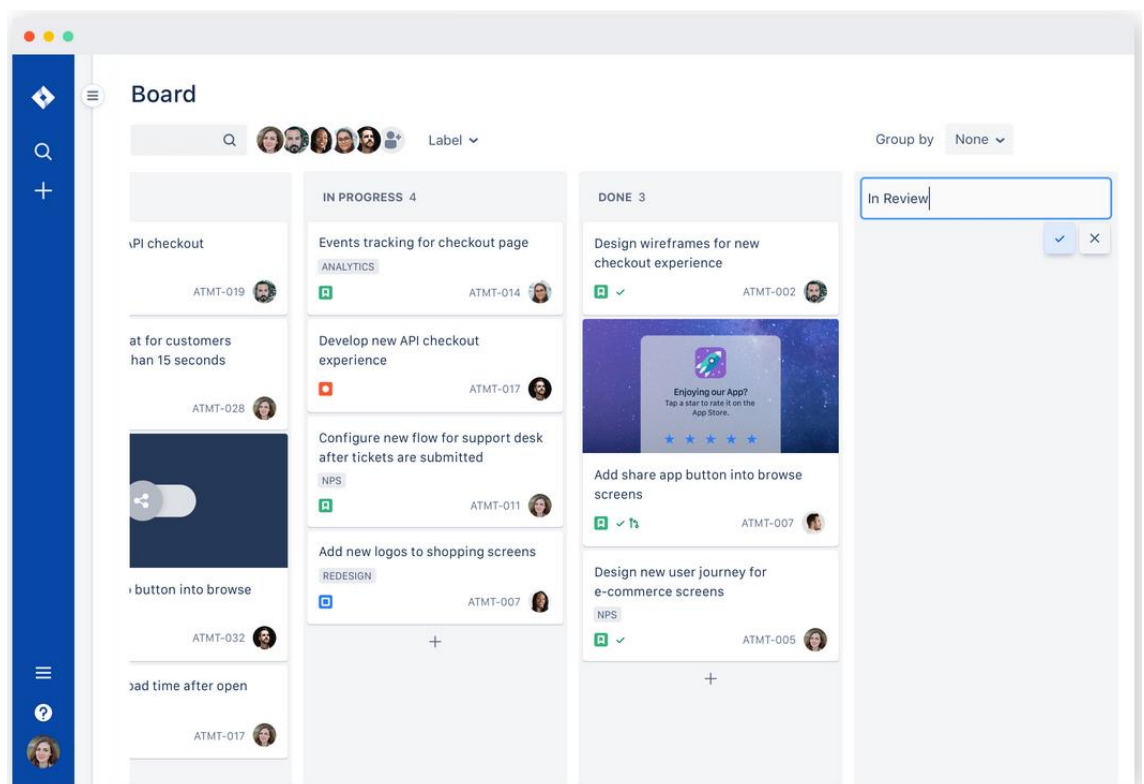


Figura 9 . Imagen de ejemplo de tablero ágil de Jira Software. [4].

3.2 Control de versiones

Para la realización del proyecto y su consiguiente seguimiento se hace uso de la herramienta de control de versiones conocida como GIT. Se define como un sistema avanzado de control de versiones en el que no se guardan los cambios de una etapa a otra si no la imagen en cada confirmación que se hace, por lo que permite rastrear el progreso del proyecto y ver cómo ha ido cambiando, por lo tanto, como el control de versiones guarda instancias del proyecto y no sus cambios, se puede experimentar tranquilamente en el proyecto teniendo la capacidad de volver al momento del desarrollo donde todo estaba correcto si fuese necesario (Astigarraga y Cruz-alonso, 2022). Además, el control de versiones te permite grabar mensajes en cada confirmación de cambio que se realiza con el fin de que el propio desarrollador o una persona externa al revisar el historial del proyecto sea capaz de entender el progreso y poder unirse al proyecto si fuese necesario en algún momento (Blischak et al. 2016).

Una de las mejores características más importantes de GIT y que a su vez le ofrece esa seguridad de utilización es que al ser un sistema distribuido te permite múltiples copias de seguridad en diferentes terminales, ya sea en local o en el propio repositorio virtual. Con esto, GIT no tiene un único punto de fallo en el que se podrían perder todos los datos, teniéndolos siempre accesible la última versión del proyecto que se encuentre en el repositorio virtual. Con esta capacidad, se permiten implementar múltiples flujos de trabajo que pueden confluir finalmente en el mismo repositorio virtual (GIT Software Freedom Conservancy, 2022).

Para experimentar con el proyecto, GIT tiene un área en el repositorio local llamada área de ensayo o staging en la que se permite revisar los cambios y las confirmaciones realizadas antes de completar la confirmación que se va a actualizar en el repositorio virtual. Esta herramienta permite también poder actualizar solamente los archivos que creamos necesarios para subirlos al repositorio virtual (GIT Software Freedom Conservancy, 2022). Para guardar los cambios en GIT o los progresos realizados se sigue mediante un proceso de confirmaciones o commit, estas son operaciones para actualizar los archivos en el servidor remoto. De tal manera, las confirmaciones o

commits de GIT se pueden ver y compilar de manera local en la copia del proyecto que se tenga en el ordenador físico y mediante un comando de GIT, poder enviar la confirmación al servidor externo y actualizar el proyecto para que sea accesible a todos los participantes (Atlassian Bitbucket, 2022).

Todo el potencial de GIT tiene acceso a través de comandos que se ejecutan por terminal, aunque GIT ofrece una interfaz para poder ver y realizar los cambios, el mayor potencial se alcanza a través de la terminal. A continuación, se puede ver una tabla con algunos de los comandos más habituales a la hora de la utilización de GIT.

Tarea de GIT	Descripción	Comando
Crear nuevo repositorio local		git init
Clonar un repositorio remoto en uno local	Crear una copia de un repositorio local	git clone /path/to/repository
	Crear una copia de un servidor remoto	git clone username@host:/path/to/repository
Añadir archivos	Añadir uno o más archivos a área de ensayo o área de staging	git add <filename> git add *
Confirmar cambios o commit	Confirma los cambios en local pero no en el remoto	git commit -m "commit message"
	Confirma cualquier archivo de área de staging y cualquiera que se haya cambiado desde entonces.	git commit -a
Confirmar cambios en servidor remoto	Envía los mensajes confirmados de nuestra rama local al servidor remoto	git push origin master
Ver el estado del repositorio	Lista del estado de los archivos que han sido cambiados y necesitan confirmación	git status

Ramas	Crea una nueva rama y se cambia a ella.	git checkout -b <branchname>
	Cambia de una rama a otra	git checkout <branchname>
	Lista todas las ramas disponibles y te dice en cual te encuentras	git branch
Actualiza el repositorio local desde el servidor remoto	Obtiene y combina los cambios del servidor remoto con el local	git pull
	Une una rama con la rama actual en la que te encuentras	git merge <branchname>
	Cambia la base de origen de la rama actual a la rama indicada	git rebase <branchname>

Tabla 2. Tabla resumen de los comandos más importantes de GIT.

Fuente: elaboración propia

La distribución de trabajo en el control de versiones GIT se hace a través de ramas o branches. Una rama o branch, según la documentación oficial de GIT, es un apuntador a una confirmación o commit que hayamos realizado en ese entorno de trabajo. GIT tiene una rama por defecto en cada proyecto que se crea, por defecto es la rama master. Esta rama en los entornos de trabajo profesionales es el entorno de producción, es decir, el trabajo que el cliente final tiene a su alcance y se expone en los servidores de despliegue y demás. Por otro lado, hay otras ramas que se suelen crear en los trabajos profesionales, una de ellas es la llamada rama develop, en ella, se aplican los cambios de las tareas que se han realizado durante un sprint de scrum, cuando la empresa crea oportuno, esa rama develop se mergea a master y así, las mejoras y errores corregidos ya pueden salir a producción una vez testeados. Por otro lado, otra rama que suele aparecer en los entornos profesionales es la rama release, esta rama es un tanto especial ya que solamente existe durante un corto periodo de tiempo. Esta rama se crea cuando se ha hecho una demostración del contenido que se encuentra en develop antes de mergear a master y se ha encontrado algún error que corregir. Los errores se corrigen en la rama release que se saca con base en develop y posteriormente cuando los errores estén solucionados, será la rama release la que se mergee a master y posteriormente desaparezca. Esto nos evita posibles conflictos en la rama de develop, ya que mientras se

arreglan esos errores, hay tareas que se siguen desarrollando. Por último, pueden existir ramas específicas en de cada tarea que se vaya a realizar, ya sea porque tiene demasiados cambios planteados, o es una tarea que se va a alargar en el tiempo y se quiere guardar su progreso sin interferir en la rama de desarrollo.

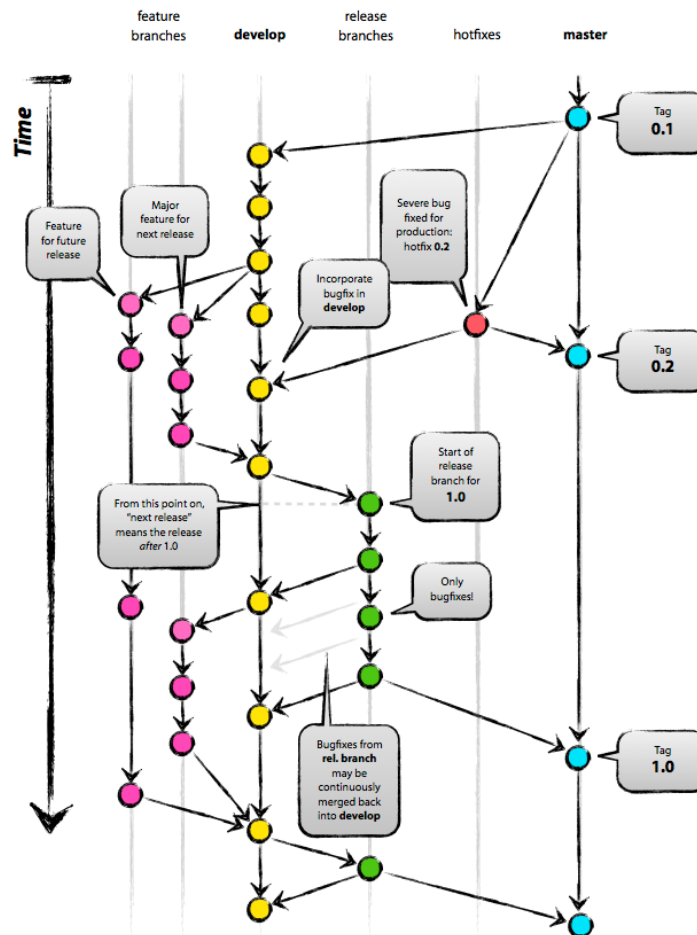


Figura 10 . Flujo de trabajo de ramas de GIT (Aula de software libre, 2021) [6]

3.3 Lenguajes de programación

Durante el desarrollo del problema y del módulo planteado, se han utilizado diferentes lenguajes de programación para ello, por ejemplo, se ha hecho uso de los lenguajes de desarrollo web como HTML, CSS y JavaScript, así como algún Framework del último lenguaje mencionado. A continuación, se explica en qué consisten estos lenguajes.

3.3.1 JavaScript

JavaScript se define como un lenguaje de programación secuencial con el que se consiguen implementar funciones complejas en una página web, así como animaciones, actualizaciones, reproductores de video etc. (mdn web docs, 2021). Por otro lado, Ramos (2021) nos indica que JavaScript es un lenguaje encargado de dar interactividad y dinamismo a una página web, indicando que es un lenguaje que lee el navegador, independientemente del dispositivo en el que se encuentre el navegador, ya sea ordenador, móvil Android o iOS, el lenguaje es siempre el mismo. Añadido a todo esto, en la mdn web docs nos indica que JavaScript es un lenguaje del lado del cliente, solamente interpretado por el navegador independientemente de la plataforma en la que se encuentre el navegador, tratándose de un lenguaje imperativo, débilmente tipado y muy dinámico.

Tradicionalmente este lenguaje tiene a confundirse con el lenguaje de programación Java ya que sus nombres son muy parecidos, pero no tienen nada que ver. JavaScript se diseñó con una sintaxis similar al lenguaje C y basándose en él, aunque adopta nombres parecidos a Java, son lenguajes utilizados para fines totalmente diferentes. Los lenguajes C y Java están orientados al lado del servidor y JavaScript está orientado al lado del cliente (Wikipedia, 2022).

3.3.2 React

React es el lenguaje en el que se basa el desarrollo de este módulo. React, tal y como se define en la propia página web del lenguaje, realmente es una biblioteca de JavaScript para construir interfaces de usuario interactivas de forma sencilla. Así pues, React, utiliza un concepto basado en componentes, se ocupa de renderizar interfaces de usuario y reserva sus acciones solamente cuando el DOM (Virtual Document Object) cambia (Sirotko, 2022). Añadido al concepto que nos ofrece Sirotko, la propia librería de React en su documentación, nos explica que el lenguaje se basa en crear componentes encapsulados que manejan un estado, para poder realizar los cambios en el DOM tiene que cambiar el estado del componente, y se convierten en interfaces de usuario complejas, como los componentes tienen la lógica construida en JavaScript se pueden

pasar datos de forma sencilla a través de la aplicación y mantener el estado del componente fuera del DOM.

React está principalmente orientado al frontend y a la creación de interfaces de usuario, aunque también existe la posibilidad de renderizar React desde el lado del servidor utilizando Node, otra librería de JavaScript.

3.3.3 TypeScript

TypeScript es un lenguaje fuertemente tipado basado en JavaScript agregándole sintaxis adicional para que el editor utilizado para redactar el código detecte los errores más temprano y para combatir lo débilmente tipado que es el lenguaje JavaScript base (Microsoft, 2022). Asimismo, TypeScript, desarrollado por la empresa Microsoft, añade tipos y objetos basados en clases, teniendo un propio compilador desarrollado también en TypeScript y compilado a JavaScript con Licencia Apache 2 (Wikipedia, 2022).

En este caso, para el desarrollo del módulo se ha implantado este lenguaje para poder controlar el tipado de todas las variables y objetos utilizados en las vistas creadas, con el fin de mantener un orden para que las llamadas API que se realicen sean más claras y así el trabajo a realizar por parte del backend sea lo más sencillo posible.

3.3.4 Otros lenguajes

Añadidos a los lenguajes de programación mencionados, se hacen usos de otros lenguajes, pero en una medida claramente menor. En primer lugar, se hace uso del lenguaje HTML, o, mejor dicho, JSX, que es una extensión de JavaScript para introducir código HTML en React para renderizarlos en el DOM (Meta Platforms, Inc., 2022). En segundo lugar, se ha hecho uso del lenguaje CSS, siendo un lenguaje basado en reglas en la que cada usuario las especifica para cada elemento, estableciendo así los estilos visuales que se van a aplicar a los elementos que se han renderizado en el DOM (Mdn web docs, 2022). Por último, se ha hecho uso del lenguaje PHP, siendo PHP un lenguaje de código abierto principalmente para desarrollar el lado del servidor de la aplicación (The PHP group, 2022), el objetivo principal del uso de este lenguaje ha sido crear claves de traducciones que tiene la propia empresa integradas en ficheros PHP para poder

disfrutar del mecanismo de traducciones, que tiene implantado la empresa, para que en el nuevo módulo también se tengan todos los lenguajes de los usuarios disponibles.

3.4 Herramientas de desarrollo

En cuanto a herramientas de desarrollo, principalmente se han utilizado un IDE de desarrollo de código como PHPStorm y un software de contenedores como Docker. En primer lugar, PHPStorm es un IDE de desarrollo, desarrollado por la empresa JetBrains, especializado en programación en lenguaje PHP y lenguajes web. Según la propia empresa JetBrains, este IDE de desarrollo cuenta con todas las funcionalidades de WebStorm y compatibilidad completa con bases de datos SQL para poder desarrollar un backend totalmente funcional, por otro lado, también reconoce los lenguajes de frontend como JavaScript y cualquier framework o librería que derive de él, con ello, este IDE de desarrollo ofrece la experiencia y la capacidad completa de desarrollar una aplicación web con soporte backend y frontend completos.

En segundo lugar, se ha utilizado el software de contenedores Docker. Docker es un marco de software que permite construir, ejecutar y administrar contenedores en servidores y la nube (Red Hat, Inc, 2022). Un contenedor se define como un paquete de software que empaqueta todo el código para que la aplicación que integra se pueda ejecutar de manera rápida y eficaz en cualquier entorno informático, independientemente del sistema operativo o sistema de ficheros en el que se esté ejecutando. Los contenedores aíslan el código y el software de su entorno y garantiza su correcto funcionamiento a pesar de las diferencias posibles en sistemas de ficheros o comandos de terminal (Docker Inc., 2022).

3.5 Despliegue de aplicación

Para desplegar la aplicación cuando el módulo entero se encuentre operativo, con toda la parte del frontend construida y toda la lógica del backend preparada, se va a hacer uso de los servicios de Amazon Web Services (AWS) para que esté disponible para los clientes. AWS se define como una plataforma en la nube que ofrece más de 200 servicios de datos a nivel global. Añadido a esto, también entregan infraestructura para ejecutar aplicaciones en la nube de forma segura, confiable y con un alto rendimiento, también tiene disponibles servicios de computación, redes de hasta 400 Gigabytes por segundo de velocidad e infraestructura de almacenamiento para optimizar el rendimiento (Amazon Web Services, Inc., 2022).

Estos servidores están preconfigurados con una lista de sistemas operativos como Windows y distribuciones de Linux como Amazon Linux 2, Ubuntu, etc., con el fin de poder adaptarse a cualquier aplicación o sistema operativo en el que esté pensado desplegar la aplicación desarrollada (Amazon Web Services, Inc., 2022).

4 Implementación y desarrollo del módulo

4.1 Instalación y preparación del entorno

Para que sea posible el desarrollo del módulo, utilizando los lenguajes de programación mencionados anteriormente y los temas necesarios hace falta la instalación de varios temas y plugins dentro del proyecto ya existente.

En primer lugar, el tema utilizado para la realización del módulo es el tema Metronic en su versión 8. Metronic es un diseño basado en Bootstrap 5, siendo Bootstrap 5 un framework de CSS en el que hay estilos predeterminados para diferentes componentes, que utiliza clases ya predefinidas para dar estilos a los diferentes elementos que se pueden utilizar en una página web (Keenthemes, 2022). Para su instalación, es necesario comprar la licencia de uso, que en este caso la empresa ha autorizado, y posteriormente de la compra de la licencia se da acceso a las vistas de demostración que tiene el tema para diferentes estilos. Una vez se elige la demostración con el estilo que sea requerido en los diseños de las vistas, en este caso es la demo1, en el proyecto se crea una carpeta llamada metronic, con dentro la carpeta demo1 que proporciona el propio tema y otras carpetas necesarias que Metronic aporta. En una de esas carpetas, se encuentra un script llamado build, en el que al ejecutarlo se cargan una serie de archivos CSS y JavaScript con los temas y posibles animaciones ya programadas para cada elemento, con ello, al correr el script build solamente tenemos que aplicar las clases que nos indica Metronic en su documentación para tener esos estilos y animaciones requeridas.

Seguido al tema de Metronic, hace falta la instalación de los lenguajes de programación que se van a utilizar. Para la utilización del lenguaje React, es necesario en el proyecto un sistema de gestión de paquetes llamado Node Pack Manager (npm) que ayuda al desarrollador a gestionar los paquetes relacionados con JavaScript (Deyimar, A., 2021). React al estar basado en Node, utiliza el mismo gestor de paquetes. En npm, se ejecuta el comando de instalación de React (`npm install --save React`) para tener

disponible el lenguaje y todos los componentes y reglas dentro del contenedor. Seguido a React, se necesita TypeScript para tener un lenguaje fuertemente tipado y poder controlar posibles errores de variables, para su instalación se sigue el mismo proceso que con React, se ejecuta el comando de npm de instalación del lenguaje (`npm install --save typescript@version`) y ya está importado directamente en los archivos de extensión `.ts` o `.tsx` asociados al lenguaje TypeScript.

Para casos y plugins especiales, se requiere la utilización del framework de JavaScript llamado JQuery. Este framework tiene incompatibilidades con React, ya que React actúa sobre los elementos de un virtual DOM y JQuery actúa directamente sobre el elemento en el DOM de la página, que es mucho más lento que el funcionamiento de React, pero es necesario para la incorporación de algunos plugins como DataTable, que son tablas construidas para mostrar datos e interactuar con ellos, y el plugin DatePicker o DateRangePicker, que sirven para elección de fechas o un rango de fechas con un fin determinado. Por otro lado, es un framework que se está quedando antiguo y la mayoría de cosas que se pueden hacer con este plugin se pueden hacer fácilmente con JavaScript puro

Por último, para la utilización del lenguaje en React y de TypeScript se necesita un compilador que te permita ejecutar los cambios en la web y ver los posibles fallos para ello, dentro del contenedor docker del proyecto se encuentra npm que tiene un compilador propio, para ello existen comandos de compilación como `npm run build`, que ejecuta la compilación de los archivos una sola vez, y el comando `npm run watch`, que ejecuta la compilación de los archivos cada vez que detecta cambios dentro de ellos, con el fin de ahorrar tiempo en las ejecuciones de la compilación.

4.2 Planteamiento del desarrollo

El desarrollo del módulo se plantea realizarlo siguiendo la metodología ágil Scrum comentada en el apartado anterior. Para ello, el módulo, que contiene ocho submódulos, llamados programas, ejercicios, calentamientos, WODs, enfriamientos, métricas,

equipamiento y menús, que se van a ir desarrollando uno a uno con sus vistas correspondientes.

En primer lugar, antes de la realización del módulo se plantea realizar la estructura básica de la página, con un menú lateral, una barra superior con información del usuario, y una barra de herramientas con opciones correspondientes al módulo en el que te encuentras.

Seguido a la realización de la estructura básica, se plantea el desarrollo del submódulo programas. Éste es el submódulo con mayor extensión, mayor número de vistas y mayor complejidad de funcionamiento.

Por último, seguido a la realización del submódulo de programas, se plantea realizar los siguientes submódulos de manera progresiva ya que siguen la misma estructura y funcionamiento solo que con diferentes objetos, están compuestos por una vista general para mostrar todos los objetos, luego una vista de detalle del objeto y una vista de edición del objeto. Por ello, al realizar un submódulo tiene que estar realizada la estructura de los demás.

4.3 Desarrollo del módulo

El desarrollo del módulo sigue la planificación mencionada en el apartado anterior. En primer lugar, la estructura básica, en segundo lugar, el primer submódulo del planificador y por último los submódulos restantes que siguen la misma estructura.

4.3.1 Estructura básica

Para la realización de la estructura básica, se empezó por realizar la barra lateral de navegación en React. Para ello se migró la barra lateral existente de la empresa en lenguaje PHP a React. Para esta migración, en primer lugar, se realizó la conversión del archivo en lenguaje PHP a un archivo en lenguaje React, posteriormente, migrado el

archivo de PHP a React, se intentó seguir la lógica de la barra lateral e intentar seguir los principios del lenguaje y sacar algún componente reutilizable para poder realizar la barra lateral por componentes. La barra lateral se compone de entradas que corresponden a los módulos y subentradas que corresponden a los submódulos de esos módulos. Al ser una lista, se determinó que debería existir un componente llamado TableEntry, en él se determinaría la lógica de si se trata de un módulo o un submódulo para darle los estilos pertinentes y los links necesarios para que funcione la navegación. Por último, este componente también diferenciará si el módulo que tiene que pintar en pantalla, está construido en React o no, para poder diferenciar el tipo de link a ejecutar en navegación.

Seguida a la barra lateral, se produjo la migración a React de la barra superior. Esta barra superior ya estaba migrada a React por parte de la empresa, solamente se tenía que realizar el tipado de las variables para poder seguir el léxico del lenguaje TypeScript e integrarlo en la vista de la estructura básica. Este componente se centra en el uso del estado para poder determinar si el usuario quiere silenciar notificaciones, recibir correos o no, saber si se encuentra en el ámbito de test que ofrece la empresa para realizar cambios y otras funcionalidades. Según el estado en el que se encuentre la barra superior, la vista pintará unos iconos u otros y seguirá la lógica determinada.

Por último, en la construcción de esta estructura básica, se realizó la construcción de la barra de herramientas. Esta barra de herramientas está ligada al estado de la barra lateral y de la barra superior. En esta barra, se pintan los botones para realizar acciones correspondientes al módulo en el que se encuentra el usuario y botones que redirigen a los ajustes de dicho módulo.

Estos tres componentes de la estructura básica funcionan conjuntamente entre sí. Se basan en el uso del estado de los componentes que nos proporciona React para ello. Se crea una vista llamada BasicStructure que contiene los 3 componentes, la barra lateral, la barra superior y la barra de herramientas. En BasicStructure se crea un estado que hace referencia al módulo y submódulo en el que se encuentra el usuario, iniciándose al principio a vacío. Conforme se va navegando en la barra lateral, el estado de módulo actual y submódulo actual va cambiando, la barra de herramientas lee este estado y según donde se encuentre, dibuja en el DOM unos botones u otros, los que sean

necesarios según la funcionalidad del módulo. La barra superior, también lee este estado, aunque también tiene otra funcionalidad añadida como la lectura de las cookies del navegador. La barra superior hace lectura de estas cookies, ya que, dentro del panel de administración de la empresa, va a haber partes hechas en React y partes hechas con la estructura antigua en PHP. Por ello, la lectura de estas cookies permite que la estructura y el funcionamiento concuerde entre la parte de React y la parte de PHP por si la navegación del usuario se ejecuta entre dos módulos de lenguajes diferentes.

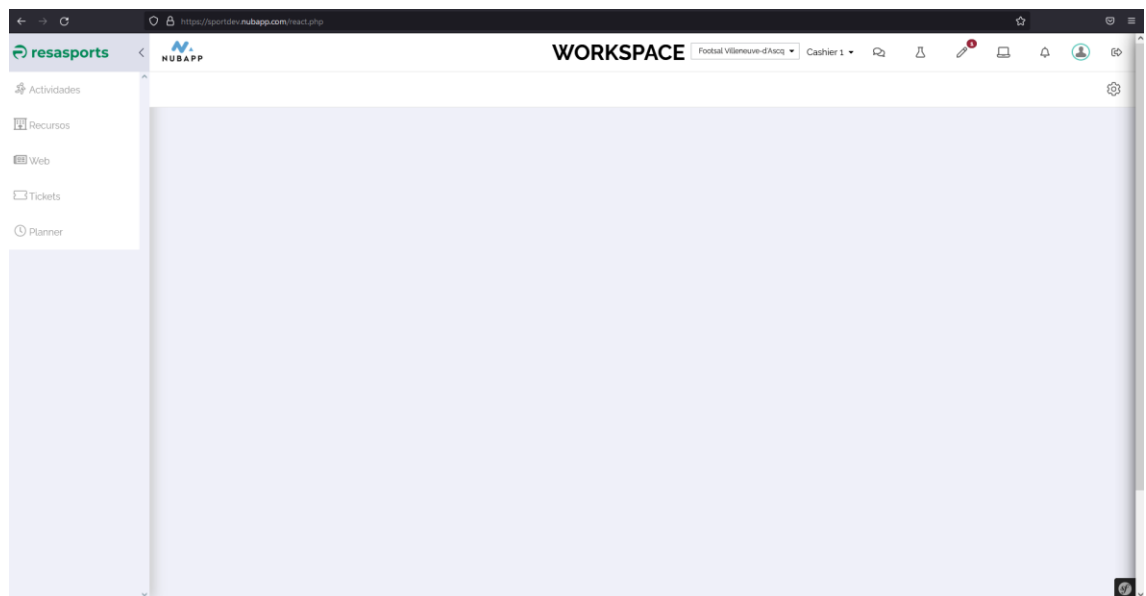


Figura 11 Estructura básica del planificador.

4.3.2 Submódulo de programas

Este submódulo es el más largo y más complejo del módulo planificador. En él se realiza toda la lógica y funcionalidad que se propone en este planificador de entrenamientos. En primer lugar, se realizó una vista en la que se muestran los programas que tiene el centro creados en una tabla. Para esta vista, se realizó un componente llamado SelectDatatable que renderiza una tabla con checkbox para poder seleccionar si se quiere borrar un elemento o todos los elementos de la tabla. Este componente se realizó ya que se iba a utilizar en las vistas principales de cada submódulo para poder visualizar los elementos.

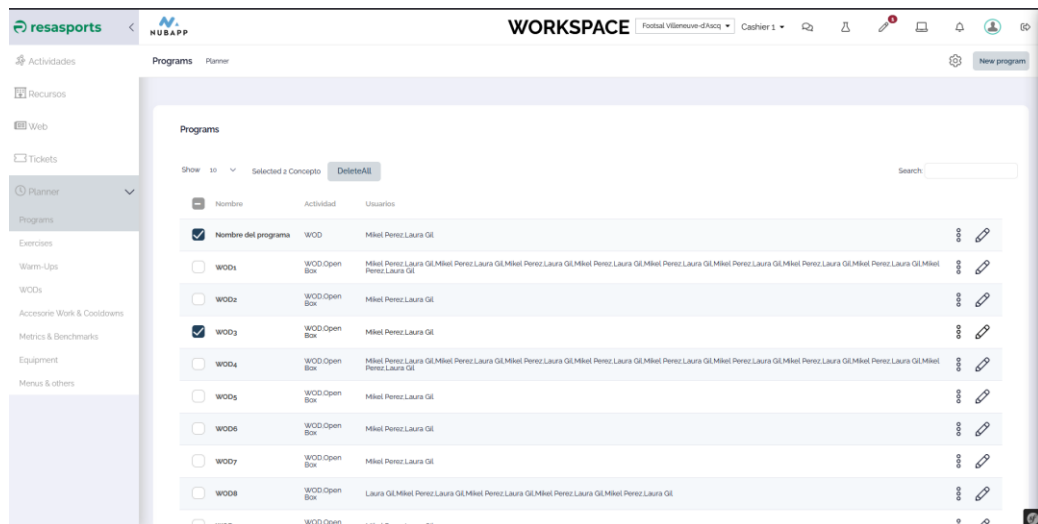


Figura 12. Vista principal del módulo

Seguido a la vista principal con la SelectDatatable, se produjo la realización de la vista de detalles del programa, en ella, se muestra la información específica de cada programa como el nombre y una descripción y la planificación por semanas que tiene el programa asignado, pudiendo editar esta planificación añadiendo semanas o editando los bloques a realizar en cada día. La vista constará de dos componentes card, que se explicará más adelante, en el que la primera card contendrá la información principal del programa como el nombre y la descripción y los botones para realizar acciones con el programa, y la segunda card contendrá las semanas correspondientes a cada programa con su información por días.

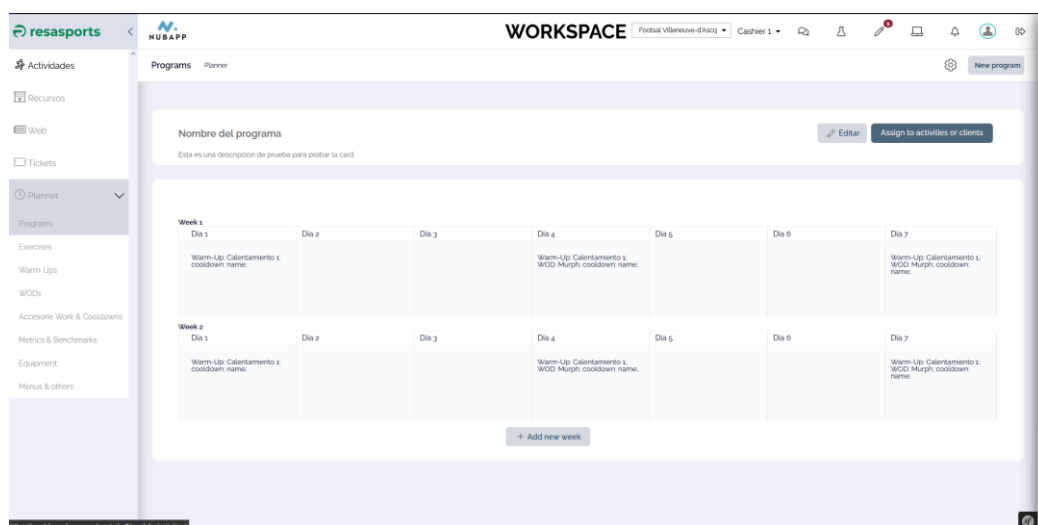


Figura 13 . Vista de detalle de programas

Para esta planificación semanal, se creó un componente llamado WeekCalendar, en el que se renderiza una tabla de siete días, con la información de la semana correspondiente al programa, este componente se realizó ya que en todas las vistas de detalles de programa se sigue la misma estructura. Para la edición de un día de la semana, se incorporó un formulario lateral como un componente adicional que se renderiza cuando se pulsa el botón de editar el día y se desrenderiza cuando se le da al botón de cerrar el formulario, así se evita la carga de componentes innecesarios para no tener tiempos de carga superiores a los necesarios. Este componente de formulario lateral se reutilizará también en todas las vistas de detalles de programa.

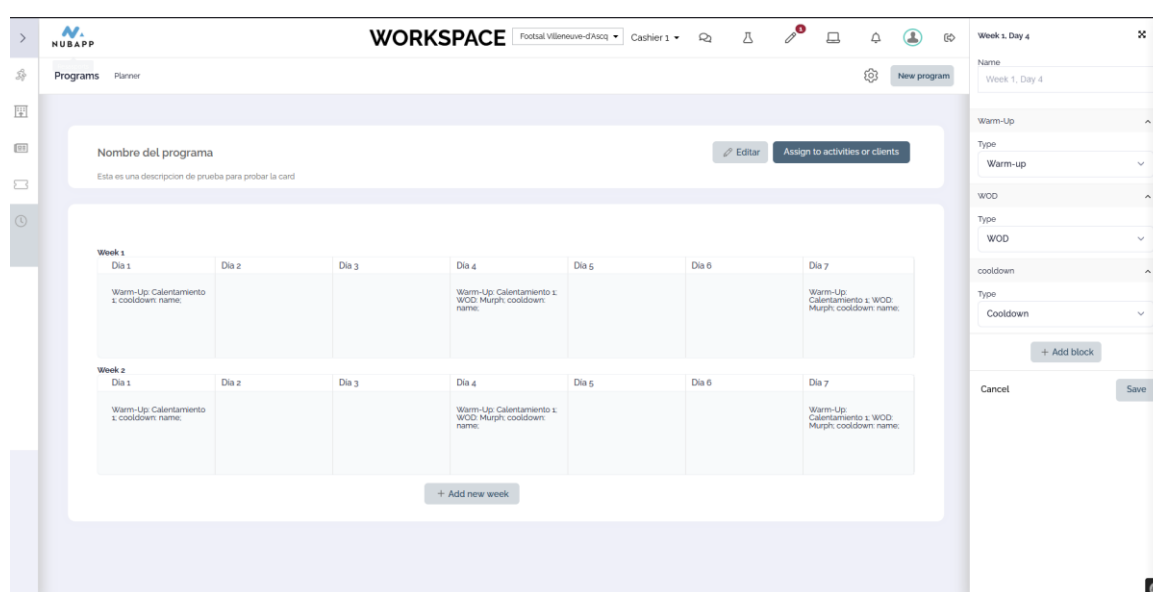


Figura 14 . Vista de detalles de programa con el formulario lateral abierto

Como se ha mencionado anteriormente, este módulo es el más completo de todos ya que toda la funcionalidad propuesta en el planificador, como crear programas, asignar programas a un usuario o una actividad, editar programas, editar días de programas pudiendo añadir o eliminar bloques dentro del día, poder editar los bloques del día pudiendo cambiarles el tipo de bloque que es, pudiendo ser de tipo calentamiento, WOD, enfriamiento, métrica, menú o equipamiento. Todas estas opciones de bloque abren el abanico de opciones al planificador, solamente no se centrará en planificar entrenamientos, también se puede extender en planificar un programa de métricas con bloques de tipo métrica, y también es posible planificar alguna

dieta deportiva con los bloques de tipo menú. Aunque se basa en los elementos de los otros submenús, se trata de el submódulo más importante del planificador.

4.3.3 Submódulos restantes

Los submódulos a desarrollar después del submódulo de programas son más sencillos de desarrollar, ya que contienen solamente una vista principal igual que la del primer submódulo, una vista de editar el elemento y una vista de detalle de elemento.

Los submódulos a desarrollar son: ejercicios, calentamientos, enfriamientos, WODs, métricas, equipamiento y menús, siendo el submódulo de métricas un poco especial al tener la vista de editar elemento con un componente adicional.

Las vistas principales de todos los submódulos son iguales, están compuestas de una tabla que está construida con el componente `SelectDatatable` mencionado en el apartado anterior. En la figura 12 se observa la vista principal del submódulo de programas, todas las demás son iguales solamente cambiando el título de la página.

Las vistas de detalle de elemento existen en los submódulos de ejercicios, calentamientos, WODs y enfriamientos. Esta vista, como se puede observar en la figura 15, consiste en coger la información de cada elemento y mostrarla en pantalla, añadiendo un video informativo sobre el elemento si fuese necesario. Para esta vista, se creó un componente llamado `ElementDetails` en el que se diferenciaba que tipo de elemento se trata y según el tipo saca la información necesaria con los botones de navegación.

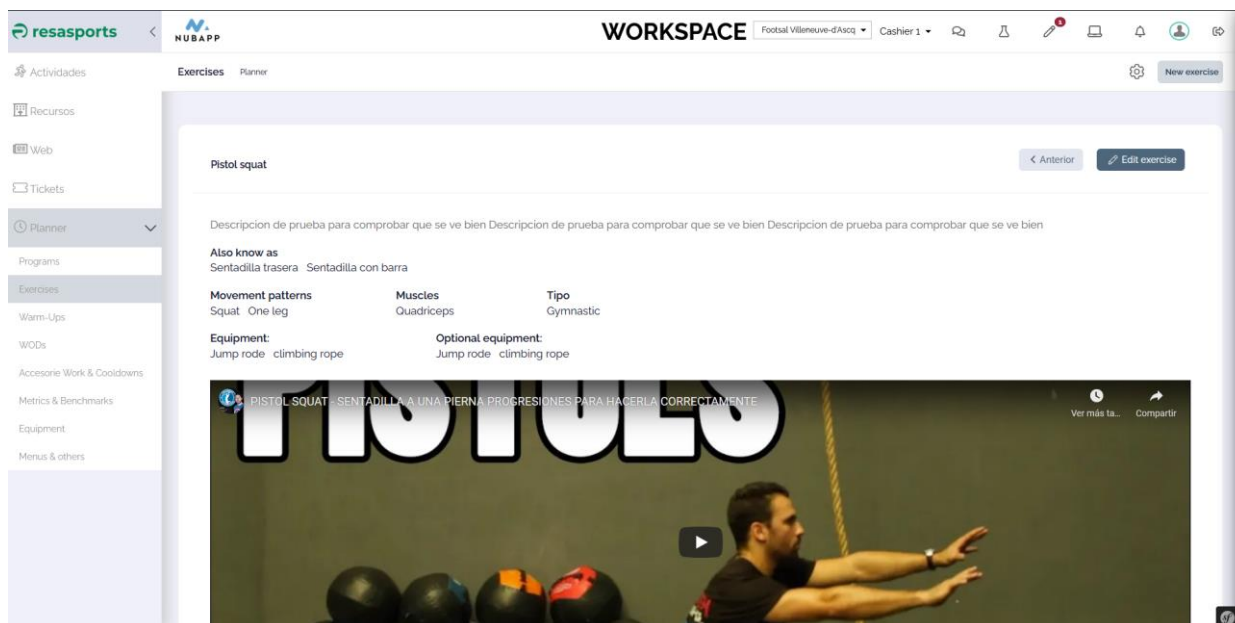


Figura 15. Ejemplo de vista de detalles de un ejercicio

Las vistas de detalles de calentamientos, WODs y enfriamientos son idénticas a la vista de la figura 15, solamente cambiando los campos a mostrar y los botones de navegación que navegan sobre el submódulo en el que se encuentran.

Para acabar con el desarrollo de estos submódulos, se realiza una vista de edición de elemento. Para estas vistas, que se utilizan en todos los submódulos, se realiza otro componente reutilizable llamado EditForm que sigue un poco la misma lógica que el componente de ElementDetails. En EditForm, se diferencia que tipo de elemento se quiere editar, y se muestra un formulario en el que se editan los campos de ese elemento. Se puede observar esta vista en la figura 16.

Workspace | Fútbol Vilanova d'Alcoi | Cashier 1

Editar ejercicio

Nombre: Pistol squat

Descripción: Descripción de prueba para comprobar que se ve bien Descripción de prueba para comprobar que se ve bien Descripción de prueba para comprobar que se ve bien

Asociado a: Sentadilla traza, Sentadilla con barra

Movement pattern: Squat - One leg

Muscle: Quadriceps

Tip: ☒ Conditioning ☒ Gymnastic ☐ Strength ☐ Weightlifting ☐ Nutrition

Equipment: Jump rope, climbing rope

Optional equipment: Jump rope, climbing rope

Video URL: https://www.youtube.com/watch?v=8W0q4p1p0v0

PRELIMINAR - SENTADILLA A UNA PIERNA (PISTOL SQUAT) PARA HACERLA CORRECTAMENTE

Figura 16. Formulario de edición de ejercicios

Las vistas de edición de los submódulos restantes siguen la misma estructura, solamente cambian los campos del formulario. La vista de edición de métricas tiene un añadido al formulario de edición, en ella se muestran los diferentes cálculos de métricas que el usuario tiene, pudiendo añadir o eliminar un cálculo, editarlo o activarlo o desactivarlo. Para ello se creó un componente MetricItem en el que se renderiza una fila con los datos del cálculo de métricas correspondientes.

Workspace | Fútbol Vilanova d'Alcoi | Cashier 1

Editar métrica

Nombre: Record en sentadilla

Descripción: Descripción de la métrica para saber que es

Ejercicios: Back squat

Tip: ☐ Conditioning ☒ Gymnastic ☐ Strength ☐ Weightlifting ☐ Nutrition

Activo	Métrica	Unidad	Eliminar
<input type="checkbox"/>	Altura	Centimeters	
<input checked="" type="checkbox"/>	Reps unbroken a 30cm	Reps	
<input type="checkbox"/>	Resp unbroken a 30cm	Centimeters	
<input checked="" type="checkbox"/>	Reps unbroken a 30cm	Reps	

+ Add metric

Figura 17. Vista de editar métricas con su componente adicional

4.3.4 Modals y pop-ups de metronic

Para una correcta experiencia de usuario y tener un módulo con interacción se ha incorporado al planificador el uso de los Modal que ofrece el framework Bootstrap. Un modal es un diálogo que se abre encima de la vista para interactuar con el usuario y ofrecer una funcionalidad. Para la incorporación de estos Modal, se requiere una instalación del paquete dentro de npm con su requerido argumento de *npm install @bootstrap/modal*.

Los Modal incorporados en el planificador consisten en un diálogo con un formulario para poder crear un nuevo programa, otro diálogo en el que el programa que se está visualizando puede ser asignado a una actividad o un usuario, siendo éste el Modal con mayor complejidad. También se han incorporado diálogos de alerta al querer eliminar un elemento, con un texto de peligro y advertencias de que la acción no se va a poder revertir. Por último, se añade un formulario para que en cada bloque de un día se puedan añadir notas para el entrenador y notas para los atletas que vayan a realizar el ejercicio.

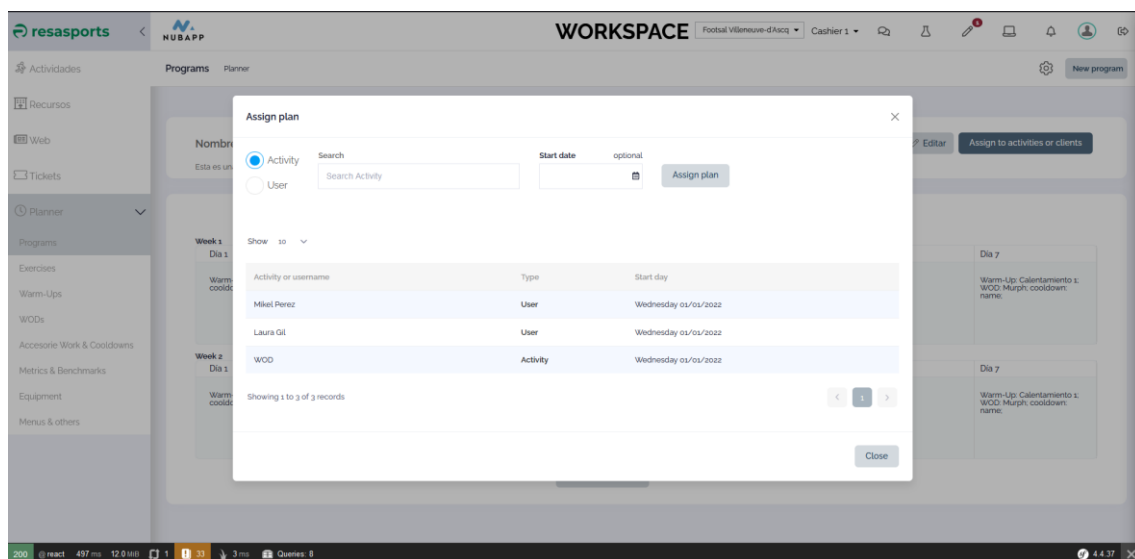


Figura 18 . Ejemplo de un modal. Asignar programa a actividad o usuario.

4.3.5 Componentes reutilizables para todo el panel de administración

Siguiendo los principios de programación que marca el lenguaje React, y lo explicado en el apartado en el que se presenta este lenguaje, el uso de este lenguaje permite la separación de las vistas en componentes para que se puedan reutilizar en

diferentes sitios dentro del proyecto. En la realización de este módulo, se han sacado unos componentes que pueden ser reutilizados en todo el panel de administración de la empresa, con el fin de poder ser utilizados en el proceso de migración de todo el panel a esta nueva tecnología.

En primer lugar, se realizó un componente llamado Card. Este componente renderiza un contenedor para poder introducir en él el contenido que sea necesario. Este componente tiene estilos propios determinados por el tema Metronic y se le han añadido unos estilos adicionales para seguir con el diseño planificado por los diseñadores para este planificador. A este componente, se le pasará por parámetro, es decir, sus props en React, una serie de atributos que corresponden a las posibles oportunidades que ofrece para renderizar en su interior. Por ejemplo, se puede decidir el título, los botones que puede tener en la cabecera, las clases que le puedes añadir para dar estilos propios, y atributos para eliminar elementos como quitar la cabecera, etc.

```
interface cardOptions {
  title?: string;
  toolbarButtons?: ReactElement[];
  classList?: string[];
  id?: string;
  noHeader?: boolean;
  headerClasses?: string[];
  bodyClasses?: string[];
}

export const Card = ({title, toolbarButtons, children, classList = [], id, noHeader, headerClasses, bodyClasses}: React.PropsWithChildren<cardOptions>) => {

  classList?.push("card")
  function printButtons() {
    if (toolbarButtons) {
      return (
        <div className="card-toolbar">
          {toolbarButtons.map(button =>
            button
          )}
        </div>
      )
    }
  }

  return (
    <div className={classList?.join(' ')} id={id}>
      {!noHeader && <div className="card-header " + headerClasses?.join(' ')>
        <div className="card-title">
          {title}
        </div>
        {printButtons()}
      </div>
      <div className="card-body " + bodyClasses?.join(' ')>
        {children}
      </div>
    </div>
  );
}
```

Figura 19 . Código del componente Card.

Seguido al componente Card, se desarrolla el componente Botón. Este componente es muy necesario ya que existen una cantidad de botones dentro del proyecto innumerable, por lo tanto, un componente que renderice los botones siempre de la misma manera y estilo, pudiendo personalizarlos para botones especiales resulta

muy útil. Al componente Botón, se le pasa por props diferentes atributos como las clases necesarias, el icono que puede llevar incorporado, la posición del icono con relación al texto, que tipo de botón es, una función a realizar al hacer clic en él, etc.

```
interface props {
  classes?: string[];
  key?: any;
  id?: string;
  icon?: string;
  onClickFunction?: any;
  title: string;
  type?: "button" | "submit" | "reset";
  noButtonClass?: boolean
  btnRef?: React.RefObject<HTMLButtonElement>
  iconAfter?: boolean
}

export const Button = ({classes = [], key, id, icon, onClickFunction, title, type, btnRef, noButtonClass, iconAfter=false}: props) => {

  icon && classes.push('btn_gap');
  !noButtonClass && classes.push("btn")

  return <button key={key} className={classes?.join(' ')} onClick={onClickFunction} id={id} type={type} ref={btnRef}>
    {icon && !iconAfter && <i className={icon}/>}
    <span>{title}</span>
    {icon && iconAfter && <i className={icon}/>}
  </button>
}
```

Figura 20 . Código del componente Botón.

Continuando con el desarrollo de componentes reutilizables para todo el panel de administración, se determinó que un elemento que se repetía mucho dentro de los formularios que se han creado y todos los existentes ya dentro del proyecto anterior es el elemento input. Para ello, se ha desarrollado un componente reutilizable llamado Input en el que se le puede determinar cualquier propiedad del elemento input de HTML y añadirle clases que determinen el estilo propio si fuese necesario.

```
interface InputProps{
  id?: string
  classes?: string[]
  type?: string
  value?: string
  checked?: boolean
  placeholder?: string
  keyDownFunction?: any
}

export const Input = ({id, classes, type, value, checked, placeholder, keyDownFunction}: InputProps) => {

  return (
    <input type={type} className={classes?.join(' ')} defaultChecked={checked} value={value} id={id} placeholder={placeholder} onKeyDown={keyDownFunction}/>
  )
}
```

Figura 21. Código de componente Input

Por último, y ligado al anterior componente reutilizable de Input, se determinó desarrollar un componente llamado TextInputGroup, que consiste en un Input con uno o dos label encima del input, según sea necesario, que determina el campo del formulario al que estas accediendo con el fin de generar una mejor experiencia de usuario. En este componente, se determinan las clases y el texto de los label, las clases de todo el conjunto

y las propiedades del input anteriormente mencionadas, añadiendo la funcionalidad de si con editables y el contenido predeterminado. Pudiendo añadir también, elementos hijos dentro de este conjunto si fuese necesario.

```
interface InputProps {
  id?: string
  classes: string[]
  editable: boolean
  content: string[] | string
}

interface TextInputGroupProps {
  input: InputProps
  labelText: string
  labelClasses: string[]
  optionalLabel?: boolean
  classes: string[]
  id?: string
}

export const TextInputGroup = ({input, labelText, labelClasses, classes, id, children, optionalLabel=false}: React.PropsWithChildren<TextInputGroupProps>) => {
  const (t) = useTranslation()

  return (
    <div id={id} className={classes.join(' ')}>
      <div id={'labels'} className="flex_item flex-row justify-content-between">
        <label className={labelClasses.join(' ')}>{labelText}</label>
        {optionalLabel && <label className={'optional_label'}>{capitalize(t('optional'))}</label>}
      </div>
      <span className={input.classes.join(' ')} id={input.id} data-kt-autosize={true}
        contentEditable={input.editable} suppressContentEditableWarning={true}>
        {Array.isArray(input.content) ? input.content.map(item => {
          return (
            <span className="margin-right-1rem margin-bottom-halfrem">
              {item}
            </span>
          )
        }) : input.content}
      </span>
      {children}
    </div>
  )
}
```

Figura 22 Código de componente *TextInputGroup*

4.4 Despliegue

Una vez desarrollado el módulo o parte de él, el próximo paso es sacar esa funcionalidad a producción para que los usuarios finales y clientes de la empresa puedan disfrutar de este servicio. Ahora mismo el producto se encuentra en fase de test a la espera de todo el desarrollo del backend correspondiente. En esta fase de test, el código desarrollado se encuentra en un servidor de pruebas de la empresa, alojado en un servidor de Amazon Web Services. Este servidor, llamado Sportdev, es una réplica del servidor de producción, al que accede el cliente final, y en él se realizan las pruebas y las demostraciones pertinentes al equipo para su posterior aprobación. Al estar el planificador a la espera del desarrollo del backend, se encuentra alojado en este servidor y por el momento el cliente final no tiene acceso a esta nueva funcionalidad. Se plantea que el producto salga al mercado a finales del mes de junio o en el mes de julio como muy tarde, cuando todo el código esté desarrollado y perfectamente testeado. En ese

momento, y cuando se hayan realizado todas las pruebas y demostraciones, el código se subirá a la rama de producción de GIT dentro del proyecto de la empresa, y desde ese momento cuando se haga una actualización de las aplicaciones, el planificador estará disponible para los clientes.

5 Conclusiones

El módulo desarrollado siguiendo las técnicas de la metodología ágil Scrum y con los lenguajes de desarrollo más actuales y modernos ofrece una funcionalidad que no se encuentra en el mercado en este nivel de negocio.

Los lenguajes utilizados ofrecen una velocidad de respuesta y tiempo de carga óptimos para que el usuario tenga una experiencia favorable. El tema utilizado para la vista de los componentes es amigable a la par que intuitivo, con el fin de mejorar la experiencia de usuario mencionada anteriormente. La metodología utilizada para el desarrollo, ha evitado que los problemas que han surgido durante la realización del módulo, al haber sido previstos y tener la capacidad de adaptación sobre ellos, alarguen los plazos previstos para terminar el producto, estando a la espera solamente de la parte de backend para que el producto esté totalmente operativo.

El producto, aporta a la empresa un nuevo servicio que ofrecer y por lo tanto la posibilidad de que más clientes quieran contratar sus servicios. El objetivo de la empresa es tener el producto para poder mostrarlo en una convención sobre el deporte y el CrossFit en Francia a finales de junio, país en el que el CrossFit es uno de los deportes con mayor popularidad.

La salida al mercado de este nuevo producto tiene como fin una expansión internacional de mayor calibre a la que tiene la empresa actualmente, llamando la atención de centros deportivos con más funcionalidad y deportes aparte del CrossFit, por ello se añadió la funcionalidad de poder planificar dietas y otros posibles planes que puedan dividirse semanalmente. Añadida a esta expansión internacional, y con la mejora de sus servicios en cuanto al planificador, la empresa pretende sacar ventaja a sus principales competidores en cuanto al software de gestión de centros deportivos, sacando esta nueva funcionalidad innovadora dentro del mercado.

6 Trabajos futuros

Una vez desarrollado el planificador completamente, con todo el frontend y el backend testeado y operativo, surgen varias tareas a realizar.

En primer lugar, se pretende una migración de todo el frontend del panel de administración de la empresa a estas nuevas tecnologías. Ahora mismo se encuentra realizado con técnicas de programación antiguas, de hace más o menos diez años, y los tiempos de carga y la experiencia de usuario que ofrece no son óptimas, por ello, se pretende realizar esta migración de una manera progresiva, adaptando al mismo tiempo el backend a unas nuevas tecnologías.

Seguida a esta migración del frontend, se pretende una migración del backend a nuevas tecnologías. El backend del panel actualmente se encuentra también con tecnologías antiguas, con diversas llamadas AJAX para cualquier interacción que ralentiza todo el proceso. Por ello, otra tarea a realizar en un futuro es cambiar todas esas llamadas y crear controladores para los casos de uso requeridos, siguiendo el patrón MVC, modelo, vista, controlador, que es muy eficaz en el ámbito del desarrollo web.

Por último, como objetivo a largo plazo se pretende sacar la funcionalidad del nuevo planificador en las aplicaciones móviles que ofrece. Se sacará una versión del planificador para la aplicación de administradores de centros, en la que puedan crear, borrar, editar y asignar los programas que crean necesarios, y una versión del planificador para los usuarios, en el que puedan ver sus programas asignados y poder controlar sus métricas y sus planes.

7 Bibliografía

1. Amazon Web Services, Inc. (Ed.) (2022). *Amazon EC2. Capacidad de computación segura y de tamaño ajustable para prácticamente cualquier carga de trabajo*. https://aws.amazon.com/es/ec2/?trk=2d5aad89-991b-4184-98b5-1f562e3102c8&sc_channel=ps&sc_campaign=acquisition&sc_medium=ACQ-P/PS-GO/Brand/Desktop/SU/Core-Main/Core/ES/ES/Text&skwid=AL!4422!3!561218200782!p!!q!!aws&ef_id=Cj0KCQjwyYKUBhDJARIsAMj9lkGWc4zpwRdWLQGNKYOCW46Sm19PxPOIJRfkv7C8AAi843wYx_pj/EaAmQyEALw_wcB:G:s&skwid=AL!4422!3!561218200782!p!!q!!aws
2. Arufe, F. (10/11/2021). CrossFit: el deporte que a su vez es marca y su modelo de negocio. *Economía 3*. <https://economia3.com/2021/10/21/464195-crossfit-el-deporte-que-a-su-vez-es-marca-y-su-modelo-de-negocio/>
3. Astigarraga, J., Cruz-Alonso, V. 2022. ¿Se puede entender cómo funcionan Git y GitHub! Ecosistemas 31(1): 2332. <https://doi.org/10.7818/ECOS.2332>
4. Atlassian (Ed.) (2022). *Jira Software vs. Azure DevOps (Formerly VSTS)*. https://www.atlassian.com/software/jira/comparison/jira-vs-azure-devops?&aceid=&adposition=&adgroup=93058433740&campaign=9124878438&creative=524412163862&device=c&keyword=jira&matchtype=e&network=g&placement=&ds_kids=p51242176358&ds_e=GOOGLE&ds_eid=70000001558501&ds_e1=GOOGLE&gclid=Cj0KCQjw1N2TBhCOARIsAGVHQc4xgqhspVOjC5PWXy37AqdksvzAgJNVWk0U--kUD21e40hkDFE_64MaAliUEALw_wcB&gclid=aw.ds
5. Atlassian Bitbucket (Ed.) (2022) *Tutorials. Guardar cambios*. <https://www.atlassian.com/es/git/tutorials/saving-changes>

6. Aula de software libre (Ed.) (2021). *Flujo de trabajo con Git (git flow)*. La importancia de la organización del flujo de trabajo. <https://aulasoftwarelibre.github.io/taller-de-git/gitflow/>
7. Beck, M. y Tang, T. (2016). Exercise addiction in CrossFit: Prevalence and psychometric properties of the Exercise Addiction Inventory. *Addictive Behaviors Reports*, 3, 33-37. <http://dx.doi.org/10.1016/j.abrep.2016.02.002>
8. Blischak JD, Davenport ER, Wilson G (2016) Introducción rápida al control de versiones con Git y GitHub. *PLoS Comput Biol* 12(1): e1004668. <https://doi.org/10.1371/journal.pcbi.1004668>
9. Clinica Miguel Medina (Ed.) (2019). *La importancia de la nutrición en el deporte*. <https://novasalutclinica.com/es/post/importancia-de-la-nutricion-en-el-deporte>
10. CrossFit, LLC (2022). *About the games*. <https://games.crossfit.com/about-the-games>
11. CrossFit, LLC (Ed.) (2022). *¿What is crossfit?* <https://www.crossfit.com/what-is-crossfit/>
12. Deyimar, A. (2021). (8 de julio de 2021) ¿Qué es npm? Una introducción básica para principiantes. *Hostinger tutoriales*. <https://www.hostinger.es/tutoriales/que-es-npm>
13. Docker Inc. (Ed.) (2022). *Use containers to Build, Share and Run your applications*. <https://www.docker.com/resources/what-container/>
14. Frías, JM., Cantero, J. y Álvarez, I. (2020). El crossfit como nueva tendencia deportiva. En Teva, MR y García, M. (Coord.) *Innovación y nuevas tendencias en el ámbito de la actividad física*. (p 21-34). Wanceulen editorial.
15. GIT Software Freedom Conservacy (Ed.) (2022). *Git –distributed-is-the-new-centralized*. <https://git-scm.com/about>
16. Glassman, G. (2022). (1 de noviembre de 2002). ¿Qué es el fitness y quien está en forma?. *CrossFit*. <https://journal.crossfit.com/article/what-is-fitness-espanol>
17. JavaScript (Ed.) (11 de mayo de 2022). En Wikipedia. <https://es.wikipedia.org/wiki/JavaScript>

18. Keenthemes. (Ed.) (2022). *Rapidly build modern web with metronic*.
<https://preview.keenthemes.com/metronic8/demo1/documentation/getting-started.html>
19. Mdn web docs (Ed.) (2021). *¿Qué es JavaScript?*
https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript#una_definici%C3%B3n_de_alto_nivel
20. Mdn web docs. (Ed.) (2022). *¿Qué es el CSS?*.
https://developer.mozilla.org/es/docs/Learn/CSS/First_steps/What_is_CSS
21. Meta Platforms, Inc. (Ed.) (2022). *Presentando JSX*.
<https://es.reactjs.org/docs/introducing-jsx.html>
22. Meta Platforms, Inc. (Ed.) (2022). *React*. <https://es.reactjs.org/>
23. Microsoft. (Ed.) (2022). *TypeScript es JavaScript con sintaxis para tipos*.
[https://www.typescriptlang-
org.translate.google/? x tr sl=en& x tr tl=es& x tr hl=es& x tr pto=sc](https://www.typescriptlang.org.translate.google/?x_tr_sl=en&x_tr_tl=es&x_tr_hl=es&x_tr_pto=sc)
24. Newcomb, T. (2020). Nobull ha llevado los zapatos de entrenamiento a las calles y senderos en un crecimiento cada vez mayor. *Forbes*. [https://www-
forbes-com.translate.google/sites/timnewcomb/2020/11/05/nobull-has-
taken-training-shoes-to-the-streets-trails-in-ever-expanding-
growth/? x tr sl=en& x tr tl=es& x tr hl=es& x tr pto=op,sc](https://www-forbes-com.translate.google/sites/timnewcomb/2020/11/05/nobull-has-taken-training-shoes-to-the-streets-trails-in-ever-expanding-growth/?x_tr_sl=en&x_tr_tl=es&x_tr_hl=es&x_tr_pto=op,sc)
25. Nubapp S.L (2022). *Entrena, nosotros cuidamos de tu gimnasio*.
<https://resawod.com/es/software/>
26. Organización Mundial de la Salud (Ed.) (2020). *Actividad física*.
<https://www.who.int/es/news-room/fact-sheets/detail/physical-activity>
27. Ortiz, A. (2020). *Tema 2. Scrum. 2.2 Entendiendo el proceso*.
28. Queiron CrossFit. (Ed.) (2022). *Wod*. [https://www.queironcrossfit.com/wod-
blog/](https://www.queironcrossfit.com/wod-blog/)
29. Ramos, R. (2021). *¿Qué es JavaScript y para qué sirve?* *Agencia de marketing Rafa Ramos*. <https://soyrafamos.com/que-es-javascript-para-que-sirve/>
30. Red Hat, Inc. (Ed.) (2022). *What is docker?*
<https://opensource.com/resources/what-docker>

31. Reebok. (30/01/2022). En *Wikipedia*.
https://es.wikipedia.org/wiki/Reebok#Adquisici%C3%B3n_por_parte_de_Aldas
32. Reebok. (Ed.) (2022). *Ropa cross-training*. <https://www.reebok.es/mallas-run-essentials-allover-print-3-4/GJ6163.html>
33. Salvatierra, G. (2014). *Estudio del nuevo fenómeno deportivo crossfit* [Trabajo de Fin de Grado]. Universidad de León.
https://buleria.unileon.es/bitstream/handle/10612/4185/8_SALVATIERRA_CAYETANO_GORKA_DICIEMBRE_2014.pdf?sequence=1&isAllowed=y
34. Schwaber, K. y Sutherland, J. (2020). *La guía Scrum. La guía definitiva Scrum: Las reglas del juego*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
35. Sirotko, A. (2022). (18 de febrero de 2022). What is React? *Flatlogic*.
<https://flatlogic.com/blog/what-is-react/>
36. StudioIP, LLC. (2022) *Wodzilla, workout tracker*
<https://www.wodzillapp.com/>
37. The PHP group. (Ed.) (2022). *¿Qué es PHP?*
<https://www.php.net/manual/es/intro-what-is.php>
38. Velites. (2019). *Temporizador y cronómetro para tus WODS*.
<https://academy.velitessport.com/timer-app/>
39. Wilson, J.M.; Marin, P.J.; Rhea, M.R.; Wilson, S.M.; Loenneke, J. P.; y Anderson, J. C. (2012). Concurrent training: a meta-analysis examining interference of aerobic and resistance exercises. *The Journal of Strength & Conditioning Research*, 26(8), 2293-2307.