

LIFTR

Fitness and food tracker

Sergio Alba Alcalde
Ing. Informática (Ing. Software)
Desarrollo Web

Gregorio Robles Martínez
01/2021



Esta obra está sujeta a una licencia de Reconocimiento-
NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	LIFTR – Fitness and food tracker
Nombre del autor:	Sergio Alba Alcalde
Nombre del consultor:	Gregorio Robles Martínez
Fecha de entrega (mm/aaaa):	01/2021
Área del Trabajo Final:	Desarrollo Web
Titulación:	<i>Ing. Informática</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>El estilo de vida de hoy en día conlleva un horario apretado, entre el trabajo, estudios y otras actividades que requieren nuestra completa atención, es fácil perder de vista la comida que hemos ingerido o cuánto ejercicio estamos haciendo realmente. Es por este motivo que mantener un diario de nuestro progreso es tan importante, porque ayuda a alcanzar objetivos de manera consistente y sube la moral.</p> <p>LIFTR fue creado precisamente con este propósito. Incluye diarios de alimentos y ejercicios, seguimiento de objetivos y todas las herramientas necesarias para tener control absoluto sobre tu progreso. LIFTR funciona en cualquier dispositivo gracias al <i>responsive design</i> y tiene una interfaz simple y fácil de usar.</p> <p>La aplicación fue desarrollada con ReactJS, NodeJS, Express y MongoDB. El despliegue fue realizado en Azure, específicamente bajo una instancia de <i>App Service</i>, integrada con Azure DevOps para implementar CI/CD.</p>	

Abstract (in English, 250 words or less):

Today's lifestyle brings a challenging schedule, with work, studies and many other activities requiring our complete attention, it is easy to lose track of what we may have eaten or how much exercise we are really doing. This is why logging progress is so important, it helps reach goals consistently, and keeps our morale up.

LIFTR was created exactly for this purpose. It includes food and exercise diaries, goal tracking and all the tools you need to be on top of your fitness progress. LIFTR runs on any device thanks to its responsive design and has a simple and easy to use interface.

The application was developed with ReactJS, NodeJS, Express and MongoDB. The deployment was done on Azure, specifically under an App Service instance, integrated with Azure DevOps to implement CI/CD.

Palabras clave (entre 4 y 8):

Fitness, goals, tracker, MERN, React, MongoDB, Express, Azure

Índice

1. INTRODUCCIÓN	7
1.1 Contexto y justificación del Trabajo	7
1.2 Objetivos del Trabajo	8
1.3 Enfoque y método seguido	8
1.4 Planificación del Trabajo	8
1.4.1 PEC 1	9
1.4.2 PEC 2	9
1.4.3 PEC 3	9
1.4.4 PEC 4	9
1.5 Breve resumen de productos obtenidos	9
1.6 Breve descripción de los otros capítulos de la memoria	10
2. EVALUACIÓN DE RIESGOS	11
3. ANÁLISIS Y REQUISITOS DEL PROYECTO	13
3.1 Descripción del sistema	13
3.2 Requisitos	13
3.2.1 Identificación de usuarios	13
3.2.2 Requisitos de usuarios	13
3.3 Casos de uso	15
3.4 Arquitectura global del sistema	27
3.5 Punto de vista de la información	28
3.6 Punto de vista de la computación	29
4. DISEÑO DE LA APLICACIÓN	31
4.2 Wireframes	31
4.2 Navegación	37
4.3 Elementos de diseño	38
4.3.1 Responsive design	38
4.3.2 Elementos	38
5. IMPLEMENTACIÓN	41
5.1. Organización	41

5.2. Control de versiones	42
5.3. Entorno de desarrollo	43
5.4. Implementación de las partes críticas	44
5.4.1 Autenticación / autorización	44
5.4.2 Estado, “props” y su propagación	46
5.4.3 Diario de ingesta y de ejercicios	48
5.4.4 Objetivos	49
5.4.5 Control de errores y validación de formularios	53
6. PRUEBAS	55
7. PROGRESO CON RESPECTO A LA PLANIFICACIÓN	63
7.1. PEC 2	63
7.2. PEC 3	63
7.3. PEC 4	63
8. FUTURO	64
9. CONCLUSIONES	65
10. GLOSARIO	66
11. BIBLIOGRAFÍA	67

Lista de figuras

Ilustración 1	9
Ilustración 2	9
Ilustración 3	9
Ilustración 4	9
Ilustración 5	27
Ilustración 6	28
Ilustración 7	29
Ilustración 8	30
Ilustración 9 – Vista de Login	31
Ilustración 10 – Vista de registro	31
Ilustración 11 – Vista <i>Home</i>	32
Ilustración 12 – Vista <i>Diary</i>	32
Ilustración 13 - Vista búsqueda ejercicios	33
Ilustración 14 - Vista de búsqueda de alimentos (resultado)	33
Ilustración 15 - Vista Workout	34
Ilustración 16 - Vista de creación de rutinas	34
Ilustración 17 - Vista de búsqueda de ejercicios	35
Ilustración 18 - Vista de <i>Goals</i>	35
Ilustración 19 - Nuevo Objetivo	36
Ilustración 20	37
Ilustración 21	38
Ilustración 22	39
Ilustración 23	40
Ilustración 24	41
Ilustración 25	42
Ilustración 26	43
Ilustración 27	44
Ilustración 28 – Estado y Props en componentes	46
Ilustración 29 – Workout Session, Timer	47
Ilustración 30 – Vista principal y del diario de ejercicios	49
Ilustración 31 – Formularios de adición y edición	50
Ilustración 32 - Vista de objetivos	51
Ilustración 33 - Vista de error 500	53
Ilustración 34 - Validación de formularios	54
Tabla 1	11
Tabla 2	11
Tabla 3	14
Tabla 4	14
Tabla 5	14
Tabla 6	14
Tabla 7	14
Tabla 8	14
Tabla 9	14
Tabla 10	14
Tabla 11	15
Tabla 12	15
Tabla 13	15
Tabla 14	15

Tabla 15	15
Tabla 16	15
Tabla 17	16
Tabla 18	16
Tabla 19	17
Tabla 20	17
Tabla 21	17
Tabla 22	18
Tabla 23	18
Tabla 24	18
Tabla 25	19
Tabla 26	20
Tabla 27	20
Tabla 28	20
Tabla 29	21
Tabla 30	22
Tabla 31	22
Tabla 32	22
Tabla 33	23
Tabla 34	24
Tabla 35	24
Tabla 36	25
Tabla 37	25
Tabla 38	26
Tabla 39	26
Tabla 40	55
Tabla 41	55
Tabla 42	55
Tabla 43	55
Tabla 44	56
Tabla 45	56
Tabla 46	57
Tabla 47	57
Tabla 48	57
Tabla 49	57
Tabla 50	58
Tabla 51	58
Tabla 52	59
Tabla 53	59
Tabla 54	60
Tabla 55	60
Tabla 56	61
Tabla 57	61
Tabla 58	61
Tabla 59	61
Tabla 60	62
Tabla 61	62

1. Introducción

1.1 Contexto y justificación del Trabajo

Se propone la realización de una aplicación web especializada en el seguimiento de la dieta y el ejercicio. La aplicación utiliza datos introducidos por el usuario para generar recomendaciones asociadas a la ingesta de calorías y ejercicios o rutinas. Se pondrán a disposición también herramientas que permiten la creación de rutinas personalizadas. También ofrece un apartado de novedades y noticias.

La aplicación web será una SPA (*Single Page Application*), desarrollada con el popular *framework* de *front end* React. Una SPA es una aplicación que, dentro del navegador, no requiere de recargar la página cuando se interactúa con un botón, un link, o cuando se completa un formulario. Muchas aplicaciones hoy en día son SPAs, donde Gmail, Github o Facebook son buenos ejemplos. Este último es precisamente el desarrollador del *framework* de interfaz utilizado en este proyecto, React. Una diferencia fundamental de estas aplicaciones con aplicaciones multi-página es que únicamente se intercambian datos a lo largo de una sesión de uso, es decir, que el servidor no tiene que encargarse de producir HTML y enviarlo al usuario. La implicación principal de que una aplicación web sea una SPA es que son renderizadas en el lado del cliente, en lugar de en el servidor (multi-página), quitando cierta carga al servidor web.

React es un *framework* de gestión de componentes HTML en JavaScript que implementa una filosofía de encapsulamiento de elementos de la página en “componentes”. Estos componentes se programan en JavaScript y son, en definitiva, funciones que devuelven HTML.

Un aspecto importante será también que la implementación de la aplicación permita a un usuario utilizar el servicio incluso cuando este no disponga de acceso a internet. Para alcanzar este objetivo, se plantea que la aplicación sea una PWA, o *progressive web application*, que, mediante un *middleware* en el navegador, permite el “cacheo” de los elementos de la aplicación e incluso llamadas a la API realizadas anteriormente. Este *middleware* se denomina *Service Worker* y es un componente implementado en los propios navegadores, y especificado por el desarrollador de la aplicación, que permite que una aplicación web actúe de manera similar a una aplicación nativa, con la capacidad de aparecer entre las aplicaciones instaladas en dispositivos móviles, o como un programa nativo en Windows o OSX.

1.2 Objetivos del Trabajo

Implementación de una aplicación web que permita el seguimiento de la ingesta calórica y actividad física, así como la planificación de objetivos y metas.

Sub-objetivos

- Ofrecer herramientas de seguimiento y planificación de ingesta calórica y de ejercicio.
- Proporcionar los conocimientos necesarios para planificar una dieta y ejercicios, saludables y seguros.
- Proveer al usuario de una experiencia sencilla e intuitiva.
- Motivar al usuario a conseguir y cumplir con sus objetivos.

1.3 Enfoque y método seguido

Desde un principio se ha decidido crear un nuevo producto desde cero. El método de gestión del proyecto sigue el marco del desarrollo en cascada, con fases bien definidas, donde la fase actual depende de la anterior. Con lo que respecta a la metodología de gestión de tareas, se ha optado por el uso de una herramienta de organización de tareas (Microsoft To-Do), basada en las pautas establecidas por la planificación temporal (Microsoft Project).

1.4 Planificación del Trabajo

Para la gestión temporal del proyecto se ha empleado Microsoft Project, que ofrece la sencilla creación de líneas temporales y diagramas de Gantt. Estas funcionalidades permiten la obtención de visibilidad global sobre las fases del proyecto y su progreso.

Por otro lado, se ha recurrido al uso de Microsoft To-Do para el seguimiento, priorización y cumplimiento de tareas a lo largo de la implementación de la aplicación.

1.4.1 PEC 1



Ilustración 1

1.4.2 PEC 2



Ilustración 2

1.4.3 PEC 3

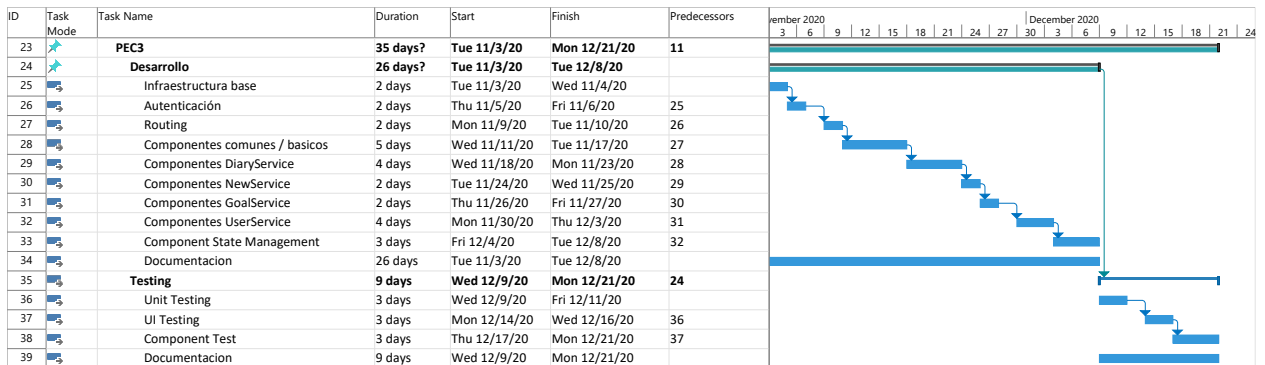


Ilustración 3

1.4.4 PEC 4



Ilustración 4

1.5 Breve resumen de productos obtenidos

De todo el proceso descrito a continuación, se han obtenido los siguientes productos:

Repositorio – con el código fuente de la aplicación ([link](#))

Aplicación LIFTR – Con las funcionalidades, estructura, tecnologías y estilos descritos en los próximos apartados.

1.6 Breve descripción de los otros capítulos de la memoria

Los capítulos a continuación están ordenados de manera temporal a lo largo del proceso de desarrollo.

Capítulos 2 - 4

Se abre con una exposición del análisis, requisitos y riesgos del proyecto, justificando y argumentando las decisiones de diseño tomadas. Así mismo, se presentan los roles de usuario identificados, sus casos de uso detallados, y los requisitos funcionales y no funcionales de la aplicación. Estos capítulos incluyen también diagramas UML de los modelos de datos, estilos o la arquitectura del sistema.

Capítulos 5 - 6

Corresponden a la fase de implementación, que detalla cómo se han desarrollado los componentes críticos y las decisiones de diseño que se han debido tomar para cumplir con los requisitos de la fase anterior. A continuación, se documentan las pruebas realizadas tras finalizar la fase de desarrollo.

Capítulos 7 - 8

Se cierra la memoria con reflexiones acerca de cómo extender las funcionalidades del producto de cara al futuro y unas conclusiones.

2. Evaluación de riesgos

Existe un gran número de factores que pueden poner en riesgo la entrega de un proyecto. Es por este motivo que es importante realizar un análisis de los riesgos para saber cuáles son, cómo pueden afectar al proyecto y cómo podemos remediarlos.

En primer lugar, se identifican, analizan y cuantifican los riesgos que pueden afectar al desarrollo del proyecto:

Tabla 1

ID	Riesgo	Tipo	Probabilidad	Impacto
1	Bugs o errores	Proyecto	Alta	Medio / Bajo
2	Baja médica	Personal	Baja	Medio / Alto
3	El mal funcionamiento de herramientas de trabajo	Técnico	Baja	Medio
4	Baja del servicio del proveedor	Técnico	Baja	Alto
5	Error de cálculo de tiempo en la planificación	Proyecto	Media	Medio
6	Falta de conocimientos	Personal	Media	Medio
7	Identificación de requisitos errónea	Proyecto	Media	Medio
8	Falta de usabilidad / calidad del producto	Producto	Media	Alto
9	Pruebas insuficientes	Proyecto	Baja	Medio

Se aplicará una estrategia proactiva de gestión, donde se busca realizar un estudio previo de los riesgos, se mide la posibilidad de que suceda y se hace una estimación del impacto que pueden tener sobre el proyecto. A partir de este análisis podemos delinear una serie de puntos clave para la correcta mitigación de los riesgos identificados hasta ahora. También se asociará el punto clave con el riesgo asociado o relacionado.

Tabla 2

ID	ID Riesgo	Descripción
1	1	Se utilizarán patrones de diseño y una estrategia de entrega basada en DevOps para la minimización de <i>bugs</i> o errores que puedan aparecer de manera inesperada.
2	1	Se implementarán <i>tests</i> para comprobar el correcto funcionamiento de los componentes y <i>endpoints</i> de la aplicación
3	2	Se cumplirá con las normas sanitarias vigentes para minimizar el riesgo de exposición al COVID19.
4	3	Todo el trabajo quedará almacenado en repositorios de código y almacenamiento en la nube. En caso de un mal funcionamiento no habrá pérdida de datos.
5	4	Todo el desarrollo se llevará a cabo en local, de manera que no se dependa del proveedor para realizar el proyecto.
6	5	Se harán revisiones periódicas de las tareas programadas para revisar los tiempos de actuación y modificarlos.

7	5	Se realizará una evaluación previa de los costes de tiempo de todas las tareas, tanto de desarrollo como de gestión, basadas en experiencias pasadas y el asesoramiento del tutor.
8	6	Se hará una investigación previa y durante la duración del proyecto, para aumentar mis conocimientos en las tecnologías y las destrezas relacionadas con la realización del proyecto.
9	7, 8	Se realizarán actuaciones de <i>testeo</i> con usuarios en la sección demográfica objetivo para evaluar los requisitos recopilados.
10	9	Se hará una evaluación previa y continúa del cumplimiento de los estándares definidos en la planificación, así como de los requisitos recopilados.

3. Análisis y requisitos del proyecto

3.1 Descripción del sistema

La aplicación web propuesta parte de la base de ser el referente a la hora de almacenar información sobre las actividades relacionadas con el *fitness* y la halterofilia. Para alcanzar este objetivo, se pueden agrupar las funcionalidades en 3 categorías: **nutrición**, **ejercicio** e **información**. Cada una de estas categorías busca el cubrir las necesidades de los usuarios en el seguimiento diario de sus objetivos, metas, así como recopilar datos para ofrecer información útil al usuario.

La categoría de **nutrición** recoge todas las acciones y datos asociados con el diario de ingesta calórica, esto comprende el introducir nuevos datos (alimentos), procesarlos y representarlos al usuario de manera que aporte un valor significativo. En cuanto a los **ejercicios**, de estos se ofrece una amplia selección de ejercicios aeróbicos y anaeróbicos, para cubrir un amplio espectro de posibles rutinas de ejercicio. De esta selección, se deberán recopilar los datos relevantes según el tipo de ejercicio, donde para un ejercicio aeróbico nos interesan valores como el tiempo, mientras que, para aquellos anaeróbicos nos interesan datos como las repeticiones realizadas y el peso (si aplica).

En la categoría de la **información** se incluyen artículos y guías que ofrecen al usuario conocimientos y novedades sobre diversas facetas del deporte, de manera que habrá que crear las estructuras de datos, vistas y acciones asociadas a la vista de artículos.

En base a estos requisitos, se describe a continuación la especificación del sistema.

3.2 Requisitos

A continuación, se exponen los requisitos, identificados según los intereses de los stakeholders, y su descripción.

3.2.1 Identificación de usuarios

Se identifican dos tipos de usuarios en el sistema:

- **Usuarios no registrados:** Estos usuarios son los que llegan por primera vez a la aplicación y no tienen una cuenta. Se les incentivará a registrarse.
- **Usuarios registrados:** Estos usuarios estarán dados de alta en la aplicación y se identificarán con sus credenciales. Una vez autenticados, pueden empezar a hacer consultas y utilizar las funcionalidades de la aplicación.

3.2.2 Requisitos de usuarios

A partir de los usuarios identificados anteriormente, se identifican los requisitos funcionales, que especifican cómo debe actuar el sistema ante ciertos insumos. Estos insumos pueden ser interacciones de usuario o de otros sistemas externos.

Requisitos funcionales de usuarios no registrados

Tabla 3

ID	RF-UNR01	Prioridad	Media
Tipo	Funcional	Descripción	Cualquier usuario podrá darse de alta en la aplicación desde el portal.

Requisitos funcionales de usuarios registrados

Tabla 4

ID	RF-UR01	Prioridad	Baja
Tipo	Funcional	Descripción	Cualquier usuario visitante que se detecte que haya visitado anteriormente la página, será redirigido directamente al <i>login</i> .

Tabla 5

ID	RF-UR02	Prioridad	Alta
Tipo	Funcional	Descripción	Cualquier usuario registrado podrá introducir datos sobre su ingesta de alimentos en su diario personal. Esto puede ser uno o más alimentos / recetas.

Tabla 6

ID	RF-UR03	Prioridad	Alta
Tipo	Funcional	Descripción	Cualquier usuario registrado podrá actualizar su perfil de usuario con sus datos biométricos.

Tabla 7

ID	RF-UR04	Prioridad	Alta
Tipo	Funcional	Descripción	Cualquier usuario registrado podrá acceder a la sección de artículos y contenido informativo.

Tabla 8

ID	RF-UR05	Prioridad	Alta
Tipo	Funcional	Descripción	Cualquier usuario con los datos biométricos almacenados podrá acceder a calcular recomendaciones de índices como el BMI o ingesta calórica recomendada.

Tabla 9

ID	RF-UR06	Prioridad	Alta
Tipo	Funcional	Descripción	Cualquier usuario registrado podrá acceder a su historial de ingesta calórica registrada.

Tabla 10

ID	RF-UR07	Prioridad	Alta
Tipo	Funcional	Descripción	Cualquier usuario registrado podrá acceder a su historial de rutinas y ejercicios realizados.

Tabla 11

ID	RF-UR08	Prioridad	Alta
Tipo	Funcional	Descripción	Cualquier usuario registrado podrá hacer <i>login</i> y <i>logout</i> en el portal.

Tabla 12

ID	RF-UNR09	Prioridad	Alta
Tipo	Funcional	Descripción	Cualquier usuario registrado podrá acceder a la búsqueda de la base de datos de alimentos desde el portal.

Tabla 13

ID	RF-UNR10	Prioridad	Alta
Tipo	Funcional	Descripción	Cualquier usuario registrado podrá acceder a la base de datos de ejercicios disponibles desde el portal.

Requisitos no funcionales

Los requisitos no funcionales refieren a las propiedades del sistema. Estos requisitos no describen el comportamiento del sistema en cuanto a el “qué” hace, sino que describen el “cómo”, tratando aspectos como rendimiento, seguridad u otras características.

Tabla 14

ID	RNF-001	Prioridad	Alta
Tipo	No Funcional	Descripción	La aplicación debe utilizar cifrado de las comunicaciones.

Tabla 15

ID	RNF-002	Prioridad	Alta
Tipo	No Funcional	Descripción	La aplicación no necesitará recargar la página para navegar por ella.

Tabla 16

ID	RNF-003	Prioridad	Alta
Tipo	No Funcional	Descripción	La aplicación debe poder añadirse a la pantalla de inicio de un dispositivo móvil siguiendo la implementación de <i>Service Workers</i> .

3.3 Casos de uso

A continuación, una descripción formal y notablemente exhaustiva de los posibles casos de uso de los distintos actores participantes en la aplicación. En estas descripciones se exponen los sucesos de eventos entre el usuario y el sistema para un determinado caso.

Tabla 17

ID	CU-UR01	Caso de uso	Hacer <i>login</i>
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere acceder a la aplicación con su cuenta	Garantías	En caso de éxito, el usuario accederá al portal
Ámbito	Portal de LIFTR	Precondición	El usuario debe disponer de las credenciales de acceso
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El usuario accede al portal 2. El sistema detecta que es un usuario recurrente 3. Se redirige a la vista de <i>login</i> 4. El usuario introduce sus credenciales 5. El usuario es redirigido a la app 		
Extensiones	<ol style="list-style-type: none"> 2.a.1 El sistema no detecta que es un usuario recurrente 2.a.2 El usuario hará click en <i>login</i> 4.a.1 El usuario introduce credenciales erróneas 4.a.2 El sistema notifica al usuario 4.a.3 El usuario introduce las credenciales correctas 4.b.1 El usuario introduce credenciales erróneas 4.b.2 El sistema notifica al usuario 4.b.3 El usuario usa la opción <i>recuperar contraseña</i> 		

Tabla 18

ID	CU-UNR01	Caso de uso	Registrarse
Nivel de Objetivo	Usuario	Actor Principal	Usuario no Registrado (UNR)
Stakeholders e intereses	UNR: Quiere darse de alta en la plataforma	Garantías	En caso de éxito, el usuario estará dado de alta
Ámbito	Portal de LIFTR	Precondición	N/A
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El cliente accede al portal 2. Se muestra la vista de <i>Login</i> 3. El usuario hace click en <i>registrar</i> 4. El usuario rellena los campos del formulario de registro 5. Se verifican los valores de los campos 6. Se da de alta al usuario 7. El usuario es notificado 8. El sistema redirige al usuario a la página de <i>Login</i> 		
Extensiones	<ol style="list-style-type: none"> 4.a.1 El usuario ha introducido datos no válidos en alguno de los campos 4.a.2 El sistema notifica al usuario acerca de las condiciones el campo erróneo y cuál debe ser su formato 		

4.a.3 El usuario reintroduce los datos correctamente

Tabla 19

ID	CU-UR02	Caso de uso	Acceder a la sección de noticias
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere acceder a la sección de noticias	Garantías	En caso de éxito, el sistema mostrará la vista de noticias
Ámbito	Portal de LIFTR	Precondición	El usuario debe haber hecho <i>login</i>
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario se encuentra en la página principal del portal 2. El usuario navega hasta la sección de noticias 3. El sistema muestra las noticias 	
Extensiones			

Tabla 20

ID	CU-UR03	Caso de uso	Acceder al diario de ingesta
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere acceder a su diario de ingesta calórica	Garantías	En caso de éxito, el sistema mostrará el diario de ingesta del día
Ámbito	Portal de LIFTR	Precondición	El usuario debe haber hecho <i>login</i>
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario se encuentra en la página principal del portal 2. El usuario hace click sobre el botón <i>Diary</i> 3. El usuario es redirigido a la vista del diario de ingesta 	
Extensiones			

Tabla 21

ID	CU-UR04	Caso de uso	Acceder al diario de ejercicios
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere acceder a su diario de ejercicios diario	Garantías	En caso de éxito, el sistema mostrará el diario de ejercicios
Ámbito	Portal de LIFTR	Precondición	El usuario debe haber hecho <i>login</i>
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario se encuentra en la página principal del portal 2. El usuario hace click sobre el botón <i>Workout</i> 3. El usuario es redirigido a la vista del diario de ejercicios 	

Extensiones	
-------------	--

Tabla 22

ID	CU-UR05	Caso de uso	Acceder a los objetivos
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere acceder a sus objetivos fijados	Garantías	En caso de éxito, el sistema mostrará los objetivos del usuario
Ámbito	Portal de LIFTR	Precondición	El usuario debe haber hecho <i>login</i>
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El usuario se encuentra en la página principal del portal 2. El usuario hace click sobre el botón <i>Goals</i> 3. El usuario es redirigido a la vista de <i>Goals</i> 		
Extensiones			

Tabla 23

ID	CU-UR06	Caso de uso	Buscar un alimento
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere encontrar un alimento en particular	Garantías	En caso de éxito, el sistema mostrará el alimento buscado
Ámbito	Portal de LIFTR	Precondición	- El usuario debe haber hecho <i>login</i>
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El usuario se encuentra en la vista del diario de ingesta. 2. El usuario crea una nueva entrada en el diario de ingesta. 3. El sistema muestra la vista de búsqueda de alimentos 4. El usuario introduce el nombre del alimento que busca en la barra de búsqueda 5. El usuario hace click en "buscar" 6. El sistema muestra una lista de resultados 7. El usuario selecciona el resultado adecuado 		
Extensiones			

Tabla 24

ID	CU-UR07	Caso de uso	Añadir una nueva entrada al diario de ingesta
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere añadir una nueva entrada a su diario de ingesta	Garantías	En caso de éxito, el sistema mostrará el diario de ingesta del usuario con la nueva entrada
Ámbito	Portal de LIFTR	Precondición	- El usuario debe haber hecho <i>login</i>

		- El usuario se encuentra en la vista del diario de ingesta
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario hace click sobre el botón “+” para añadir una nueva entrada 2. El sistema muestra un buscador de alimentos 3. El usuario busca el alimento (CU-UR06) y lo selecciona 4. El sistema muestra el formulario para crear la nueva entrada del alimento 5. El usuario rellena los campos 6. El sistema muestra la confirmación 7. El sistema redirige al usuario a la vista del diario de ingesta
Extensiones		

Tabla 25

ID	CU-UR08	Caso de uso	Modificar una entrada del diario de ingesta
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere modificar una entrada de su diario de ingesta	Garantías	En caso de éxito, el sistema mostrará el diario de ingesta del usuario con la entrada modificada
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista del diario de ingesta
Escenario principal de éxito			<ol style="list-style-type: none"> 1. El usuario selecciona el elemento a modificar 2. El sistema muestra el elemento con los datos almacenados 3. El usuario modifica las cantidades o tamaño de porción de la entrada 4. El usuario confirma las modificaciones 5. El sistema muestra la entrada con los nuevos datos
Extensiones			<ol style="list-style-type: none"> 3.a.1 El usuario introduce datos no válidos 3.a.2 El sistema notifica el error 3.a.3 El usuario introduce datos válidos

Tabla 26

ID	CU-UR09	Caso de uso	Eliminar una entrada del diario de ingesta
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere eliminar una entrada de su diario de ingesta	Garantías	En caso de éxito, el sistema mostrará el diario de ingesta del usuario sin la entrada eliminada
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista del diario de ingesta
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario selecciona el elemento a eliminar 2. El sistema elimina los elementos y muestra el diario de ingesta sin los elementos eliminados 	
Extensiones			

Tabla 27

ID	CU-UR10	Caso de uso	Buscar un ejercicio
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere encontrar un ejercicio en particular	Garantías	En caso de éxito, el sistema mostrará el ejercicio buscado
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista del diario de ejercicios
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario se encuentra en la vista del diario de ejercicios 2. El usuario crea una nueva rutina 3. El sistema muestra los campos de búsqueda de ejercicios 4. El usuario introduce el nombre del ejercicio que busca en la barra de búsqueda 5. El usuario hace click en buscar 6. El sistema muestra una lista de resultados 7. El usuario selecciona el resultado adecuado 	
Extensiones			

Tabla 28

ID	CU-UR11	Caso de uso	Añadir una nueva entrada al diario de ejercicios
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere añadir una nueva entrada a	Garantías	En caso de éxito, el sistema mostrará el diario de ejercicios del usuario con la nueva entrada

	su diario de ejercicios		
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista del diario de ejercicios
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario hace click sobre el botón “+” para añadir una nueva entrada 2. El sistema muestra un buscador de ejercicios 3. El usuario busca el alimento (CU-UR08) y lo selecciona 4. El sistema muestra el formulario para crear la nueva entrada del ejercicio 5. El usuario rellena los campos 6. El sistema muestra la confirmación 7. El sistema redirige al usuario a la vista del diario de ejercicios 	
Extensiones			

Tabla 29

ID	CU-UR12	Caso de uso	Modificar una entrada del diario de ejercicios
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere modificar una entrada de su diario de ejercicios	Garantías	En caso de éxito, el sistema mostrará el diario de ejercicios del usuario con la entrada modificada
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista del diario de ejercicios
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario selecciona el elemento a modificar 2. El sistema muestra el elemento con los datos almacenados 3. El usuario modifica el tiempo, repeticiones, peso, u otro valor de la entrada 4. El usuario confirma las modificaciones 5. El sistema muestra la entrada con los nuevos datos 	
Extensiones		<ol style="list-style-type: none"> 3.a.1 El usuario introduce datos no válidos 3.a.2 El sistema notifica el error 3.a.3 El usuario introduce datos válidos 	

Tabla 30

ID	CU-UR13	Caso de uso	Eliminar una entrada del diario de ejercicios
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado (UR)
Stakeholders e intereses	UR: Quiere eliminar una entrada de su diario de ejercicios	Garantías	En caso de éxito, el sistema mostrará el diario de ejercicios del usuario sin la entrada eliminada
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista del diario de ejercicios
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario selecciona el elemento a eliminar 2. El sistema elimina los elementos y muestra el diario de ejercicios sin los elementos eliminados 	
Extensiones			

Tabla 31

ID	CU-UR14	Caso de uso	Consultar los artículos de noticias
Nivel de Objetivo	Usuario	Actor Principal	Usuario (Registrado o no registrado)
Stakeholders e intereses	U: Quiere ver la lista de artículos de noticias	Garantías	En caso de éxito, el sistema mostrará el listado de artículos publicados
Ámbito	Portal de LIFTR	Precondición	El usuario se encuentra en la página principal del portal
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario selecciona el botón de <i>news</i> (noticias) 2. El sistema redirige al usuario a la vista de artículos de noticias 3. El sistema muestra el listado de artículos publicados 	
Extensiones			

Tabla 32

ID	CU-UR15	Caso de uso	Añadir un nuevo objetivo
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado
Stakeholders e intereses	UR: Quiere añadir un nuevo objetivo	Garantías	En caso de éxito, el sistema mostrará el nuevo objetivo en la vista de objetivos
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la página principal del portal
Escenario principal de éxito		<ol style="list-style-type: none"> 1. El usuario selecciona el botón de <i>goals</i> 	

	<ol style="list-style-type: none"> 2. El sistema redirige al usuario a la vista de objetivos 3. El sistema muestra los objetivos (si existen) del usuario 4. El usuario utiliza un botón para añadir un nuevo objetivo 5. El sistema muestra un formulario con los datos necesarios del objetivo 6. El usuario rellena y envía el formulario 7. El sistema verifica que el formulario contiene datos correctos. 8. El sistema redirige al usuario a la vista de objetivos, mostrando el nuevo objetivo añadido
Extensiones	<ol style="list-style-type: none"> 7.a.1 El usuario ha introducido datos no válidos 7.a.2 El sistema muestra un mensaje de error, indicando los campos erróneos 7.a.3 El usuario introduce datos válidos 7.a.4 El sistema valida los datos y añade el nuevo objetivo.

Tabla 33

ID	CU-UR17	Caso de uso	Eliminar un objetivo
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado
Stakeholders e intereses	UR: Quiere eliminar un objetivo	Garantías	En caso de éxito, el sistema mostrará el objetivo en la vista de objetivos, sin el objetivo eliminado
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista de Objetivos
Escenario principal de éxito			<ol style="list-style-type: none"> 1. El sistema muestra los objetivos existentes 2. El usuario selecciona un objetivo 3. El sistema muestra un resumen del estado del objetivo 4. El usuario usa el botón <i>delete</i> 5. El sistema muestra un mensaje de confirmación al usuario 6. El usuario confirma la eliminación 7. El sistema elimina la entrada y muestra un mensaje de confirmación de la operación 8. El sistema redirige al usuario a la vista de objetivos, sin el objetivo eliminado
Extensiones			

Tabla 34

ID	CU-UR17	Caso de uso	Modificar las preferencias de usuario
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado
Stakeholders e intereses	UR: Quiere modificar sus preferencias	Garantías	En caso de éxito, el sistema mostrará las nuevas preferencias del usuario
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista de modificación de preferencias
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El sistema muestra las preferencias actuales 2. El usuario modifica el campo de la preferencia(s) que desea 3. El sistema actualiza las preferencias del usuario 4. El sistema redirige al usuario a la vista <i>Home</i> 		
Extensiones	<ol style="list-style-type: none"> 2.a.1 El usuario introduce datos erróneos o no válidos 2.a.2 El sistema indica los campos erróneos 2.a.3 El usuario rectifica los datos erróneos o no válidos 2.a.4 El usuario envía el formulario 		

Tabla 35

ID	CU-UR18	Caso de uso	Crear una rutina
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado
Stakeholders e intereses	UR: Quiere crear una rutina	Garantías	En caso de éxito, el sistema mostrará el objetivo en la vista de ejercicios con la nueva rutina añadida
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista de ejercicios
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El sistema muestra las rutinas existentes 2. El usuario hace click sobre el botón de añadir rutina 3. El sistema muestra un formulario para que rellene el usuario 4. El usuario rellena y envía el formulario. 5. El sistema verifica los datos del formulario y muestra un mensaje de confirmación. 6. El sistema redirige al usuario a la vista de ejercicios, con la nueva rutina añadida 		
Extensiones	<ol style="list-style-type: none"> 4.a.1 El usuario introduce datos erróneos o no válidos 		

	4.a.2 El sistema indica los campos erróneos 4.a.3 El usuario rectifica los datos erróneos o no válidos 4.a.4 El usuario envía el formulario
--	---

Tabla 36

ID	CU-UR19	Caso de uso	Modificar una rutina
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado
Stakeholders e intereses	UR: Quiere modificar una rutina	Garantías	En caso de éxito, el sistema mostrará el objetivo en la vista de ejercicios con la rutina modificada
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista de ejercicios
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El sistema muestra las rutinas existentes 2. El usuario selecciona una rutina y hace click en el botón de edición 3. El sistema muestra un formulario para que el usuario haga las modificaciones deseadas 4. El usuario rellena y envía el formulario. 5. El sistema verifica los datos del formulario 6. El sistema redirige al usuario a la vista de ejercicios, con la rutina modificada 		
Extensiones	<p>4.a.1 El usuario introduce datos erróneos o no válidos 4.a.2 El sistema indica los campos erróneos 4.a.3 El usuario rectifica los datos erróneos o no válidos 4.a.4 El usuario envía el formulario</p>		

Tabla 37

ID	CU-UR20	Caso de uso	Eliminar una rutina
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado
Stakeholders e intereses	UR: Quiere eliminar una rutina	Garantías	En caso de éxito, el sistema mostrará el objetivo en la vista de ejercicios sin la rutina eliminada
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista de ejercicios
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El sistema muestra las rutinas existentes 2. El usuario selecciona una rutina y hace click sobre el botón de eliminación 3. El sistema elimina la rutina. 		

	4. El sistema redirige al usuario a la vista de ejercicios, sin la rutina eliminada
Extensiones	4.a.1 El usuario cancela la operación

Tabla 38

ID	CU-UR21	Caso de uso	Hacer una sesión
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado
Stakeholders e intereses	UR: Quiere hacer una sesión de una rutina	Garantías	En caso de éxito, el sistema mostrará la vista de <i>Session</i>
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista de ejercicios
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El sistema muestra las rutinas existentes 2. El usuario selecciona una rutina y hace click sobre el botón de ejecución 3. El sistema carga la rutina y muestra la vista de <i>Session</i> 		
Extensiones			

Tabla 39

ID	CU-UR22	Caso de uso	Ver una sesión pasada
Nivel de Objetivo	Usuario	Actor Principal	Usuario Registrado
Stakeholders e intereses	UR: Quiere ver una sesión pasada	Garantías	En caso de éxito, el sistema mostrará los datos de la sesión seleccionada
Ámbito	Portal de LIFTR	Precondición	<ul style="list-style-type: none"> - El usuario debe haber hecho <i>login</i> - El usuario se encuentra en la vista de ejercicios
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El sistema muestra las sesiones anteriores 2. El usuario selecciona una sesión y hace click sobre el botón de expansión. 3. El sistema muestra la sesión y sus datos.. 		
Extensiones			

3.4 Arquitectura global del sistema

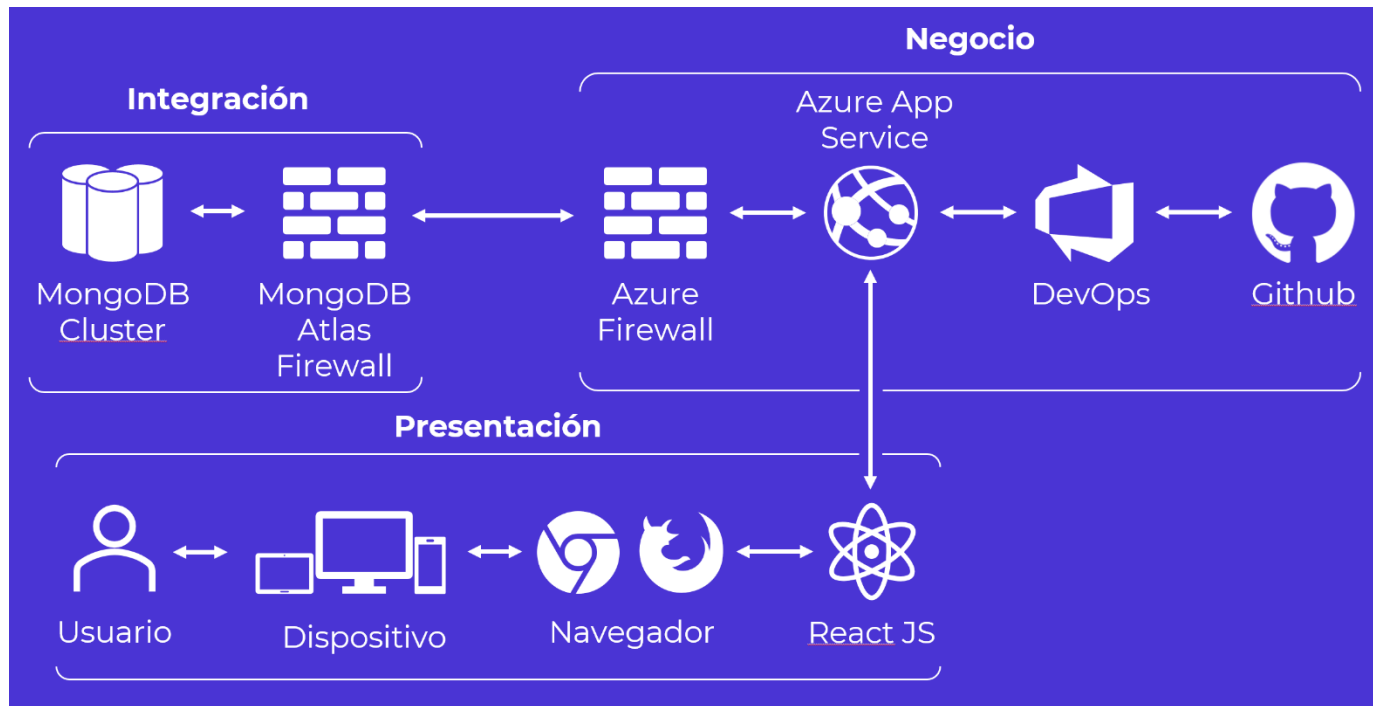


Ilustración 5

Describe la organización del sistema y los componentes que la comprenden, así como las relaciones entre sí. Se utiliza un clúster de MongoDB Atlas para la capa de persistencia, a la cual se conecta el *back end* alojado en Azure. El *back end* está basado en NodeJS y lo sirve un App Service de Azure. El servicio de App Service permite no tener que preocuparse de la infraestructura ni el *hardware* subyacente y únicamente concentrar los esfuerzos en el desarrollo.

El App Service está conectado con el servicio de DevOps de Azure, que permite la creación de *pipelines* de CI/CD, que agiliza y automatiza el proceso de despliegue de la aplicación, además de que facilita la reiteración de las nuevas versiones con un sistema de integración (CI) automatizado. GitHub, por otro lado, es la herramienta utilizada para el control de versiones la cual se integra directamente con Azure DevOps.

3.5 Punto de vista de la información

El siguiente diagrama de clases UML describe las clases del sistema presentado, los atributos de cada uno, métodos u operaciones y las relaciones entre sí.

La información de la aplicación se centra en el usuario. Cada usuario dispone de su propio diario de ingesta y de ejercicios. Ambos tipos de diarios tienen funcionalidades y comportamientos similares que pueden ser abstraídos a una superclase, de la que después heredan cada una de las subclases para cada tipo específico. Dentro de estos diarios encontramos entradas, cada una para su tipo respectivo. Las entradas de los diarios de ingesta calórica recogen datos acerca de qué alimento se ha ingerido, en qué cantidad y cuando. Las entradas del diario de ejercicios almacenan el ejercicio realizado y los parámetros relevantes para el tipo de ejercicio.

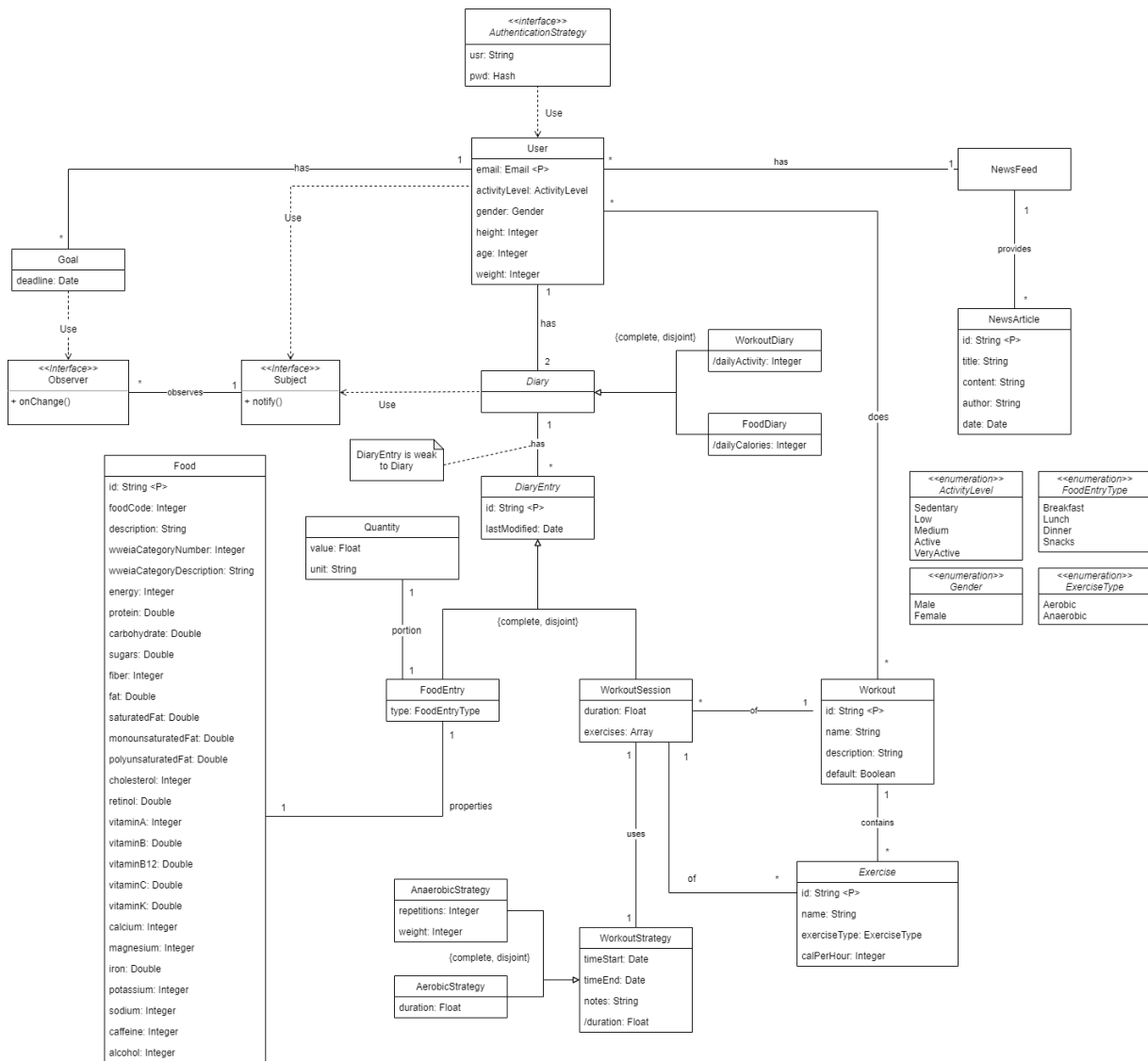


Ilustración 6

Por último, encontramos el proveedor de noticias y el apartado de objetivos. Este último utiliza el patrón observador para monitorizar cambios en algún valor (biométrico o de otro tipo) del usuario.

3.6 Punto de vista de la computación

El punto de vista computacional describe la funcionalidad básica del sistema, así como su descomposición y organización en objetos que interactúan entre sí para proporcionar dicha funcionalidad. Se han agrupado las funcionalidades en distintos objetos computacionales y se especifican las interfaces de cada objeto, con las posibles funciones que se pueden llevar a cabo para cada uno.

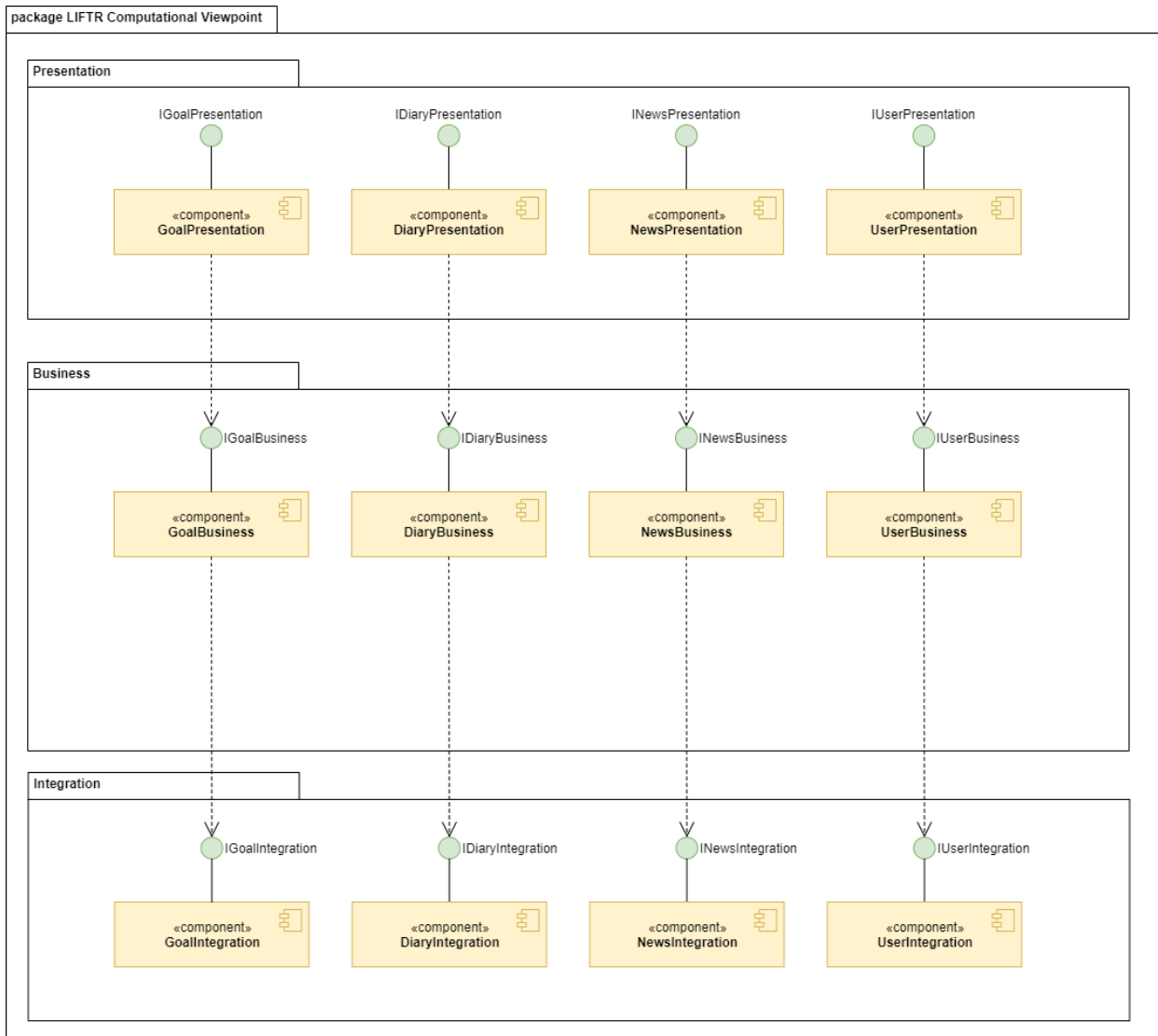
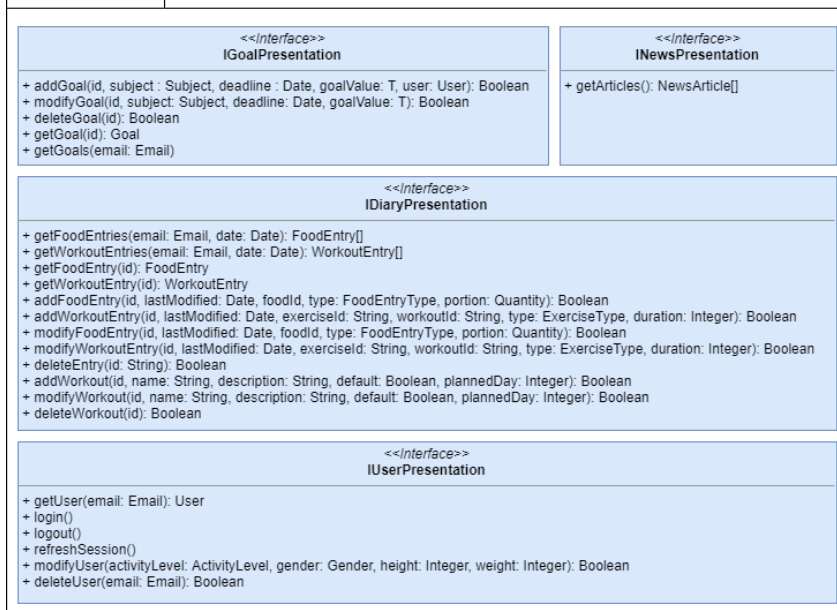
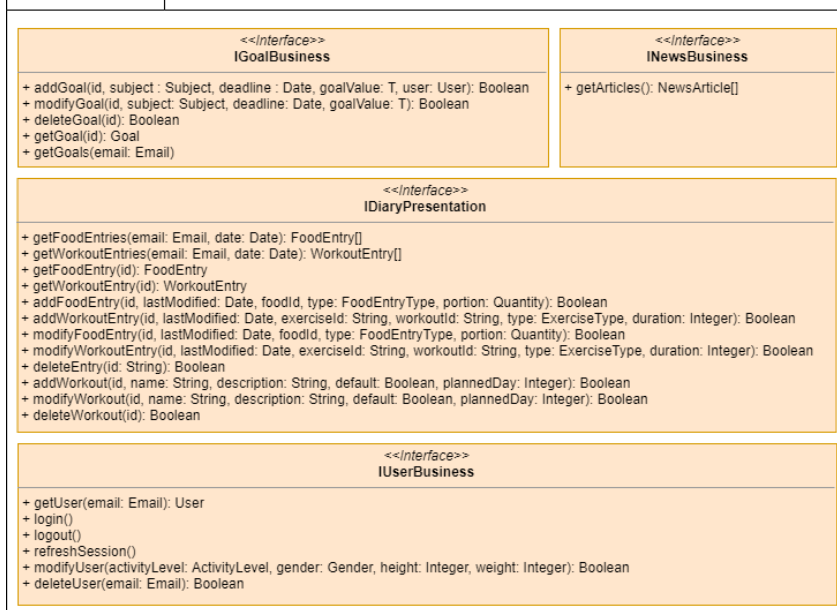


Ilustración 7

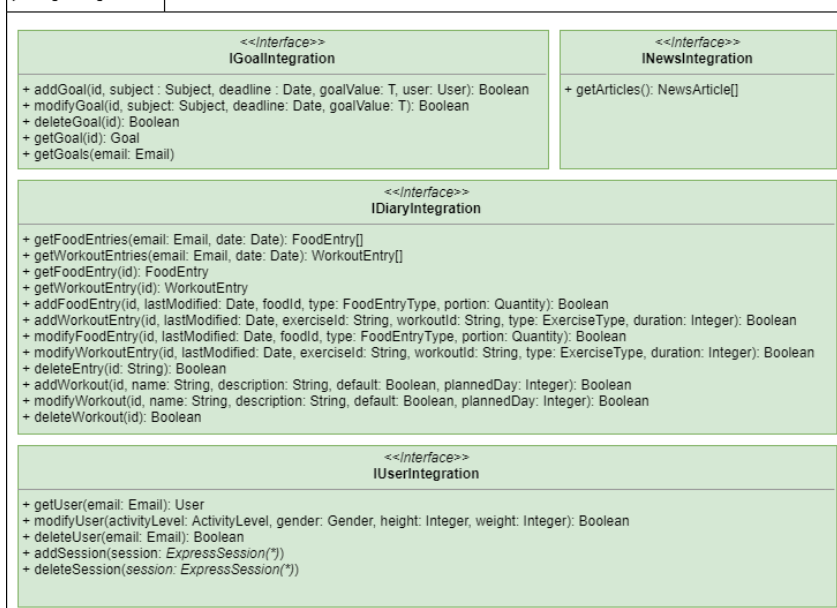
package Presentation



package Business



package Integration



*ExpressSession es un tipo del framework Express

4. Diseño de la aplicación

4.2 Wireframes

Bocetos de las vistas que describen de manera general la distribución de elementos en las pantallas (se busca que sea multiplataforma, de manera que se tendrá esto en cuenta).

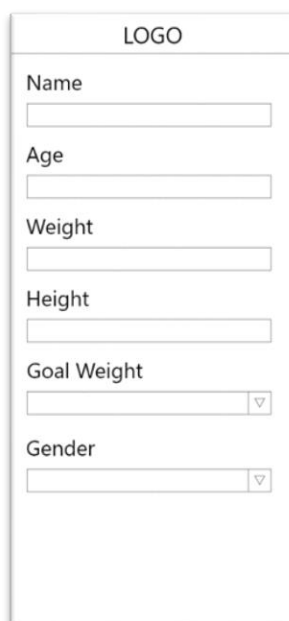


A vertical wireframe for a login screen. At the top center is a rectangular box labeled "LOGO". Below it, there are two input fields: the first is labeled "Email input" and the second is labeled "Password Input". At the bottom center is a button labeled "Signup".

Vista de registro en la aplicación, permite registrarse y darse de alta.

Una vez validados los datos, se redirige a un formulario para recoger datos biométricos sobre el usuario.

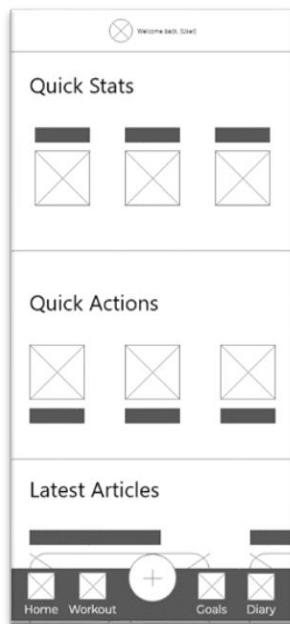
Ilustración 9 – Vista de Login



A vertical wireframe for a registration form. At the top center is a rectangular box labeled "LOGO". Below it are several input fields and dropdown menus: "Name" (text input), "Age" (text input), "Weight" (text input), "Height" (text input), "Goal Weight" (text input with a dropdown arrow), and "Gender" (text input with a dropdown arrow).

Datos biométricos básicos del usuario para establecer un punto de partida y próximos pasos.

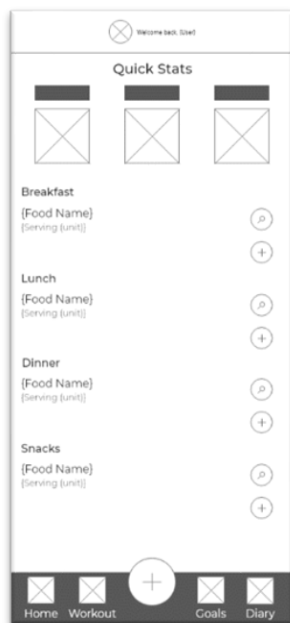
Ilustración 10 – Vista de registro



Elementos comunes de las distintas categorías con datos importantes / relevantes para el usuario.

Incluye novedades, artículos, etc.

Ilustración 11 – Vista *Home*



Listado de alimentos, en cada comida, consumidos y registrados por el usuario.

Contiene estadísticas acerca de las calorías restantes a ingerir, algún objetivo prioritario a cumplir (macronutriente) o dato relevante.

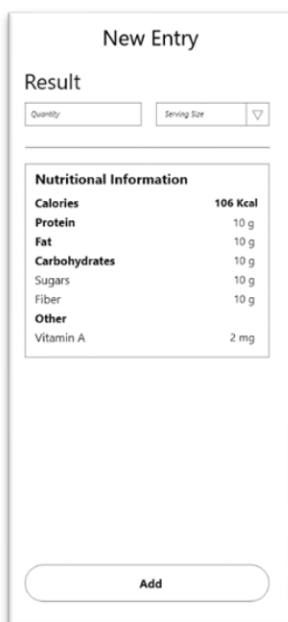
Ilustración 12 – Vista *Diary*



Inserción de una nueva entrada en el diario de ingesta, permite buscar alimentos en la base de datos.

Los resultados se muestran y permiten seleccionar un alimento para la nueva entrada al diario.

Ilustración 13 - Vista búsqueda ejercicios



Vista de un resultado concreto de la búsqueda de un alimento en la vista anterior, donde se especifica la cantidad y se muestran los valores nutricionales.

Permite añadir y confirmar la entrada en el diario.

Ilustración 14 - Vista de búsqueda de alimentos (resultado)

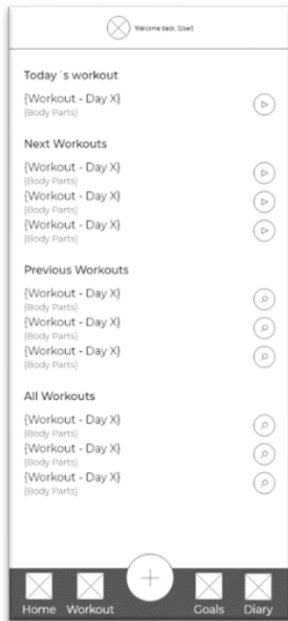


Ilustración 15 - Vista Workout

Vista con las rutinas, historial de rutinas / sesiones anteriores realizadas.

Permite empezar una rutina, añadir una rutina, modificarlas o ver las sesiones / rutinas realizadas anteriormente.

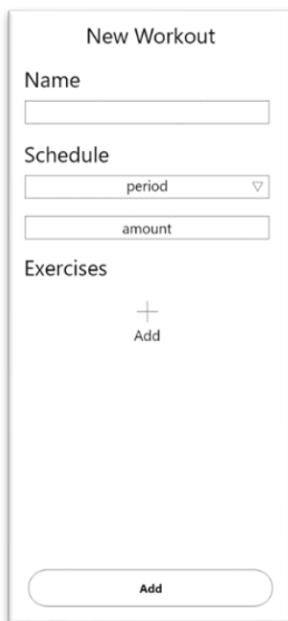


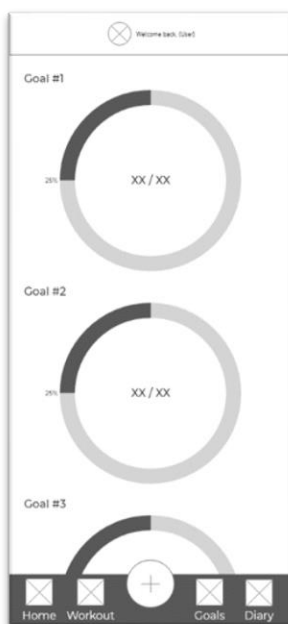
Ilustración 16 - Vista de creación de rutinas

Permite dar un nombre a la rutina y añadir ejercicios.



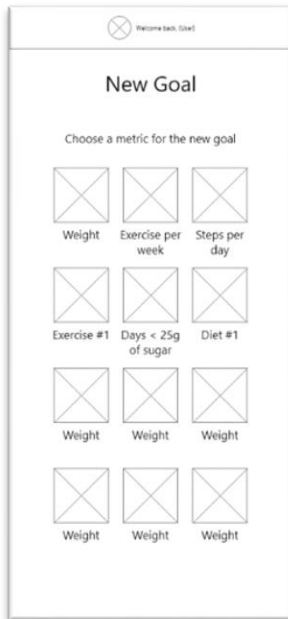
Permite buscar ejercicios en la base de datos, así como añadirlos a la rutina.

Ilustración 17 - Vista de búsqueda de ejercicios



Muestra los objetivos actualmente creados y el progreso específico de cada uno. Cada objetivo muestra el progreso actual sobre una métrica o atributo.

Ilustración 18 - Vista de Goals



Muestra las métricas disponibles sobre las que crear objetivos. El usuario selecciona una.

Ilustración 19 - Nuevo Objetivo

Nota: Las vistas de modificación de entradas serán las mismas que las utilizadas para la creación, pero con ciertos campos deshabilitados (donde aplique).

4.2 Navegación

El siguiente diagrama muestra las vistas descritas anteriormente y la navegación entre ellas.

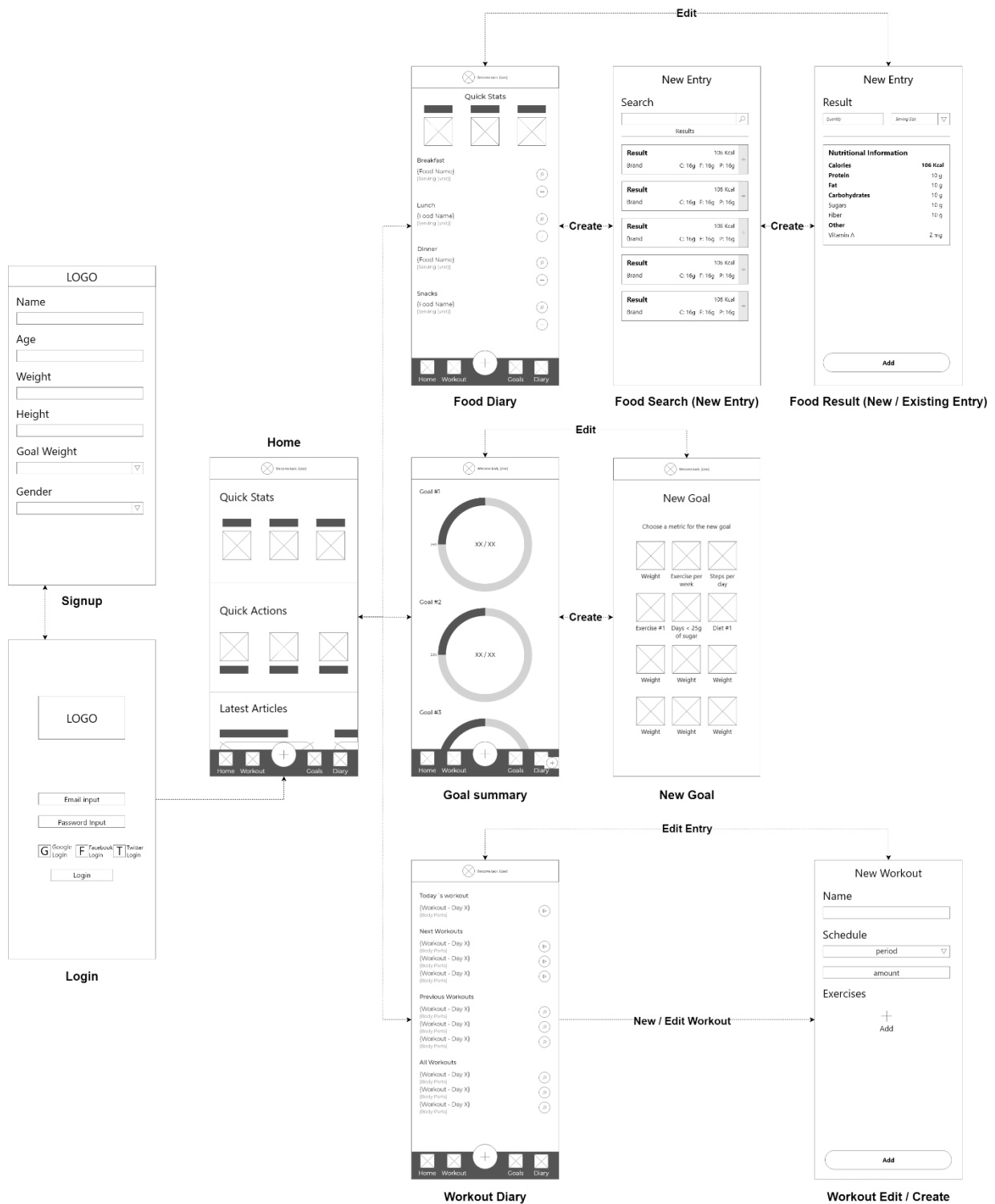


Ilustración 20

4.3 Elementos de diseño

Descripción de los elementos elegidos para el diseño final, divididos en componentes, que siguen unas pautas de diseño preestablecidas para que haya más cohesión. La aplicación utiliza la librería de componentes UI de [ant.design](#), que ofrece más de 50 componentes personalizables y reutilizables que facilitan la rápida implementación y reiteración de versiones, especialmente en equipos reducidos de desarrolladores.

4.3.1 Responsive design

El diseño de la aplicación se basa en la filosofía *mobile-first*, donde se da prioridad a un diseño accesible y de fácil uso en dispositivos móviles, que serán la “demografía” de dispositivos objetivo. Para alcanzar este objetivo, se hace uso del componente *Grid*, que divide la vista en 24 columnas homogéneas y que permite que los componentes cambien el tamaño según el tamaño de la pantalla del dispositivo.

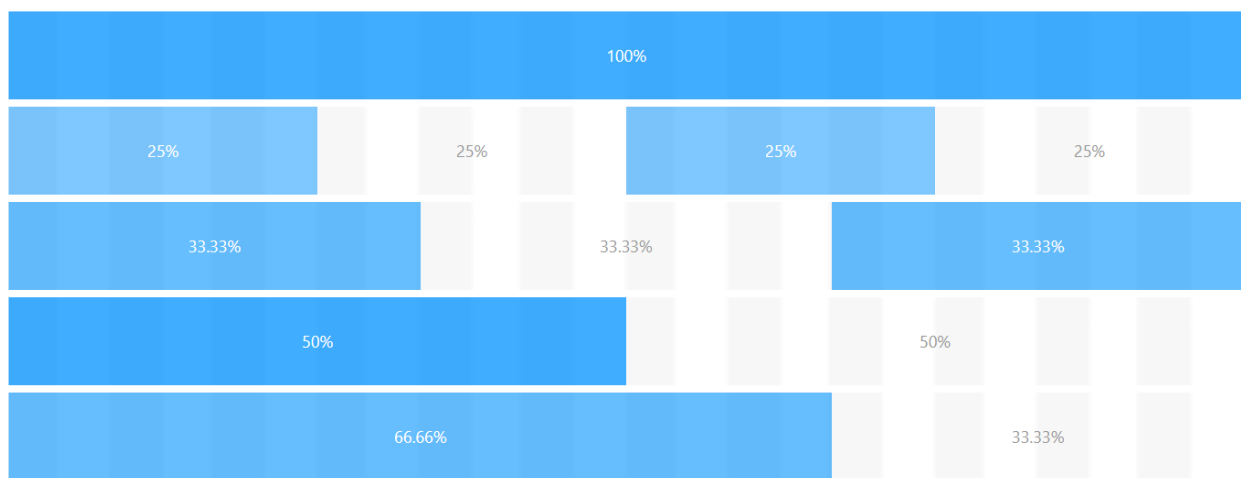


Ilustración 21

Esta flexibilidad permite el planteamiento de un diseño enfocado primariamente a dispositivos móviles, pero que se adapta automáticamente también a pantallas grandes. Al mismo tiempo, se adaptará automáticamente la propia distribución de los grupos de elementos que componen las vistas.

4.3.2 Elementos

La percepción del usuario de la aplicación empieza con el UI y sus elementos. Aplicaciones muy conocidas, como Uber o Facebook, son rápidamente reconocibles por el estilo de sus interfaces, colores y otros elementos que los diferencian de la competencia.

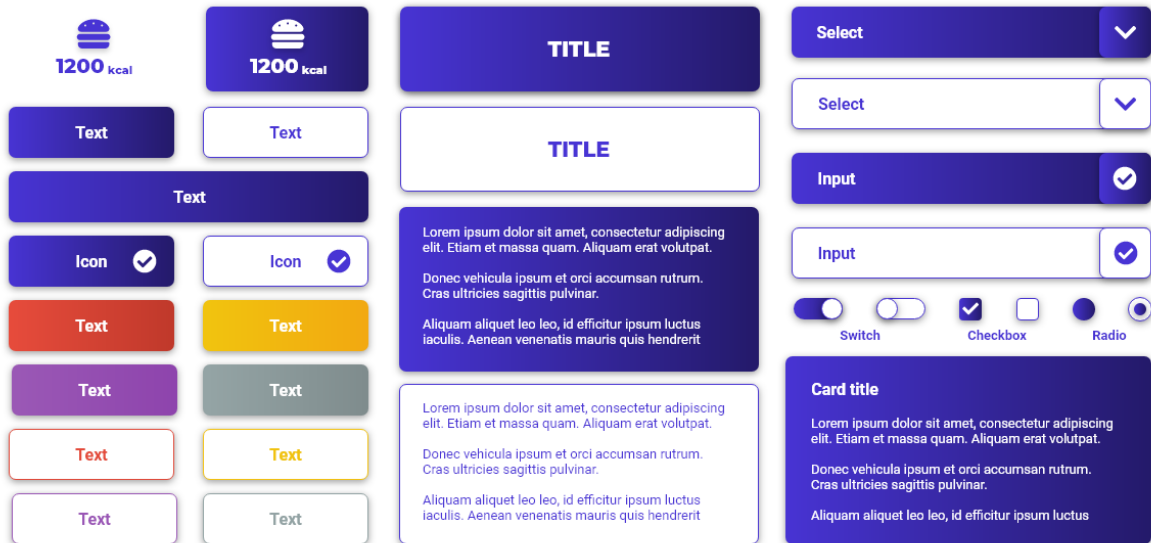


Ilustración 22

Los elementos utilizan fondos con degradados lineales y texto en blanco y viceversa, para aportar contraste y resaltar el texto o los elementos contenidos. Para cada versión de un elemento, existe su contrapartida con fondo transparente y con los matices y borde del color primario.

Los bordes de los contenedores de cada tipo de elemento, sea un botón o un campo, tienen un radio de borde de 7px y el borde de los elementos de fondo transparente tiene un grosor de 1px.

4.3.2.1 Colores

Los colores son una parte fundamental de la percepción del usuario sobre el mundo y los objetos que nos rodean. Es por este motivo que se trata de uno de los puntos centrales del diseño, porque determinan la percepción del usuario no sólo de los elementos de la pantalla, sino de toda la aplicación en general.

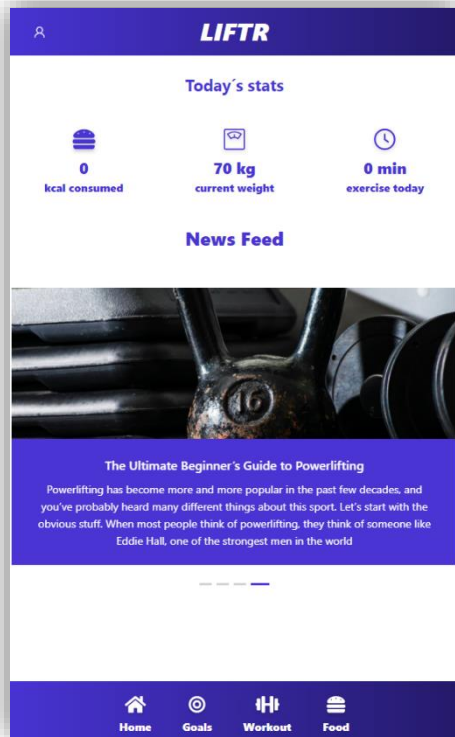
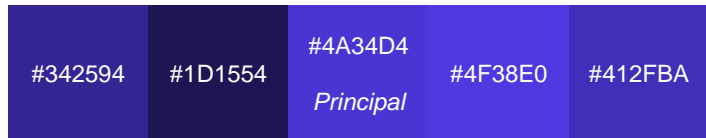


Ilustración 23

En el caso de esta aplicación, se ha elegido un color principal (#4834D4) y se ha utilizado el contraste negativo entre este y el blanco, combinando el uso del color como fondo (sólido o degradado) con texto e iconos en blanco y fondo blanco con texto e iconos en el color principal.

La paleta de colores utilizada comprende, por lo tanto, la escala de tonalidades del color principal:



Para alertas u otros elementos de *feedback*, se utilizan colores según la importancia o la "intención" del mensaje:



5. Implementación

A partir del diseño anterior y de los requisitos recopilados, en la etapa de implementación es cuando se ven en funcionamiento. Esta es la fase que requiere de más esfuerzo y en la que pueden surgir los obstáculos o problemas que marquen que el producto final sea adecuado y de buena calidad.

A continuación, se exponen las herramientas utilizadas para el desarrollo, los aspectos más relevantes de la implementación de la aplicación y qué obstáculos han surgido a lo largo de su creación.

5.1. Organización

Se ha utilizado Microsoft To-Do para el seguimiento de tareas y priorización de las mismas. Durante el desarrollo, sobretodo cuando se realiza en solitario, es fácil quedarse estancado y no obtener progreso. El uso de herramientas de gestión permite obtener una visión global de cómo progresa todo el desarrollo y ajustar los esfuerzos allá donde sea más prioritario, con el objetivo de cumplir con los intervalos temporales establecidos durante las fases de análisis y diseño.

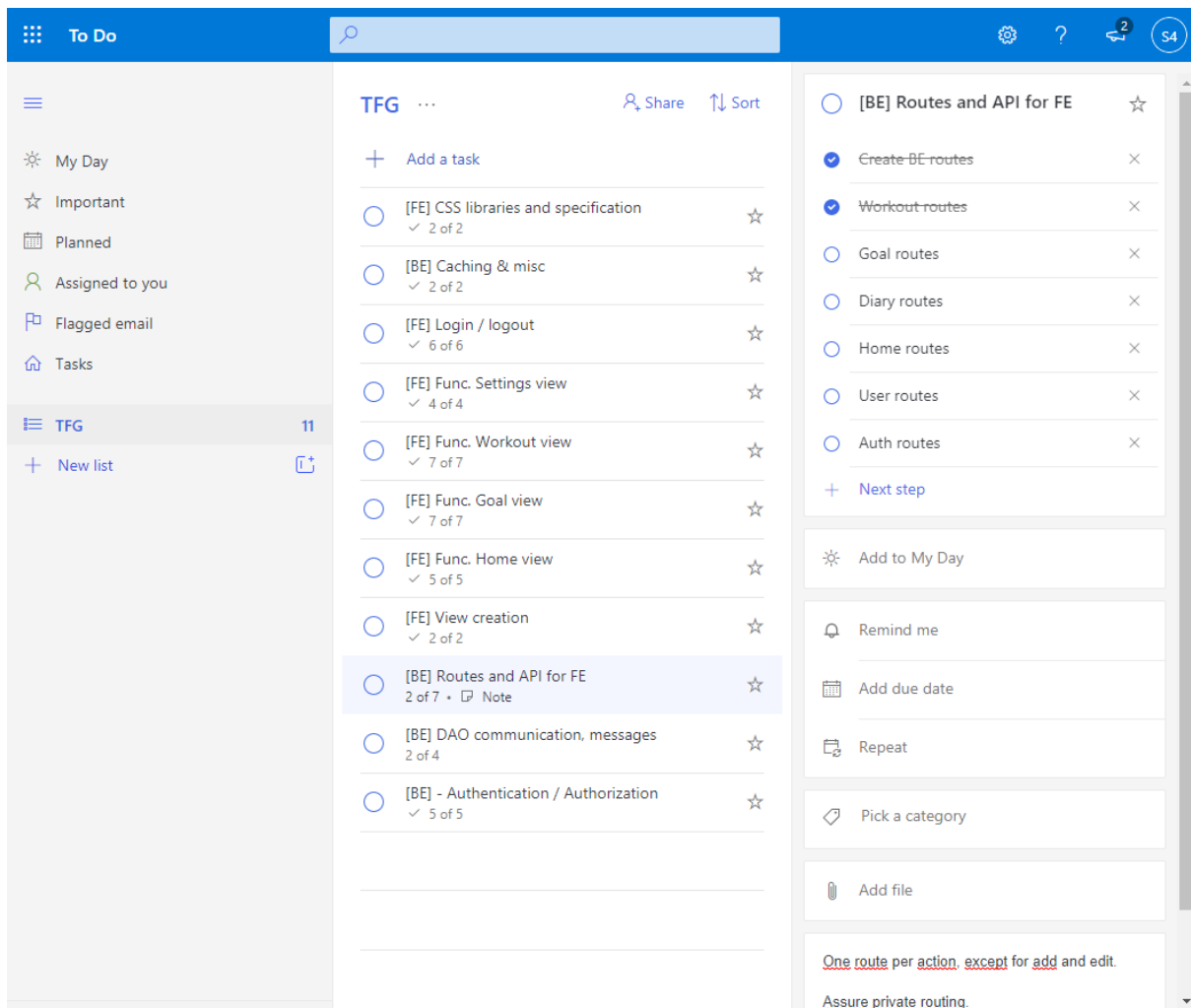


Ilustración 24

5.2. Control de versiones

El control de versionado de código es esencial a día de hoy a la hora desarrollar cualquier aplicación de cierto tamaño. Herramientas como Git o SVN hacen posible la creación de versiones incrementales que, en caso de que encontremos una situación “no recuperable” o en la que necesitemos revertir a un estado funcional anterior, nos permiten recuperar ese estado de manera instantánea, sin importar los cambios realizados de manera posterior al *commit* en cuestión.

Para la realización de LIFTR se ha elegido como alojamiento Github, que implementa Git como la infraestructura subyacente. A continuación, una captura de la línea de *commits* del *master branch*:

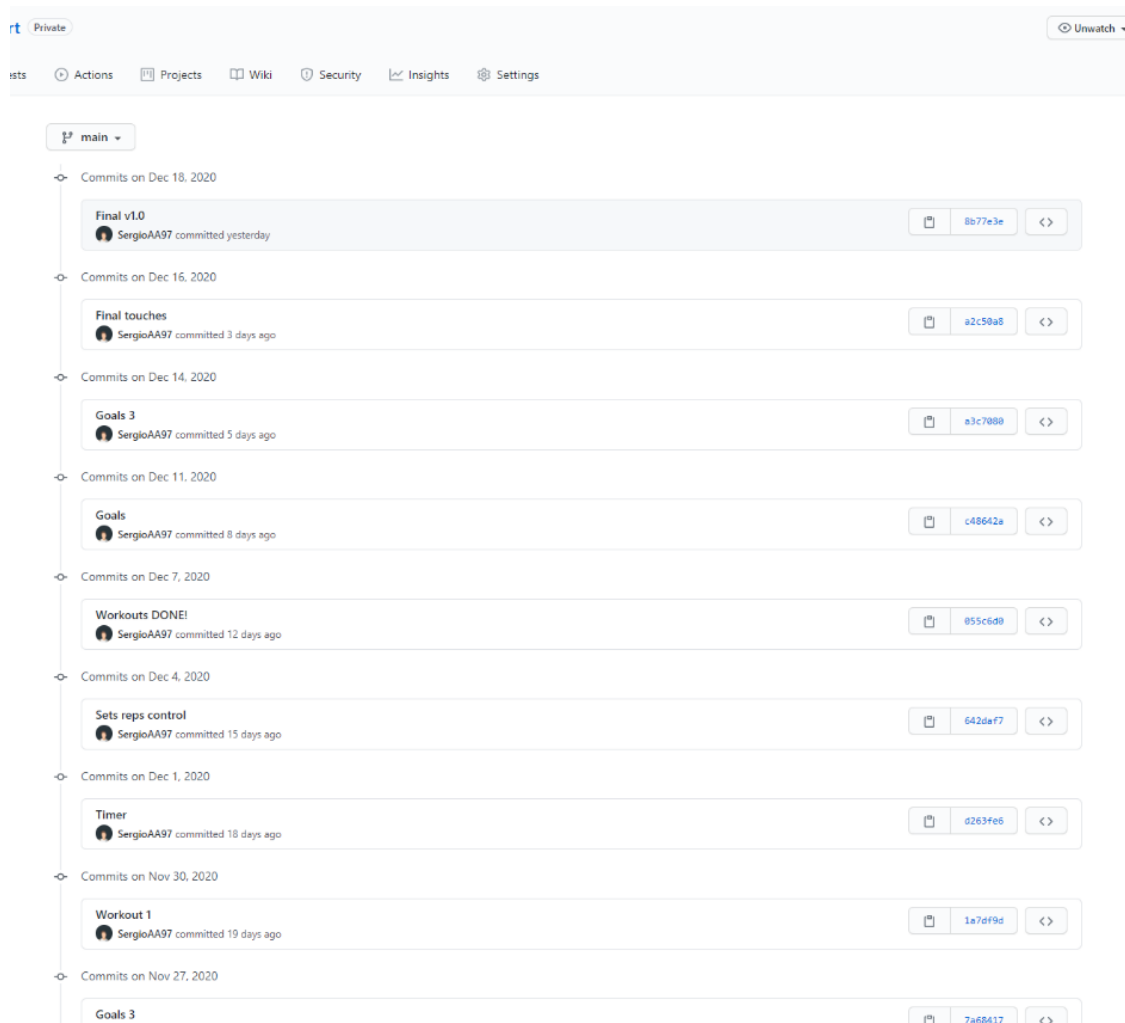


Ilustración 25

5.3. Entorno de desarrollo

Como comentado en la sección de arquitectura de sistema, la infraestructura de la aplicación se basa en el *MERN stack*. Se trata de un acrónimo de la combinación de sus tecnologías o *frameworks*, compuesto por **M**ongoDB, **E**xpress, **R**eact y **N**odeJS. En su conjunto, estas tecnologías cubren todas las capas de la aplicación, desde la base de datos (MongoDB) a la capa de presentación (React).

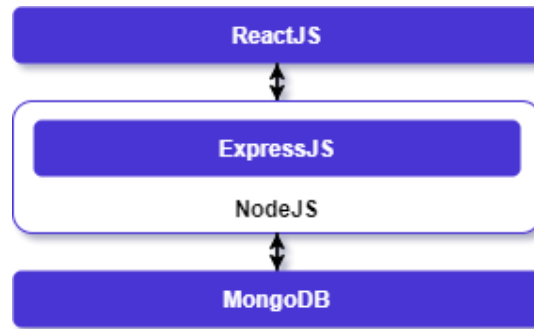


Ilustración 26

MongoDB es una base de datos no relacional multiplataforma. En lugar de utilizar tablas, como ocurre en una base de datos relacional, se utilizan “documentos” basados en una variante de JSON (BSON) para su almacenamiento. Cada documento puede contener datos, como atributos del mismo, de cualquier tipo. Como ventajas principales frente a una base de datos relacional como MySQL es que, dado que utilizamos JavaScript como el lenguaje para el *back end* y el *front end*, permite la integración de código JS directamente en las peticiones a la base de datos. Además, la importación de un documento JSON/BSON es directa en JS, lo que quiere decir que podemos guardar la respuesta de la BBDD directamente en una variable y empezar a tratarla como si fuese un TAD normal.

NodeJS y Express son los encargados de la gestión del *back end* o la capa de negocio. Express es un *framework* que facilita la implementación de APIs y aplicaciones web con NodeJS, con un gran número de middlewares e interfaces ya implementadas que facilitan la gestión de peticiones y el envío de sus respuestas. Adicionalmente se ha implementado *Passport* para la capa de autenticación. Passport es un *framework* con el objetivo de hacer sencilla y versátil la implementación de autenticación en NodeJS. Soporta una gran multitud de proveedores de autenticación, como Auth0 o Amazon Cognito. En este caso, se ha implementado únicamente una estrategia local, que utiliza la BBDD de MongoDB para almacenar las sesiones de usuarios conectados, los *logins/logouts* y los registros de usuarios nuevos.

Por último, React es un *framework* basado en JS para interfaces de usuario. React se basa en la creación de componentes encapsulados con estado, que permiten crear interfaces interactivas de usuario ricas en contenido, funcionalidad y, sobretudo, fácilmente reutilizables y gestionables.

Como IDE, se ha utilizada Visual Studio Code, el editor de código *open source* de Microsoft. La razón principal es la sencilla integración con herramientas de control de versiones y con los servicios de Azure donde se alojará la aplicación.

5.4. Implementación de las partes críticas

5.4.1 Autenticación / autorización

Como comentado en el punto anterior, se ha utilizado *Passport* para la implementación de la capa de autenticación / autorización de la aplicación. La estrategia utilizada se basa en la creación de *tokens* JWT (JSON Web Tokens) para la gestión de sesiones de usuario y en la comprobación de credenciales contra la BBDD MongoDB. El usuario recibe este *token* una vez se ha realizado el *login* en la aplicación o directamente en la API.

```
_id: ObjectId("5f7b7b0d255d192fbccb824c")
username: "user"
password: "$2b$10$f6a8K744HaFZrkEDHUPTA0sej4c1PIQk6zSIX0Uka1KwtULyhbV0q"
role: "user"
__v: 2
> entries: Array

_id: ObjectId("5f7cc42b14f0aa7cfc06eec8")
username: "newUser"
password: "$2b$10$yWfZmR5EZVjTMajbsGKV4u0hkifqfynlF2YqAw5pDvLMukcMIjk/0"
role: "user"
__v: 0
```

Ilustración 27

La estrategia local utiliza la base de datos para almacenar tanto las sesiones de usuarios como las credenciales de los mismos. Dado que la seguridad es uno de los aspectos más importantes a la hora de ofrecer servicios en línea, las credenciales de usuario nunca se almacenan en claro. En su lugar, se almacena un *hash* generado por la librería *bcrypt*, que nos permite crear los *hashes* de las *passwords* con encriptación AES256. La ventaja principal de esta metodología es que la contraseña nunca queda almacenada en claro y aunque se consiga control de la base de datos no se podrán robar credenciales de los usuarios almacenados. El código de firma de *token* JWT, según la [especificación oficial](#):

```
const signToken = (userID) => {
  return JWT.sign(
    {
      iss: "LIFTr",
      sub: userID,
    },
    "SomeGreatS3cre3tLIFTr",
    {
      expiresIn: "2h",
    }
  );
};
```

A continuación, se encuentran extractos de las operaciones de *login* y de comprobación de sesión:

```
userRouter.post(
  "/login",
  passport.authenticate("local", { session: false }),
  (req, res) => {
    if (req.isAuthenticated()) {
      const { _id, username, role } = req.user;
      const token = signToken(_id);
      res.cookie("access_token", token, { httpOnly: true, sameSite: true });
      console.log(`User ${username} has logged in`);
      res.status(200).json({ isAuthenticated: true, user: { username, role } });
    }
  }
);
```

```
userRouter.get(
  "/authenticated",
  passport.authenticate("jwt", { session: false }),
  (req, res) => {
    const { username, role } = req.user;

    res.status(200).json({
      isAuthenticated: true,
      user: {
        username,
        role,
      },
    });
  }
);
```

5.4.2 Estado, “props” y su propagación

En React todos los elementos de una interfaz se encapsulan en componentes. Cada componente puede ser uno o más elementos HTML con CSS y JS asociado a los mismos. React expone una interfaz que permite interactuar con otros componentes y con los “ciclos de vida” de un componente mismo. Este ciclo está compuesto por una serie de eventos relacionados con la creación, renderización y eliminación del elemento en el DOM. Cada uno de estos componentes puede albergar un “estado”, que le permite almacenar de manera temporal datos, y propagar este estado a otros componentes (“props”).

El “estado” de un componente se entiende, en este contexto, como uno o varios objetos almacenados en un componente, que permiten cambiar los datos con los que un componente determinado trabaja. Los “props” son los parámetros que recibe un elemento de uno que se encuentra más alto en su escala jerárquica en el DOM. Generalmente, un componente propagará su estado a otros mediante “props”.

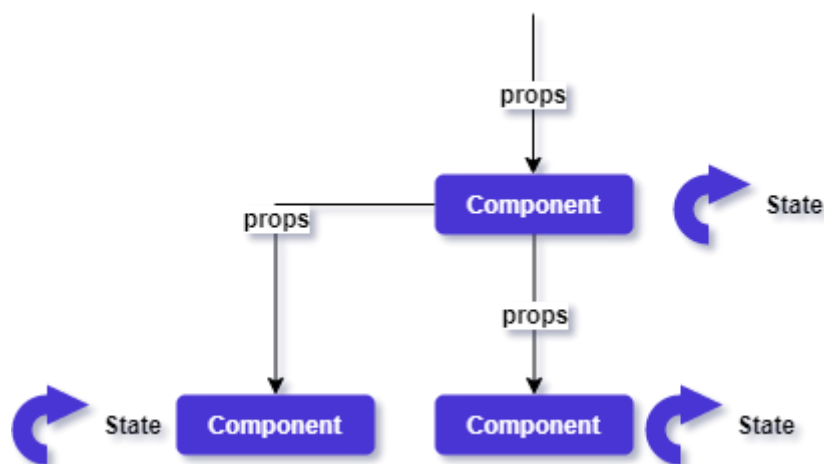


Ilustración 28 – Estado y Props en componentes

Hasta la versión 16.7 de React, sólo era posible tener estado en componentes derivados de la clase `React.Component`, sin embargo, se introdujeron los *hooks* en la versión 16.8, que permiten acceder a todas estas funcionalidades desde componentes funcionales. Estos *hooks*, son, al fin y al cabo, funciones de alto orden (HO) que abstraen la funcionalidad que anteriormente iba en una clase, a un componente funcional. A excepción del componente principal (`App.js`), todos los demás componentes son funcionales y utilizan *hooks* para crear el estado y llamar a las funciones del ciclo de vida del componente. Estos *hooks* permiten también la abstracción de lógica de la aplicación a una función independiente del componente. Un ejemplo de esto es el *custom hook* del temporizador de las sesiones de ejercicio, `useTimer`.

```
const Timer = () => {
  const [time, startTimer, stopTimer, resetTimer] = useTimer();

  let dateTime = new Date(time);

  return (
    <>
      <Row justify="center" className="inv-font">
        <TimeSlot value={dateTime.getMinutes()} label="m" />
        <TimeSlot value={dateTime.getSeconds()} label="s" />
      </Row>
    </>
  );
};
```



```
    <TimeSlot value={dateTime.getMilliseconds()} label="ms" />
  </Row>
)
}
```

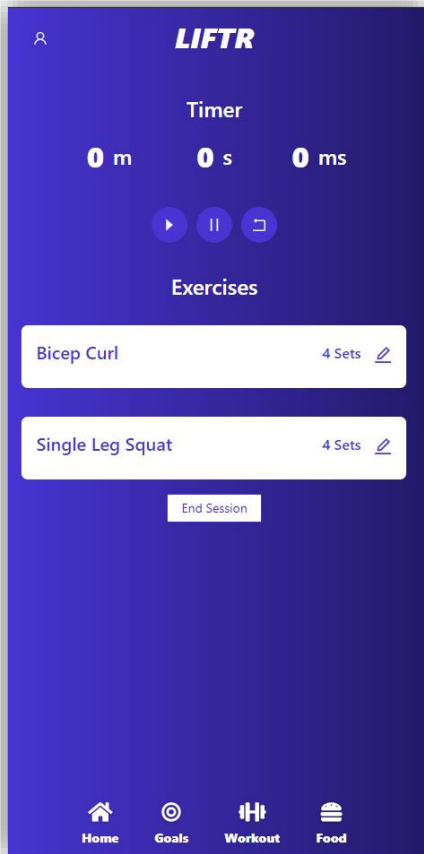


Ilustración 29 – Workout Session, Timer

Este *custom hook* devuelve un vector con el tiempo actual y las funciones para iniciarlo, pararlo y reiniciarlo. El código de *useTimer* se puede encontrar en mi [repositorio](#).

El problema principal de este planteamiento se encuentra en la profundidad (en términos del DOM) a la que se encuentra un componente que recibe el estado, con respecto al componente que lo emite y gestiona. Si existen varios componentes intermedios, esto obligará al desarrollador a pasar el estado como “props” a todos los componentes intermedios. Esto causa gran dependencia entre componentes y dificulta la correcta gestión del estado en proyectos muy grandes.

Para remediar este problema, se creó la interfaz de “Context”, que crea un objeto que almacenará datos compartidos entre diversos componentes. La diferencia principal entre un objeto “Context” y un componente normal, es que el “Context” puede propagar su estado de manera selectiva y completamente independiente a la jerarquía del DOM, siempre que el “Context” se encuentre en un nivel superior a los consumidores del estado.

A continuación, podemos ver un extracto de la implementación del “DiaryContext”, que es el objeto “Context” encargado de la gestión de los datos relacionados con los diarios de ingesta calórica y de ejercicios:

```
export const DiaryContext = createContext();

export default ({ children }) => {
  const [foodEntries, setFoodEntries] = useState([{}]);
  const [foodStats, setFoodStats] = useState(null);
  const [exerEntries, setExerEntries] = useState([{}]);
  const [exerStats, setExerStats] = useState(null);
}
```

La función “createContext” inicializa el objeto. Por otro lado, las funciones “useState”, son los *hooks* de estado, que devuelven el valor del estado actual y una función *setter*, para modificarlo, en forma de un vector. Por último, “{children}” es un “prop” que contiene los elementos “hijo” del componente.

En conjunto, estas funcionalidades han permitido una eficiente implementación de la gestión del estado en la aplicación, donde únicamente los componentes que deben recibir datos los

reciben, sin pasar por cada componente intermedio, y que estos se almacenan de manera centralizada y elegante en un componente superior.

5.4.3 Diario de ingesta y de ejercicios

El pilar central del objetivo de la aplicación son los diarios de ingesta y los de ejercicios. Estos diarios permiten al usuario crear entradas de ingesta con una amplia selección de alimentos distintos. El usuario selecciona para qué “comida” del día quiere registrar la entrada y esta es guardada en la base de datos. A partir de estos datos, la aplicación actualiza la ingesta diaria, en términos de contenido de calorías y macronutrientes (proteína, grasa y carbohidratos). Los datos de referencia, en cuanto al límite calórico del día, se registra en las vistas de objetivos o *Goals*. La implementación de esta sección es relativamente sencilla, donde se hacen peticiones POST al *back end* para enviar los datos relativos a la ingesta y actualizar la vista.

Con lo que respecta el diario de ejercicios, este permite la creación de rutinas personalizadas o utilizar las creadas por defecto. Para registrar una rutina, lo único que el usuario debe hacer es utilizar el botón de empezar o “Play”. Esto inicia una sesión de la rutina. En esta vista el usuario puede utilizar un temporizador integrado, para calcular la duración de un determinado ejercicio, por ejemplo, y puede registrar el peso y repeticiones (en caso de una rutina anaeróbica) o la duración del mismo (en caso de una rutina aeróbica). Al presionar sobre el botón de finalización de sesión, esta queda registrada en la base de datos. El usuario es después devuelto a la vista del diario de ejercicios o *Workout*, donde le aparecerá la nueva sesión en la lista de historial de sesiones.

A continuación, un extracto del formulario de creación de una nueva entrada en el diario de ingesta:

```
let nutrients = UtilService.initializeNutrients(food);
const diaryContext = useContext(DiaryContext);

let { type } = useParams();
let history = useHistory();

const onFinish = (data) => {
  const { quantity } = data;
  let entry = {
    type,
    created: Date.now(),
    lastModified: Date.now(),
    item: {
      ref: food._id,
      quantity,
    },
  },
};
FoodDiaryService.postEntry(entry).then((x) => {
  diaryContext.refreshEntries();
  history.push("/diary")
});
};
```

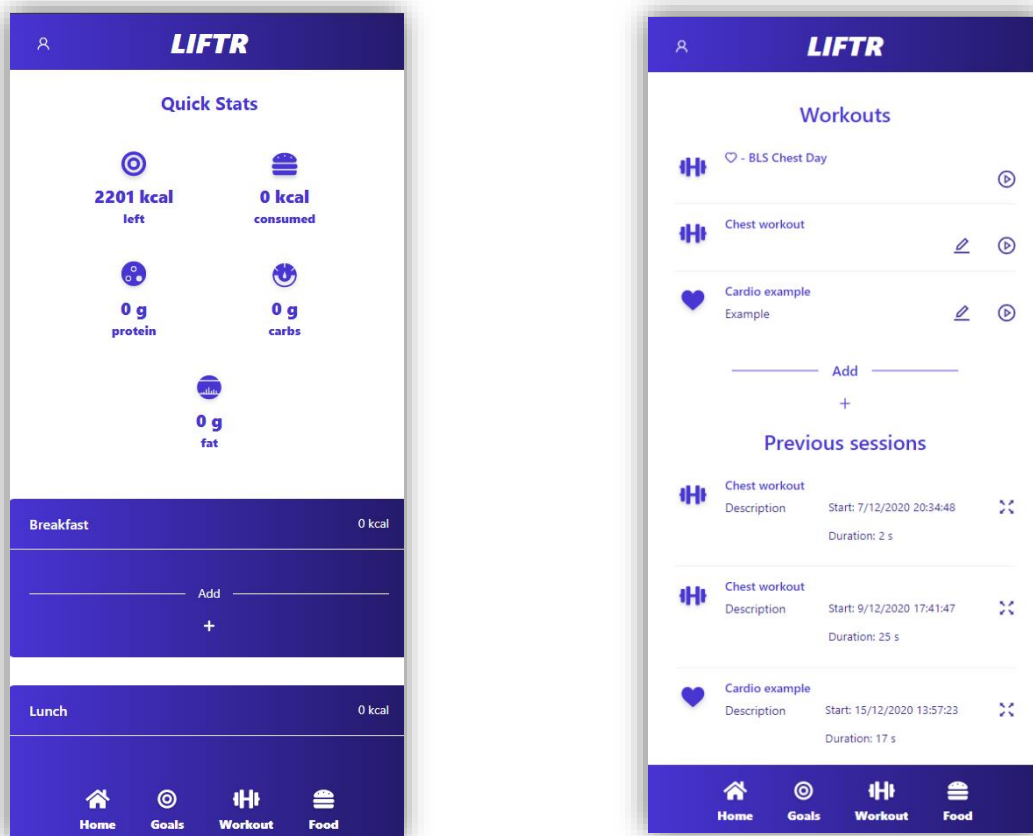


Ilustración 30 – Vista principal y del diario de ejercicios

La mayor dificultad de esta sección recae sobre la existencia de dos tipos distintos de rutinas, donde hay que cambiar el algoritmo y los campos, según el tipo. Para resolver este obstáculo, se ha implementado una estrategia por tipo, siguiendo el patrón *Strategy* de diseño. Cada algoritmo calcula y recopila los datos relevantes para sus tipos de ejercicio.

Cabe mencionar que se terminó por suprimir la vista de creación y edición de rutinas. Se ha terminado integrando directamente en la propia vista del diario, donde se abrirá el formulario de edición o creación, según si se hace *click* sobre el “+” o el botón de edición de una rutina (Ilustración 30).

Otro punto importante era la necesidad de implementar un formulario capaz de permitir que un usuario añada tantos ejercicios como quiera. Para esto, se ha implementado un formulario dinámico que permite añadir campos nuevos de ejercicio. Estos campos, además, tendrán uno o dos *inputs*, según si se trata de ejercicios anaeróbicos o aeróbicos. El botón para borrar una rutina sólo estará disponible cuando se entre en el modo edición.

5.4.4 Objetivos

La sección de objetivos permite al usuario la creación y eliminación de ciertos objetivos relacionados con los datos del usuario. Se definen objetivos de tipo básico o *core*, que están asociados a los objetivos de calorías y macronutrientes, y objetivos personalizables o “LIFTR”. Estos últimos permiten el seguimiento de factores como horas de un tipo de ejercicio a la semana o un peso objetivo. También se pueden definir fechas límite para estos objetivos. En esta vista, el usuario puede calcular su ingesta calórica de referencia en base a la fórmula de

Katch-McArdle (William D. McArdle, 2010). En cualquier caso, se avisa de que estos valores deben ser consultados con un médico y que son únicamente eso, de referencia.

```
const onSumbit = (values) => {
  console.log({ gender, height, mass, activityLevel, age });
  console.log(finalTDEE({ gender, height, mass, activityLevel, age }));
  let finalTdee = finalTDEE({ gender, height, mass, activityLevel, age });
  let finalBmr = calculateBMR({ mass, height, age, gender });

  const [proteinMul, fatMul, carbMul] = getMacros(goal);
  setResult({
    tdee: finalTdee,
    bmr: finalBmr,
    activityMultiplier: getActivityMultiplier(activityLevel),
    carbohydrates: Math.round((finalTdee * carbMul) / 4),
    fat: Math.round((finalTdee * fatMul) / 9),
    protein: Math.round((finalTdee * proteinMul) / 4),
  });
};
```

En el extracto de código anterior vemos la función encargada de actualizar los valores de referencia del usuario. Esta función recupera los valores del formulario, hace los cálculos y actualiza el estado del componente, actualizando así la vista con los valores calculados.

Para implementar estos objetivos, era esencial consolidar el estado de la aplicación bajo un objeto proveedor “Context”, de manera que pudiese ser accedida por los diarios de ingesta y

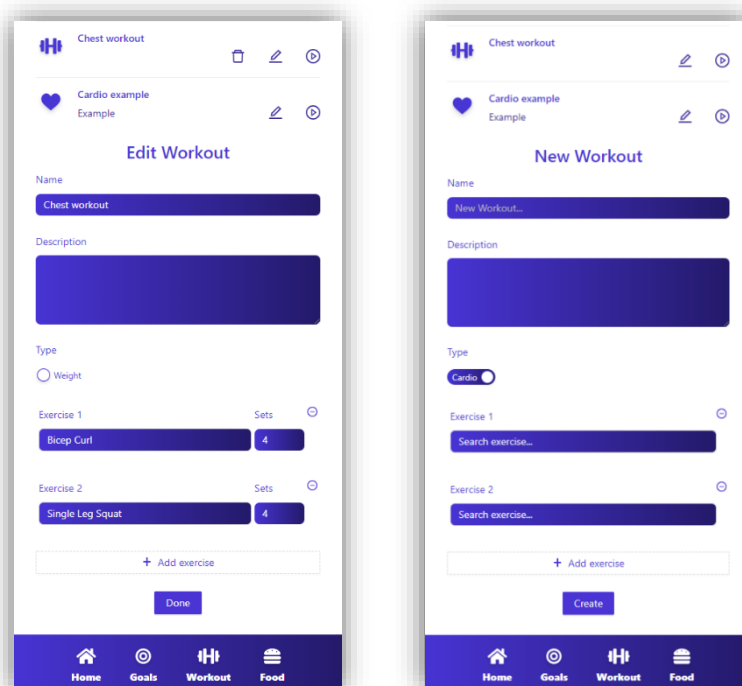


Ilustración 31 – Formularios de adición y edición

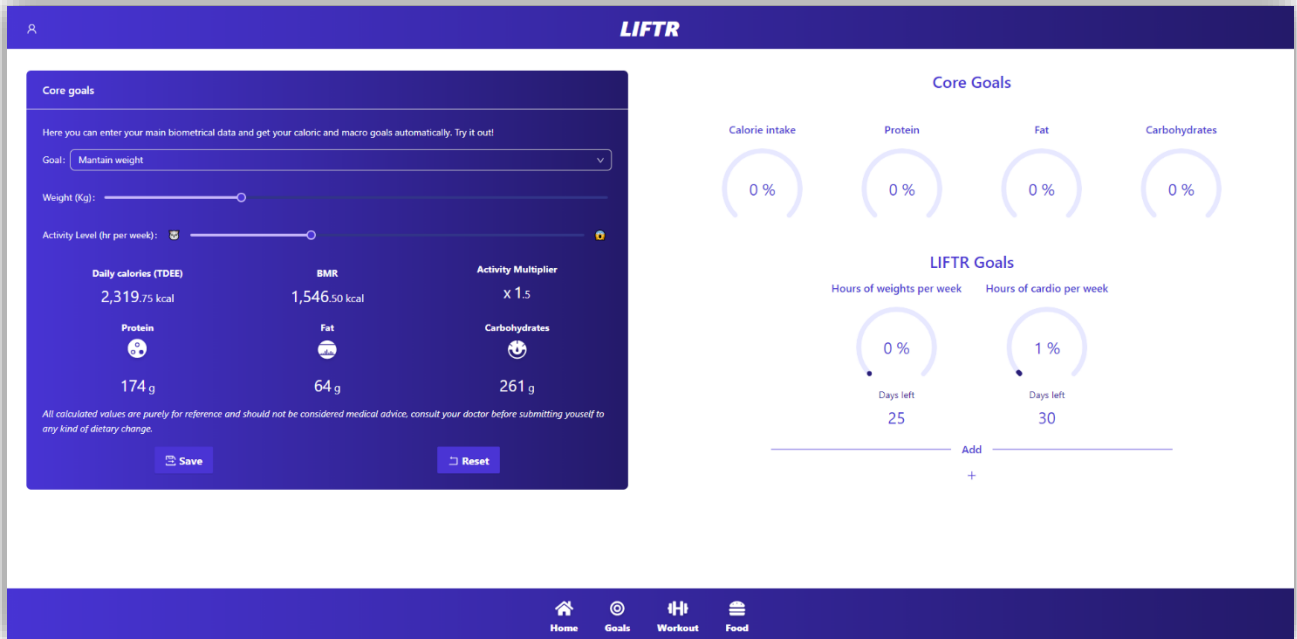


Ilustración 32 - Vista de objetivos

ejercicios. Los atributos del usuario de la base de datos son gestionados y almacenados en el estado de este objeto, que los inyecta como parámetro (“props”) a los componentes adecuados en las vistas de los diarios.

Podemos ver a continuación un extracto de la función encargada de crear un objetivo “LIFTR”:

```
const addCustomGoal = async (values, name) => {
  try {
    let endDate = new Date(Date.now());
    console.log(endDate);
    endDate.setDate(endDate.getDate() + parseInt(values.deadline));

    console.log(values, name, endDate);
    const newGoal = {
      name,
      startDate: new Date(Date.now()),
      endDate,
      goalValue: parseInt(values.goalValue),
    };
    console.log(newGoal);
    await GoalService.postCustomGoal(newGoal);
    let newGoals = [...customGoals, newGoal];
    setCustomFields(newGoals);
    setCustomGoals(newGoals);
  } catch (err) {
    console.error(err);
  }
};
```


5.4.5 Control de errores y validación de formularios

Una gestión efectiva de los errores y excepciones de la aplicación es esencial a la hora de garantizar el dominio sobre cualquier factor que puede causar un problema en la plataforma. Pese a la arquitectura en tres capas, únicamente deberemos gestionar las excepciones de la capa de negocio y la de presentación, puesto que la capa de persistencia se encuentra alojada en un entorno autogestionado (MongoDB Atlas). Por consiguiente, a continuación, se enumeran los posibles puntos de fallo de la aplicación y las medidas tomadas para cada uno:

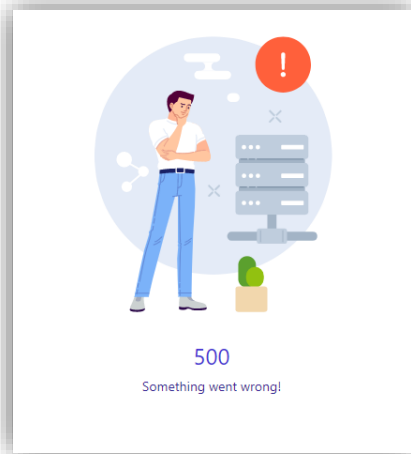


Ilustración 33 - Vista de error 500

1. Errores de conexión o disponibilidad

En el caso de que el servidor de la capa de negocio se desconecte, la capa de presentación detectará la caída y mostrará un mensaje 500. Lo mismo ocurre con la capa de negocio y la BBDD, que verifica que se conecta correctamente antes de iniciar el servidor.

2. Errores de autenticación

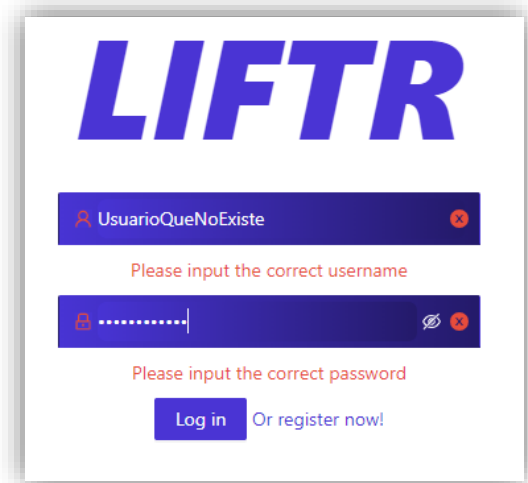
Únicamente los usuarios con un *login* válido podrán usar la aplicación. La duración de una sesión es de 2 horas desde que se genera el *token* JWT. Cuando este caduca o la sesión no es válida, el usuario es devuelto a la vista de *login* para renovar la sesión.

3. Validación de formularios

Los formularios son un punto donde la validación de campos es fundamental para que estos datos concuerden con los tipos esperados por la capa de persistencia. El *framework* de ant.d de componentes tiene su propia implementación de formularios, a los que se les puede pasar un vector de reglas (*rules*) que un campo debe cumplir.

```
<Form.Item
  name="username"
  help={status === "error" ? "Please input the correct username" : ""}
  hasFeedback={status}
  rules={[
    { required: true, message: "Please input your username!" },
    {
      type: "string",
      pattern: "^[A-Za-z0-9]+(?:[ _-][A-Za-z0-9]+)*$",
      message: "Please input a valid username!",
    },
  ]}
  className="gradient-primary rounded-corners"
  validateStatus={status}
>
```

Cuando este no se cumple, se genera un mensaje de error:



The image shows a login form for 'LIFTR'. At the top, the logo 'LIFTR' is displayed in a large, bold, blue font. Below the logo, there are two input fields. The first field is for the username, containing the text 'UsuarioQueNoExiste'. Below this field, a red error message reads 'Please input the correct username'. The second field is for the password, shown as a series of dots. Below this field, another red error message reads 'Please input the correct password'. At the bottom of the form, there are two buttons: a blue 'Log in' button and a text link 'Or register now!'.

Ilustración 34 - Validación de formularios

6. Pruebas

A lo largo del desarrollo se han realizado diversas pruebas a las distintas partes de la aplicación, con el objetivo de mejorar la experiencia de usuario, resolver problemas y depurar el código.

A continuación, se muestra la batería de pruebas realizadas sobre la última versión de la aplicación:

Tabla 40

ID	1		
Descripción	Login a la aplicación		
N	Entrada	Esperado	Verif.
1	Introduce sólo nombre	Mensaje de error en formulario	OK
2	Introduce sólo contraseña	Mensaje de error en formulario	OK
3	Introduce una combinación no válida	Mensaje de error en formulario	OK
4	Introduce credenciales no existentes	Mensaje de error en formulario	OK
5	Introduce credenciales existentes	Acceso	OK

Tabla 41

ID	2		
Descripción	Barra de navegación superior		
N	Entrada	Esperado	Verif.
1	Click sobre icono de usuario	Se muestra el desplegable con las preferencias y <i>logout</i>	OK
2	Click sobre preferencias	Se muestra la vista de preferencias de usuario	OK
3	Click sobre <i>logout</i>	Se devuelve a la vista de <i>Login</i> , se termina la sesión en <i>back end</i>	OK
4	Click sobre logo	Se devuelve a la vista Home	OK

Tabla 42

ID	3		
Caso	Barra de navegación inferior		
N	Entrada	Esperado	Verif.
1	Click sobre <i>Home</i>	Se muestra la vista Home	OK
2	Click sobre <i>Goals</i>	Se muestra la vista Goals	OK
3	Click sobre <i>Workout</i>	Se muestra la vista Workout	OK
4	Click sobre <i>Food</i>	Se muestra la vista Food	OK

Tabla 43

ID	4		
Descripción	Barra de navegación inferior		
N	Entrada	Esperado	Verif.
1	Click sobre <i>Home</i>	Se muestra la vista Home	OK
2	Click sobre <i>Goals</i>	Se muestra la vista Goals	OK
3	Click sobre <i>Workout</i>	Se muestra la vista Workout	OK
4	Click sobre <i>Food</i>	Se muestra la vista Food	OK

Tabla 44

ID	5		
Descripción	<i>News Feed</i>		
N	Entrada	Esperado	Verif.
1	Click sobre un artículo	Se muestra la vista de Artículo, con el artículo seleccionado como contenido	OK
2	Click sobre botones de carrusel	Se navega entre los artículos	OK

Tabla 45

ID	6		
Descripción	<i>Formulario de Core goals</i>		
N	Entrada	Esperado	Verif.
1	Calcular sin cambiar valores	Se calculan los valores de referencia según las preferencias de usuario	OK
2	Cambio del desplegable <i>Goal</i>	Se modifica el objetivo actual (Perder peso, mantenerlo o subir)	OK
3	Cambio de <i>slider Weight</i>	Se calcula en base al nuevo peso introducido	OK
4	Cambio de <i>slider Activity Level (hr per week)</i>	Se calcula en base al nuevo índice de actividad	OK
5	Calcular con valores cambiados	Se calculan los valores de referencia según los valores introducidos y las preferencias del usuario	OK
6	Guardar nuevos valores (<i>Save</i>)	Se almacenan los nuevos valores de referencia, se actualiza el peso actual si ha cambiado	OK
7	Descartar nuevos valores (<i>Reset</i>)	Se descartan los valores calculados y se reestablece el formulario	OK

Tabla 46

ID	7		
Descripción	Indicadores <i>Core goals</i>		
N	Entrada	Esperado	Verif.
1	Indicador de calorías (<i>Calore intake</i>)	Se actualiza según las entradas del diario de ingesta y las preferencias de usuario	OK
2	Indicador de proteína (<i>Protein</i>)	Se actualiza según las entradas del diario de ingesta y las preferencias de usuario	OK
3	Indicador de grasa (<i>Fat</i>)	Se actualiza según las entradas del diario de ingesta y las preferencias de usuario	OK
4	Indicador de carbohidratos (<i>Carbohydrates</i>)	Se actualiza según las entradas del diario de ingesta y las preferencias de usuario	OK

Tabla 47

ID	8		
Descripción	Añadir un <i>LIFTR Goal</i>		
N	Entrada	Esperado	Verif.
1	No se introducen valores	Mensaje de error en el formulario	OK
2	Sólo se introduce la fecha límite	Mensaje de error en el formulario	OK
3	Sólo se introduce el valor objetivo	Mensaje de error en el formulario	OK
4	Se introduce una cadena de texto en lugar de un entero	Mensaje de error en el formulario	OK
5	Se introduce un decimal en lugar de un entero	Mensaje de error en el formulario	OK
6	Se introducen los valores correctamente	Se actualiza según las entradas del diario de ingesta	OK

Tabla 48

ID	9		
Descripción	Eliminar un <i>LIFTR Goal</i>		
N	Entrada	Esperado	Verif.
1	Se hace click sobre añadir objetivo	Se despliegan los botones de los objetivos posibles, desactivados	OK
2	Se hace click sobre uno de los botones	El botón se vuelve rojo y muestra un icono de una papelera	OK
3	Se hace click sobre el icono de la papelera	Se elimina el indicador de objetivo LIFTR, se elimina de la BBDD, el botón vuelve a estar activado.	OK

Tabla 49

ID	10		
Descripción	Indicadores <i>LIFTR Goal</i>		
N	Entrada	Esperado	Verif.
1	Indicador de peso objetivo (<i>Goal Weight</i>)	Se actualiza según el formulario <i>Core Goals</i> y las preferencias del usuario	OK
2	Indicador de horas de pesas a la semana (<i>Hours of weights per week</i>)	Se actualiza según las sesiones del diario de rutinas	OK

3	Indicador de horas de pesas a la semana (<i>Hours of cardio per week</i>)	Se actualiza según las sesiones del diario de rutinas	OK
4	N/A	Se muestran solo los indicadores de objetivos existentes	OK

Tabla 50

ID		11	Descripción		Ver rutinas de ejercicio y sesiones
N	Entrada	Esperado	Verif.		
1	Se entra en la vista <i>Workout</i>	Se muestran las rutinas por defecto y las personalizadas (si existen)	OK		
2	Se entra en la vista <i>Workout</i>	Se muestran las sesiones anteriores del usuario	OK		

Tabla 51

ID		12	Descripción		Crear una nueva rutina
N	Entrada	Esperado	Verif.		
1	Se hace click sobre el botón “+”	Se muestra el formulario de creación de rutinas	OK		
2	Se deja el formulario en blanco	Se muestra un error en el formulario	OK		
3	Sólo se rellena el campo de “Name”	Se crea la nueva rutina con el nombre indicado y el tipo por defecto	OK		
4	Se introduce un número en el campo “Name”	Se muestra un error en el formulario	OK		
5	Se introduce una cadena de texto en el campo “Name”, sin descripción ni ejercicios	Se crea la nueva rutina con el nombre indicado	OK		
6	Se introduce un nombre válido, una descripción, pero sin ejercicios	Se crea la nueva rutina con el nombre y descripción indicados, con el tipo por defecto	OK		
7	Se introduce el nombre, una descripción y se añade un campo de ejercicios, pero sin rellenarlo	Se muestra un error en el formulario	OK		
8	Se añade un campo nuevo de ejercicios y se borra.	Se elimina el campo dinámico de ejercicio	OK		
9	Se introduce el nombre, una descripción, se selecciona el tipo “Cardio” y se añade un campo de ejercicios, seleccionando un ejercicio	Se crea la nueva rutina con el nombre, tipo, descripción y ejercicio indicados	OK		
10	Se introduce el nombre, una descripción, se selecciona el tipo “Weight” y se añade un campo de ejercicios, seleccionando un ejercicio, pero sin rellenar los “Sets”	Se muestra un error en el formulario	OK		
11	Se introduce el nombre, una descripción, se selecciona el tipo “Weight” y se añade un campo de ejercicios, seleccionando un	Se crea la nueva rutina con el nombre, tipo, descripción y ejercicio indicados	OK		

	ejercicio y rellenando el campo "Sets"		
--	--	--	--

Tabla 52

ID		13		
Descripción		Modificar una rutina		
N	Entrada	Esperado	Verif.	
1	Se hace click sobre el botón de edición	Se muestra el formulario de edición de rutinas, con los valores de la rutina seleccionada	OK	
2	Se borran todos los campos	Se muestra un error en el formulario	OK	
3	Sólo se rellena el campo de "Name"	Se modifica la rutina correctamente, eliminando los campos sobrantes	OK	
4	Se introduce un número en el campo "Name"	Se muestra un error en el formulario	OK	
5	Se introduce una cadena de texto en el campo "Name", sin descripción ni ejercicios	Se modifica la rutina correctamente, eliminando los campos sobrantes	OK	
6	Se introduce un nombre válido, una descripción, pero sin ejercicios	Se crea la nueva rutina con el nombre y descripción indicados	OK	
7	Se introduce el nombre, una descripción y se añade un campo de ejercicios, pero sin rellenarlo	Se muestra un error en el formulario	OK	
8	Se introduce el nombre, una descripción, se selecciona el tipo "Cardio" y se añade un campo de ejercicios, seleccionando un ejercicio	Se modifica la rutina correctamente, eliminando los campos sobrantes	OK	
9	Se introduce el nombre, una descripción, se selecciona el tipo "Weight" y se añade un campo de ejercicios, seleccionando un ejercicio, pero sin rellenar los "Sets"	Se modifica la rutina correctamente, eliminando los campos sobrantes	OK	
10	Se introduce el nombre, una descripción, se selecciona el tipo "Weight" y se añade un campo de ejercicios, seleccionando un ejercicio y rellenando el campo "Sets"	Se modifica la rutina correctamente, eliminando los campos sobrantes	OK	
11	Se hace click nuevamente sobre el botón de edición	Se cierra el formulario	OK	

Tabla 53

ID		14		
Descripción		Eliminar una rutina		
N	Entrada	Esperado	Verif.	
1	Se hace click sobre el botón de edición de una rutina propia	Se muestra el formulario de edición y el botón de eliminar	OK	
2	Se hace click sobre el botón de eliminación	Se elimina la rutina de la BBDD y se actualiza la vista	OK	

Tabla 54

ID	15		
Descripción	Ver una sesión anterior		
N	Entrada	Esperado	Verif.
1	Se entra en la vista <i>Workout</i>	Se muestran las rutinas por defecto y las personalizadas (si existen)	OK
2	Se selecciona una sesión	Se muestran los datos de la sesión correctamente en una ventana modal	OK

Tabla 55

ID	16		
Descripción	Hacer una sesión de una rutina		
N	Entrada	Esperado	Verif.
1	Se entra en la vista <i>Workout</i>	Se muestran las rutinas por defecto y las personalizadas (si existen)	OK
2	Se hace click sobre "Play" al lado de la rutina deseada	Se muestra la vista <i>Workout Session</i> , con un temporizador y un campo por cada ejercicio de la rutina a realizar. Se toma la fecha y tiempo de comienzo de rutina.	OK
3	Se hace click sobre "Play" en el temporizar	El temporizador empieza	OK
4	Se pierde el <i>focus</i> de la ventana	El temporizador continúa, sin interrupciones	OK
5	Se hace click sobre "Pause" en el temporizador	El temporizador se pone en pausa, manteniendo el tiempo anterior	OK
6	Se hace click sobre "Play" nuevamente	El temporizador continúa desde el tiempo de pausa	OK
7	Se hace click sobre "Reset" en el temporizador	El temporizador vuelve a 0	OK
8	Se hace click sobre el botón de edición de un ejercicio	Se abre el formulario de repeticiones y peso, o duración (según tipo de rutina)	OK
9	Se dejan los campos por defecto	Se añade un nuevo <i>Set</i> o entrada al ejercicio (según tipo de rutina)	OK
10	Se introduce una cadena en lugar de un número	Se convierte a número, si no es posible, da error	OK
11	Se introduce un número	Se registra el nuevo <i>Set</i> o entrada al ejercicio (según tipo de rutina)	OK
12	Se hace click sobre el botón "End session"	Se devuelve al usuario a la vista de <i>Workout</i> , registrando la nueva sesión y el tiempo de finalización, calculando la duración	OK

Tabla 56

ID	17		
Descripción	Visualización de datos actuales del diario de ingesta (<i>Food</i>)		
N	Entrada	Esperado	Verif.
1	Se entra en la vista <i>Food</i>	Se muestran las calorías restantes, las consumidas y la proteína, grasa y carbohidratos consumidos el día de hoy	OK

Tabla 57

ID	18		
Descripción	Crear una nueva entrada en el diario de ingesta		
N	Entrada	Esperado	Verif.
1	Se hace click sobre “+”	Se navega a la vista de búsqueda de alimentos	OK
2	Se busca un elemento “Apple”	Se muestran los resultados que contengan “Apple” en el nombre	OK
3	Se selecciona un elemento	Se muestra la vista de resultados del elemento, con su contenido nutricional	OK
4	Se deja el campo de cantidad en blanco	Se muestra un error en el formulario	OK
5	Se introduce una cadena	Se transforma a número, si no se puede, se muestra error	OK
6	Se introduce un número	Se registra la nueva entrada, devolviendo al usuario al diario de ingesta, con los nuevos datos	OK

Tabla 58

ID	19		
Descripción	Editar una entrada en el diario de ingesta		
N	Entrada	Esperado	Verif.
1	Se hace click sobre el botón de edición	Se navega a la vista de edición de entradas	OK
2	Se deja el campo de cantidad en blanco	Se muestra un error en el formulario	OK
3	Se introduce una cadena de texto en el campo de cantidad	Se muestra un error en el formulario	OK
4	Se introduce un número válido en la cantidad	Se actualiza la entrada y se devuelve al usuario a la vista del diario de ingesta	OK

Tabla 59

ID	20		
Descripción	Eliminar una entrada en el diario de ingesta		
N	Entrada	Esperado	Verif.
1	Se hace click sobre el botón de edición	Se navega a la vista de edición de entradas	OK
2	Se hace click sobre el botón de eliminación	Se elimina la entrada de la BBDD y se devuelve al usuario a la vista del diario de ingesta, con los valores actualizados	OK

Tabla 60

ID	21		
Descripción	Editar las preferencias de usuario		
N	Entrada	Esperado	Verif.
1	Se cambia el género	Se actualiza el perfil con el nuevo género	OK
2	Se cambia el peso	Se actualiza el perfil con el nuevo peso	OK
3	Se cambia la altura	Se actualiza el perfil con la nueva altura	OK
4	Se cambia la edad	Se actualiza el perfil con el nuevo peso	OK

Tabla 61

ID	22		
Descripción	Registrarse		
N	Entrada	Esperado	Verif.
1	Se dejan los campos en blanco	Se muestra un error en el formulario	OK
2	Se rellena sólo el campo "Username"	Se muestra un error en el formulario	OK
3	Se rellena sólo el campo "Password"	Se muestra un error en el formulario	OK
4	Se rellena sólo el campo "Weight"	Se muestra un error en el formulario	OK
5	Se rellena sólo el campo "Height"	Se muestra un error en el formulario	OK
6	Se rellena sólo el campo "Age"	Se muestra un error en el formulario	OK
7	Se escribe un usuario con símbolos	Se muestra un error en el formulario	OK
8	Se escribe una contraseña de menos de 8 caracteres	Se muestra un error en el formulario	OK
9	Se rellenan solo los campos "Username" y "Password"	Se muestra un error en el formulario	OK
10	Se rellenan los campos "Username", "Password" y "Weight"	Se muestra un error en el formulario	OK
11	Se rellenan los campos "Username", "Password" y "Height"	Se muestra un error en el formulario	OK
12	Se rellenan los campos "Username", "Password" y "Age"	Se muestra un error en el formulario	OK
13	Se rellenan los campos "Username", "Password", "Weight" y "Height"	Se muestra un error en el formulario	OK
14	Se introduce una cadena de texto en "Weight", "Height" o "Age"	Se muestra un error en el formulario	OK
15	Se introduce un decimal en "Weight", "Height" o "Age"	Se muestra un error en el formulario	OK
16	Se introduce un usuario que ya existe	Se muestra un error en el formulario	OK
17	Se rellenan los campos correctamente	El usuario es dado de alta con los datos indicados y es redirigido a la vista de <i>Login</i>	OK

7. Progreso con respecto a la planificación

7.1. PEC 2

Durante este periodo se ha podido mantener el ritmo de trabajo necesario para cumplir con la mayoría de los puntos de la planificación, realizada en la entrega anterior, para esta segunda entrega. La realización del análisis de requisitos del proyecto, estimado en 2 semanas, se ha llevado a cabo en el tiempo estipulado y sin mayores contratiempos. La parte de diseño, estimada en 11 días, se ha realizado casi por completo. Se ha decidido no realizar todos los diseños finales y, en su lugar, finalizar el diseño de los elementos de las vistas. La inversión de esfuerzo y tiempo en la realización de diseños finales para todas las vistas era mucho mayor que el beneficio, puesto que durante la fase de implementación y de testeo podría darse el caso de que el diseño es vistoso, pero poco funcional. Sin embargo, si se puede consolidar el estilo, colores y diseño de los elementos que compondrán las vistas.

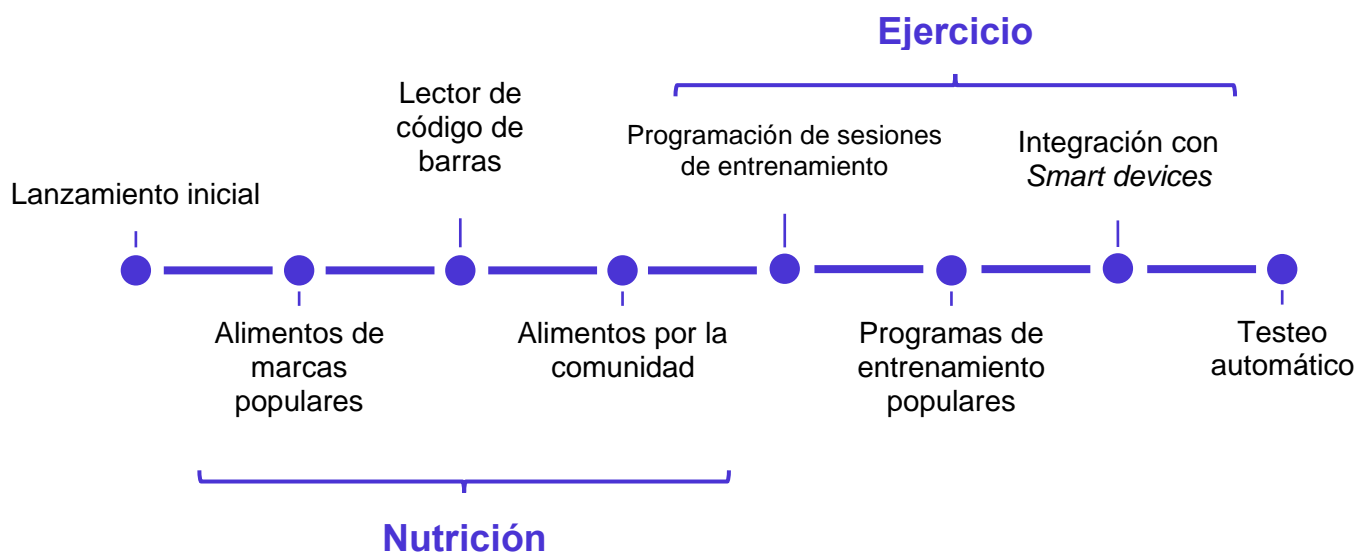
7.2. PEC 3

Durante este periodo se ha podido llevar a cabo todo el desarrollo dentro del plazo establecido. Pese a ciertos obstáculos e imprevistos que han causado retrasos, los objetivos establecidos para esta entrega han sido cumplidos. Sin embargo, si se han tenido que recortar ciertas funcionalidades o acciones para poder cumplir con la entrega. Algunas de estas funcionalidades quedan anotadas para su posible posterior implementación, pero no llegarán a estar presentes en la entrega final. Con lo que respecta el producto final, la aplicación es fiel a los diseños realizados y a los estilos propuestos, cumple con los casos de uso y requisitos; y es completamente funcional.

7.3. PEC 4

Este periodo marca el final y la entrega del trabajo. Se ha realizado todo el trabajo de corrección, revisión y publicación de la memoria, así como la generación de la presentación. De esta última, se ha realizado también un vídeo con un explicativo de la presentación y una demostración del funcionamiento de la aplicación. Se han cumplido con todos los objetivos de esta entrega, sin ningún tipo de obstáculo mayor.

8. Futuro



El producto presentado hasta ahora ofrece los pilares básicos necesarios para que la aplicación sea una herramienta de gran utilidad en el seguimiento de la alimentación y el ejercicio, sin embargo, presenta ciertas carencias en funcionalidades más avanzadas. Algunas de estas son las expuestas en la figura anterior.

La expansión de la base de datos de alimentos, así como la posibilidad de crear nuevos alimentos propios, extiende la funcionalidad del diario de ingesta a virtualmente cualquier alimento. Estos datos además pueden ser retroalimentados a la propia aplicación, poniendo así a disposición del resto de la comunidad de usuarios estos nuevos alimentos.

De manera general, los usuarios tendrán sus propias rutinas de ejercicio. Estas rutinas suelen llevarse a cabo los mismos días de la semana, semana tras semana. Por otro lado, este conjunto de rutinas se puede agrupar en un "programa". Si se permitiese a los usuarios programar sus entrenamientos, así como crear y compartir sus programas con el resto de la comunidad, no sólo se enriquecería el abanico de funcionalidades de este módulo, sino que fomentaría la interacción entre usuarios dentro de la propia plataforma, consolidando así el interés del usuario hacia esta.

9. Conclusiones

La realización del trabajo de fin de grado es el resultado de los conocimientos adquiridos durante la carrera, ya que se han trabajado las diversas facetas de las asignaturas que lo componen. El proceso ha sido muy satisfactorio, dado que me ha permitido conocer de primera mano lo que representa la realización de un proyecto de desarrollo de principio a fin, con una idea y una metodología propias.

Las tecnologías empleadas en este proyecto las llevo empleando desde hacer varios años, pero este trabajo me ha hecho afrontar nuevos campos de conocimiento como la autenticación, el CI/CD y todo el trabajo asociado a la realización de un diseño para su posterior implementación, con tiempo y recursos limitados. Estos factores han tenido gran influencia sobre el resultado final, donde se debía tener cautela a la hora de aumentar la complejidad del sistema, ya que esto podría conllevar el no poder entregarlo a tiempo.

Personalmente, estoy muy satisfecho con el trabajo realizado, dado que este incluye las funcionalidades y planteamiento planteadas inicialmente, además de los materiales acompañantes, como son la presentación y esta memoria. Cabe mencionar también que la planificación fue cumplida y que el análisis de riesgos inicial fue apta para garantizar el éxito del trabajo.

10. Glosario

Término	Definición
ReactJS	<i>Framework</i> de JavaScript para el desarrollo de interfaces en páginas web.
NodeJS	Entorno de ejecución de JavaScript para el desarrollo de aplicaciones de servidor.
Express	<i>Framework</i> de JavaScript para el desarrollo de servidores web con NodeJS.
MongoDB	Base de datos NoSQL basada en documentos BSON.
App Service	Servicio de publicación de aplicaciones de Microsoft Azure.
CI/CD	<i>Continuous Integration</i> , integración automatizada desde el repositorio de código. <i>Continuous Deployment</i> , despliegue automatizado de versiones al servicio de publicación o App Service.
SPA	<i>Single Page Application</i> , aplicación de una sola página, no tiene que recargar al navegar entre vistas.
PWA	<i>Progressive web application</i> , tecnología web que adapta aplicaciones web de navegador a un entorno nativo del SO.
Service Worker	<i>Script</i> de navegador que se ejecuta de manera separada a una página web, permite la implementación de funcionalidades <i>offline</i> , características de aplicaciones nativas
API	<i>Application Programming Interface</i> , intermediario de <i>software</i> que permite la comunicación entre aplicaciones.
Bug	Error de programación o sistema.
<i>Endpoint</i>	Nodo remoto que comunica con la red en la que está conectado.
<i>Front End</i>	Capa de presentación.
<i>Back End</i>	Capa de negocio y persistencia.
<i>Wireframe</i>	Boceto del diseño de una vista de la capa de presentación.
UI	<i>User Interface</i> , interfaz de usuario.
<i>Feedback</i>	Respuesta o notificación de un sistema al usuario para indicar un cambio de estado en la aplicación o evento.
<i>Commit</i>	Guardado de cambios en Git.
<i>Branch</i>	Rama de desarrollo de Git.
<i>Framework</i>	Librería de funcionalidades o código, con una metodología de diseño específica, que permite la rápida implementación de soluciones software.
<i>MERN</i>	Conjunto de las tecnologías MongoDB, Express, ReactJS, NodeJS.
<i>JSON</i>	<i>JavaScript Object Notation</i> , formato de intercambio de datos.
<i>BSON</i>	Variante de JSON.
<i>BBDD</i>	Base de datos.
<i>TAD</i>	Tipo de dato abstracto.
<i>IDE</i>	<i>Integrated development environment</i> , entorno de herramientas para el desarrollo de software.
<i>JWT</i>	<i>JSON Web Token</i> , estándar de codificación de <i>tokens</i> de autenticación en JSON.
<i>Bcrypt</i>	Librería criptográfica en JavaScript.

<i>Props</i>	Propiedades o parámetros que recibe un componente de ReactJS.
<i>DOM</i>	<i>Document Object Model</i> , interfaz de programación para documentos HTML y XML.
<i>Hook</i>	En software, es un concepto que permite la modificación del comportamiento de un programa sin cambiar el código de la clase.
<i>Context</i>	Un componente de inyección de props en ReactJS.

11. Bibliografía

- (s.f.). Obtenido de PassportJS: <http://www.passportjs.org/>
- (s.f.). Obtenido de Express JS: <https://expressjs.com/>
- Ant Design Documentation*. (s.f.). Obtenido de Ant Design: <https://ant.design/>
- Azure Documentation*. (s.f.). Obtenido de Microsoft Azure: <https://docs.microsoft.com/en-us/azure/?product=featured>
- Documentation NodeJS*. (s.f.). Obtenido de NodeJS: <https://nodejs.org/en/docs/>
- Facebook. (s.f.). *React Documentation*. Obtenido de ReactJS: <https://reactjs.org/docs/getting-started.html>
- Local Authentication Using Passport in Node.js*. (08 de April de 2020). Obtenido de SitePoint: <https://www.sitepoint.com/local-authentication-using-passport-node-js/>
- Moraites, A. (13 de November de 2018). *Deploy a React app with Node.js*. Obtenido de Medium.com: <https://medium.com/@achillesmoraites/serve-a-react-app-with-express-server-c5986769bac>
- Service Workers: an Introduction | Web Fundamentals*. (s.f.). Obtenido de Google Developers: <https://developers.google.com/web/fundamentals/primers/service-workers>
- The MongoDB 4.4 Manual*. (s.f.). Obtenido de MongoDB: <https://docs.mongodb.com/manual/>
- William D. McArdle, F. I. (2010). *Essential Exercise Physiology*. Lippincott Williams and Wilkins.

12. Anexos

- Contenido del fichero .zip de la entrega:
 1. Código fuente de la última versión en el día de la publicación.
 2. Memoria en formato PDF.
 3. Presentación en formato PPT y PDF.
 4. Vídeo de la presentación en formato MP4.