



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

GRADO EN INGENIERÍA INFORMÁTICA

Tecnologías de la información

TRABAJO FIN DE GRADO

Plataforma de entrenamiento basada en inteligencia artificial y
tecnologías web para la recomendación de rutinas de ejercicio
personalizadas

Tamara Redondo Soto

Septiembre, 2024



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

Departamento de Tecnologías y Sistemas de Información

Tecnologías de la información

TRABAJO FIN DE GRADO

**Plataforma de entrenamiento basada en inteligencia
artificial y tecnologías web para la recomendación de
rutinas de ejercicio personalizadas**

Autora: Tamara Redondo Soto

Tutor(a): Santiago Sánchez Sobrino

Septiembre, 2024

Plataforma de entrenamiento basada en inteligencia artificial y tecnologías web
© Tamara Redondo Soto, 2024

Este documento se distribuye con licencia CC BY-NC-SA 4.0. El texto completo de la licencia se puede obtener en <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

La copia y distribución de esta obra está permitida en todo el mundo, sin regalías y por cualquier medio, siempre que esta nota sea preservada. Se concede permiso para copiar y distribuir traducciones de este libro desde el español original a otro idioma, siempre que la traducción sea aprobada por el autor del libro y tanto el aviso de copyright como esta nota de permiso, sean preservados en todas las copias.

Este texto ha sido preparado con la plantilla L^AT_EX para Trabajo Fin de Estudios en Ingeniería Informática para la UCLM publicada por [Jesús Salido](#) en el repositorio público Zenodo, DOI: [10.5281/zenodo.4561708](https://doi.org/10.5281/zenodo.4561708), como parte del curso «[L^AT_EX esencial para preparación de TFG, tesis y otros documentos académicos](#)» impartido en la Escuela Superior de Informática de la Universidad de Castilla-La Mancha. Si la empleas para preparar tu TFG, te agradeceré que la cites [51] e incluyas en tus referencias como se indica en Zenodo con el DOI suministrado para todas las versiones.



TRIBUNAL:

Presidente: _____

Vocal: _____

Secretario(a): _____

FECHA DE DEFENSA: _____

CALIFICACIÓN: _____

PRESIDENTE

VOCAL

SECRETARIO(A)

Fdo.:

Fdo.:

Fdo.:

*A mis padres
por su incondicional amor y apoyo durante todo este proceso.
Gracias por ser mi mayor fuente de fortaleza.*

Resumen

El deporte desempeña un papel crucial para el desarrollo personal, brindando un medio apropiado para el crecimiento y superación en abundantes facetas de la vida. Sumado a los beneficios evidentes para la salud física, el deporte fomenta la disciplina, instruida mediante la práctica continua, la resiliencia, madurada a través de los desafíos y las adversidades, la autoconfianza, definiendo aspectos determinantes de la mente y el carácter.

De manera creciente, se ha apostado por difundir la importancia de un estilo de vida activo, logrando así una tendencia mundial hacia una mayor motivación en la práctica de actividades físicas. Por consiguiente, el aumento en la conciencia sobre la importancia del ejercicio ha llevado a un incremento en el número de personas que forman parte en estas actividades y un número muy bajo de personal cualificado necesario para asesorar personalmente a los deportistas. En ese sentido, esta situación también ha contribuido en la propagación de desinformación tanto entre los usuarios como en el personal de entrenamiento. La facilidad de acceso a información no verificada ni acorde a su perfil, genera seguridad en los usuarios para gestionar su propio entrenamiento, dando lugar a consecuencias tales como: lesiones graves por falta de conocimiento propio sobre la técnica, planificación ineficiente de las rutinas, hasta falta de motivación y enfoque.

Este trabajo fin de grado aborda la problemática con el desarrollo de una plataforma de entrenamiento versátil, que más allá de servir rutinas de ejercicios predefinidas, ofrece rutinas personalizadas adaptadas al perfil del usuario y al entorno de entrenamiento elegido y así poder aprovechar al máximo su potencial. El sistema de recomendaciones se apoya en la Inteligencia Artificial para una mayor personalización de las rutinas. A su vez, la plataforma permite realizar un seguimiento del progreso con una interfaz amigable y se incluyen mecanismos de gamificación para mayor motivación. Y finalmente, se integra un asistente virtual para ofrecer respuestas personalizadas y se estudia la posible implementación de una aplicación para relojes inteligentes.

Palabras clave: TFG, Inteligencia Artificial, fitness, sistema web, UCLM.

Abstract

Sport plays a crucial role in personal development, providing a suitable means for growth and overcoming various facets of life. In addition to the evident health benefits, sports promote discipline, learned through continuous practice, resilience, matured through challenges and adversities, and self-confidence, defining crucial aspects of the mind and character.

Increasingly, there has been a push to promote the importance of an active lifestyle, resulting in a global trend towards greater motivation in engaging in physical activities. Consequently, heightened awareness of the importance of exercise has led to an increase in the number of people participating in these activities, but with a very low number of qualified personnel available to provide personal guidance to athletes. This situation has also contributed to the spread of misinformation among both users and training staff. The ease of access to unverified information not tailored to individual needs provides users with a false sense of confidence in managing their own training. This can lead to consequences such as serious injuries due to lack of proper technique knowledge, inefficient planning of routines, and even a lack of motivation and focus.

This final degree project addresses the challenges in developing a versatile training platform that goes beyond providing predefined exercise routines by offering personalized routines tailored to the user's profile and chosen training environment, thereby maximizing their potential. The recommendation system leverages Artificial Intelligence for greater personalization of routines. Additionally, the platform allows for progress tracking with a user-friendly interface and includes gamification mechanisms for increased motivation. Finally, a virtual assistant is integrated to offer personalized responses, and the potential implementation of a smartwatch application is explored.

Keywords: TFG, Artificial Intelligence, fitness, web system, UCLM.

Agradecimientos

Primero y ante todo, quiero agradecer de corazón a todos los profesores de la Escuela Superior de Informática de Ciudad Real. Su constante entrega y compromiso han sido de gran importancia en mi desarrollo académico. Un reconocimiento especial a mi tutor, Santiago Sánchez Sobrino, por darme la oportunidad de realizar este Trabajo Fin de Grado y por su constante orientación y apoyo a lo largo de todo el proceso.

A mis padres, Pilar y Agustín, que siempre han estado ahí para mí. Gracias por depositar tanta confianza en mí y brindarme tan excelente educación. No habría logrado llegar tan lejos sin vosotros.

A Alex, mi otra mitad, por estar a mi lado en cada paso del camino. Tu presencia y apoyo incondicional han sido indispensables.

Finalmente, quiero agradecer a mis compañeros de clase, por contribuir a que esta etapa universitaria sea memorable.

*Tamara Redondo Soto
Ciudad Real, 2024*

Acrónimos

LISTA DE ACRÓNIMOS

ACID	:	Atomicity, Consistency, Isolation, Durability:, 45, 46
AI	:	Artificial Intelligence, 26
API	:	Application Programming Interface, 6, 10, 15, 38, 39, 40, 41 42, 46, 57, 63, 78, 79
AR	:	Augmented Reality, 17
ATRC	:	Advanced Technology Research Center, 20
AWS	:	Amazon Web Services, 6, 77
BERT	:	Bidirectional Encoder Representations from Transformers, 17
BLOOM	:	Big Learning Optimization Model, 20
BSON	:	Binary javaScript Object Notation, 15
CLI	:	Command Line Interface, 36
Cloc	:	Count Lines of Code, 76
CRUD	:	Create, Read, Update, Delete, 15, 42, 46
CSS	:	Cascading Style Sheets, 9, 11, 12
DALL-E	:	Deep Learning Image Generation Model, 17
DBMS	:	Database Management System, 11, 15, 43, 45
DNS	:	Domain Name System, 57
DOM	:	Document Object Model, 12, 13
DTO	:	Data Transfer Object, 41, 42
ECDSA	:	Elliptic Curve Digital Signature Algorithm, 48
ECS	:	Elastic Container Service, 77
EE	:	Enterprise Edition, 36
FN	:	Forma Normal, 45
Gemma	:	Granular and Efficient Model, 19
Gen AI	:	Generative Artificial Intelligence, 16
GPT	:	Generative Pre-trained Transformer, 17, 19, 79
GPU	:	Graphics Processing Unit, 20
GQA	:	Generated Questions and Answers, 19, 56
HMAC	:	Hash-based Message Authentication Code, 48
HTML	:	HyperText Markup Language, 9, 11, 12, 35, 40
HTTP	:	Hypertext Transfer Protocol, 6, 10, 15, 40, 41, 49, 53, 78
IA	:	Inteligencia Artificial, 16, 17, 26
IDE	:	Integrated Development Environment, 36
iOS	:	iPhone Operating System, 24
ISO	:	International Organization for Standardization, 67
IU	:	Interfaz de Usuario, 9
JDBC	:	Java Database Connectivity, 37, 42
JDK	:	Java Development Kit, 37
JDT	:	Java Development Tools, 37

JPA	: Java Persistence API, 37, 42
JRE	: Java Runtime Environment, 37
JS	: JavaScript, 12
JSON	: JavaScript Object Notation, 15, 49, 56, 57, 61, 59
JSP	: JavaServer Pages, 37
JWT	: Json Web Token, 39, 48, 49, 50
KISS	: Keep It Simple, Stupid, 7
LLM	: Large Language Model, 17, 18, 51, 53, 55
MMLU	: Massive Multitask Language Understanding, 19
NLP	: Natural Language Processing, 18
OMS	: Organización Mundial de la Salud, 1, 21
ORM	: Object-Relational Mapping, 37, 42
OS	: Operating System, 23
PaLM	: Pathways Language Model, 17
PDA	: Personal Digital Assistant, 10
PI	: Proprietary Information, 17
PLN	: Procesamiento de Lenguaje Natural, 52
POJO	: Plain Old Java Object, 41
POM	: Project Object Model 36
RAG	: Retrieval-Augmented Generation 55
RDB	: Relational Database, 45
RDBMS	: Relational Database Management System, 15, 46
REST	: Representational State Transfer, 6, 15, 40, 57
RFC	: Request for Comments, 48
RSA	: Rivest-Shamir-Adleman, 48
RTE	: Real-Time Environment, 12
SAS	: Statistical Analysis System, 18
SE	: Standard Edition, 37
SMoE	: Sparse Mixture of Experts, 18
SPA	: Single Page Application, 13, 40
SQL	: Structured Query Language, 15, 38, 42, 46
TFG	: Trabajo Fin de Grado, 4, 38, 77
TI	: Tecnología de la Información, 17
TII	: Technology Innovation Institute, 20
UML	: Unified Modeling Language, 43
URI	: Uniform Resource Identifier, 41, 49
URL	: Uniform Resource Locator, 15, 41, 48
UX	: User Experience, 9
VLM	: Visual Language Model, 20
VSC	: Visual Studio Code, 38
XML	: eXtensible Markup Language, 15
1FN	: Primera Forma Normal, 45
2FN	: Segunda Forma Normal, 45
3FN	: Tercera Forma Normal, 45

Índice general

Resumen	V
Abstract	VII
Agradecimientos	IX
Acrónimos	XI
Índice de figuras	XVII
Índice de tablas	XIX
Índice de listados	XXI
1. Introducción	1
1.1. Motivación	1
1.1.1. Estado actual del ámbito deportivo	1
1.1.2. Identificación de tendencias	2
1.1.3. Consecuencias	3
1.1.4. Propuesta	3
1.1.5. Impacto	3
1.2. Estructura del documento	4
2. Objetivo	5
3. Estado del Arte	9
3.1. Estado del arte	9
3.1.1. Arquitecturas de las aplicaciones Web	10
3.1.2. Tecnologías Web	11
3.1.3. Estado actual de la Inteligencia Artificial	16
3.1.4. Aplicaciones y Sistemas Web	21
4. Metodología	31
4.1. Metodologías de Desarrollo Software	31
4.1.1. Metodologías Ágiles	31
4.1.2. Metodología Scrum	32
4.2. Planificación del trabajo	34
4.3. Pila tecnológica	35
4.3.1. Entornos de desarrollo	35
4.3.2. Frontend	36
4.3.3. Backend	37
4.3.4. Orquestación	38

4.3.5. Herramientas de gestión y planificación	38
5. Arquitectura	39
5.1. Visión general	39
5.1.1. Arquitectura frontend y backend del sistema	40
5.1.2. Diseño de la base de datos	43
5.1.3. Validación de formularios	46
5.1.4. Autenticación basada en token	47
5.1.5. Arquitectura del Modelo de Lenguaje Grande	51
5.2. Evaluación de Modelos de Lenguaje Grande	56
5.3. Módulos	63
5.3.1. Sistema de Recomendaciones Personalizadas	63
5.3.2. Creación de formularios	64
5.3.3. Creación de rutinas	65
5.3.4. Mostrar las rutinas	65
5.3.5. Perfil del usuario	65
5.3.6. Creación del análisis	66
5.3.7. Calendario de rutinas	66
5.3.8. Logros	66
5.3.9. Asistente virtual	67
6. Resultados	69
6.1. Sprint 0	69
6.1.1. Estudio de los LLM y sistemas web	69
6.1.2. Elección de la pila tecnológica	69
6.2. Sprint 1	69
6.2.1. Diseño de Onboarding	70
6.2.2. Modificar la información de perfil	70
6.3. Sprint 2	71
6.3.1. Prueba de concepto	71
6.4. Sprint 3	71
6.4.1. Métricas relevantes	72
6.5. Sprint 4	73
6.6. Sprint 5	73
6.6.1. Sistema de notificaciones	73
6.6.2. Asistente virtual	75
6.7. Costes y recursos del proyecto	75
6.8. Estadísticas de código	76
6.8.1. Herramienta cloc	76
7. Conclusiones	77
7.1. Objetivos alcanzados	77
7.2. Justificación de competencias adquiridas	78
7.3. Trabajos derivados y futuros	79
7.4. Valoración personal	79
Bibliografía	81
A. Repositorio	87
B. Manual de Usuario	89
B.1. Inicio de sesión	89
B.2. Registro de usuario	89

B.3. Registro de usuario. Parte 2	89
B.4. Rutina de ejercicios	90
B.5. Perfil de usuario	91
B.6. Análisis personalizado	91
B.7. Calendario	92
B.8. Logros	92
B.9. Chat	94
B.10. Cerrar sesión	94

Índice de figuras

3.1. Organizaciones que han incorporado IA	16
3.2. Aplicación Google Fit	25
3.3. Aplicación Jefit	26
3.4. Aplicación Caliber Fitness	27
3.5. Aplicación Freeletics	27
3.6. Aplicación Fitbod	28
4.1. Cronograma de Trabajo - Diagrama Gantt	35
5.1. Arquitectura general de la aplicación web	39
5.2. Diagrama de clases general del proyecto	44
5.3. Proceso de generación de prompts y funcionamiento de los LLM	55
5.4. Diagrama de secuencia del flujo de creación de rutinas	64
6.1. Login y Registro de usuarios	70
6.2. Proceso de OnBoarding	71
6.3. Visualización de rutinas y registro de actividad	72
6.4. Métricas relevantes	72
6.5. Perfil de usuario	73
6.6. Análisis personalizado	74
6.7. Calendario de rutinas	74
6.8. Visualización de la rutina del día seleccionado	74
6.9. Sistema de logros	75
6.10. Asistente virtual	75
6.11. Número de líneas de código en el Backend	76
6.12. Número de líneas de código en el Frontend	76
B.1. Inicio de sesión	89
B.2. Registro de usuario	90
B.3. Primera pregunta del formulario	90
B.4. Última pregunta del formulario	90
B.5. Pantalla de carga de la generación de la rutina	91
B.6. Disposición de los ejercicios de la rutina	91
B.7. Perfil de usuario	92
B.8. Modificación del perfil de usuario y actualización de las rutinas	92
B.9. Disposición y actualización del análisis personalizado	93
B.10. Disposición del calendario	93
B.11. Visualización de una rutina completada	94
B.12. Disposición de los logros y trofeos	95
B.13. Descripción asociada a un logro	95
B.14. Disposición del chat	96

Índice de tablas

3.1.	Tabla comparativa Modelos de Lenguaje Grande más actuales	22
3.2.	Tabla comparativa aplicaciones y sistemas web analizados	29
5.1.	Métricas de rendimiento de Mistral, Llama 2 y Llama 3	63
6.1.	Costes y recursos del proyecto	76

Índice de listados

5.1.	Validadores del registro de usuario	47
5.2.	Validación del campo username en el formulario de registro	48
5.3.	Token generado en el endpoint de autenticación	48
5.4.	Validación del JWT	50
5.5.	Verificación de la firma JWT	51
5.6.	Implementación de roles en el mensaje inicial	54
5.7.	Envío del historial de mensajes al modelo de lenguaje	54
5.8.	Agregación de mensajes al historial	54
5.9.	Rutina generada por Mistral en la prueba 1	58
5.10.	Primera rutina generada por Mistral en la prueba 2.(1 de 6)	58
5.11.	Información adicional no solicitada. Mistral	59
5.12.	Rutina generada por Llama 2 en la prueba 1	60
5.13.	Rutina generada por Llama 2 en la prueba 2.(1 de 1)	61
5.14.	Información adicional no solicitada. Llama 2	61
5.15.	Rutina generada por Llama 3 en la prueba 1	62
5.16.	Primera rutina generada por Llama 3 en la prueba 2.(1 de 9)	62
5.17.	Información adicional no solicitada. Llama 3	62

CAPÍTULO 1

Introducción

1.1. MOTIVACIÓN

El presente Trabajo Fin de Grado surge de la necesidad de lidiar con el desconocimiento que afecta al ámbito del entrenamiento deportivo, enfatizando la importancia de la asistencia y gestión de un entrenamiento individualizado para cada perfil de usuario. El proyecto aspira a ser un incentivo de cambio, utilizando la innovación y la tecnología para ofrecer una solución íntegra, mejorando la calidad del asesoramiento y la experiencia de entrenamiento de los individuos y enfrentando las limitaciones actuales del sector.

1.1.1. Estado actual del ámbito deportivo

La falta de actividad física, o estilo de vida sedentario, representa un desafío global significativo que afecta la salud de la población en todo el mundo. Para abordar este problema, la Organización Mundial de la Salud (OMS) ha destacado la importancia de mantener una rutina de ejercicios [45], generando una mayor conciencia global sobre los beneficios que aporta el ejercicio, respaldando a los gobiernos en el desarrollo de políticas nacionales y promoviendo la cooperación internacional. También, ha impulsado la construcción de infraestructuras, programas educativos y de formación y ha incentivado la investigación en este ámbito. Aunque en su informe no proporciona estadísticas ni cifras precisas, se señala un aumento mundial en la conciencia sobre los beneficios del deporte, lo que muestra una inclinación creciente hacia un modo de vida más activo y saludable.

Al referirnos a la participación deportiva, en 2022 más del 50 % de individuos mayores de 15 años participaron en actividades deportivas, lo que representa un aumento significativo comparado con los niveles anteriores a COVID-19 [18, p. 25].

En términos demográficos, la realización de ejercicio es más alta entre los jóvenes, con más del 80 %, y disminuye con la edad, llegando alrededor del 16 % entre las personas mayores de 75 años. Se ha observado un incremento en la participación en actividades deportivas en la mayoría de los grupos de edad. Además, los hombres tienden a practicar deporte con más frecuencia que las mujeres, mostrando una marcada disparidad en la práctica a lo largo del año.

A lo largo del tiempo, las preferencias y tendencias en el ámbito deportivo han evolucionado notablemente [15], mostrando cambios en las preferencias de los deportistas y en las innovaciones tecnológicas y metodológicas en el entrenamiento físico. Tradicionalmente, disciplinas como el fútbol y el tenis han dominado el panorama deportivo global, captando a millones de participantes y espectadores gracias a su popularidad y accesibilidad. No obstante, en los últimos años se ha observado una inclinación hacia modalidades deportivas más dinámicas y especializadas entre los más jóvenes. Algunos ejemplos destacados incluyen el *CrossFit*, un programa de entrenamiento funcional de alta intensidad que mezcla ejercicios aeróbicos, levantamiento de pesas y gimnasia para incrementar la resistencia, fuerza y flexibilidad. El *CrossFit* ha ganado popularidad debido a su enfoque integral en el rendimiento físico y su habilidad para motivar y desafiar a los participantes a superar sus límites y mejorar su condición física.

Además, el *fitness* y los centros deportivos han tenido un aumento significativo en su popularidad en distintas edades, convirtiéndose en una parte esencial del estilo de vida saludable y activo de numerosas personas. Este fenómeno se debe a varios factores que han contribuido a su creciente relevancia y atractivo.

Uno de los factores clave en el panorama deportivo actual es la popularidad del *running* y caminar. El *running*, continúa atrayendo a jóvenes que buscan mejorar su resistencia y salud física. Por otro lado, caminar ha adquirido importancia especialmente entre adultos mayores como una actividad accesible y beneficiosa, permitiéndoles mantenerse activos de manera relajada.

Otro factor importante es el acceso a información deportiva [36]. La televisión continúa siendo la principal fuente de información deportiva, seguida por internet y plataformas digitales deportivas, las cuales están en alza, principalmente entre la población más joven. La prensa general y deportiva, así como el uso de dispositivos móviles y redes sociales, juegan un papel importante en la obtención de información deportiva. Estos datos muestran una tendencia hacia el uso de medios digitales y tecnológicos para mantenerse informado sobre noticias y eventos deportivos. Este acceso varía dependiendo del género y la edad, siendo mayor en hombres y en personas más jóvenes, observándose diferencias generacionales evidentes, con una menor aceptación entre individuos mayores de 75 años.

La participación en actividades deportivas también es influenciada por el factor de la motivación que refleja la necesidad física y emocional de las personas. Una de las principales motivaciones es el deseo de mantenerse en forma, un objetivo que es especialmente importante para las mujeres, quienes consideran la actividad física como una forma de cuidar su salud y bienestar general. Esta motivación no se limita a los beneficios estéticos, también se incluye la prevención de enfermedades y la promoción de un estilo de vida dinámico y saludable. En el mundo de las estadísticas deportivas, esto se refleja en la gran cantidad de participación femenina en deportes como el *running* y el *fitness*, los cuales no solo ayudan a mantener la condición física, sino que también mejoran la salud cardiovascular y el bienestar general.

Por otro lado, la diversión y el entretenimiento son factores esenciales, más comúnmente mencionados por los hombres. Para muchas personas, el deporte no solo representa una manera de mantenerse activo, es una forma de disfrutar a través de la competitividad o la recreación física. En el ámbito deportivo, se da importancia al aspecto social, pues las prácticas deportivas suelen convertirse en oportunidades para socializar y compartir experiencias en compañía de amigos, familiares y comunidades locales. Esto está relacionado con la popularidad de la práctica de deportes como el fútbol, el padel y el tenis entre los hombres, los cuales permiten competir, disfrutar y crear lazos sociales significativos.

1.1.2. Identificación de tendencias

El incremento global en la conciencia sobre los beneficios del ejercicio físico revela un aumento constante en el número de personas que muestran interés en las actividades deportivas. Además, la creciente accesibilidad a instalaciones deportivas, así como numerosos eventos, junto con la difusión de información a través de medios tradicionales digitales como sitios web, redes sociales y blogs, ha propiciado otra forma de difusión acercando a un número creciente de personas que se tome el mundo del deporte como parte esencial de su estilo de vida.

Sin embargo, este crecimiento también ha traído consigo desafíos significativos. Existe una notable desinformación dentro del sector deportivo, que puede afectar la manera en que las personas acceden y comprenden las prácticas deportivas y de acondicionamiento físico.

Es crucial ahora examinar cómo afecta la desinformación en los usuarios y en el entrenamiento. En este último aspecto, la labor del entrenador se ve cada vez más infravalorada, evidenciada por la tendencia a prescindir del monitor. Este menosprecio masivo conlleva la percepción errónea de que cualquier persona puede entrenar por su cuenta, sin ninguna supervisión. De igual modo, existe una falsa confianza en la cantidad de información disponible en internet, tanto para resolver posibles

dudas como para orientar un programa de entrenamiento autodidacta. Esta sobreabundancia de datos puede ser tanto una ventaja como una desventaja para los usuarios, ya que, si bien brinda acceso a una gran cantidad de información, también presenta la dificultad significativa de discernir entre información rigurosa y engañosa. Este escenario genera una situación potencialmente peligrosa [29].

1.1.3. Consecuencias

Factores como la mala ejecución técnica, la elección de ejercicios inadecuados y el desconocimiento de las bases del entrenamiento, posibilitan un aumento en el riesgo de lesiones y sobreentrenamiento. Además de la falta de evaluaciones iniciales detalladas y un análisis exhaustivo para cada deportista que impiden obtener la información necesaria para planificar de manera efectiva las sesiones de entrenamiento. Resultan en programas sin fundamento sólido y aumentando el riesgo de lesiones y sobreentrenamiento.

Asimismo, la falta de resultados visibles, motivación y la percepción errónea sobre el tiempo requerido para ver cambios, además de estas lesiones frecuentes y la pérdida de continuidad por carecer de guía, pueden generar frustración, aumentando la probabilidad de abandono por parte de los usuarios.

1.1.4. Propuesta

En sintonía con las consecuencias mencionadas previamente en la sección 1.1.3, el trabajo fin de grado tiene como objetivo diseñar y desarrollar una plataforma de entrenamiento basada en inteligencia artificial y tecnologías web para mejorar la experiencia de la realización del ejercicio físico mediante la recomendación de rutinas personalizadas y adaptadas a los usuario.

Para prevenir la elección de ejercicios inadecuados, se ha implementado un sistema de recomendaciones considerando la recopilación de datos mediante información personal, objetivos que quiere conseguir y restricciones y condiciones médicas, entre otros aspectos, que logren la presentación de rutinas personalizadas para cada perfil. Así mismo, para superar la barrera del desconocimiento de las bases del entrenamiento, el sistema contará con un asistente virtual disponible para responder todas las cuestiones que puedan surgir durante la ejecución de la rutina. Además, se proporcionarán descripciones detalladas para cada ejercicio. Para tratar la pérdida de continuidad, se permitirá realizar un seguimiento de progreso para que los usuarios puedan visualizar sus avances continuamente. Finalmente, para eludir posibles factores que aumenten la probabilidad de abandono, se incorporarán técnicas de gamificación para que la experiencia sea más atractiva y motivadora.

La propuesta se lleva a cabo mediante la integración de un modelo de inteligencia artificial basado en aprendizaje automático para la recomendación personalizada de rutinas de entrenamiento. Se han realizado pruebas exhaustivas para validar la efectividad del sistema de recomendaciones y del asistente virtual. Además, se emplean métodos de investigación para recopilar datos iniciales sobre las necesidades y preferencias de los usuarios, lo que guía la personalización de las funcionalidades de la plataforma. El desarrollo tecnológico incluye la implementación de técnicas de gamificación para mejorar la experiencia del usuario y fomentar la adherencia al programa de entrenamiento.

1.1.5. Impacto

La creación de una plataforma web de *fitness* tiene el potencial de causar importantes repercusiones socio-económicas favorables. En primer lugar, al facilitar el acceso a programas de ejercicio físico personalizados y efectivos, la plataforma podría desempeñar un papel fundamental en la mejora de la salud pública. Esto puede resultar en una reducción potencial de enfermedades crónicas causadas por la inactividad física, como problemas cardíacos, obesidad y diabetes, que mejora la calidad de vida de los individuos y disminuye los gastos asociados con la atención médica en salud pública y privada.

Además, fomentar la actividad física regular puede resultar en un incremento de la productividad laboral. Las personas físicamente más activas tienden a tener niveles más altos de energía, mejor

concentración y menores tasas de ausencia laboral por enfermedades. Esto no solamente mejora la eficiencia y el rendimiento de los empleados, sino que también colabora en una economía más activa y competitiva.

Desde una perspectiva de empleo, la creación y puesta en marcha de la plataforma podría abrir posibilidades de trabajo en diversos sectores, incluyendo el desarrollo de software, diseño de contenido, soporte técnico y marketing digital. También, podría haber un aumento en la demanda de profesionales del *fitness* y entrenadores personales que puedan aportar y respaldar el uso de la plataforma, lo que sería beneficioso para la industria del *fitness* en su conjunto.

La innovación tecnológica y la competitividad del mercado serían otro aspecto clave. Implementar tecnologías innovadoras en el ámbito del *fitness* no solo destaca la plataforma como líder en su industria, sino que también puede atraer inversiones extra y promover un ambiente favorable para la investigación y desarrollo en salud y bienestar.

1.2. ESTRUCTURA DEL DOCUMENTO

Este proyecto ha optado por seguir la estructura que presenta un TFG en la ESI-UCLM. La estructura del documento incluye siete capítulos y dos anexos, donde cada uno cumple una función concreta:

- **Capítulo 1. Introducción.** En el primer capítulo, se aborda la motivación detrás del desarrollo del Trabajo Fin de Grado. Además, se incluye un análisis del estado actual del tema a abordar, identificando sus principales tendencias y consecuencias y se presenta la propuesta que busca resolver dichas consecuencias, además de justificar el impacto potencial que puede tener dicha propuesta en el sector.
- **Capítulo 2. Objetivo.** El segundo capítulo detalla los objetivos del proyecto. Definiendo el objetivo general, así como varios subobjetivos que desglosan dicho objetivo general en módulos más precisos y tratables.
- **Capítulo 3. Estado del Arte.** En el tercer capítulo, se realiza una revisión exhaustiva del estado del arte, que proporciona una visión general de las estrategias actuales, en forma de tecnologías y herramientas, para abordar las distintas fases del trabajo. Además de analizar plataformas comparables a la propuesta del proyecto con el fin de recopilar información relevante.
- **Capítulo 4. Metodología.** Se centra en explicar la metodología empleada en el desarrollo del proyecto, la planificación llevada a cabo de los hitos principales y la pila tecnológica donde se incluyen las tecnologías y herramientas utilizadas.
- **Capítulo 5. Arquitectura.** El quinto capítulo explica de manera detallada la arquitectura general que sigue el proyecto. Además, se detalla el diseño y la implementación elegida para cada solución propuesta, analizando cómo se aborda el problema, su solución escogida y las posibles ventajas que aporta.
- **Capítulo 6. Resultados.** En el sexto capítulo, se presentan los resultados obtenidos del desarrollo del proyecto. Se evalúa el funcionamiento de la aplicación a través de capturas de pantalla y se analizan tanto los recursos y costes asociados como las estadísticas de código.
- **Capítulo 7. Conclusiones.** El séptimo y último capítulo ofrece una conclusión final del proyecto. Se resumen los objetivos alcanzados y se discuten las dificultades encontradas y la aplicación de las competencias adquiridas durante la realización del proyecto. Además se exploran trabajos derivados y futuros y se finaliza con una valoración personal.
- **Anexo A. Repositorio.** Donde se facilita el enlace al repositorio de GitHub que contiene el código fuente del proyecto.
- **Anexo B. Manual de usuario.** Proporciona un manual de usuario que define las instrucciones a seguir para la utilización del sistema desarrollado.

CAPÍTULO 2

Objetivo

Este trabajo se sitúa en el ámbito interdisciplinario de la salud y la tecnología, centrándose en la aplicación de la informática para mejorar el bienestar físico y mental. El **objetivo principal** de este Trabajo Fin de Grado, en cuanto a su finalidad específica, es el diseño y desarrollo de una plataforma de entrenamiento basada en inteligencia artificial y tecnologías web para mejorar la experiencia de la realización del ejercicio físico mediante la recomendación de rutinas personalizadas y adaptadas a los usuarios.

Se enfoca en abordar la primordial necesidad de personalizar y poner en rigor la práctica deportiva mediante el diseño y desarrollo de una plataforma tecnológica avanzada usando un modelo grande de lenguaje basado en Inteligencia Artificial. También se busca superar otros desafíos actuales como mitigar la elección de ejercicios inadecuados, prevenir lesiones frecuentes, reducir la barrera del desconocimiento de las bases del entrenamiento, evitar la perdida de continuidad y disminuir la probabilidad de abandono.

Para alcanzar el objetivo principal, se han identificado varios **subobjetivos**:

1. Estudio de las tecnologías y herramientas que soporten el desarrollo y despliegue del servicio, incluyendo los sistemas web que tratan de solucionar una problemática similar a la que se enfrenta este proyecto.
 - **Revisión Bibliográfica.** Realizar una exhaustiva investigación por diversos medios, artículos, publicaciones académicas y documentos científicos que aborden problemáticas similares en el ámbito de la salud y el *fitness*. Así, esta revisión permite distinguir las mejores prácticas y los avances tecnológicos que puedan ser empleados en el desarrollo de la plataforma tecnológica.
 - **Análisis Comparativo de Sistemas Web.** Estudiar sistemas web y aplicaciones informáticas existentes que brindan soluciones afines, con el propósito de comprender sus perspectivas, funcionalidades y limitaciones. De esta manera, este análisis comparativo facilita un fundamento sólido para la implementación de características efectivas e innovadoras en la propuesta.
 - **Análisis de Tecnologías Actuales.** Para asegurar el funcionamiento adecuado se deben evaluar diferentes opciones tecnológicas (lenguajes de programación, frameworks, sistemas de gestión de bases de datos, entornos y herramientas de desarrollo) más afines que se integren de manera efectiva con las necesidades en este contexto.
 - **Análisis Comparativo de Modelos de Lenguaje Grande.** Evaluar distintos modelos centrados en el procesamiento del lenguaje natural para identificar las fortalezas y debilidades de cada uno y escoger el más adecuado para llevar a cabo el proyecto.
2. Diseño de una arquitectura web que facilite la escalabilidad automática de la plataforma. La escalabilidad automática es esencial para asegurar que la aplicación pueda hacer frente a aumentos inesperados de tráfico. Facilita la ampliación o reducción de la infraestructura en función de la demanda.

- **Administración de contenedores.** Emplear contenedores, como los proporcionados por Docker, para la gestión automática la escalabilidad de los servicios.
 - **Servicios basados en la nube.** Plataformas como AWS posibilitan desplegar la aplicación de forma escalable haciendo uso de contenedores, facilitando la gestión y la adaptación automática a las necesidades.
3. Diseño y desarrollo de un sistema web, desplegado sobre la arquitectura anterior, que permita al usuario la selección y gestión de sus rutinas de ejercicios.
- **Interfaz de Usuario para la Visualización de Rutinas.** Diseñar una página inicial donde los usuarios puedan visualizar la rutina diaria.
 - **Sistema de Seguimiento de Ejercicios y Rutinas Completadas.** Incorporar un sistema de seguimiento de ejercicios realizados que permita a los usuarios registrar la finalización de cada ejercicio. Asimismo, facilitar el seguimiento y registro de las rutinas completadas.
 - **Sistema de Seguimiento de Progreso del usuario.** Registrar datos a medida que se realizan las rutinas que permitan efectuar un seguimiento del progreso y rendimiento del usuario a lo largo del tiempo.
 - **Actualización del perfil y generación de nuevas rutinas.** Permitir a los usuarios la creación de nuevas rutinas de ejercicio personalizadas, teniendo en cuenta la información actualizada de su perfil.
 - **Calendario interactivo para Gestión de Rutinas y Sesiones de Entrenamiento.** Crear un calendario que permita a los usuarios visualizar las rutinas realizadas y las fechas de referencia de las rutinas de ejercicio programadas. Además de la posibilidad de visualización detallada de dichas rutinas realizadas.
4. Diseño y desarrollo de un sistema de gamificación integrado en el sistema web.
- Incorporar técnicas de gamificación para una experiencia atractiva y motivadora.
 - **Sistema de Puntos, Logros y Recompensas.** A medida que los usuarios completan ejercicios y rutinas, desbloquean logros distribuidos en categorías. Por ende, cada categoría completada recibe una recompensa.
 - Enviar notificaciones personalizadas cada vez que se alcanza un logro para mantener el compromiso de los usuarios con sus rutinas.
5. Especificación e implementación de una API que pueda ser consumida por los sistemas de la plataforma.
- **Desarrollar los métodos que gestionan las peticiones de la API.** Implementar funciones concretas en el backend para gestionar las solicitudes entrantes y llevar a cabo operaciones sobre los datos almacenados en el repositorio. Cada método debe ser diseñado para manejar un tipo específico de solicitud HTTP, como GET, POST, PUT o DELETE, en el contexto de una arquitectura REST, y debe incorporar la lógica necesaria para comunicarse con la base de datos, procesar los datos recibidos y devolver respuestas apropiadas al cliente.
6. Desarrollo de un módulo funcional inteligente que permita obtener recomendaciones de rutinas de ejercicios personalizadas adaptadas a las características de los usuarios y estructuradas para su representación en el sistema web.
- **Recopilación de Datos Personales.** Creación de formularios para obtener información relevante de los usuarios.
 - **Algoritmo de Recomendación:** Implementar un algoritmo que analice los datos recopilados y genere rutinas de ejercicios personalizadas adaptadas a los objetivos y condiciones particulares de cada usuario.

- **Estructuración de las rutinas.** Estructurar las rutinas proporcionadas por el modelo en un formato que pueda ser fácilmente incorporado y mostrado en el sistema web. Esto implica la organización de los ejercicios en secuencias lógicas.

7. Desarrollo de una Plataforma Web Centralizada.

- Diseñar una interfaz amigable y adaptable para pantallas de escritorio y dispositivos móviles.
- Utilizar los principios KISS (Keep It Simple, Stupid) para mantener la simplicidad y usabilidad de la interfaz.
- Aplicar buenas prácticas de diseño para garantizar una experiencia de usuario efectiva e intuitiva.

8. Desarrollo de Módulos Adicionales.

- Aplicar buenas prácticas de desarrollo de software para garantizar robustez y fiabilidad del sistema.
- Identificar y documentar las restricciones técnicas y de diseño que afecten al desarrollo y funcionalidad del sistema.
- **Estudio y análisis de funcionalidad adicional.** Realizar un estudio de viabilidad para la integración de una aplicación para Relojes Inteligentes.

Además del desarrollo de un producto tecnológico innovador, también conlleva la investigación exhaustiva y el análisis de las necesidades y desafíos actuales en el área de la práctica deportiva. Por ello, el trabajo se posiciona en la categoría de “Diseño y desarrollo de «artefactos»”, destacando la relevancia de crear soluciones prácticas y eficientes para estos desafíos localizados. La combinación de investigación y tecnología busca no solo potenciar la experiencia de entrenamiento, sino también fomentar la innovación en el campo del deporte y la tecnología aplicada.

Para cumplir los objetivos mencionados, es crucial enfocarse en los requisitos que la solución debe satisfacer, los cuales guiarán el desarrollo de la plataforma de rutinas de ejercicios. Estos requisitos incluyen tanto las funcionalidades que la solución debe ofrecer como las limitaciones que definen su alcance. En el contexto de un proyecto orientado al desarrollo de un «artefacto», los requisitos se clasifican generalmente en funcionales y no funcionales.

Los **requisitos funcionales** deben asegurar que las funcionalidades clave se implementen de manera efectiva, satisfaciendo las necesidades de los usuarios que eligen hacer ejercicio en casa y aquellos que optan por gimnasios sin contar con asistencia profesional. Es necesario crear un sistema sólido de recomendaciones personalizadas que se apoye en un modelo grande de lenguaje y que tenga en cuenta variables como la edad, el nivel de actividad física y el entorno de entrenamiento. Los usuarios deben poder gestionar su perfil personal y generar nuevas rutinas de ejercicios basadas en su perfil a través de un sistema web centralizado, con capacidad de registro y seguimiento del progreso mediante una interfaz adaptable para dispositivos móviles y de escritorio. Además de incorporar la gamificación para mantener la motivación del usuario a través de notificaciones e integrar un asistente virtual que resuelva consultas específicas sobre las rutinas de ejercicio.

En cuanto a los **requisitos no funcionales**, se da mayor importancia a la usabilidad mediante una interfaz intuitiva y de fácil uso, asegurando un rendimiento óptimo. La seguridad de la información personal será primordial, mediante el uso de cifrado y autenticación robustos, al mismo tiempo que la plataforma se diseñará con una arquitectura escalable y fácil de mantener para permitir futuras actualizaciones y la integración con dispositivos como relojes inteligentes. Asimismo, se garantizará la accesibilidad, la disponibilidad continua y la adecuación a los estándares web.

CAPÍTULO 3

Estado del Arte

3.1. ESTADO DEL ARTE

Para entender de manera más completa los mecanismos de programación web, es importante tener conocimiento del funcionamiento de los mismos.

El proceso de creación y mantenimiento de páginas web, conocido como desarrollo web, implica una amplia variedad de acciones desde el diseño y la creación de códigos, hasta la gestión de servidores, de los contenidos y de las bases de datos. En definitiva, el desarrollo web implica la construcción integral de un sitio web y sus componentes usando lenguajes de programación y aplicaciones.

Para una adecuada división de los procesos de programación, es importante familiarizarse con dos conceptos: frontend y backend [24].

1. **Frontend.** Es la parte del desarrollo web que se ocupa de la interfaz del usuario (IU), esto es, todo aquello que los usuarios ven y con lo que interactúan de manera directa en un sitio web o aplicación. Esta área del desarrollo web engloba tanto la apariencia visual como la interactividad funcional y la experiencia del usuario (UX).
2. **Backend.** Es la parte del desarrollo web que maneja la lógica y la operatividad interna de un sitio web, esto es, todo lo que sucede en segundo plano y no es perceptible para el usuario. Este sector se dedica a administrar servidores, bases de datos y aplicaciones para garantizar el correcto funcionamiento del sitio web.

En consecuencia, hay tres enfoques primordiales en el desarrollo web [32]:

1. **Desarrollador Web Frontend.** Un desarrollador frontend pone atención a la interfaz de usuario de un sitio web. Utiliza sus habilidades en HTML y CSS para manejar la apariencia y la experiencia de uso de un sitio web, frecuentemente ajustándose a diferentes tamaños de pantalla de los dispositivos móviles y de escritorio.

Estos desarrolladores suelen utilizar JavaScript para diseñar una interacción ágil para los usuarios y también, están familiarizados con las bibliotecas y frameworks de JavaScript que pueden agilizar la creación de aplicaciones interactivas para los usuarios finales. Además, se enfoca en mejorar el rendimiento del sitio web, en la optimización de los motores de búsqueda y la accesibilidad del sitio web.

La mayoría de los desarrolladores frontend también trabajan como diseñadores, pero no es un requisito para ser considerado como un profesional del área del frontend. Igualmente, muchos diseñadores web son expertos en HTML y CSS y los pueden emplear en sus diseños, pero no se autodenominan desarrolladores.

2. **Desarrollador Web Backend.** Un desarrollador de backend se enfoca en las aplicaciones del lado del servidor. Esto incluye trabajar con software de servidor web, bases de datos y sistemas operativos, así como dominar varios lenguajes de programación y frameworks específicos para garantizar la eficiencia y seguridad de las aplicaciones en el servidor.

Los desarrolladores de backend suelen colaborar con los integrantes encargados del equipo del lado del cliente del proyecto. El desarrollo del backend puede abarcar la creación de interfaces de programación de aplicaciones (API) que respalden los servicios del frontend, sin requerir de un acoplamiento estrecho de ambos lados.

3. **Desarrollador Web Full-Stack.** El desarrollo web Full-stack se encarga de las tareas tanto del desarrollo frontend como del backend por igual.

Según la complejidad del sitio web, un desarrollador full-stack puede ser responsable de todas las fases del desarrollo web, desde el lado del servidor hasta la interfaz del usuario. Aportan al trabajo de rendimiento ya que estos profesionales se encargan de visualizar todos los aspectos del desarrollo web desde una perspectiva general, y con suficientes conocimientos en ambas áreas para emitir opiniones valiosas.

3.1.1. Arquitecturas de las aplicaciones Web

Una aplicación web es alojada en un servidor web y accesible para usuarios que se conectan desde cualquier ubicación a través de navegadores. Esta arquitectura incluye un servidor web, una conexión de red, y uno o más clientes [48].

El servidor web proporciona a los clientes páginas de información formateada cuando las solicitan. Los requisitos se envían mediante una conexión de red, utilizando el protocolo HTTP. Cuando se solicita esta petición a través de HTTP y el servidor web la recibe, éste busca la página web en su sistema de archivos y la devuelve al navegador que la solicitó.

Las aplicaciones web se fundamentan en el modelo cliente/servidor que gestionan servidores web, y que emplean páginas web como interfaz. Los componentes de una aplicación web son:

1. **Interfaz.** Es accesible para los usuarios a través de navegadores, dispositivos móviles, PDAs y otros dispositivos. Toda la funcionalidad está disponible mediante el navegador, siendo limitada y dirigida por la propia aplicación.
2. **Lógica de negocio.** Es la parte más importante de la aplicación, ya que establece los procesos y operaciones necesarios para su funcionamiento. Consiste en todas las operaciones necesarias para proveer el servicio.
3. **Administración de los datos.** Implica la manipulación de bases de datos y archivos.

En la arquitectura de un sistema web tradicional, se utiliza comúnmente un modelo de capas donde cada capa tiene responsabilidades específicas en el procesamiento y tratamiento de la información.

1. **Modelo de dos capas.** La información pasa por dos capas entre la interfaz y la administración de los datos. La mayoría de la lógica de negocio se ejecuta en el lado del cliente, llamado "fat client". En este enfoque, la aplicación encargada de la interfaz de usuario integra la mayoría de la lógica de negocio, mientras que el servidor se enfoca principalmente en la administración de los datos.
2. **Modelo de n-capas.** La información atraviesa varias capas, siendo el más común el modelo de tres capas.

Modelo de tres capas.

Diseñado para resolver las limitaciones de la arquitectura de dos capas, añadiendo una capa intermedia, la capa de negocio, entre la interfaz y la administración de datos. Esta capa permite separar la lógica de negocio de la interfaz de usuario y los datos, facilitando la administración y la integración de datos de diversas fuentes.

Se divide en tres capas o niveles, cada una con una función específica y bien definida:

1. **Nivel de presentación.** Funciona en ambos, tanto en el cliente como en el servidor, es fundamental en la comunicación entre el usuario y la aplicación. Recopila y envía la información del usuario al servidor para su procesamiento, después simplifica la interacción con la capa

de negocio para el análisis y manejo de la información. Las tecnologías más usadas incluyen HTML, CSS, JavaScript, y frameworks de frontend como React, Angular, y Vue.js.

2. **Nivel de negocio.** El nivel de proceso, ubicado en el servidor web, cumple el rol principal en el flujo de datos de la aplicación. Este nivel recibe datos del cliente, se comunica con el nivel de administración de datos para realizar ciertas operaciones y entrega los resultados obtenidos de vuelta al nivel de presentación. Engloba Node.js, Django, Spring, y otros frameworks de backend como tecnologías comunes.
3. **Nivel de administración de datos.** También conocido como servidor de datos, realiza funciones importantes en el sistema de aplicación. Esta capa almacena y gestiona la información de los usuarios de manera segura, garantizando su integridad y protección evitando accesos no autorizados. Como tecnologías comunes tenemos sistemas de gestión de bases de datos (DBMS) como MySQL, PostgreSQL, MongoDB, Oracle, y otros.

3.1.2. Tecnologías Web

Las tecnologías web son un conjunto de herramientas, protocolos y estándares que posibilitan el desarrollo, la implementación y el funcionamiento de las aplicaciones y servicios en la World Wide Web, WWW. Estas tecnologías son esenciales para la creación y el mantenimiento de sitios y aplicaciones web abarcando diversas áreas del desarrollo y gestión de contenido en línea.

En el desarrollo web actual, existen tres tecnologías que son necesarias para la creación y funcionamiento de sitios web: HTML, CSS y JavaScript. Cada una de estas tecnologías establecen la base y desempeñan roles cruciales en la organización del contenido, el diseño visual y la interactividad del usuario.

1. **HTML.** El Lenguaje de Marcado de Hipertexto, HyperText Markup Language [41], es fundamental en el desarrollo web al establecer la estructura básica del contenido en una página web. El lenguaje HTML estructura la información utilizando etiquetas y atributos que determinan la disposición y presentación de los elementos.

Estos elementos se utilizan para encerrar diferentes partes del contenido y darles una apariencia o comportamiento específico. Las etiquetas HTML son marcadores, "<,>", que se encargan de delimitar estos elementos HTML. Dentro de estas etiquetas, se puede encontrar el contenido del elemento, en forma de texto, otras etiquetas o incluso imágenes o videos. Además de su contenido, un elemento HTML puede contener atributos, que ofrecen información adicional sobre el elemento. Estos atributos se definen dentro de la etiqueta de apertura y se emplean para configurar propiedades particulares del elemento, tales como el estilo, la funcionalidad o la relación con otros recursos.

Para crear una página web, se requiere un documento HTML que contenga tres elementos principales: <html>, <head> y <body>. Esta última etiqueta, el cuerpo, contiene toda la parte visible de una página web. Para organizar su contenido, los desarrolladores emplean principalmente elementos genéricos como <div> y . Estos elementos no ofrecen ningún significado semántico concreto más allá de su función de contenedor.

Con el paso del tiempo, HTML ha progresado para adaptarse a las crecientes exigencias del diseño y desarrollo web contemporáneos. La evolución más importante ha sido el paso del HTML tradicional a **HTML5**. Este cambio no se limita a ser solo una actualización; evoluciona el desarrollo web con funciones más avanzadas y capacidades mejoradas.

Una de las incorporaciones de HTML5 fueron los elementos semánticos [53] que resolvieron el problema del uso genérico de <div> para estructurar y organizar el contenido de las páginas web: <header>, <nav>, <section>, <article>, <aside>, y <footer>, que clarifican su contenido y ofrecen una estructura más nítida y relevante. Estas formas más descriptivas de organizar el contenido web, hacen que sea más fácil la lectura por parte de los motores de búsqueda y la

interpretación por parte de los lectores de pantalla, ambos desempeñan funciones clave para mejorar la accesibilidad y utilidad de la web para diferentes tipos de usuarios.

2. **CSS.** Por sus siglas, Cascading Style Sheets, se refiere a las Hojas de Estilo en Cascada [61]. CSS es un lenguaje que se emplea para definir el aspecto visual de un documento HTML. Mientras que HTML define la estructura básica del contenido, CSS se encarga de añadir estilos visuales, como colores, fuentes, márgenes, y la disposición de los elementos en la página. Esto posibilita la separación de la estructura del contenido (HTML) y la presentación del mismo (CSS), lo que ayuda a mantener y adaptar más fácilmente el diseño web.

Su característica de cascada se relaciona con la manera en que se aplican las reglas de estilo cuando hay varias reglas que afectan al mismo elemento. En un orden jerárquico, las reglas más específicas son prioritarias sobre las más generales, y las definiciones finales pueden reemplazar a las anteriores. Esta capacidad de herencia permite a los diseñadores web crear estilos sofisticados y complejos.

El lenguaje CSS se divide en módulos, que representan bloques de contenido o elementos de interfaz con un propósito y una función específicos. La versión más actualizada de estos módulos es CSS3, que incorpora mejoras sustanciales y capacidades nuevas como flexbox, grid layout, transiciones y animaciones.

3. **JavaScript.** Lenguaje de programación esencial en el desarrollo web, reconocido por su capacidad de añadir interactividad a las páginas web. JavaScript, JS, se ejecuta en el navegador de forma directa, lo que hace más sencilla la creación de interfaces reactivas y la manipulación en tiempo real del DOM. A diferencia de HTML, que proporciona la estructura, y CSS, que maneja el estilo visual, JavaScript posibilita añadir funcionalidades dinámicas que mejoran la experiencia del usuario. Abarcando tareas como validación de formularios en el frontend, diseño de efectos de animación, modificación del contenido según las acciones del usuario y comunicación asincrónica con servidores.

JavaScript continúa siendo la columna vertebral del desarrollo web, y su relevancia ha aumentado en 2024. Además de ser el lenguaje de programación con más oportunidades de empleo. Siendo el lenguaje de programación más utilizado por undécima vez consecutiva y comúnmente usado por el 63,61 % de los Perfiles Tech [30].

Con la llegada de Node.js, JavaScript ha logrado tener presencia en el lado del servidor, lo que ha permitido un proceso de desarrollo más consistente y competente. Frameworks como React, Angular y Vue.js han transformado el proceso de desarrollo frontend, ofreciendo marcos y utilidades que agilizan la programación y aumentan la calidad del código.

■ Entornos de Ejecución de JavaScript

Un entorno de ejecución (RTE) de JavaScript es esencialmente el entorno en el que se ejecuta el código JavaScript. Este entorno proporciona todas las herramientas, bibliotecas y la infraestructura necesarias para que el código JavaScript funcione correctamente en un ordenador.

- **Node.js.** Node.js¹ es un entorno en tiempo de ejecución de código abierto multiplataforma que se fundamenta en el lenguaje de programación JavaScript [62]. Ha transformado el desarrollo web al permitir que JavaScript se ejecute en el servidor, aunque no exclusivamente, posibilitando la creación de aplicaciones en tiempo real y altamente escalables. Cada día es utilizado por más desarrolladores en todo el mundo, y en la actualidad ya se puede ver como un elemento esencial de Internet.

Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web. Node.js utiliza un enfoque de programación no bloqueante y basado en eventos, lo cual lo vuelve ideal para aplicaciones que necesitan alta concurrencia y tiempos de respuesta rápidos, como juegos en línea, chat en tiempo

¹<https://nodejs.org/en>

real y sistemas de colaboración. Asimismo, respaldado por el gestor de paquetes npm, brinda una extensa biblioteca de módulos y herramientas.

- **Deno.** Es un entorno de ejecución muy actual, desarrollado en el lenguaje de programación Rust [43], que compite directamente con Node.js. Los programadores emplean Deno² para desarrollar aplicaciones web backend y herramientas de línea de comandos seguras y escalables. Deno fue creado con el propósito de abordar los problemas de seguridad y rendimiento que enfrentan los usuarios de Node.js, al brindar soporte incorporado para TypeScript, seguridad más estricta y una biblioteca estándar incluida. Por el contrario, Node.js es una alternativa conocida en la que confían muchos desarrolladores y empresas. Aunque es probable que se vea incrementada la popularidad de Deno con el transcurso del tiempo.

■ Frameworks de JavaScript

Son fundamentales para el desarrollo web frontend actual, ya que ofrecen a los desarrolladores herramientas sólidas para construir aplicaciones web interactivas y escalables. En muchas empresas modernas, los frameworks son una parte estándar del conjunto de herramientas utilizadas, por lo que la experiencia en frameworks es un requisito común para los trabajos de desarrollo frontend en la actualidad.

En el desarrollo web, no existe un framework superior a los demás; la decisión se basa de las necesidades particulares de cada proyecto. Cada framework posee sus propias fortalezas y características que lo hacen apropiado para distintas circunstancias. Entre los frameworks de código abierto más utilizados en este año 2024 se encuentran los siguientes [56]:

- **React.** A día de hoy, continúa siendo uno de los más demandados para la creación de interfaces de usuario. React³ es comúnmente referido como una biblioteca JavaScript, no como un framework completo. Busca desarrollar interfaces de usuario que hagan más fácil la creación de aplicaciones web, mediante componentes reutilizables y reactivos que se ajustan dinámicamente a los cambios. React utiliza el Virtual DOM como una técnica para optimizar la eficiencia en la manipulación del DOM real.
- **Angular.** Es un framework [60], diseñado principalmente para TypeScript, empleado para desarrollar aplicaciones web móviles y de escritorio, adopta el patrón de diseño Modelo-Vista-Controlador (MVC), el cual divide de forma clara la lógica de presentación de los datos y la interacción del usuario. Esto hace que sea más sencillo colaborar entre varios equipos de desarrollo y mejora la organización del código.

Angular⁴ utiliza un enfoque distinto al Virtual DOM. En lugar de manipular un Virtual DOM de manera directa, como lo hace React, emplea su sistema de detección de cambios. Angular emplea un sistema bidireccional de enlace de datos [7], de modo que cualquier modificación en los datos se manifestará de forma automática en la interfaz de usuario y viceversa. Haciendo más sencillo el desarrollo de aplicaciones interactivas y en tiempo real.

Aunque sobresale por su capacidad para crear interfaces mediante una sintaxis de plantillas potente, Angular se dificulta por su pronunciada curva de aprendizaje debido a la combinación de una estructura compleja, su diseño enfocado a la utilización de TypeScript y la introducción de conceptos avanzados.

- **Vue.js.** Es un framework utilizado para desarrollar aplicaciones web de una sola página, SPA. Su característica principal es su capacidad de encapsular y reutilizar código mediante componentes. Es conocido por su enfoque progresivo y de adopción incremental, lo que implica que puede ser integrado de forma gradual en proyectos existentes sin necesidad de

²<https://deno.com/>

³<https://es.react.dev/>

⁴<https://angular.dev/>

una reescritura completa. Al igual que React, Vue⁵ emplea un Virtual DOM para manejar las actualizaciones del DOM de manera eficiente. En suma, Vue proporciona un equilibrio entre la simplicidad de React y la complejidad de Angular.

- **Svelte.** Se trata más de una herramienta que de un framework o una librería, en la cual escribimos todo el código frontend que luego se convierte en JavaScript altamente optimizado. Los otros frameworks llevan a cabo gran parte del trabajo en el navegador del usuario durante la ejecución de la aplicación, por el contrario, Svelte⁶ brinda un enfoque diferente en el desarrollo de aplicaciones web [40]: lleva a cabo la mayoría de su trabajo durante la compilación. Esto implica que en vez de manejar el DOM en tiempo de ejecución, Svelte compila componentes en código JavaScript altamente optimizado a lo largo del proceso de compilación, resultando en mejoras significativas en el rendimiento de la aplicación. Esto lleva a una mejora en el rendimiento y un código mucho más limpio. A pesar de ser más reciente comparado con React y Angular, Svelte ha llamado la atención de los desarrolladores por su enfoque directo y eficiente.

▪ **TypeScript**

TypeScript (TS)⁷ se trata de un lenguaje de programación que se sitúa en una capa superior de JavaScript [47]. Esto significa que TypeScript incorpora al lenguaje diversas funcionalidades adicionales que facilitan la escritura de código con menos errores, más sencillez, coherencia y facilidad de prueba, por lo tanto, un código más limpio y robusto. Dado a que TypeScript es una extensión de JavaScript, todo el código válido en JavaScript también es compatible con TypeScript, pero no a la inversa.

Asimismo, al usar TypeScript, se pueden tomar ventaja de las características extra del lenguaje, como interfaces, clases y decoradores, que pueden hacer más sencillo el proceso de desarrollo de aplicaciones más sólidas y escalables. Para los desarrolladores, implica generar software de mayor calidad de manera más rápida y eficiente.

TypeScript es compatible con la mayoría de los frameworks populares de JavaScript, como Angular, React y Vue.js, lo cual prácticamente impone el uso de TypeScript si trabajas con estos frameworks. Considerando además que la mayoría de las empresas de desarrollo de software utilizan estos frameworks o librerías en sus proyectos principales.

▪ **Spring Framework**

Spring Framework es un framework de código abierto que simplifica la creación de aplicaciones en Java, Kotlin y Groovy. A pesar de ser muy potente, el proceso inicial de configuración y de preparación de aplicaciones para producción resulta bastante tedioso. Spring Boot hace que el proceso sea extremadamente sencillo al ofrecer un conjunto de herramientas que permiten crear aplicaciones de Spring que sean fáciles de configurar.

Spring Boot⁸ es una tecnología que nos facilita la creación de aplicaciones autocontenidas, omitiendo al desarrollador la complejidad de la arquitectura, para que puedan dedicarse exclusivamente al desarrollo, dejando a Spring Boot tareas como la configuración de dependencias, desplegar el servicio o aplicación a un servidor de aplicaciones.

Para ello, Spring Boot utiliza un servidor de aplicaciones embebido, que por defecto es Tomcat, pero también pueden ser otros. Además, Spring Boot también nos ofrece gestor de dependencias como maven o gradle, configuraciones automáticas y otras funciones para la personalización de la aplicación.

▪ **Bases de Datos**

⁵<https://vuejs.org/>

⁶<https://svelte.dev/>

⁷<https://www.typescriptlang.org/>

⁸<https://spring.io/projects/spring-boot>

- **MySQL.** MySQL⁹ es un sistema de gestión de bases de datos (DBMS) de código abierto desarrollado por Oracle [4]. Se basa en un modelo de base de datos relacional, en el que los datos se organizan en tablas compuestas por filas y columnas, con relaciones entre tablas bien definidas. Además, emplea el lenguaje SQL para ejecutar consultas y gestionar datos. Sus componentes esenciales abarcan un servidor de base de datos, diversos motores de almacenamiento y clientes que facilitan la interacción con la base de datos. MySQL es una solución estable que resulta útil en ciertos escenarios, como bases de datos de páginas web, aplicaciones y administración de productos comerciales.
- **MongoDB.** MongoDB¹⁰ es un sistema de gestión de bases de datos no relacionales y de código abierto, que emplea documentos flexibles en vez de tablas y filas para almacenar diferentes tipos de datos [27]. Debido a ser NoSQL, no necesita un sistema de gestión de bases de datos relacionales (RDBMS), lo que permite ofrecer un modelo de almacenamiento de datos flexible para almacenar y consultar variedad de datos de forma sencilla, por lo que también proporciona un entorno altamente escalable para aplicaciones y servicios multiplataforma. Las unidades básicas de MongoDB son los documentos, o colecciones de documentos. Estos documentos, en formato JSON binario (BSON), tienen la capacidad de guardar distintos datos y ser compartidos entre varios sistemas.

- **Interfaces de Programación de Aplicaciones, APIs**

- **REST.** Identificamos como RESTful [22] a todas aquellas API que cumplen con todos los principios del diseño REST. Por otro lado, denominamos RESTless a aquellas que no siguen completamente con dichos criterios.

Una API RESTful (Representational State Transfer) [50] se define por utilizar operaciones HTTP estándar para llevar a cabo operaciones CRUD en recursos identificados por URLs. Estas interfaces carecen de estado, stateless, lo que implica que cada petición del cliente debe incluir todos los datos necesarios para que el servidor pueda procesarla sin depender de solicitudes previas. Los recursos son presentados en formatos convencionales como JSON o XML, lo que simplifica la comunicación entre diversos sistemas. Las APIs RESTful son creadas con la intención de ser sencillas, escalables y de facilitar una interacción eficiente entre clientes y servidores distribuidos en la web.

- **GraphQL.** GraphQL¹¹ es tanto un lenguaje de consulta API como un tiempo de ejecución [31] para procesar esas consultas utilizando los datos existentes, abordando incluso las consultas más complejas. Tiene la capacidad de solicitar y recibir solamente la información que se pide. Por ello, se emplea para mejorar la comunicación eficiente entre clientes y servidores, posibilitando a los clientes solicitar los datos que necesitan de manera específica.

GraphQL proporciona todos los datos en un único endpoint. Al contrario de las APIs REST tradicionales, que requieren de varias solicitudes a distintos endpoints para obtener datos relacionados, GraphQL permite la obtención de todo lo requerido en una única petición. Esto se consigue a través de consultas flexibles que solicitan los campos requeridos únicamente y permiten anidar las peticiones para obtener información relacionada en una sola transmisión de datos desde el servidor. GraphQL está emergiendo como una alternativa más eficiente a REST para la creación de APIs.

Ningún software o herramienta por sí sola puede satisfacer todas las necesidades de una empresa. Las compañías de éxito emplean múltiples aplicaciones y crean una pila de tecnología para alcanzar sus metas empresariales. Utilizar las herramientas inadecuadas o continuar con alternativas ineficientes puede resultar en pérdidas millonarias a largo plazo. Las pérdidas son aún mayores si no se cuenta

⁹<https://www.mysql.com/>

¹⁰<https://www.mongodb.com/>

¹¹<https://graphql.org/>

con ningún tech stack [21].

Un **Stack Tecnológico** o pila tecnológica 4.3 es el grupo de herramientas, plataformas, aplicaciones y piezas de software que una empresa emplea para desarrollar sus productos, realizar sus operaciones de negocio y supervisar el rendimiento. Un buen stack tecnológico debe incluir tecnologías modernas y actualizadas en todas sus áreas principales: frontend, backend e infraestructura.

Contar con un tech stack apropiado y acorde a las necesidades y estrategia de la empresa es crucial. Una pila tecnológica adecuadamente creada no solo mejora la eficiencia y la productividad, sino que también simplifica la integración entre diferentes sistemas y la escalabilidad para adaptarse al crecimiento.

3.1.3. Estado actual de la Inteligencia Artificial

La Inteligencia Artificial (IA) es un área de la informática que se dedica a crear sistemas y programas capaces de realizar tareas que típicamente necesitan inteligencia humana [46]. Estos sistemas están diseñados para adquirir conocimiento de datos, adaptarse a nuevas situaciones, y llevar a cabo decisiones y acciones según metas concretas. La integración de la Inteligencia Artificial en el desarrollo web es una tendencia en aumento que promete revolucionar la interacción con las aplicaciones y servicios accesibles a través de internet [34].

En el presente año, la tasa de adopción de la Inteligencia Artificial ha incrementado a un 72 % [35]. En contraste con los últimos seis años, donde alrededor del 50 % de las organizaciones han adoptado la inteligencia artificial, gráfico 3.1.

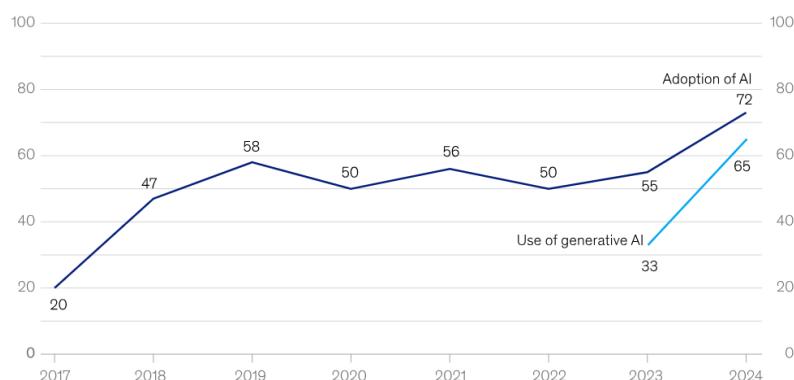


Figura 3.1: Organizaciones que han incorporado IA en al menos una función empresarial.

Fuente: <https://www.mckinsey.com/>

La Inteligencia Artificial Generativa es el siguiente paso en la Inteligencia Artificial [2], esta se centra en la creación de sistemas capaces de producir contenido original y creativo, como texto, imágenes o videos, imitando la forma en que piensa su humano. El interés en este enfoque ha aumentado debido a su potencial para generar nuevas oportunidades en diferentes sectores, como el entretenimiento y la investigación científica. El aumento del interés por la Inteligencia Artificial Generativa ha mostrado un rango más amplio de capacidades en el campo de la IA, demostrando su impacto significativo en nuestra sociedad actual.

2023 fue el año en que la Inteligencia Artificial Generativa (gen AI) fue descubierta por el mundo, mientras que en 2024 las organizaciones empezaron a utilizar esta tecnología y a beneficiarse económicamente de ella. Ya que las empresas están experimentando ventajas tangibles al emplear la Inteligencia Artificial de vanguardia, con reducciones de gastos e incrementos de ganancias reportados en las unidades de negocio que hacen uso de esta tecnología.

Las expectativas sobre el impacto de Gen AI siguen siendo muy elevadas y afirman que provocará un cambio considerable e innovador en sus sectores en los próximos años. Viéndose un aumento al 65 % en el uso regular de la IA generativa, casi el doble del porcentaje que hace diez meses, gráfico 3.1.

Las empresas están implementando la Inteligencia Artificial en más áreas de operaciones de la empresa. El 50 % de las organizaciones han implementado la IA en dos o más funciones empresariales, en contraste con menos de un tercio el pasado año. Según la organización media, la Inteligencia Artificial de última generación se emplea principalmente en dos funciones: en marketing y ventas y en desarrollo de productos y servicios, áreas donde se ha demostrado que su adopción puede generar un alto valor, al igual que en Tecnología de la Información (TI).

Además, se nota que su uso también está llegando a la vida personal de los individuos. Comparando con el año pasado, actualmente es más usual que las personas utilicen la Inteligencia Artificial en su trabajo y es aún más frecuente que la utilicen tanto en el ámbito laboral como en el personal.

Las empresas también son conscientes de los múltiples riesgos relacionados con esta tecnología. Los riesgos abarcan desde problemas de gestión de datos, como la privacidad de los datos, el sesgo o la violación de derechos de autor (PI), hasta la gestión de modelos que suelen relacionarse con resultados inexactos o falta de explicación, además de la seguridad y la incorrecta utilización de la tecnología. La inexactitud es el principal riesgo reconocido al utilizar IA generativa, afectando en diversos casos de uso en toda la cadena de valor de la IA, como los recorridos de los clientes, resúmenes, codificación y contenido creativo. Este año las organizaciones están más enfocadas en trabajar activamente la atenuación de esta inexactitud.

3.1.3.1. Inteligencia Artificial Generativa: Introducción a los Modelos de Lenguaje Grande

Los Modelos de Lenguaje Grande, LLM, son diseñados para comprender y generar lenguaje humano con un alto grado de sofisticación. Estos modelos utilizan Inteligencia Artificial Generativa para producir texto coherente y relevante en respuesta a las entradas de texto facilitadas por los usuarios [28]. Este enfoque avanzado permite a los LLMs no solo interpretar y comprender el significado de palabras y frases, sino también generar respuestas contextuales adecuadas que imitan de manera notable el estilo y la estructura del lenguaje.

Los Modelos de Lenguaje Grande operan a través de Redes Neuronales Profundas entrenadas con grandes volúmenes de datos textuales y, en ocasiones, datos visuales. Utilizan un Enfoque Autorregresivo (AR) para predecir frases y palabras basándose en el contexto proporcionado por el usuario. Este entrenamiento a gran escala les ayuda a comprender el lenguaje natural de forma más eficiente y a generar respuestas relevantes y coherentes.

Seleccionar el Modelo de Lenguaje Grande más adecuado puede ser complicado por la gran diversidad de opciones disponibles y sus diferentes capacidades y aplicaciones, como GPT-3, BERT, T5, PaLM y DALL-E 2 que son algunos de los LLMs más populares y utilizados en la actualidad. Para tomar una decisión informada, es crucial tener en cuenta varios aspectos clave [6]:

1. **La misión en la aplicación.** Es crucial identificar qué necesidad específica abordará el LLM en la aplicación: ¿Qué dificultad ayudará a resolver el LLM?. Entre las funcionalidades más habituales se encuentran la capacidad de sintetizar información, organizar datos, responder preguntas, generar código, redactar contenido, seguir instrucciones, llevar a cabo tareas multilingües y transformar texto en representaciones vectoriales (embeddings).
2. **El idioma.** Antes de elegir un LLM, es fundamental determinar si la solicitud funcionará en un solo idioma o en varios antes de hacer una elección: ¿Necesito el LLM para poder comprender en varios idiomas?. Los LLM pueden estar enfocados en uno o múltiples idiomas.
3. **La longitud de la Ventana de Contexto.** Se debe garantizar que la capacidad de *tokens* del LLM sea adecuada para las necesidades de entrada y salida de texto. ¿Cuál será la longitud de la entrada y salida de texto? El tamaño de la *Context Window* determina la cantidad de información contextual que el modelo puede abarcar en una sola interacción. Los modelos procesan el texto en *tokens*, y la eficacia y desempeño del modelo dependerá de la cantidad de *tokens* en tareas que implican comprender y generar texto largo.

4. **Precios.** Los costos de utilizar un LLM pueden tener diferencias importantes. Existen modelos de código abierto, Open Source, sin costo directo y otros que requieren pago, cuyo precio depende de la cantidad de *tokens* procesados.
5. **Rendimiento del Modelo.** Es necesario analizar y contrastar cómo se desempeñan los Modelos de Lenguaje Grande en tareas específicas y qué tan precisos son al generar respuestas, con el fin de seleccionar el más adecuado para las necesidades de la aplicación.

Los puntos de referencia, Benchmarks, son estudios y evaluaciones sistemáticas que contrastan el rendimiento de diferentes modelos en diversas tareas de procesamiento de lenguaje natural (NLP). Cada benchmark puede centrarse en diversos aspectos de las capacidades de un modelo, como la comprensión del lenguaje, el razonamiento, la generación de código y las capacidades conversacionales.

3.1.3.2. Estudio de Modelos de Código Abierto de Última Generación

Es necesario revisar los Modelos de Lenguaje Grande más populares que han sido lanzados al mercado en los últimos meses o días para mantenerse actualizado. Estos modelos son líderes en el campo de la IA y el procesamiento del lenguaje natural, logrando importantes avances en términos de capacidad y precisión.

1. **Mistral: Mixtral 8x22B.** Mistral AI, SAS, una startup fundada en 2023 con sede en París, especializada en Inteligencia Artificial, desarrolló Mistral, un Modelo de Lenguaje Grande diseñado para comprender y generar texto en diversos idiomas. Actualmente, se pueden encontrar disponibles como opciones de código abierto, los modelos Mistral 7B, Mixtral 8x7B y Mixtral 8x22B

Su último lanzamiento, Mixtral 8x22B ¹², es un modelo generativo, disperso y preentrenado, que utiliza la técnica de Mezcla de Expertos Esparcidos (SMoE). Esta arquitectura divide el modelo en múltiples 'expertos', *Mixture of Experts*, y se centra en la activación limitada de expertos, *Sparse*. En Mixtral 8x22B, se utiliza únicamente 39.000 millones de parámetros activos del total. Este ajuste optimiza el uso de recursos, ofreciendo una relación costo-beneficio incomparable para su tamaño.

También dispone de una ventana de contexto de 64.000 *tokens*, que indica la cantidad de texto que el modelo puede procesar y referirse en un momento dado, permitiéndole manejar grandes documentos con facilidad. Por otro lado, Mixtral 8x22B posee un tamaño máximo teórico de parámetro de hasta 176.000 millones, indicando el número de variables internas que el modelo emplea para tomar decisiones y hacer predicciones; aunque, en su funcionamiento activo, utilice 39.000 millones de parámetros.

Sobresale en habilidades multilingües, dominando cinco idiomas: inglés, francés, italiano, alemán y español, así como matemáticas, tareas de codificación y habilidades de razonamiento. Y su naturaleza de código abierto impulsa la innovación, la colaboración, y con ello, los avances de la Inteligencia Artificial.

En términos de costo-rendimiento, Mixtral 8x22B supera a competidores como Llama 2 70B y Command R+ de Cohere. El modelo destaca en pruebas de razonamiento, logrando puntajes superiores a los de otros en varios benchmarks. En tareas multilingües, codificación y desafíos matemáticos, Mixtral 8x22B supera claramente a LLaMA 2 70B, aunque el actual Llama 3 asegura superar las métricas de Mixtral 8x22B.

2. **LLaMA: Llama3-400B.** Meta Platforms, Inc, empresa con sede en California, especializada en tecnología de redes sociales e inteligencia artificial. Desarrolló LLaMA "Large Language Model Meta AI", y lanzó tres generaciones de modelos hasta el momento, Llama 1, Llama 2 (Llama 2 7B, Llama 2 13B y Llama 2 70B) y Llama 3 (Llama 3 8B y Llama 3 70B).

¹²<https://mistral.ai/news/mixtral-8x22b/>

Este mes de Abril, Meta AI lanzó su más reciente serie de Modelos de Lenguaje Grande, Llama 3. Esta generación presenta dos versiones 8B y 70B, con 8.000 millones y 70.000 millones de parámetros, respectivamente, optimizadas especialmente para instrucciones. No obstante, es importante mencionar que la ventana de contexto actual se limita a 8.000 *tokens*, lo que equivale aproximadamente a 6.000 palabras, lo cual puede ser relevante para algunas aplicaciones que necesitan contextos más amplios.

Una de las novedades de Llama 3 es su Tokenizador, que amplia el vocabulario significativamente con 128.256 *tokens* en contraste con los 32.000 de su predecesor, Llama 2. Permitiendo una codificación más eficiente del texto tanto en la entrada como en la salida, lo cual podría impulsar el multilingüismo y mejorar de manera general el rendimiento. Además, Llama 3 utiliza la técnica de Atención de Consultas Agrupadas (GQA), con el fin de incrementar la escalabilidad al permitir que el modelo centre su atención en múltiples partes de la entrada al mismo tiempo. Siendo fundamental para aumentar la eficacia en contextos largos y complicados, mejorando la capacidad del modelo para comprender y analizar información extensa y variada.

Llama 3 8B se ha destacado por su rendimiento superior en comparación con modelos anteriores como Gemma 7B y Mistral 7B Instruct [49].

Meta también comunicó que tienen en proceso un modelo que será lanzado pronto tras completar su entrenamiento, el modelo Meta Llama 3 400B+¹³. Se confirma que será el modelo más grande y potente de la generación Llama 3, con un total de 400.000 millones de parámetros. Y los primeros resultados indican que superará las capacidades de sus versiones más pequeñas.

Una de las ventajas más destacables del modelo es su gran velocidad y rendimiento. Se prevé que este modelo pueda procesar y crear texto a una velocidad sin igual debido a su gran cantidad de parámetros. El modelo de 70B ya es un avance notable, pero se espera que el modelo de 400B sea alrededor de 5,7 veces mayor. El aumento en tamaño y complejidad probablemente conducirá a un aumento proporcional en la velocidad y precisión del procesamiento. Esto implica que será capaz de abordar tareas más complicadas, generar texto más coherente y preciso y procesar conjuntos de datos más extensos.

Se espera que el modelo muestre varias capacidades avanzadas, como la capacidad multimodal de procesar y generar datos en diferentes formatos como texto, imágenes y audio. Asimismo, se prevé que brinde soporte multilingüe, permitiendo la interacción fluida y comprensión entre varios idiomas, facilitando la comunicación global. Se anticipa también que tenga una ventana de contexto más extensa, para comprender y procesar secuencias de texto más largas, mejorando la precisión y contexto de sus respuestas. Y capacidades generales mejoradas, mayor precisión, fluidez y coherencia en sus tareas de procesamiento y generación de lenguaje.

Llama 3 400B+ optimizado para instrucciones, alcanzó en las primeras pruebas una puntuación de 86,1 en el benchmark MMLU (Massive Multitask Language Understanding), equiparando en rendimiento al modelo GPT-4 con menos de la mitad de los parámetros. Concluyendo ser más eficiente en recursos computacionales, consumo de energía y costo al lograr el mismo rendimiento con menos parámetros.

3. **Gemma: Gemma 2.** Google, una de las empresas líderes en el desarrollo de tecnologías de Inteligencia Artificial, ha desarrollado una familia de Modelos de Lenguaje Grande de código abierto llamada Gemma, caracterizados por ser livianos, de alto rendimiento y optimizados para inferencia rápida en diversos tipos de hardware. Actualmente cuenta con Gemma 2B, Gemma 7B, Gemma 2 9B y Gemma 2 27B.

Gemma 2¹⁴ se trata de la reciente actualización de la serie de Modelos de Lenguaje Grande de código abierto de Google. Este modelo presenta dos variantes: Gemma 9B y Gemma 27B, con 9.000 millones y 27.000 millones de parámetros respectivamente. Su longitud de ventana de

¹³<https://ai.meta.com/blog/meta-llama-3/>

¹⁴<https://ai.google.dev/gemma?hl=es-419>

contexto es de 8.000 *tokens*. Los modelos Gemma 2 fueron entrenados con alrededor del doble de datos en comparación con su primera iteración, con un total de 13 billones de *tokens* para la versión de 27B y 8 billones de *tokens* para la versión de 9B, obtenidos principalmente de datos web en inglés, código y matemáticas.

Utiliza la técnica Atención de Ventana Deslizante que se emplea para disminuir los recursos de memoria y tiempo necesarios para los cálculos de atención en diferentes modelos, como Mistral 7B. La novedad de Gemma 2 es que se añade una ventana deslizante a todas las capas restantes (local - 4.096 *tokens*), manteniendo la atención global cuadrática completa en las capas intermedias (8.192 *tokens*). Siendo una estrategia para mejorar la calidad en contextos largos ya que la mitad de las capas siguen atendiendo a todos los *tokens*, aprovechando parcialmente la atención deslizante.

Gemma 2 mejora en eficiencia y potencia en inferencia con importantes avances en seguridad con respecto a su predecesora. Destaca por su rendimiento en inferencia, compitiendo de manera efectiva con modelos mucho más grandes en cuanto a eficiencia y velocidad. De acuerdo a las pruebas realizadas por Google, el rendimiento de la Inteligencia Artificial preentrenada en ambos tamaños es comparable, e incluso superior en ciertos frameworks en versión de 9B, a Llama 3. Y superior, a Grok-1, la Inteligencia Artificial de Elon Musk [64].

La familia Gemma sigue creciendo con PaliGemma. Consiste en una versión ligera y abierta de un modelo de lenguaje de visión (VLM). Procesa imágenes y texto como entradas para responder preguntas sobre imágenes con detalles y contexto. Esto le permite analizar más a fondo las imágenes y ofrecer estadísticas útiles, como subtítulos para imágenes y vídeos cortos, detección de objetos y lectura de texto en imágenes.

4. **Falcon: Falcon 2.** El Instituto de Innovación Tecnológica (TII), principal institución científica a nivel global y la base de investigación científica aplicada del Advanced Technology Research Council (ATRC) de Abu Dabi, desarrolló el muy reconocido Modelo de Lenguaje Grande, Falcon. Falcon 2¹⁵ es la más reciente generación que ha desarrollado, que se basa en el éxito de sus modelos anteriores Falcon 7B, Falcon 40B y Falcon 180B, presentados en 2023.

En esta serie, se presentaron dos versiones innovadoras: Falcon 2 11B, un modelo LLM mejorado y accesible con entrenamiento en 5,5 billones de *tokens* y 11.000 millones de parámetros, y Falcon 2 11B VLM, que se destaca por sus capacidades de modelo visión-lenguaje para la conversión fluida de entradas visuales en salidas textuales. A pesar de que ambos modelos son multilingües, Falcon 2 11B VLM sobresale por ser el pionero en multimodalidad del TII y siendo actualmente el único en el mercado de gama alta que ofrece la capacidad de convertir imagen a texto. También, puede funcionar eficientemente en una única unidad de procesamiento gráfico (GPU), lo que lo convierte altamente escalable y fácil de integrar en infraestructuras más ligeras. Con habilidades multilingües, se centra en el inglés, pero también contiene información en otros idiomas.

Falcon 2 11B VLM aún no se encuentra disponible en su página oficial. Y en un futuro cercano, se plantea expandir los modelos Falcon 2 de última generación con una variedad de tamaños adicionales. Estos modelos se podrán perfeccionar aún más con tecnologías avanzadas de aprendizaje automático como Mezcla de Expertos (MoE), los cuales buscan llevar su rendimiento a niveles más avanzados.

Según la evaluación de Hugging Face [26], el modelo Falcon 2 11B supera en rendimiento al modelo Llama 3 8B de Meta y rinde al mismo nivel que el modelo líder Gemma 7B de Google.

5. **BLOOM.** En 2022, se dio a conocer el proyecto BLOOM¹⁶ (BigScience Large Open-science Open-access Multilingual Language Model) después de una colaboración a nivel mundial que contó con la participación de más de 1.000 investigadores voluntarios de más de 70 países y expertos

¹⁵<https://falconllm.tii.ae/falcon-models.html>

¹⁶<https://huggingface.co/bigscience/bloom>

de Hugging Face. Bloom tiene el propósito de generar texto autorregresivo, destacándose en la ampliación y finalización de secuencias de texto a partir de *prompts* dados para producir de forma coherente relatos, guiones, poesía, artículos y otros tipos de escritos.

Se caracteriza por sus buenas capacidades multilingües, con aptitud para producir texto en 46 idiomas y 13 lenguajes de programación, evidenciando un amplio y diverso dominio del lenguaje. Cuenta con 176.000 millones de parámetros, se ubica como uno de los modelos de lenguaje más potentes que se pueden encontrar en código abierto. Su amplio rango de habilidades lingüísticas incluye la generación de texto, la resolución de preguntas, la síntesis, la traducción y la generación de código, lo que la convierte en una herramienta muy completa para diversas necesidades lingüísticas.

BLOOM posee grandes capacidades multilingües gracias a su entrenamiento con un extenso conjunto de datos de 1.6TB (corpus ROOTS) que incluye los 46 lenguajes naturales y 13 lenguajes de programación, con más del 30 % de los datos en inglés. En muchos idiomas, como el español y el árabe, BLOOM es considerado el primer modelo de su tamaño.

Durante 3.5 meses, el modelo fue entrenado en la supercomputadora Jean Zay, gracias a una subvención del gobierno francés, equivalente a más de 5 millones de horas de computación. BLOOM ha logrado alcanzar un rendimiento competitivo en varios puntos de referencia.

Como conclusión, al decidir por un LLM, tal como cualquier tecnología avanzada, la recomendación es probar y comparar diversas opciones en la práctica. Esto requiere analizar y evaluar cómo cada LLM se comporta en el sistema particular, tomando en cuenta aspectos clave según las necesidades específicas del sistema, garantizando una integración efectiva y un rendimiento satisfactorio.

En la tabla 3.1, se comparan los Modelos de Lenguaje Grande analizados previamente en esta sección.

3.1.4. Aplicaciones y Sistemas Web para la gestión de rutinas y actividades

Hacer ejercicio de forma regular es una de las mejores decisiones que alguien puede tomar para mejorar su bienestar y salud general. La OMS respalda que el ejercicio físico ayuda a prevenir enfermedades crónicas, mejora la salud mental, y aumenta la longevidad [63]. De acuerdo con la OMS, mantener una rutina de actividad física puede disminuir la posibilidad de padecer enfermedades del corazón, diabetes tipo 2 y algunos tipos de cáncer, al mismo tiempo que ayuda a fortalecer los músculos y huesos, mejorar la salud mental y el descanso, y mantener un peso adecuado y saludable.

En la era digital actual, las aplicaciones y sistemas web han cambiado la manera en que las personas planifican y llevan a cabo sus rutinas de entrenamientos físicos. Estas tendencias muestran avances tecnológicos, una creciente personalización en el *fitness* y una mayor conciencia sobre la importancia de la salud mental y el bienestar integral.

Entre las tendencias de este año que prometen revolucionar nuestra forma de comprensión y práctica del deporte, se encuentran [3]:

- **Tecnología Wearable.** En la actualidad, la tecnología wearable sigue revolucionando el sector del *fitness* con dispositivos y accesorios cada vez más avanzados. Un dispositivo es clasificado como wearable cuando puede ser usado directamente en el cuerpo o como un accesorio, y está hecho para ser utilizado mientras se está en movimiento. Estos dispositivos suelen ser creados con el fin de hacer más fácil la interacción y ofrecer datos relevantes al instante.

Los relojes inteligentes son los pioneros en el campo de los wearables, comenzaron a aparecer en el mercado a principios de los 2000 y actualmente se han transformado en un accesorio necesario para aquellos que son fanáticos de esta tecnología. Han sufrido una notable evolución desde su aparición. Inicialmente se diseñaron para mostrar notificaciones básicas, funciones de agenda y recordatorios y aplicaciones simples como cronómetros, calculadoras y alarmas.

3.1. Estado del arte

Modelo	Mixtral 8x22B	Llama 3 400B+	Gemma 2	Falcon 2	BLOOM
Estado	Lanzado en Abril 2024	En fase de desarrollo	Lanzado en 28 Junio 2024	Lanzado en 14 Mayo 2024. Falcon 2 11B VLM aún no disponible.	Lanzado en Noviembre 2022
Desarrollador	Mistral AI	Meta AI	Google DeepMind	Instituto de Innovación Tecnológica (III)	Hugging Face
Técnica innovadora	Mezcla de Expertos Esparcidos (SMoE)	Atención de Consultas Agrupadas (GQA)	Atención de Ventana Deslizante	Mezcla de Expertos (MoE)	Entrenamiento con Corpus ROOTS
Multiidiomátic	Sí	Sí	No, solo Inglés	Sí	Sí, 59 lenguas
Accesibilidad	No	No especificado	PaliGemma VLM, Sí	Falcon 2 11B VLM, Sí	No
Tamaño del modelo	Código Abierto	Código Abierto	Código Abierto	Código Abierto	Código Abierto
Ventana de contexto	176.000 millones	400.000 millones	9.000 millones Gemma 2 2.27B	11.000 millones	176.000 millones
Características	>8.000 tokens	8.000 tokens	No se especifica	2.048 tokens	
Comparativas	Supera a Llama 2 70B en varios ámbitos	Rendimiento a la par con GPT-4	Rendimiento competitivo con Llama 3 y superior a Grok-1	Supera a Llama 3 8B, rinde igual que Gemma 7B	Rendimiento competitivo

Tabla 3.1: Tabla comparativa Modelos de Lenguaje Grande más actuales.

La información de la tabla se fundamenta en los datos disponibles hasta la fecha para cada modelo.

Posteriormente, se empezaron a añadir funcionalidades básicas de seguimiento deportivo con medición de pasos y calorías quemadas y sensores para el monitoreo de la salud.

Hoy en día, estos relojes brindan una amplia gama de funcionalidades que los convierten en completos centros de control, haciendo que los usuarios puedan gestionar una variedad de aplicaciones y dispositivos desde su muñeca. Permiten controlar los dispositivos conectados como son los del hogar inteligente (luces, termostatos, cerraduras, cámaras de seguridad) a través de aplicaciones y comandos de voz; es posible controlar también la reproducción de música en altavoces y televisores inteligentes. La integración de Asistentes de Voz es la que posibilita a los usuarios llevar a cabo tareas más complejas mediante comandos de voz.

Apple, Fitbit, Samsung, Google y Garmin sobresalen como las marcas líderes en dispositivos wearables. Algunos ejemplos populares de estos dispositivos incluyen relojes inteligentes como el Apple Watch, que cuenta con HomeKit para el control de dispositivos inteligentes del hogar y Siri como asistente de voz; el Samsung Galaxy Watch, que integra SmartThings para la automatización del hogar y Bixby como asistente personal; así como los dispositivos que integran el sistema operativo Wear OS¹⁷ de la marca Google y los relojes Fitbit Charge de la marca Fitbit y Garmin Vivoactive de la marca Garmin, entre otros.

Una de las mayores preocupaciones en la actualidad es definitivamente el cuidado de la salud y la Tecnología Wearable aborda este tema ofreciendo una alternativa confiable dirigida a personas que practican deportes o buscan mantener un estilo de vida activo: **las pulseras de actividad**. Están enfocadas exclusivamente en la monitorización exhaustiva de la actividad física y el rendimiento deportivo de los usuarios con el fin de mejorar la salud en términos generales. Proporcionan seguimiento continuo, notificaciones en tiempo real y no solo registran la frecuencia cardíaca y calorías quemadas, sino que también monitorizan aspectos como la calidad del sueño, la ingesta de agua, los niveles de azúcar en sangre y otros macronutrientes específicos para la vida *fitness*.

La mayoría de los dispositivos wearables inteligentes pueden conectarse con aplicaciones de *fitness* que permiten a los usuarios monitorear sus objetivos y progresos. Esto hace que sea más fácil la planificación y personalización de las sesiones de entrenamiento.

- **Aplicaciones móviles de ejercicio.** Las aplicaciones móviles de ejercicio han transformado la manera en que las personas gestionan sus rutinas de ejercicio [42]. Se han convertido en herramientas indispensables para los que buscan mantener una práctica de ejercicio efectiva y constante. Estas aplicaciones brindan una gran variedad de funcionalidades que hacen que sea más fácil para los usuarios alcanzar sus objetivos *fitness*.

Una de las principales ventajas de estas aplicaciones es la posibilidad de programar y monitorear rutinas de ejercicio de manera personalizada. Los usuarios tienen la capacidad de crear planes de entrenamiento personalizados según sus necesidades específicas, establecer metas y llevar a cabo un seguimiento de su progreso a lo largo del tiempo. Esto contribuye no solo a mantener la motivación, sino también a adaptar las rutinas según los resultados conseguidos.

Otra característica sobresaliente es la inclusión de planes de alimentación a medida. Algunas aplicaciones no solo se enfocan en el entrenamiento físico, sino que también brindan asesoramiento nutricional de acuerdo los objetivos individuales de cada persona. Esto ayuda de manera significativa a lograr resultados más efectivos y duraderos en cuanto a la salud y la condición física.

También, las aplicaciones de ejercicio ofrecen acceso inmediato a una gran variedad de recursos. Los usuarios tienen la posibilidad de ver videos instructivos, seguir rutinas guiadas por profesionales y recibir consejos útiles en todo momento y desde cualquier ubicación. Esta accesibilidad hace mucho más fácil la práctica del ejercicio, especialmente para personas con poco tiempo o que prefieren entrenar en su hogar.

¹⁷https://wearos.google.com/intl/es_es/

La interacción social es otro aspecto relevante que ofrecen muchas de estas aplicaciones. Los usuarios tienen la capacidad de conectarse con una comunidad global de amantes del *fitness*, compartir sus logros, tomar parte en retos y obtener apoyo y motivación de otros. Esto no solamente promueve un ambiente de apoyo y compañerismo, sino que contribuye a mantener la motivación a largo plazo.

Las aplicaciones están aprovechando la creciente popularidad de los wearables para incorporar datos de frecuencia cardíaca, patrones de sueño y actividad general. La combinación de datos obtenidos de wearables con las funcionalidades de las aplicaciones móviles genera un ecosistema integral para la salud física. Los usuarios pueden ver y estudiar sin dificultad su evolución a lo largo del tiempo, fijar metas basadas en datos reales y obtener recomendaciones personalizadas para mejorar su bienestar y condición física en general.

- **Entrenamiento personal.** El entrenamiento personal ha adquirido una importancia significativa, posicionándose como una de las tendencias destacadas del actual 2024 [58]. Este enfoque proporciona un plan de ejercicios adaptado, creado de manera específica para las metas y requerimientos personales de cada individuo. Contar con un entrenador personal implica la evaluación del estado físico inicial, el fijar metas alcanzables y la creación de un programa de entrenamiento personalizado. Los instructores ofrecen guía, corrección de la técnica y continua motivación, lo que no solo aumenta la efectividad de los entrenamientos, sino que también disminuye la probabilidad de sufrir lesiones.

Las aplicaciones móviles han hecho que el entrenamiento personalizado sea más accesible, ofreciendo a los usuarios la posibilidad de experimentar beneficios comparables a los de tener un entrenador personal. Estas aplicaciones permiten a los individuos recibir asesoramiento especializado y supervisión constante sin requerir de la presencia física de un entrenador. El entrenamiento personal en dispositivos móviles ofrece a los usuario una experiencia altamente personalizada y efectiva.

En la siguiente sección, se realiza el estudio y evaluación de diversas aplicaciones y sistemas web populares diseñados para la monitorización y gestión de ejercicios y rutinas de entrenamiento, que han sido seleccionadas específicamente por sus características distintivas que permiten la comparación con el proyecto que planteo desarrollar.

3.1.4.1. Análisis de la solución: Google Fit

Google Fit¹⁸ es una aplicación móvil disponible para dispositivos Android e iOS que no cuenta con una página web para acceder y monitorear actividades. Google Fit **se enfoca en el seguimiento de la actividad física diaria sin proporcionar rutinas de entrenamiento personalizadas**. Ofrece un seguimiento minucioso del progreso del usuario al registrar actividades como pasos, tiempo activo, distancia recorrida y calorías quemadas. Los usuarios pueden añadir manualmente diferentes actividades físicas y fijar metas personalizadas.

Además, presenta historiales junto con estadísticas detalladas en forma de gráficos a través de resúmenes diarios, semanales y mensuales. Su interfaz es intuitiva y accesible, emplea notificaciones y recordatorios y facilita la monitorización al conectarse con otras aplicaciones y dispositivos de salud, incluyendo el seguimiento del sueño, el peso corporal y la frecuencia cardíaca, entre otros aspectos.

3.1.4.2. Análisis de la solución: Jefit

Jefit¹⁹ es una plataforma web y aplicación móvil sólida creada para simplificar la planificación y el seguimiento de rutinas de ejercicios de forma eficiente. **Hace uso de una amplia base de datos de ejercicios con instrucciones detalladas y videos para que los usuarios diseñen**

¹⁸<https://www.google.com/fit/>

¹⁹<https://www.jefit.com/>

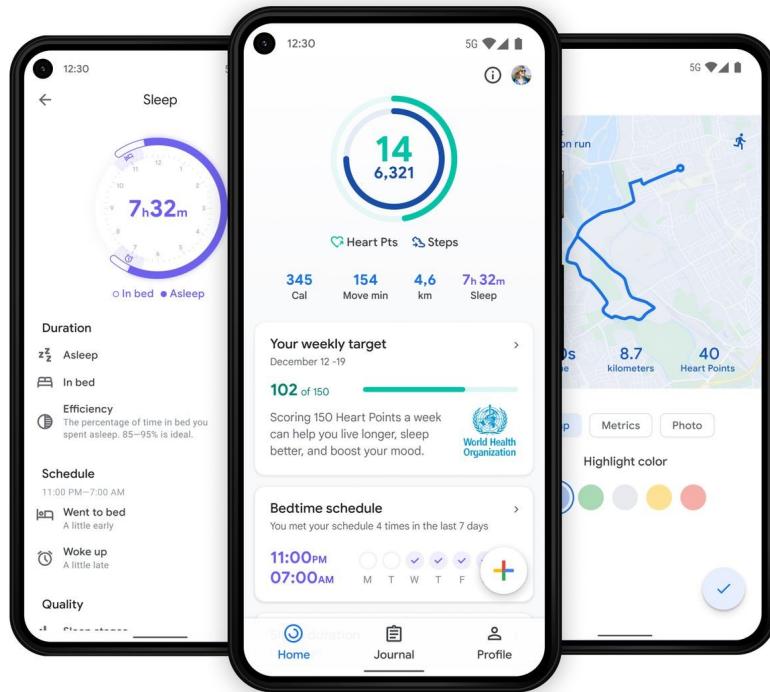


Figura 3.2: Aplicación Google Fit

rutinas personalizadas, permitiendo adaptarlas a sus objetivos específicos. La aplicación emplea el cuestionario inicial para obtener los datos del usuario y así poder brindar herramientas y funcionalidades para facilitar al usuario la creación de sus propias rutinas, teniendo el control total sobre la configuración y progresión de sus entrenamientos.

La aplicación facilita un seguimiento exhaustivo del progreso a través de registro detallado de cada sesión de ejercicios, con información sobre series, repeticiones y pesos utilizados. Esto se acompaña de gráficos y estadísticas que muestran el progreso. Aparte de su interfaz intuitiva y accesible, Jefit incluye un calendario incorporado para planificar entrenamientos y la capacidad de sincronización en la nube para acceder desde varios dispositivos. La participación de la comunidad dentro de la aplicación permite a los usuarios compartir logros y ejercicios.

3.1.4.3. Análisis de la solución: Caliber

Caliber²⁰ se trata de una plataforma en línea y aplicación móvil muy conocida que ofrece rutinas de entrenamiento personalizadas y adaptativas según las necesidades y objetivos de cada usuarios. **No emplea inteligencia artificial para la creación de rutinas personalizadas, la creación se lleva a cabo mediante el equipo de entrenadores certificados** que analizan la información inicial del usuario, sus objetivos, su nivel de experiencia y su disponibilidad de equipamiento, entre otros y diseñan rutinas personalizadas que se ajustan con el perfil del usuario. Facilitando la **adaptación dinámica de las rutinas mediante la interacción continua con los entrenadores personales**, de acuerdo con el avance y la feedback del usuario.

Proporciona seguimiento de progreso, estadísticas detalladas y facilita el registro de las sesiones de entrenamiento, con información sobre los ejercicios realizados, número de repeticiones, series. Ofrece guías y tutoriales para ejecutar adecuadamente los ejercicios incluidos en las rutinas. Asimismo, la aplicación tiene la capacidad de integración con dispositivos de monitorización y otros wearables. Cuenta con una interfaz intuitiva y accesible y brinda asesoramiento nutricional e interacción con entrenadores personales por mensaje, videollamada o consultas programadas que refuerzan los

²⁰<https://caliberstrong.com/>

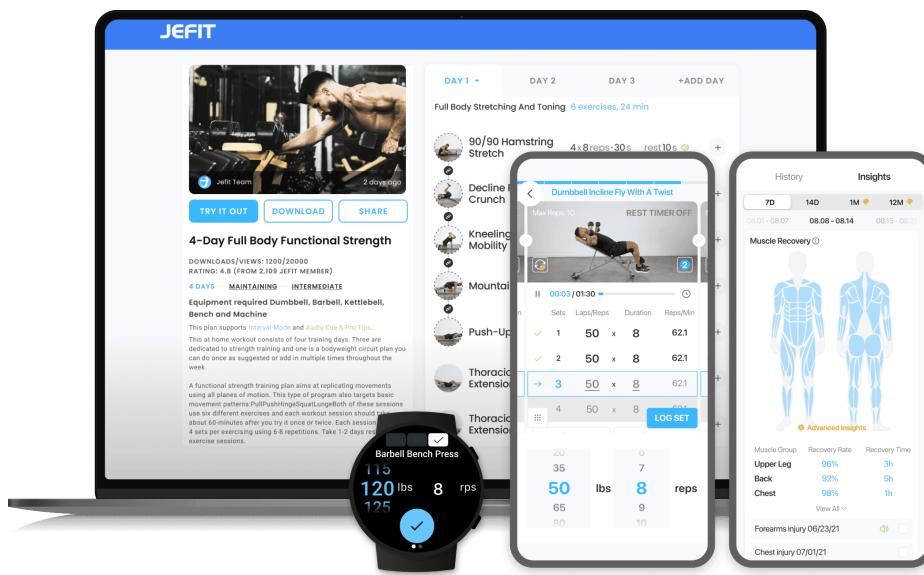


Figura 3.3: Aplicación Jefit

programas de entrenamiento.

3.1.4.4. Análisis de la solución: Freeletics

Freeletics²¹ es una plataforma web y aplicación móvil muy utilizada que provee programas de entrenamiento personalizados y adaptativos **usando inteligencia artificial a través de su AI Coach**. Este *Entrenador de IA* analiza constantemente los datos proporcionados por el usuario, estado físico, objetivos de entrenamiento y rendimiento, entre otros datos y genera rutinas de ejercicios personalizadas. Además de depender de algoritmos de inteligencia artificial, **el AI Coach también aprende y mejora a medida que recibe más información y feedback del usuario**.

Utiliza un cuestionario inicial para obtener la información de los usuarios. Proporciona seguimiento de progreso con estadísticas detalladas como tiempos de ejercicio y registro de actividades como repeticiones y series. Cuenta con un calendario, notificaciones que mantienen a los usuarios al tanto de próximas sesiones y logros alcanzados, una interfaz intuitiva y accesible y es compatible con varios dispositivos wearables.

3.1.4.5. Análisis de la solución: Fitbod

Fitbod²² es una aplicación móvil disponible para dispositivos iOS y Android que **emplea un algoritmo basado en inteligencia artificial que analiza el historial de entrenamiento del usuario y recopila datos sobre los entrenamientos realizados**, como ejercicios realizados, series, repeticiones y pesos, entre otros datos. Utilizando estos datos genera rutinas de entrenamiento personalizadas y adaptables, y va mejorando gradualmente la precisión y personalización de las mismas. De esta manera, no solo hace uso de un cuestionario inicial estático, **Fitbod se apoya en la constante evaluación de datos para ofrecer recomendaciones dinámicas y personalizadas acerca de la carga de trabajo, el volumen de entrenamiento y la intensidad, entre otros aspectos**.

3.1.4.6. Tabla comparativa de las aplicaciones y sistemas web analizados

El trabajo busca llevar a cabo un sistema web que utilice Inteligencia Artificial para generar rutinas personalizadas basadas en la información recolectada en el cuestionario inicial. Además de otros criterios importantes a considerar:

²¹<https://www.freeletics.com/es/>

²²<https://fitbod.me/>

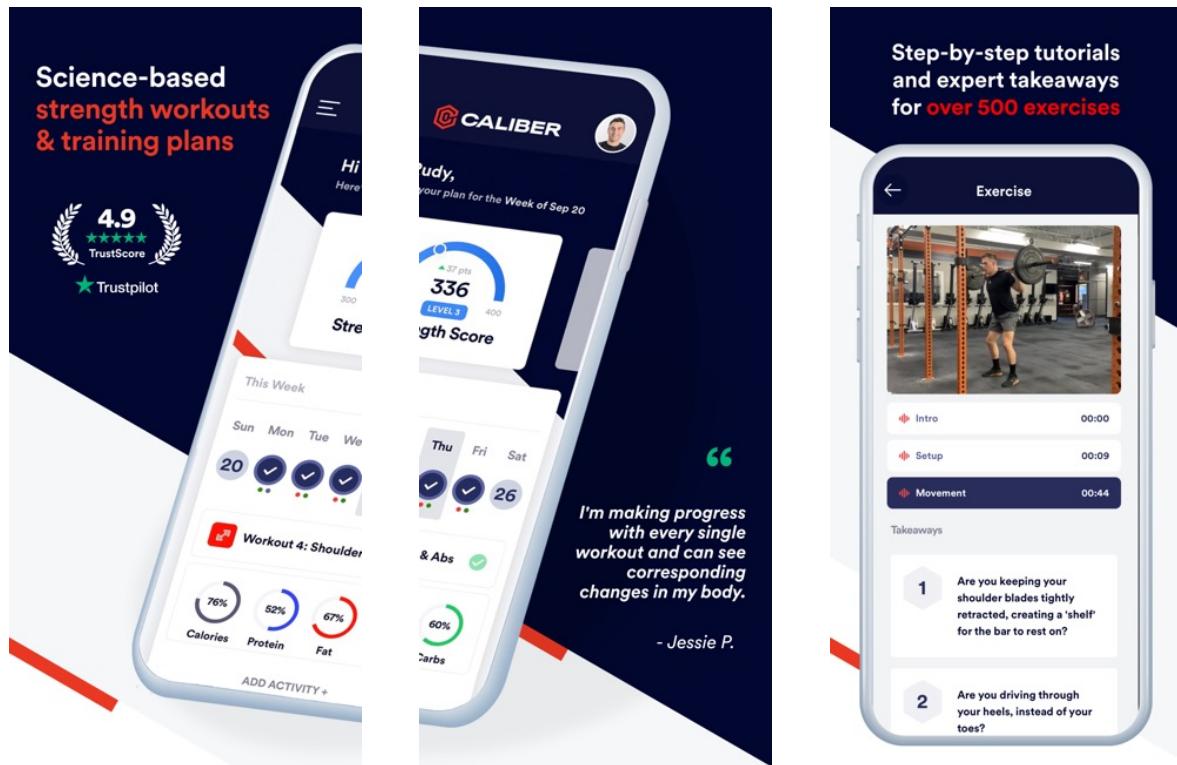


Figura 3.4: Aplicación Caliber Fitness

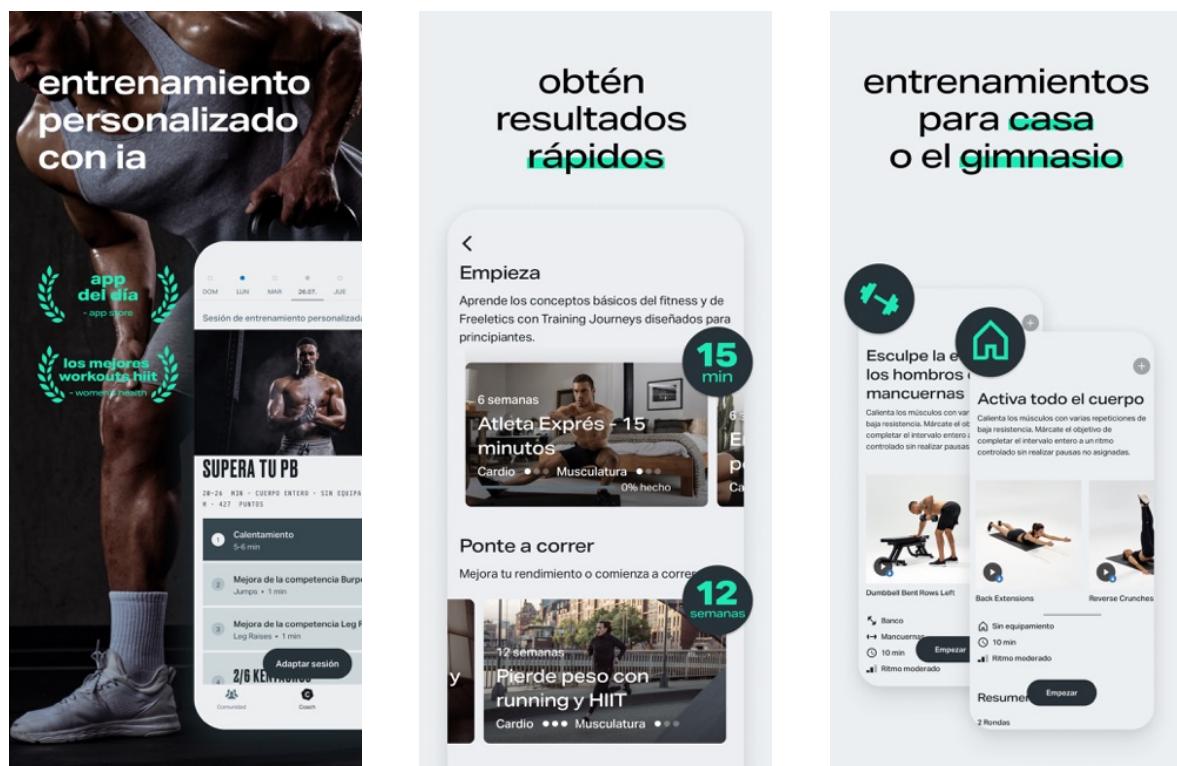


Figura 3.5: Aplicación Freeletics

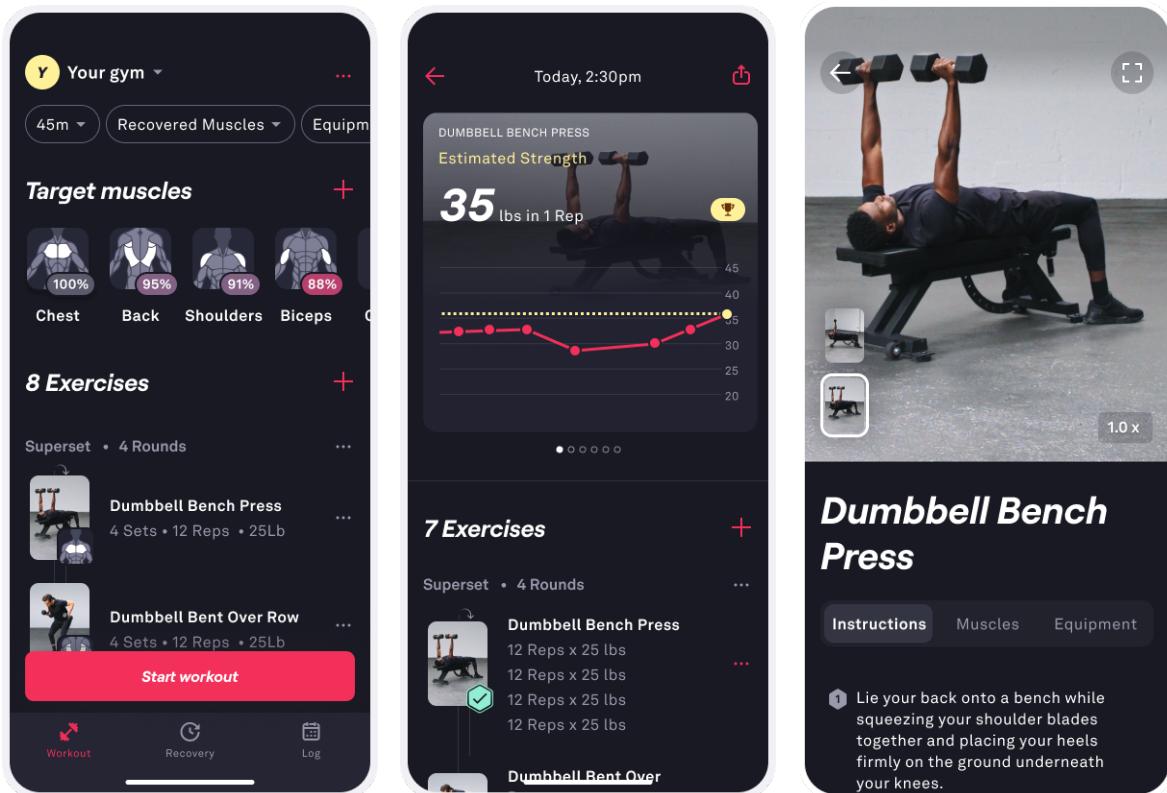


Figura 3.6: Aplicación Fitbod

- Seguimiento de Progreso:** poder registrar las rutinas de entrenamiento realizadas, con información detallada sobre los ejercicios realizados, repeticiones y series, entre otros datos. Además de incluir medidas de rendimiento como el número de rutinas completadas y el número de horas activo, entre otros aspectos, para medir el progreso en términos de una meta específica.
- Recomendaciones Basadas en Datos:** utilizar la información recopilada sobre el usuario, para sugerir recomendaciones de cambio o ajustes en cualquier ámbito para alcanzar objetivos específicos.
- Calendario:** incluir un calendario que sirva como guía para que los usuarios puedan programar sus sesiones futuras de entrenamiento.
- Interacción con Asistente Virtual:** ofrecer asesoramiento personalizado acerca de consultas relacionadas con la rutina actual.
- Uso de Notificaciones:** capacidad de mandar mensajes para comunicar al usuario sobre eventos relevantes y logros alcanzados, entre otras cosas.
- Compatibilidad Multiplataforma:** permitir el acceso y utilización de la aplicación tanto en dispositivos móviles como en ordenadores, manteniendo la accesibilidad.
- Interfaz Intuitiva y Accesible:** interfaz diseñada de manera clara, fácil de usar y accesible. Para facilitar la navegación y visualización de datos en la aplicación.

En la tabla 3.2 se compara la disposición de las características mencionadas en las aplicaciones y sistemas web analizados:

	Mi aplicación	Google Fit	Jefit	Caliber	Freeletics	Fitbod
Rutinas de entrenamiento personalizadas	Sí	-	Sí(*)	Sí	Sí	Sí
Utilización de Inteligencia Artificial	Sí	-	-	-	Sí	Sí
Cuestionario inicial	Sí	-	Sí	Sí	Sí	Sí
Seguimiento de Progreso	Sí	Sí	Sí	Sí	Sí	Sí
Recomendaciones Basadas en Datos	Sí	-	-	-	-	Sí
Calendario	Sí	-	Sí	Sí	Sí	Sí
Interacción con Asistente Virtual	Sí	-	-	-	-	-
Uso de Notificaciones	Sí	Sí	Sí	Sí	Sí	Sí
Compatibilidad Multiplataforma	Sí	-	Sí	Sí	Sí	-
Interfaz intuitiva y accesible	Sí	Sí	Sí	Sí	Sí	Sí

Tabla 3.2: Tabla comparativa aplicaciones y sistemas web analizados

*Jefit no ofrece principalmente rutinas personalizadas a los usuarios; en cambio, proporciona herramientas para que los usuarios puedan crear sus propias rutinas personalizadas.

CAPÍTULO 4

Metodología

4.1. METODOLOGÍAS DE DESARROLLO SOFTWARE

Para garantizar el éxito en el proceso de desarrollo de software, no basta con valerse de notaciones de modelado y herramientas, es esencial apoyarse en un componente crucial: una metodología de desarrollo que pueda orientar en la correcta aplicación de los demás componentes. El desarrollo de software no es algo sencillo. Una evidencia es la existencia de múltiples enfoques metodológicos que abordan diversas áreas del proceso de desarrollo [33].

Desde un punto de vista, existen enfoques más tradicionales que se centran en el control del proceso, definiendo con precisión los roles, actividades y artefactos que deben desarrollarse, como modelado, herramientas y documentación minuciosa. Estos enfoques han sido eficaces y fundamentales en muchos proyectos de gran tamaño en cuanto a tiempo y recursos, sin embargo, también han tenido dificultades en otros proyectos con entornos muy cambiantes que requieren disminuir los tiempos de desarrollo sin comprometer la calidad. Frente a estos problemas, una opción de mejora podría ser enfocarse en otras áreas, como el factor humano o el producto software. Esta es la creencia de las metodologías ágiles, las cuales valoran más al individuo, la colaboración con el cliente y el desarrollo incremental del software con iteraciones muy cortas sin descuidar las *buenas prácticas*.

4.1.1. Metodologías Ágiles

La metodología ágil [25] es un enfoque para la gestión de proyectos y desarrollo de software a través de la implementación rápida de equipos flexibles y eficientes para planificar el proceso de trabajo. *Agile* permite tomar la decisión más conveniente en cada circunstancia sin poner en riesgo el proyecto.

La aplicación de las Metodologías Ágiles sigue un ciclo de desarrollo iterativo e incremental, siendo crucial la contribución del factor humano para lograr el éxito del proyecto. Este enfoque posibilita la entrega del software en partes pequeñas y funcionales, denominadas incrementos. Cada iteración se puede ver como un pequeño proyecto abordable en el que se realizan actividades de análisis de requerimiento, diseño, implementación y pruebas para crear una parte del sistema final, reduciendo así el coste y posibles fallos. El proceso se repite múltiples veces generando un nuevo incremento en cada ciclo y la elaboración del producto completo marca el final de dicho proceso. Introducen nuevos métodos de trabajo que demandan una cantidad de procesos que no se ven afectados por tareas administrativas como la planificación, el control y la documentación, ni respaldan la idea radical de no contar con ningún proceso.

La **Metodología Agile** es la metodología de desarrollo software seleccionada para realizar el desarrollo de este Proyecto Fin de Grado. Esta decisión se basa en las numerosas ventajas que proporciona dicha metodología beneficiosas para el proyecto, entre las cuales podemos resaltar [12]: capacidad de adaptación a cambios en los requisitos durante el proceso de desarrollo, entrega continua y en plazos breves de software que cumpla con las funciones requeridas, colaboración entre el cliente

y el equipo de desarrollo, importancia de la simplicidad, enfoque en la excelencia técnica y el diseño de calidad y mejoras frecuentes del producto y proceso.

Asimismo, estas metodologías ágiles de desarrollo son especialmente beneficiosas para el proyecto debido a que son ideales para proyectos con requisitos poco claros o muy cambiantes, junto con las situaciones en las que se necesitan disminuir significativamente los tiempos de desarrollo sin comprometer la calidad. Igualmente, estas metodologías son efectivas en equipos pequeños que abordan problemas específicos, lo cual no impide su uso en el desarrollo de sistemas más grandes, dado que una adecuada división en módulos es esencial para su implementación exitosa.

Manifiesto Ágil

El Manifiesto Ágil es el documento que establece los principios fundamentales de las metodologías ágiles de desarrollo software. La propuesta era brindar otra opción a los métodos tradicionales, conocidos por ser inflexibles y estar guiados por la elaboración de documentación en cada etapa.

El Manifiesto Ágil consta de cuatro postulados y varios principios. Sus postulados son [13]:

1. *Valorar al individuo y a las interacciones del equipo de desarrollo por encima del proceso y las herramientas.* Señala que las personas son componentes fundamentales en todo proceso de desarrollo.
2. *Valorar el desarrollo de software que funcione por sobre una documentación exhaustiva.* Enuncia que el valor de la comunicación directa entre individuos al interactuar con los prototipos no puede ser reemplazada ni igualada por los documentos.
3. *Valorar la colaboración con el cliente por sobre la negociación contractual.* El cliente colabora activamente con el equipo de trabajo. El contrato en sí no agrega valor al producto, simplemente es un procedimiento formal.
4. *Valorar la respuesta al cambio por sobre el seguimiento de un plan.* El proceso de desarrollo debe incluir una rápida evolución y continuidad, priorizando sobre la capacidad de seguimiento y la garantía de planes preestablecidos.

A pesar de que la mayoría de las metodologías ágiles se adhieren a los postulados y principios del Manifiesto Ágil, su aplicación puede variar considerablemente entre ellas. Entre las metodologías ágiles disponibles en la actualidad, concretamente se ha escogido la metodología *Scrum* para llevar a cabo el proyecto.

4.1.2. Metodología Scrum

No se trata de una sigla, sino de un término utilizado en el rugby para describir la formación que se usa para la recuperación rápida del juego tras una infracción menor. La metodología de desarrollo ágil, *Scrum*, es un marco de trabajo creado para favorecer la colaboración eficaz entre equipos en proyectos, que utiliza una serie de reglas y artefactos y establece roles que mantienen la estructura adecuada para su funcionamiento adecuado [11].

Scrum sigue un enfoque incremental que se basa en la teoría de control empírico de procesos. Se sustenta en tres pilares clave:

- **Transparencia;** asegura que se pueda ver claramente todo lo que pueda influir en el resultado.
- **Inspección;** contribuye a identificar variaciones no deseadas en el proceso.
- **Adaptación;** efectúa los cambios necesarios para reducir el impacto de estas variaciones.

4.1.2.1. Método de Aplicación

Es importante considerar que para utilizar metodologías ágiles es fundamental estar en un contexto analítico y abierto a cambios. Esto permite adaptar las metodologías ágiles según las necesidades específicas de cada compañía y proyecto en particular. También es crucial proteger aspectos clave como la planificación tradicionalmente utilizada en proyectos software. Esto sugiere que, a pesar de

que las metodologías ágiles promuevan la flexibilidad y adaptabilidad, reconocen la importancia de una planificación adecuada para el éxito del proyecto.

Scrum se focaliza en identificar las funcionalidades más prioritarias y factibles de completar en poco tiempo, los ciclos de iteración, llamados *sprints*, facilitan un proceso de trabajo funcional y productivo.

Los pasos necesarios para implementar y ejecutar *Scrum* en un proyecto software son:

- **Definir los diferentes roles** El componente básico de *Scrum* consiste en un reducido equipo de personas; un *Scrum Team*. El *Scrum Team* se compone de un *Scrum master*, un *Product Owner* y los *Developers*.

1. El ***Scrum Master*** tiene la responsabilidad de implementar *Scrum* según lo descrito en la Guía de *Scrum* [52]. Su función es garantizar que el *Scrum Team* esté siguiendo la metodología, sus prácticas, valores y normas; es el líder del equipo pero no gestiona el desarrollo.

En este proyecto, este rol se lleva a cabo por la autora de este Trabajo Fin de Grado, Tamara Redondo Soto.

2. El ***Product Owner*** es un individuo único que actúa en nombre de los interesados, su rol es optimizar el valor del producto y el trabajo del equipo de desarrollo; una de sus responsabilidades incluye gestionar la lista ordenada de funcionalidades requeridas o *Product Backlog*.

En este proyecto, este rol se asume por el tutor Santiago Sánchez Sobrino.

3. Los ***Developers***, son los miembros del *Scrum Team* que se comprometen a desarrollar cualquier parte de un *Increment* utilizable en cada *sprint*. Su tarea consiste en transformar las necesidades del cliente, expresadas en el *Product Backlog*, en iteraciones funcionales del producto. El número ideal de integrantes del grupo varía entre tres y nueve personas.

En este proyecto, ha sido formado por una única persona, Tamara Redondo Soto. Desempeñando las funciones de desarrollo backend y frontend, administración de la base de datos y evaluación del sistema.

- **Planificación de Sprints** En *Scrum*, se determina un evento clave o *Sprint*, que representa una ventana de tiempo durante la cual se desarrolla una versión funcional del producto (incremento). Cada *sprint* tiene la duración máxima de un mes y se trata como un proyecto independiente.

El proyecto, siguiendo la metodología *Scrum*, se ha organizado en una serie de *sprints*. Detallando que cada *sprint* ha tenido una duración de 30 días.

En este proyecto esta elección se basa en que un *sprint* de 30 días permite equilibrar las responsabilidades académicas con el desarrollo del proyecto, asegurando la calidad del trabajo y el rendimiento académico. En equipos bien organizados de software, este período es óptimo para generar valor, destacando la importancia de una buena organización. Como única desarrolladora, este tiempo posibilita que cada entrega cumpla con los estándares definidos en la *Definition of Done*. Asimismo, un *sprint* más largo es más eficaz al gestionar incidencias y mantener la estabilidad del proyecto, mientras que la gradualidad del desarrollo minimiza el impacto de las reuniones y eventos regulares del *sprint*.

Un *sprint* se compone de las siguientes fases:

1. ***Sprint Planning***. En esta reunión se establece el plan de trabajo; qué será entregado y cómo se llevará a cabo. En otras palabras, el diseño del sistema y la estimación de cantidad de trabajo. La duración de esta tarea es de ocho horas en un *sprint* de un mes. En caso de que el *sprint* sea más corto, el tiempo se asigna de manera proporcional.
2. ***Daily Scrum***. Es una reunión diaria del equipo de desarrollo con una duración de quince minutos, en la que se informa sobre el progreso desde la última reunión, lo que se hará

hasta la próxima reunión y los problemas que han surgido.

3. **Trabajo de desarrollo.** El equipo de desarrollo comienza su labor siguiendo los planes de entrega establecidos.
4. **Sprint Review.** La Revisión del *Sprint* tiene lugar al concluir el *sprint* y dura cuatro horas o una proporción equivalente según la duración del *sprint*. En esta revisión el *Product Owner* analiza las tareas realizadas, detecta las pendientes y discute acerca del *Product Backlog*; el equipo de desarrollo señala los obstáculos enfrentados, cómo se solucionaron y presenta el producto y su funcionalidad. Esta reunión será crucial para los próximos *sprints*.
5. **Sprint Retrospective.** Consiste en una reunión del *Scrum Team* de tres horas en la que se evalúa la comunicación, el proceso y las herramientas utilizadas, identificando aciertos y errores para implementar un plan de mejoras en el próximo *sprint*.
6. **Inicio inmediato del siguiente sprint.** Después de finalizar todas las fases del *sprint*, el equipo y su entorno deben trabajar juntos para iniciar nuevamente el proceso en el que se planifica el siguiente *sprint*.

Los **Artefactos de Scrum**, son los subproductos derivados de las actividades del marco de trabajo. Cada artefacto incluye un compromiso para asegurar que proporciona información que aumente la transparencia y sirve como punto de referencia para medir el progreso:

- Para el **Product Backlog**, es el Objetivo del Producto.
- Para el **Sprint Backlog**, es el Objetivo del *sprint*.
- Para el **Increment**, es la Definición de Terminado (*Definition of Done*).

Los artefactos de *Scrum* se dividen en:

- El *Product Backlog* es una lista de requerimientos definidos, actualizados y ordenados por el *Product Owner*, en base a su valor, riesgo, prioridad y necesidad. La lista nunca se considera terminada, ya que cambia a lo largo del proyecto en desarrollo. Incluye todos los requisitos funcionales y no funcionales que el sistema debe cumplir.
- El *Sprint Backlog* es una parte del *Product Backlog* y la estrategia para llevar a cabo el *Increment* del producto. Se compone de una serie de requisitos; los más prioritarios del *Product backlog*, los pendientes por resolver del *sprint* anterior y los adicionales. El equipo de desarrollo debe aprobar el *Sprint backlog* ya que les pertenece y solo ellos pueden modificarlo.
- El Monitoreo de Progreso implica calcular la cantidad de trabajo pendiente durante el *sprint*. Se destaca por presentarse en cualquier instante, lo que posibilita al *Product Owner* evaluar el progreso del desarrollo.
- El *Increment* consiste en la suma de los elementos terminados en el *Sprint backlog*. En caso de que haya elementos no completados, es necesario regresarlos al *Product backlog* con alta prioridad para que se incluyan en el próximo *sprint*. Un elemento está terminado si es funcional. El total de elementos terminados es el producto a entregar.

4.2. PLANIFICACIÓN DEL TRABAJO

Para llevar a cabo el proyecto, se ha adoptado la metodología *Scrum*, organizando el trabajo en *sprints* numerados del 0 al 5, claramente definidos. La realización del proyecto desde su inicio hasta su fin, ha requerido un total aproximado de 171 días de trabajo independiente. La totalidad de días dedicados al proyecto es una estimación, no es una medida precisa.

Un **Diagrama de Gantt** es una metodología de representación de actividades y tareas con el objetivo de dar una visión generalizada del tiempo invertido a cada actividad considerada de forma independiente dentro de un proceso [19].

En la figura 4.1 se muestra el diagrama de Gantt como cronograma del proyecto. Cada barra horizontal representa un *sprint* específico. El calendario completo del proyecto se muestra en el eje horizontal, desde la fecha de inicio del primer *sprint* hasta la fecha de finalización del último *sprint*, permitiendo una representación visual intuitiva y rápida del tiempo que se ha asignado para cada etapa del proyecto y del total dedicado al proyecto en su conjunto.

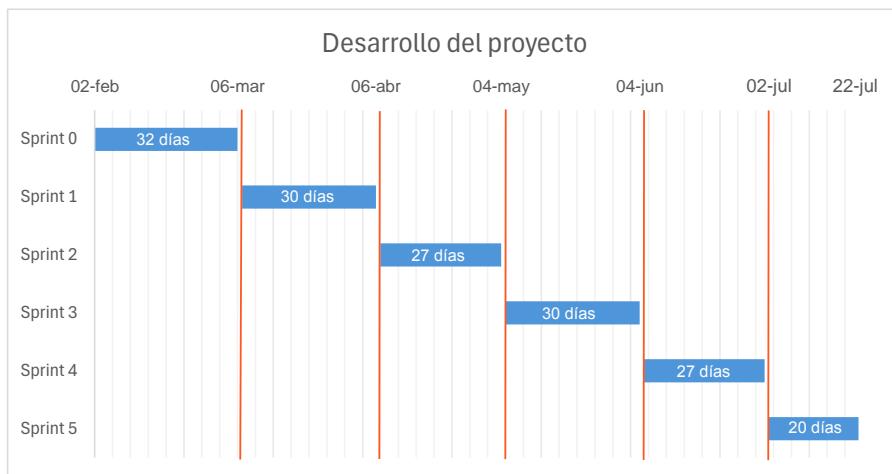


Figura 4.1: Cronograma de Trabajo - Diagrama Gantt

4.3. PILA TECNOLÓGICA

En esta sección, se describe detalladamente la selección de la tecnología utilizada en el desarrollo del proyecto. Cada componente de la pila tecnológica desempeña un papel crucial en el éxito de este proyecto web:

4.3.1. Entornos de desarrollo

- **Visual Studio Code**¹. Es un editor de código fuente diseñado para utilizar recursos mínimos y responder de forma rápida y eficiente. Funciona en Windows, macOS y Linux, y está especialmente optimizado para JavaScript, TypeScript y Node.js.

Además, se beneficia de un amplio ecosistema de extensiones que permiten personalizar y optimizar la herramienta. Estas extensiones proporcionan diversas funcionalidades de manera modular y aislada, lo que permite adaptar la experiencia según las necesidades del usuario. Además, hay extensiones disponibles para soportar otros lenguajes y entornos de ejecución como C++, C#, Java, Python y .NET, facilitando así el desarrollo en esos entornos específicos. En este caso, se han incorporado varias extensiones para ampliar las capacidades de Visual Studio Code para diferentes aspectos del desarrollo: *Angular Language Service*, *Angular Snippets (Version 16)* y *Angular 17 Snippets - TypeScript, Html, Angular Material, ngRx, RxJS & Flex Layout*, útiles para facilitar el desarrollo eficiente con el entorno de desarrollo Angular. Asimismo, se han utilizado las extensiones *Docker* y *Dev Containers* para facilitar la gestión de contenedores Docker y garantizar entornos de desarrollo consistentes y reproducibles.

El entorno de desarrollo Visual Studio Code se ha utilizado para el desarrollo de la parte frontend en el proyecto. La versión empleada ha sido la 1.90.2.

- **Chrome DevTools**². Se trata de un conjunto de herramientas para desarrolladores web incorporadas en el navegador de Google Chrome. Permite editar páginas al momento y diagnosticar problemas rápidamente, lo que facilita la creación de sitios web de mayor calidad en menos

¹<https://visualstudio.microsoft.com/es/>

²<https://developer.chrome.com/docs/devtools?hl=es-419>

tiempo. A pesar de que Visual Studio Code no cuenta con Chrome DevTools integrado ni está vinculado directamente con Chrome, es importante que todo desarrollador web utilice las Chrome DevTools en conjunto con el editor de código. Siendo esencial para un desarrollo web eficiente y completo.

- **Eclipse IDE³**. En particular se usa una configuración predefinida de Eclipse IDE: **Eclipse IDE for Enterprise Java and Web Developers**, que proporciona un conjunto de herramientas para desarrolladores enfocados en Java y aplicaciones web, incluyendo un entorno de desarrollo integrado (IDE) para Java, integración de Git para Eclipse, herramientas de desarrollo Java EE (Enterprise Edition), integración de Maven para Eclipse, un sistema de gestión de tareas Mylyn y más.

El entorno de desarrollo Eclipse se ha utilizado para el desarrollo de la parte backend en el proyecto. La versión empleada ha sido la 2023-12 (4.30.0).

- **Apache Maven⁴**. Es una herramienta de comprensión y gestión de proyectos de software. Usa el concepto de Modelo de Objetos de Proyecto (POM), Maven puede manejar la creación, los informes y la documentación de un proyecto desde una sola fuente de información. Eclipse ofrece la integración de Maven, por lo que incluye soporte nativo y no es necesario instalar herramientas adicionales para gestionar las dependencias de Maven en el proyecto Spring Boot.

La versión utilizada de Apache Maven es la 3.9.6.

4.3.2. Frontend

Además del entorno de desarrollo Visual Studio Code, para el desarrollo de la parte frontend se necesitan otras tecnologías:

- **Angular⁵**. Angular es un framework de diseño de aplicaciones y plataforma de desarrollo para crear aplicaciones de una sola página eficientes y sofisticadas.

La versión utilizada de Angular es la 18.0.1. Para desarrollar aplicaciones con Angular, es esencial contar con:

- **Angular CLI**. Es una herramienta de interfaz de línea de comandos empleada para inicializar, desarrollar, estructurar y mantener aplicaciones Angular directamente desde un shell de comandos.

La versión utilizada de Angular CLI es la 18.0.1.

- **Node.js y Package Manager (npm)** Angular emplea Node.js y npm para gestionar dependencias y ejecutar los scripts de construcción requeridos para compilar la aplicación Angular y prepararla para su despliegue.

La versión utilizada de Node.js es la 20.11.0 y la versión del Package Manager es la 10.5.0.

- **Bootstrap⁶**. Es un framework de desarrollo web de código abierto, diseñado con el propósito de ayudar en la creación de los sitios web responsivos. Proporciona una colección de sintaxis para diseños de plantillas.

En el proyecto, se ha integrado Bootstrap para aprovechar sus beneficios en la creación de la interfaz de usuario, se ha utilizado la versión 5.2.3.

- **TypeScript⁷**: TypeScript es el lenguaje de programación principal utilizado por Angular para desarrollar el código de la aplicación. Angular viene integrado con TypeScript, por lo

³<https://eclipseide.org/>

⁴<https://maven.apache.org/>

⁵<https://angular.dev/>

⁶<https://getbootstrap.com/>

⁷<https://www.typescriptlang.org/>

que no hace falta tener TypeScript por separado para desarrollar aplicaciones Angular. Sin embargo, contar de manera global Typescript en el equipo facilita la ejecución del compilador de TypeScript('tsc') desde cualquier ubicación, así como los comandos globales, sin necesidad de estar dentro del proyecto Angular.

La versión utilizada de TypeScript es la 5.4.5.

4.3.3. Backend

- **Java SE Development Kit⁸**. El Kit de Desarrollo de Java (JDK) es un software esencial para los desarrolladores de Java, ya que proporciona el intérprete Java, clases Java y herramientas de desarrollo Java (JDT), como el compilador, depurador, desensamblador, visor de applets, generador de archivos de apéndice y generador de documentación. En Eclipse, para desarrollar aplicaciones Java, es necesario utilizar un JDK en lugar de un Entorno de Ejecución de Java (JRE). La principal diferencia radica en que el JDK no solo permite ejecutar aplicaciones Java compiladas, como lo hace JRE, sino que también incluye herramientas necesarias para desarrollar y compilar nuevo código Java, lo que resulta esencial para poder crear de manera eficaz aplicaciones y applets en este entorno de desarrollo integrado.

La versión utilizada de Java es la 21.0.1.

- **Java Spring Boot⁹**. Java Spring Boot (Spring Boot) se trata de una herramienta que agiliza y facilita el desarrollo de microservicios y aplicaciones web utilizando Spring Framework.

La versión utilizada de Spring Boot es la v3.1.0.

- Spring Tools Suite (STS)¹⁰. Es un conjunto de herramientas diseñadas especialmente para simplificar el desarrollo de aplicaciones basadas en Spring dentro del entorno de Eclipse. STS ofrece soporte para el desarrollo de aplicaciones basadas en Spring y Spring Boot. Contiene características como la asistencia en la escritura de código, la integración con Spring Boot, el soporte para Maven, entre otros recursos.

La versión de Spring Tools 4 (STS) utilizada es la 4.23.1.RELEASE.

Spring Boot permite crear aplicaciones autocontenidoas, delega a Spring Boot tareas como configuración de dependencias y desplegar el servicio o aplicación a un servidor de aplicaciones. Para ello, Spring Boot usa de forma interna un servidor de aplicaciones embebido, que por defecto es Tomcat:

- **Apache Tomcat¹¹**. Se trata de un servidor web y contenedor de servlets que ejecuta aplicaciones web basadas en Java. Además, es una implementación del Java Servlet, JavaServer Pages (JSP), Java Expression Language y Java WebSocket.

La aplicación está integrada con Apache Tomcat, en la versión 10.1.8

Asimismo, Spring Boot simplifica la configuración de la base de datos y gestiona las dependencias de forma automática a través de Maven, esto incluye los **Conectores JDBC¹²** necesarios para conectar la base de datos. En proyectos que utilizan *spring-boot-starter-data-jpa*, Spring Boot configura automáticamente JPA para trabajar con Hibernate como proveedor de persistencia por defecto, gestionándolo internamente.

- **Java Hibernate¹³**. Es una herramienta de mapeo objeto-relacional (ORM) para el lenguaje de programación Java. Ofrece un marco para mapear un modelo de dominio orientado a objetos a una base de datos relacional. Eclipse emplea Hibernate para simplificar la

⁸<https://www.oracle.com/java/technologies/downloads/>

⁹<https://spring.io/projects/spring-boot>

¹⁰<https://spring.io/tools>

¹¹<https://tomcat.apache.org/>

¹²<https://www.mysql.com/products/connector/>

¹³<https://hibernate.org/>

interacción con una base de datos MySQL. Además, Eclipse está configurado para usar el dialecto MySQLDialect, el cual optimiza las consultas SQL generadas por Hibernate para que sean compatibles y eficientes con la base de datos MySQL específica en uso.

La versión de Hibernate ORM core es la 6.2.2.Final, asimismo, Hibernate está utilizando MySQLDialect en la versión 8.0.36.

- **MySQL**¹⁴. Es un sistema de administración de bases de datos relacionales. Es un software de código abierto desarrollado por Oracle.
 - **MySQL Workbench**¹⁵. Es una herramienta visual integrada para la administración de bases de datos que proporciona un conjunto de herramientas para diseñar, administrar, modelar y configurar bases de datos MySQL.
- **Postman**¹⁶. Se trata de una herramienta para desarrolladores web que les permite realizar peticiones HTTP a cualquier API. En este proyecto se ha utilizado para llevar a cabo pruebas de APIs.

4.3.4. Orquestación

- **Docker**¹⁷. Docker es una plataforma de software que facilita la creación, prueba e implementación ágil de aplicaciones. Docker empaqueta software en unidades estandarizadas llamadas contenedores que contienen todo lo necesario para que el software se ejecute.

4.3.5. Herramientas de gestión y planificación

- **Github**¹⁸. Es un servicio basado en la nube que ofrece alojamiento para sistemas de control de versiones (VCS) basados en Git. Permite mantener un seguimiento detallado del progreso. En este proyecto se ha utilizado para llevar a cabo el control de versiones.
- **Microsoft Teams**¹⁹. Es una plataforma unificada de comunicación y colaboración que ofrece chat persistente en entornos laborales, reuniones por video, almacenamiento de archivos con funciones colaborativas, y la capacidad de integrar diversas aplicaciones. Se ha utilizado para la interacción del alumno con el tutor académico de este TFG.
- **Overleaf**²⁰. Es una plataforma vía web que facilita la creación y visualización de documentos escritos en L^AT_EXdirectamente desde el navegador. Además, permite compartir y colaborar en tiempo real sobre el documento, realizar un control de versiones y exportarlos a diferentes formatos. Ha sido utilizado para el desarrollo de la memoria de este TFG.
- **Visual Paradigm**²¹. Es una herramienta de modelado y diseño de software adoptada en la industria del desarrollo de software. Proporciona una plataforma completa para la creación de modelos visuales que incluyen desde diagramas de clases y de flujo hasta modelos detallados de arquitectura y diseño de sistemas. Se ha utilizado para el modelado de software y diseño de bases de datos.
- **Balsamiq Mockups**²². Es una herramienta de diseño utilizada principalmente para crear maquetas o prototipos de interfaces de usuario y sitios web. Se caracteriza por su enfoque en la simplicidad y rapidez, permitiendo diseñar esquemas visuales de manera intuitiva y ágil. Se ha utilizado para el diseño de interfaces de usuario.

¹⁴<https://www.mysql.com/>

¹⁵<https://www.mysql.com/products/workbench/>

¹⁶<https://www.postman.com/>

¹⁷<https://www.docker.com/>

¹⁸<https://github.com/>

¹⁹<https://www.microsoft.com/es-es/microsoft-teams/log-in>

²⁰<https://es.overleaf.com>

²¹<https://www.visual-paradigm.com/>

²²<https://balsamiq.com/>

CAPÍTULO 5

Arquitectura

En esta sección, se lleva a cabo un análisis exhaustivo de los componentes arquitectónicos que constituyen la columna vertebral de este proyecto. Se realiza una revisión minuciosa de cada uno de los módulos que integran el servicio web. Se han diseñado varios diagramas que proporcionan una perspectiva completa y organizada de las distintas partes que forman la configuración del proyecto. Estos diagramas facilitan la comprensión tanto la interrelación entre los componentes como el flujo de datos y procesos que sostienen la funcionalidad del sistema.

5.1. VISIÓN GENERAL

El sistema en términos generales consta de cuatro capas distribuidas entre el frontend y backend, con una capa de seguridad sumada entre ambos componentes. La figura 5.1 representa el diagrama completo del sistema web.

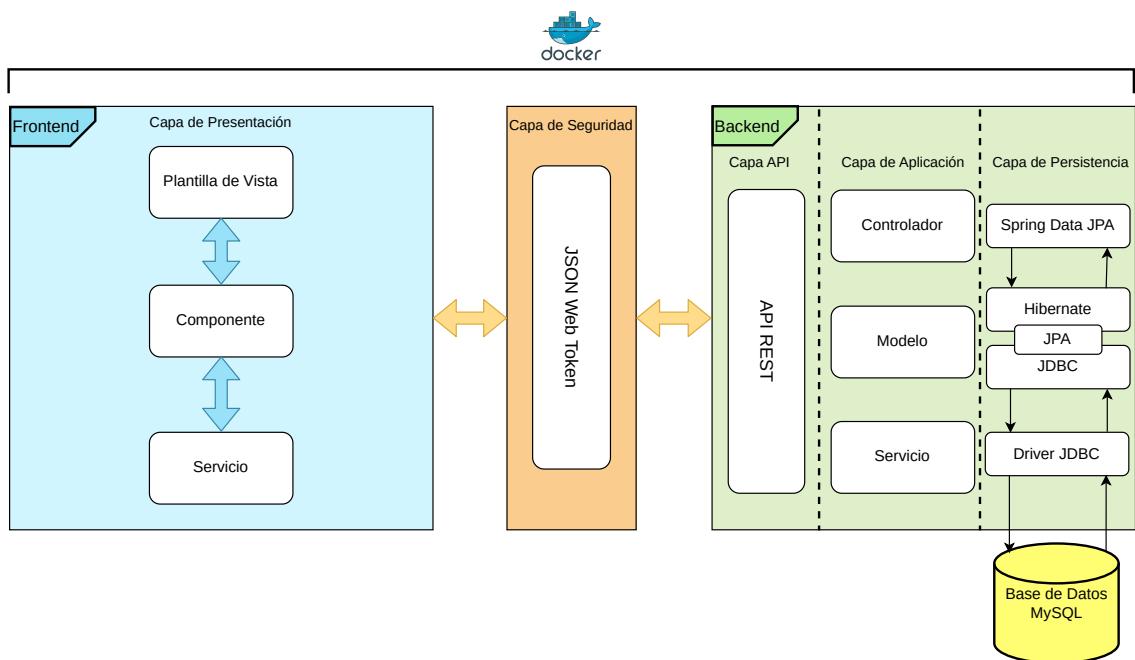


Figura 5.1: Arquitectura general de la aplicación web. (Fuente: T.Redondo, CC BY-NC-ND)

El sistema es protegido por una capa de seguridad que emplea un sólido mecanismo de gestión de identidad y acceso para proteger sus recursos. En esta implementación en particular, se utilizan JSON Web Tokens, **JWT**, para asegurar la autenticación y autorización de los usuarios de manera escalable. Los JWT se emplean para proteger las rutas de la API, garantizando que únicamente los usuarios verificados puedan acceder a los recursos protegidos.

Asimismo, la capa de persistencia interactúa de forma directa con la base de datos para gestionar volúmenes extensos de información, donde los datos de la aplicación se almacenan de manera estructurada y eficiente, conectándose entre sí en una unidad lógica.

Para garantizar la consistencia y el despliegue eficiente del entorno de desarrollo y producción, se emplea **Docker**. Docker posibilita la creación de contenedores estandarizados que contienen tanto la aplicación como todas sus dependencias, garantizando así la ejecución consistente del software en cualquier entorno. Cada componente de la arquitectura, debe ejecutarse en su propio contenedor. La orquestación de estos contenedores se gestiona mediante un único servicio a través de Docker Compose, lo que facilita y organiza el despliegue de la aplicación.

5.1.1. Arquitectura frontend y backend del sistema

La aplicación del cliente utiliza Angular como el principal *framework* para el desarrollo de aplicaciones de una sola página (SPA). La **capa de presentación** se estructura en plantillas de vista, componentes y servicios.

Un **componente** controla una parte en pantalla llamada vista. Una **vista** es la agrupación más pequeña de elementos de visualización que se pueden crear y destruir al mismo tiempo. Angular se trata de una vista dirigida por una o varias directivas. Una clase componente y su plantilla asociada describen una vista. Una plantilla es una forma de HTML que indica a Angular cómo renderizar el componente. Una vista está específicamente representada por una instancia de Referencia de Vista, *ViewRef*, que está asociada con un componente. Las vistas suelen estructurarse en forma de jerarquías, lo que permite modificar, mostrar u ocultar secciones completas de la interfaz de usuario o páginas como una unidad. Una vista que pertenece directamente a un componente se conoce como vista de *host*.

Angular crea, actualiza y destruye componentes conforme el usuario navega por la aplicación. La aplicación puede responder en cada etapa de ese ciclo de vida mediante el uso de enlaces opcionales de ciclo de vida, como *ngOnInit()*. En teoría, el propósito principal de un componente es hacer más fácil la experiencia del usuario. Esto se logra exponiendo propiedades y métodos para el enlace de datos, actuando de intermediario entre la vista, definida por la plantilla, y la lógica de la aplicación, que generalmente incluye algún tipo de modelo. La lógica de aplicación de un componente se define dentro de una clase específica. Esta clase define cómo se comporta el componente y cómo interactúa con la vista a través de una API de propiedades y métodos.

Angular diferencia los componentes de los **servicios** para potenciar la modularidad y la posibilidad de reutilización. Al separar la funcionalidad relacionada con la vista de un componente de otros tipos de procesamiento se puede lograr que los componentes sean rápidos y eficientes. Un servicio es un término amplio que engloba cualquier valor, función o característica que requiera una aplicación. Un servicio suele ser una clase con un propósito bien definido y limitado. Necesita realizar una tarea concreta y hacerla bien. Un componente puede encomendar ciertas responsabilidades a los servicios, tales como obtener datos del servidor, validar la entrada del usuario o registrar información en la consola. Cuando se establecen estas tareas de procesamiento dentro de una clase de servicio inyectable, se asegura que estén accesibles para todos los componentes. Además, es posible aumentar la flexibilidad de la aplicación inyectando varios proveedores del mismo tipo de servicio, según sea necesario en distintos escenarios.

Por otro lado, la aplicación del servidor utiliza Spring Boot como framework principal para controlar la lógica de negocio y gestionar la interacción con la base de datos. Se estructura en varias capas. La comunicación principal entre el frontend y el backend se realiza a través de APIs, que son herramientas diseñadas para conectar sistemas o softwares y permitir su interacción. La **capa API** expone los servicios web mediante una API REST, facilitando que el frontend se comunique con la lógica del servidor mediante peticiones HTTP. La API REST permite que el cliente y el servidor puedan interactuar sin que el cliente necesite conocer los detalles internos de implementación del servidor. Del mismo modo, el servidor puede manejar los datos enviados por el cliente sin preocuparse

por cómo serán utilizados. Spring Boot brinda un soporte de gran calidad para desarrollar servicios web RESTful.

Los principales métodos HTTP utilizados para interactuar con la API y gestionar recursos incluyen:

- **HTTP GET.** Se trata del protocolo de solicitud HTTP estándar. Su función es solicitar al servidor una información o recurso específico. Este método no necesita un cuerpo de solicitud.
- **HTTP POST.** Se utiliza para la creación de un recurso. En casos donde es necesario enviar información al servidor. Este método incluye el cuerpo de la solicitud.
- **HTTP PUT.** Se emplea para actualizar un recurso ya existente mediante la sustitución total de éste en el servidor con el nuevo estado enviado en el cuerpo de la solicitud. En caso de que el recurso exista, se reemplaza por completo con los datos proporcionados en la solicitud. En el caso contrario, se crea un nuevo recurso con la URL especificada.
- **HTTP PATCH.** Comparte similitudes con el método PUT al ser empleado para actualizar recursos existentes en un servidor. No obstante, se emplea para asignar cambios parciales a un recurso. En lugar de enviar todo el recurso al completo, solo se envían las partes que han sido modificadas.
- **HTTP DELETE.** Sirve para eliminar un recurso ya existente. Este método carece de un cuerpo de solicitud.

En una API REST, los recursos se identifican mediante una URI, Uniform Resource Identifier. Una URI es una representación de un recurso mediante una cadena de caracteres. Los servicios REST se basan en la manipulación de recursos. Los servicios RESTFUL disponen de recursos direccionables, y las URLs son el medio principal para lograr la direccionabilidad en REST. Una URL, Uniform Resource Locator, es un tipo de URI que no solo identifican de manera única el recurso, sino que también facilita su localización para acceder a él o compartir su ubicación.

En el proyecto se emplea el **patrón DTO**, *Data Transfer Object*, para estructurar y transportar la información entre las distintas capas de la aplicación. El objetivo del patrón DTO es crear un objeto plano (POJO) con varios atributos que puedan ser enviados o recuperados del servidor en una única invocación, permitiendo que un DTO contenga información de múltiples fuentes o tablas y concentrarlas en una única clase simple. La utilización de estos DTOs ofrecen varias ventajas significativas; permiten desacoplar la presentación, la capa de servicio y el modelo de dominio. La capa de presentación y la capa de servicio comparten contratos de datos en lugar de clases, estos contratos son representaciones neutrales de los datos intercambiados entre componentes. En el desarrollo de aplicaciones Java, las clases de entidad se utilizan para modelar las tablas en una base de datos relacional. Sin embargo, exponer directamente estas entidades a una interfaz remota puede no ser ideal. Por eso, se emplean los DTOs para separar las responsabilidades; aislan los datos utilizados en la lógica de negocio de aquellos específicamente necesarios para la interfaz de usuario. Esto también resulta en un aumento de la seguridad en la aplicación y previene la sobrecarga de datos innecesarios en la comunicación [39].

Mapper es una técnica que permite transferir información entre DTOs y entidades de manera bidireccional. Los *mappers* se pueden utilizar de diferentes formas; la forma más desacoplada es la creación de *mappers* mediante la Interfaz de Mapeo de *MapStruct* de Java, que automatiza este proceso y reduce la necesidad de código de mapeo manual. Alternativamente, se puede realizar el mapeo de forma manual mediante código personalizado sin depender de herramientas automáticas de generación de código. En el proyecto se ha optado por esta segunda alternativa, la cual resulta beneficiosa cuando se requiere un control detallado sobre el proceso de transformación de datos y el tamaño y la complejidad del proyecto no justifican la agregación de una capa adicional de *mappers* independiente. A pesar de que la opción escogida demanda un esfuerzo adicional en comparación con el uso de herramientas de mapeo automático.

Continuando con la **capa de aplicación**, la responsable de implementar y coordinar la lógica de negocio de la aplicación, está integrada por tres componentes: controladores, modelos y servicios.

El **controlador**, o *Controller*, actúa como intermediario entre el usuario y el sistema, gestionando las solicitudes HTTP entrantes y coordinando la respuesta. Generalmente, el controlador recibe datos del usuario a través de la solicitud, delega la tarea de procesamiento de la lógica de negocio a un servicio, y luego envía una respuesta adecuada de vuelta al usuario.

El modelo, o **Model**, por otro lado, define las representaciones del modelo de datos, tales como una clase, una entidad o cualquier otra estructura de datos que represente un concepto dentro del ámbito de la aplicación. Puede incluir tanto datos como la lógica asociada a esos datos, así como métodos para llevar a cabo operaciones sobre esos datos y se utiliza para representar y manipular datos dentro de la aplicación. Los modelos de datos sirven como una base sobre la cual se construyen los DTOs. A diferencia de los modelos de datos, los DTOs no contienen lógica de negocio ni métodos complejos.

El servicio, o **Service**, es responsable de la lógica de negocio y de manejar los datos recuperados de la capa de persistencia. Efectúa operaciones complejas en los datos como filtrado, ordenamiento y validación antes de ser enviados de vuelta al controlador o al repositorio.

Finalmente, la **capa de persistencia** tiene la responsabilidad de acceder a la base de datos y llevar a cabo operaciones *CRUD* (Crear, Leer, Actualizar, Eliminar). Ofrece una capa de abstracción sobre la base de datos para evitar que la capa de servicio se involucre en los detalles de almacenamiento de datos. Se puede acceder a la capa de persistencia mediante el controlador, pero es una práctica que se recomienda, y se utiliza en el proyecto, el delegar todo el acceso a la base de datos al objeto de servicio. Este enfoque promueve el principio de separación de intereses.

Para acceder a los datos almacenados en una base de datos relacional, existen diferentes opciones clasificadas en dos categorías: el acceso a bajo nivel y el acceso mediante un ORM (Object-Relational Mapping).

El **acceso a bajo nivel** implica el uso directo del lenguaje SQL, *Structured Query Language*, para llevar a cabo consultas y operaciones en la base de datos. Se requiere una biblioteca que posibilite establecer la conexión con la base de datos y la transferencia de los datos. En el lenguaje Java, esta librería se llama JDBC y es el método más básico y estándar de acceder a los datos.

Cuando se opta por trabajar con objetos en lugar de consultas SQL directas, se recurre a un **ORM**. Es decir, se requiere alguna biblioteca adicional que se encargue de generar y ejecutar las consultas, así como de mapear los resultados obtenidos a los objetos correspondientes del código. Acceder a través de un ORM implica emplear una capa de abstracción. De este modo, es posible operar con los datos como si se tratara de objetos de Java, evitando la escritura de código SQL. En Java, existen varias bibliotecas que aplican el concepto de ORM, las cuales examinaremos a continuación:

- **JDBC**, *Java Database Connectivity*, es la API principal de Java para el acceso a los datos de bases de datos relacionales. Permite ejecutar consultas SQL directamente en la base de datos y utilizar los resultados en la aplicación. JDBC ofrece un control exhaustivo sobre las consultas SQL y el manejo de los datos, sin embargo, esto implica escribir más código y ocuparse de detalles de bajo nivel. Para que JDBC pueda operar, requiere un *driver* específico de la base de datos que actúe como intermediario y que implemente la especificación JDBC particular de esa base de datos. Las librerías de ORM no implementan directamente JDBC; en lugar de eso, dependen de él y lo utilizan internamente para gestionar la comunicación con las bases de datos.
- **JPA**, *Jakarta Persistence*, es una especificación que define la manera en que los objetos en Java deben ser mapeados a tablas en una base de datos relacional. Esto es, JPA no se utiliza de forma directa, sino que determina una serie de conceptos que deben ser implementados por una biblioteca específica.

- **Hibernate** es la implementación más conocida y utilizada de la especificación JPA. Al usar Hibernate, se añade una capa adicional de complejidad, ya que entre la aplicación y los datos se interponen tanto Hibernate como JPA, además de la capa de bajo nivel y el *driver*.

Spring Data JPA es un componente de Spring Data. Ofrece un conjunto de abstracciones y convenciones para hacer más sencilla la implementación de repositorios de datos. Asimismo, produce de manera automática implementaciones para los métodos de consulta habituales en las interfaces de repositorio, lo cual puede resultar en un ahorro de tiempo. No obstante, Spring Data JPA necesita utilizar alguna implementación de JPA, por lo general Hibernate, para llevar a cabo las operaciones de base de datos.

5.1.2. Diseño de la base de datos

El diseño de una base de datos es el proceso en el que define la estructura de los datos necesaria para un determinado sistema de información. Implica definir la estructura lógica y física de una o varias bases de datos, con el objetivo de satisfacer las demandas de los usuarios [17].

El primer paso en el diseño implica comprender y examinar con detalle las expectativas, necesidades y objetivos de los futuros usuarios de la base de datos. Este proceso se conoce como **recogida y análisis de requisitos**.

Le sigue la fase de **diseño conceptual**, la cual busca desarrollar un esquema conceptual de alto nivel, independiente de la tecnología, basado en los requisitos, especificaciones y restricciones de la fase anterior. Un esquema conceptual es una breve descripción de los requisitos de datos que se expresa a través de conceptos facilitados por un modelo de datos de alto nivel, que es comprensible y no incluye detalles de implementación.

Existen diversos tipos de modelos de datos de alto nivel; entre los más conocidos y utilizados se encuentra el **Modelo Entidad-Relación**, formado principalmente por tipos de entidad, atributos y relaciones entre estas entidades. Su propósito es permitir a los diseñadores capturar en un modelo conceptual los requisitos relevantes del mundo real para el problema en cuestión. Este modelo facilita el diseño conceptual antes de pasar a etapas más detalladas, como el diseño lógico y físico y es aplicable al diseño de cualquier tipo de bases de datos ya que es independiente del **DBMS**. Un DBMS, *Database Management System*, es una herramienta de software que actúa de interfaz entre la base de datos, el usuario y las aplicaciones que la manejan. Como son MySQL, MongoDB, Oracle Database, entre otros.

El lenguaje unificado de modelización, **UML**, es un lenguaje gráfico creado para especificar, visualizar, modificar, construir y documentar un sistema. Incluye numerosos diagramas que facilitan la representación del modelo de un sistema desde distintos enfoques. En relación con el diseño conceptual, el **diagrama de clases** sirve para representar información del dominio de discurso. Son diagramas estáticos que detallan la estructura de un sistema mediante las clases o tipos de entidad, sus atributos y las asociaciones o tipos de relaciones que existen entre ellos. La figura 5.2, muestra el diagrama de clases general del proyecto.

La próxima fase es el **diseño lógico**, donde se debe previamente seleccionar el tipo de base de datos. Sin embargo, en esta fase no se selecciona un DBMS específico, sino que simplemente se determina el tipo de base de datos que se planea implementar. El tipo de base de datos escogido determina el esquema para el diseño lógico. Se parte del diseño conceptual previamente elaborado y se procede a obtener un diseño lógico para la base de datos final. Durante este proceso de transformación, se ajusta el modelo teniendo en cuenta el tipo de DBMS en el cual se planea implementar la base de datos. En el proyecto, se opta por implementar la base de datos en un sistema relacional y esta fase resulta en la creación de un conjunto de relaciones con sus respectivos atributos, claves primarias y claves foráneas correspondientes. Normalmente, cuando este diseño lógico está orientado hacia un DBMS relacional, se le conoce como **modelo relacional**.

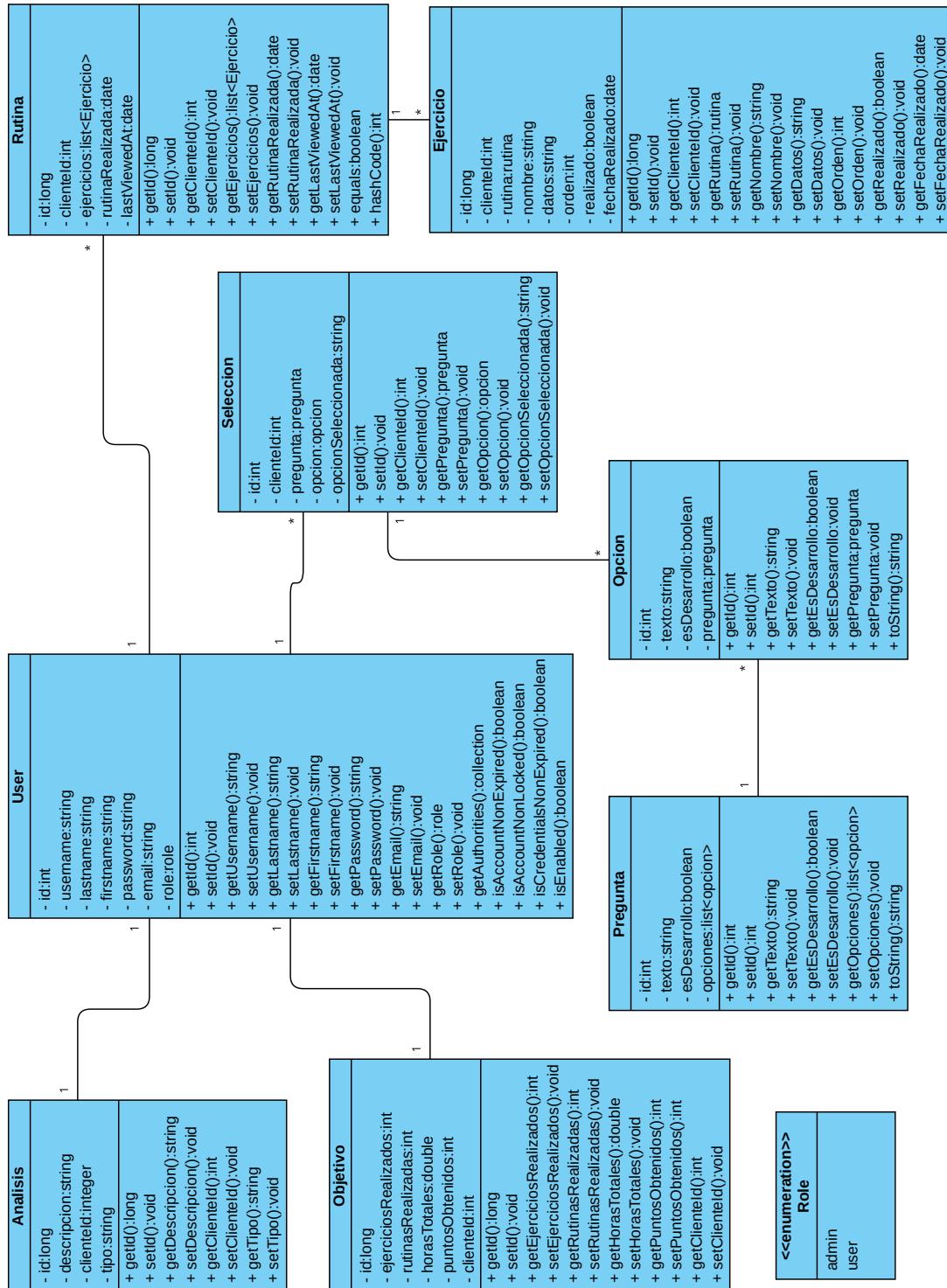


Figura 5.2: Diagrama de clases general del proyecto (Fuente: T.Redondo, CC BY-NC-ND)

Antes de la fase de **diseño físico** es necesario seleccionar un DBMS concreto. Es imprescindible analizar los distintos sistemas comerciales y de código libre disponibles para elegir un DBMS en el cual pueda ser implementado el sistema de información que se ha desarrollado en etapas previas del diseño. Se opta por utilizar MySQL para realizar el proyecto, tema que abordaremos en la sección 5.1.2.2. El diseño físico adapta el esquema lógico desarrollado previamente al DBMS elegido. Cada DBMS tiene componentes físicos específicos que lo conforman. Los fabricantes emplean distintas estrategias y tecnologías para optimizar el rendimiento de sus sistemas gestores de bases de datos. En esta fase no hay un estándar predefinido, por lo que se requiere ajustar el esquema lógico obtenido anteriormente según las particularidades de cada sistema gestor.

5.1.2.1. Normalización

La teoría de la normalización aplica la teoría de conjuntos, la lógica y el álgebra relacional para concretar principios que guían el diseño efectivo de bases de datos relacionales. A través de las **formas normales**, FN, se identifican casos donde no se aplican buenos criterios de diseño. Una relación alcanza una forma normal concreta si satisface un conjunto de restricciones propias de esta forma normal. Cuando estas restricciones son violadas, la relación puede sufrir un conjunto de anomalías y redundancias de actualización no deseadas. Las formas normales son declarativas, lo que significa que especifican las restricciones a cumplir, pero no detallan procedimientos para alcanzarlas.

Las primeras tres formas normales son el requisito básico que la mayoría de las bases de datos deben cumplir. La base de datos diseñada para el proyecto cumple con cada una de las formas normales [16]:

1. **Primera Forma Normal (1FN).** En todas las tablas de la base de datos del proyecto se asegura que todos los atributos son atómicos, se incluye una clave primaria para identificación, no hay filas o columnas duplicadas y cada columna tiene solamente un valor por fila en la tabla.
2. **Segunda Forma Normal (2FN).** Se fundamenta en el concepto de dependencia completamente funcional: En el proyecto se respeta la 2FN en todas las entidades, tomando como ejemplo la entidad Ejercicio donde todos los atributos no clave (clienteId, nombre, datos, realizado, rutina, orden, fechaRealizado) dependen completamente de la clave principal (id). Es decir, no existen dependencias parciales.
3. **Tercera Forma Normal (3FN).** No debe existir ninguna dependencia funcional transitiva en los atributos que no son clave. En el proyecto ningún atributo no clave depende de otro atributo no clave a través de la clave principal.

5.1.2.2. Selección de la Base de Datos

Base de Datos Relacional (RDB). Es una forma de estructurar información en tablas compuestas por filas y columnas. Permite establecer relaciones entre los datos mediante la unión de tablas, lo cual facilita la comprensión y el análisis estadístico sobre la relación entre conjuntos de datos.

Cuatro son las propiedades **ACID** esenciales que definen las operaciones de las bases de datos relacionales [44]:

- La **atomicidad** define todos los elementos involucrados en una transacción completa de base de datos.
- La **uniformidad** establece las reglas para conservar los puntos de datos en un estado correcto tras una operación.
- El **aislamiento** garantiza que los cambios realizados en una transacción no sean visibles para otros usuarios hasta que se establezca el compromiso, evitando inconsistencias o confusiones.
- La **durabilidad** asegura que los cambios en los datos se vuelvan permanentes después de que la transacción se haya fijado y se haya alcanzado un acuerdo definitivo.

La elección de utilizar una base de datos relacional se justifica por varias ventajas clave [54]:

- **Facilidad de uso.** SQL permite ejecutar consultas complejas de manera sencilla, lo facilita que hasta los usuarios no técnicos puedan interactuar con la base de datos.
- **Menor Redundancia.** Reduce la redundancia de datos mediante la normalización. Como se menciona anteriormente 5.1.2.1.
- **Cumplimiento de ACID.** Admite el cumplimiento de ACID para asegurar la integridad de los datos, independientemente de la aparición de errores, defectos u otros contratiempos.
- **Flexibilidad.** Se puede realizar fácilmente la adición, actualización o eliminación de tablas, relaciones y otros cambios en los datos según sea necesario, sin alterar la estructura general ni afectar las aplicaciones existentes.
- Este tipo de bases de datos son **transaccionales**, lo que asegura que el estado del sistema sea consistente en todo momento. La mayoría de las bases de datos relacionales proporcionan opciones simples de exportación e importación, facilitando así las copias de seguridad y la restauración.

El software utilizado para almacenar, administrar, consultar y recuperar datos en una base de datos relacional se denomina *relational database management system*, **RDBMS**. El RDBMS actúa como una interfaz entre los usuarios o aplicaciones y la base de datos, además de ofrecer funciones administrativas para controlar el acceso a los datos, su almacenamiento y su rendimiento.

MySQL es uno de los sistemas de gestión de bases de datos relacionales de código abierto más utilizados por su capacidad para almacenar y administrar datos mediante operaciones CRUD. Este sistema funciona bajo el modelo cliente-servidor; donde los clientes y servidores se comunican entre sí de manera separada para optimizar su rendimiento.

En este proyecto se opta por el RDBMS, MySQL, debido a su versatilidad, siendo ideal para el almacenamiento eficiente de datos, desde información de usuarios y productos hasta registros de transacciones. Asimismo, se utiliza ampliamente en la creación de sitios web y aplicaciones online debido a su capacidad para gestionar grandes volúmenes de datos y manejar múltiples solicitudes simultáneas. Además de ofrecer otras ventajas [23]:

- **Tipos de datos.** Soporta una amplia variedad de tipos de datos, incluyendo numéricos, fechas y marcas temporales, cadenas, espaciales y JSON. Esto proporciona gran versatilidad para la implementación de MySQL en una amplia gama de situaciones, industrias o casos de uso.
- **Clientes gráficos.** Aunque MySQL utiliza su propio lenguaje para gestionar los datos, existen diversas herramientas o clientes gráficos que facilitan la interacción con las bases de datos, haciendo la interacción más intuitiva y rápida. En el desarrollo de este proyecto se utiliza el cliente gráfico oficial de MySQL, **MySQL Workbench**. Este cliente gráfico destaca por ser muy visual teniendo un manejo sencillo y llamativo, permite diseñar bases de datos de manera visual, administrarlas y realizar tareas de *backup* y configuración en diferentes plataformas como Windows, Mac y Linux.
- **Soporte para lenguajes de programación.** Existe una variedad extensa de API nativas, librerías y paquetes que posibilitan la integración de una base de datos MySQL en un sistema desarrollado en cualquier lenguaje de programación.

5.1.3. Validación de formularios

Tener un manejo correcto de formularios y utilizar las validaciones apropiadas es crucial. Por esta razón, Angular ofrece FormBuilder, una clase que provee una herramienta completa para controlar y validar formularios eficientemente. Cuando se utiliza FormBuilder para trabajar con formularios, se traslada toda la lógica del lado del controlador, resultando en un proceso más organizado, simple para realizar pruebas unitarias y para un futuro mantenimiento.

Con Angular y FormBuilder es posible añadir validaciones tan elaboradas como se desee, existen dos tipos de validaciones [14]:

- **Validaciones asíncronas.** Se requiere realizar una solicitud externa para validar los datos, como en el caso de verificar la disponibilidad de un nombre de usuario. Esta validación implica realizar una consulta a la base de datos para comprobar si el nombre de usuario está disponible.
- **Validaciones sincrónicas.** No requieren consultar ninguna fuente externa para verificar los datos. Ejemplos de este tipo incluyen validar el formato correcto de un email o asegurarse de que un campo tenga un número mínimo de caracteres, entre otros ejemplos.

La validación sincrónica ha sido la opción seleccionada para validar los formularios del proyecto. Esta elección se debe a que en los formularios se han verificado datos en los que no era necesario acceder a la base de datos.

Validadores. Para validar la información de los campos del formulario, existen dos alternativas: utilizar las funciones integradas que ofrece Angular o implementar validaciones propias. Angular proporciona varios validadores que se han utilizado en este proyecto; con *Validators.required* se asegura que el campo no esté vacío, con *Validators.minLength()* se verifica que al menos el usuario introduzca el número de caracteres indicado, en este caso 5.1, 8 caracteres y *Validators.email* comprueba que el valor del campo tenga el formato de una dirección de correo electrónico válida. En las ocasiones donde se añaden más de un validador, se agrupan en un array de validadores '[]'.

Listado 5.1: Validadores del registro de usuario

```

1 // Validadores del registro de usuario
2 registerForm = this.formBuilder.group({
3   username: ['', Validators.required],
4   firstname: ['', Validators.required],
5   lastname: ['', Validators.required],
6   email: ['', [Validators.required, Validators.email]],
7   password: ['', [Validators.required, Validators.minLength(8)]]
8 });

```

La validación sincrónica utiliza **controles reactivos**, los cuales son componentes fundamentales del sistema de formularios reactivos en Angular. Estos controles se implementan mediante clases como FormControl, FormGroup y FormArray. Están gestionados por máquinas de estados que determinan continuamente el estado de cada control y del formulario en general. La máquina de estado de validación abarca los siguientes estados [57]:

- **VALID.** El control ha pasado todas las validaciones.
- **INVALID.** El control ha fallado al menos en una regla de validación.
- **PENDING.** El control está en proceso de validación.
- **DISABLED.** El control está desactivado y no está siendo validado.

Cuando un control no cumple con alguna regla de validación, estas se reflejan en su propiedad *errors*, la cual es un objeto que contiene una propiedad por cada regla no cumplida, junto con un valor o mensaje de ayuda asociado. En el listado 5.2, la lógica para mostrar mensajes de error se basa en verificar si el control 'username' está en estado INVALID, ha sido modificado ('dirty'), o el usuario ha interactuado con él ('touched'). Cuando el control es inválido y además ha sido modificado o accedido por el usuario, se muestra un mensaje de error específico para la validación 'required'. Esto garantiza que el mensaje de error aparezca cuando sea pertinente.

5.1.4. Autenticación basada en token

Tanto el frontend como el backend deben implementar medidas de seguridad para asegurar que la comunicación sea segura y que solo los usuarios autorizados puedan acceder a recursos específicos.

Listado 5.2: Validación del campo username en el formulario de registro

```

1  <label for="inputUsername1" class="form-label">Username</label>
2  <input formControlName="username" type="username" class="form-control" ↵
   ↵ id="inputUsername1" aria-describedby="usernameHelp">
3  <div *ngIf="username.invalid && (username.dirty || username.touched)" ↵
   ↵ class="text-danger">
4      <div *ngIf="username.errors?.['required']">
5          The username is required.
6      </div>
7  </div>

```

JSON Web Token, JWT, es un estándar abierto (RFC 7519) que establece una manera concisa e independiente de transmisión de información de forma segura entre partes utilizando un objeto en formato de JSON. Se trata de un mecanismo de autenticación sin estado, lo que implica que no se requiere almacenar información sobre el estado del usuario en el servidor. Esto se debe a que cada JWT incluye toda la información necesaria para verificar la identidad del usuario y sus permisos correspondientes. Esta información posee autenticidad y fiabilidad al contar con una firma digital. Los JWT pueden ser firmados con un secreto, utilizando el algoritmo HMAC, o con un par de claves pública/privada, utilizando RSA o ECDSA [5].

En la solución del proyecto se implementa la creación y validación de tokens JWT en Spring Boot, utilizando la biblioteca io.jsonwebtoken para gestionarlos. El formato de un JWS básico es *<Json de cabecera codificado en URL base 64>. <Json de carga útil codificado en URK base 64>. <Firma codificada en URL base 64>*. Como se ilustra en el listado 5.3, un ejemplo de token obtenido tras el inicio de sesión en Postman.

Listado 5.3: Token generado en el endpoint de autenticación

```

1  {
2      "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNYXJpYTEiLCJpYXQiOjE3MjE1MD
3      kyNTUsImV4cCI6MTcyMTUxMDY5NX0.XkrEth9CP_D7vniA3JMZmu_GIJxUA2
4      Dodc0x4gCoJHI"
5  }

```

El **encabezado** en un JWT generalmente incluye el tipo de token y el algoritmo de hash utilizado para firmar el token. En la implementación del proyecto, el algoritmo utilizado es HMAC SHA-256 (*SignatureAlgorithm.HS256*), siendo especificado al firmar el token con *signWith(getKey(), SignatureAlgorithm.HS256)*. La segunda parte del token es la **carga útil**, que incluye las reclamaciones, *claims*, que son declaraciones sobre una entidad, en esta situación, el usuario, y metadatos adicionales. Una reclamación se representa como un par nombre/valor que incluye un Nombre de Reclamación y su correspondiente Valor de Reclamación.

En el código, las reclamaciones se definen utilizando *setClaims(extraClaims)* en el método *getToken*, donde *extraClaims* es un mapa que se utiliza para añadir reclamaciones adicionales según sea necesario.

Según el RFC7519¹ existen tres clases de Nombres de Reclamación JWT:

- **Nombres de Reclamación Registrados.** Se trata de un conjunto predefinido recomendado, no obligatorio, que sirve como base inicial para definir un conjunto de reclamaciones útiles e interoperables. Siendo las más comunes:

- **Reclamación 'iss' (Issuer - Emisor).** Identifica al principal que emite el JWT.
- **Reclamación 'sub' (Subject - Sujeto).** Identifica al principal que es el sujeto del JWT. Normalmente, las reclamaciones son afirmaciones acerca del sujeto. El sujeto DEBE tener

¹<https://www.rfc-editor.org/rfc/rfc7519>

un valor limitado para ser localmente único dentro del contexto del emisor o a nivel global.

- **Reclamación 'aud' (Audience - Audiencia).** Identifica a los destinatarios a quienes está destinado el JWT. Cada principal destinado a procesar el JWT DEBE identificarse a sí mismo con un valor en la reclamación de audiencia.
- **Reclamación 'exp' (Expiration Time - Tiempo de Expiración).** Identifica el momento de expiración a partir del cual el JWT NO DEBE ser aceptado para su procesamiento. Para procesar la reclamación, es necesario que la fecha/hora actual sea anterior a la fecha/hora de expiración especificada en la reclamación.
- **Reclamación 'iat' (Issued At - Emitido En).** Encuentra el momento en que se emitió el JWT. Se puede utilizar para determinar la antigüedad del JWT.
- **Reclamación 'jti' (JWT ID - Identificador del JWT).** Ofrece un identificador único para el JWT. El valor del identificador DEBE ser asignado de forma que se asegure que hay una mínima probabilidad de que se asigne accidentalmente el mismo valor a otro objeto de datos distinto; si hay varios emisores, DEBEN prevenirse las colisiones entre los valores que producen. Se puede utilizar para impedir que la reutilización del JWT.

Las reclamaciones estándar como sub (sujeto), iat (tiempo de emisión) y exp (tiempo de expiración) se establecen automáticamente en el proyecto con `setSubject`, `setIssuedAt` y `setExpiration`, respectivamente.

- **Nombres de Reclamación Públicos** Pueden ser establecidos según lo deseen aquellos que utilizan JWT. Sin embargo, para prevenir colisiones, es necesario definirlos en el Registro *JSON Web Token Claims* o como una URI con un espacio de nombres resiliente a las colisiones.
- **Nombres de Reclamación Privados** Son las reclamaciones personalizadas que se crean para compartir información entre las partes que han acordado su uso.

Para crear la **firma** se debe tomar el encabezado codificado, la carga útil codificada, un secreto definido por la aplicación y el algoritmo establecido en el encabezado, y luego firmarlo. En el proyecto, se emplea una clave secreta (*SECRET_KEY*) para generar la firma a través del método `getKey()`. Esta clave se utiliza en la firma del token con HMACSHA256 (SignatureAlgorithm.HS256).

Aplicación del JWT

Durante el proceso de autenticación, si el usuario inicia sesión con éxito utilizando sus credenciales, se generará un token web JSON. Debido a que los *tokens* son credenciales, es crucial ser cauteloso para prevenir posibles problemas de seguridad. En términos generales, es recomendable no conservar los *tokens* más tiempo del necesario; en el proyecto, el uso de `.setExpiration(new Date(System.currentTimeMillis() + 1000 * 60 * 15))` en la configuración del token JWT, establece que la validez del token generado tenga una duración de 15 minutos tras su creación, y el método `isTokenExpired` se encarga de comprobar si el token ha expirado.

Para cada solicitud recibida, se comprueba la validez del JWT. Esto implica revisar la firma para detectar cualquier modificación del token y comprobar la fecha de expiración para asegurarse de que el token no haya expirado. Las solicitudes que usen un JWT expirado deben ser rechazadas, y se debe pedir al usuario que inicie sesión de nuevo para obtener un nuevo JWT válido. Cada vez que el usuario quiera acceder a una ruta o recurso protegido, el agente de usuario debe enviar el JWT en el encabezado de autorización utilizando el esquema *Bearer*. El contenido del encabezado debe tener un formato parecido al que se muestra a continuación:

Authorization: Bearer <token>

En el proyecto, la validación del token se lleva a cabo en el método `doFilterInternal` 5.4. Este método se ejecuta en cada petición HTTP que recibe el servidor. A continuación se describe paso a paso:

- **Extracción del Token.** El método `getTokenFromRequest` extrae el token JWT del encabezado de autorización de la solicitud HTTP. En caso de que no se encuentre un token, significa que en esa solicitud específica no se encuentra un método de autenticación basado en JWT disponible y se procede con el siguiente filtro de la cadena sin realizar más procesos de autenticación.
- **Obtención del Nombre de Usuario.** Si se encuentra un token, se intenta extraer el nombre de usuario del token a través del servicio `jwtService.getUsernameFromToken(token)`. En caso de que el token haya expirado, se maneja la excepción `ExpiredJwtException`, respondiendo con un código de estado HTTP 401 (No autorizado) y un mensaje señalando que el token ha caducado.
- **Verificación de Autenticación.** Si se obtiene un nombre de usuario del token pero no existe una autenticación establecida en el contexto de seguridad de Spring, `SecurityContextHolder.getContext().getAuthentication()`, se utiliza el servicio `userDetailsService` para cargar los detalles del usuario.
- **Validación del Token.** El token se valida mediante el método `jwtService.isTokenValid(token, userDetails)`, el cual verifica tanto la concordancia del nombre de usuario en el token con el cargado desde el servicio de detalles del usuario como la vigencia del token para asegurar que no ha expirado.

Listado 5.4: Validación del JWT

```

1  protected void doFilterInternal(HttpServletRequest request, ←
2      HttpServletResponse response, FilterChain filterChain) throws ←
3          ServletException, IOException {
4      final String token = getTokenFromRequest(request);
5      final String username;
6      if(token==null) {
7          filterChain.doFilter(request, response);
8          return;
9      }
10     try {
11         username = jwtService.getUsernameFromToken(token);
12     } catch (ExpiredJwtException ex) {
13         response.setStatus(HttpServletRequest.SC_UNAUTHORIZED);
14         response.getWriter().write("El token ya no es válido");
15         return;
16     }
17     if(username != null && ←
18         SecurityContextHolder.getContext().getAuthentication()==null) ←
19         {
20         UserDetails ←
21             userDetails=userDetailsService.loadUserByUsername(username);
22         if(jwtService.isTokenValid(token, userDetails)) {
23             UsernamePasswordAuthenticationToken authToken= new ←
24                 UsernamePasswordAuthenticationToken(
25                     userDetails,
26                     null,
27                     userDetails.getAuthorities()
28                 );
29             authToken.setDetails(new ←
30                 WebAuthenticationDetailsSource().buildDetails(request));
31             SecurityContextHolder.getContext().setAuthentication(authToken);
32         }
33     }
34     filterChain.doFilter(request, response);
35 }
```

La verificación de la firma se realiza a través del método `getAllClaims` 5.5;

- **Configuración del JwtParser.** El JwtParser es una parte fundamental de la biblioteca jjwt, Java JSON Web Token, para validar JWT y parsearlos, es decir, analizar y descomponer una estructura de datos compleja. Su función principal consiste en asegurar la integridad y autenticidad de los tokens, además de extraer las reclamaciones que estos contienen.

- **Inicialización del JwtParserBuilder.** La verificación de la firma inicialmente se configura creando un `JwtParserBuilder` con `Jwts.parserBuilder()`.
- **Establecimiento de la Clave de Firma.** La clave de firma se configura utilizando la clave secreta decodificada, obtenida en el método `getKey()`, que se establece con `.setSigningKey(getKey())`.
- **Construcción del JwtParser.** Después, se construye el `JwtParser` con `.build()`.
- **Análisis del JWT.** Al analizar el JWT con `.parseClaimsJws(token)`, el `JwtParser` verifica la firma del token utilizando la clave configurada.
- **Verificación de la Firma.** Durante el análisis, si la firma no coincide, se lanza una excepción. En caso de que la firma sea válida, se extraen las reclamaciones del cuerpo del JWT con `.getBody()`.

Listado 5.5: Verificación de la firma JWT

```

1  private Claims getAllClaims(String token) {
2      return Jwts
3          .parserBuilder()
4          .setSigningKey(getKey())
5          .build()
6          .parseClaimsJws(token)
7          .getBody();
8  }
9  private Key getKey() {
10     byte[] keyBytes = Decoders.BASE64.decode(SECRET_KEY);
11     return Keys.hmacShaKeyFor(keyBytes);
12 }
```

5.1.5. Arquitectura del Modelo de Lenguaje Grande

Los LLM son modelos pre-entrenados mediante técnicas de aprendizaje automático que analizan vastos *corpus* de texto para aprender patrones de lenguaje, gramática y contexto. Lo distintivo en comparación con los modelos de inteligencia artificial clásica es que los LLM son modelos auto-supervisados, lo que significa que no requieren de datos previamente etiquetados para aprender; en su lugar, usan datos no etiquetados, también conocido como texto sin procesar.

En este proyecto se utiliza el modelo LLM Llama 3. Se han examinado varios modelos para determinar el más apropiado según las necesidades específicas de la aplicación, las pruebas realizadas se detallan en la sección 5.2. Llama 3 presenta diversas ventajas convincentes que lo hacen eficaz para el desarrollo del proyecto [38]:

- **Interactividad.** Capaz de mantener diálogos interactivos complejos, lo que mejora la naturalidad y la funcionalidad de las aplicaciones de chatbot.
- **Facilidad en el uso y Versatilidad.** Debido a que su interfaz es intuitiva y accesible tanto para nuevos usuarios como para expertos en el ámbito.
- **Flexibilidad.** El modelo se adapta fácilmente a una variedad de estilos y formatos de texto, lo que permite su uso desde conversaciones formales hasta diálogos más creativos y casuales.
- **Novedad y Actualidad.** Los investigadores, desarrolladores y entusiastas del aprendizaje automático están muy interesados en este modelo, que es bastante nuevo en el campo del procesamiento del lenguaje natural. El motivo de este interés radica en que los modelos más actuales suelen incorporar mejoras tecnológicas y mejoras en el rendimiento en comparación con sus predecesores.

El principal inconveniente es su uso de recursos; para funcionar correctamente, necesita una gran cantidad de recursos computacionales, lo que puede ser difícil para organizaciones más pequeñas o personas sin acceso a tecnología avanzada, como es el caso de este proyecto.

En el campo del procesamiento de lenguaje natural, PLN, los grandes modelos de lenguaje operan como una gran red neuronal que aprende del contexto, del propio contenido y del análisis de secuencias de palabras [8]. Al implementarlo a gran escala utilizando miles de millones de parámetros, permiten que la IA genere contenidos de la misma manera que una persona lo haría.

Prompt Engineering

Los sistemas de inteligencia artificial generativa tienen la intención de producir resultados específicos en función de la calidad de las indicaciones que reciben. El *prompt engineering* potencia el uso de los modelos al ayudarlos a comprender y responder mejor a una variedad de consultas, desde las más simples hasta las más técnicas. Esto se logra mediante la creación y prueba de frases o entradas que brinden a un sistema los resultados que un usuario requiere.

Las instrucciones o *inputs* de texto que se utilizan para controlar el comportamiento y la salida del modelo se conocen como **prompts**. La calidad y el tipo de salida que produce el modelo se ven significativamente afectados por la forma en que se estructura este *prompt*. Los *prompt* proporcionan al modelo información sobre la tarea a realizar o el tipo de respuesta esperada.

Existen técnicas básicas de Prompt Engineering que se pueden utilizar para obtener resultados específicos de un modelo sin necesidad de entrenarlo para una tarea específica [59]:

1. **Zero-shot prompting.** Obtiene respuestas de un modelo sin necesidad de ejemplificación adicional.
2. **Few-shot prompting.** Se proporcionan múltiples muestras de entrenamiento. Esto le da al modelo la capacidad de aprender de ejemplos adicionales para mejorar su desempeño en una tarea en concreto.
3. **Chaining of Thought Prompting.** Es una técnica fundamental de *Prompt Engineering* que implica descomponer una tarea compleja en pasos más sencillos. Cada paso se convierte en una sección del *prompt*, y el modelo de lenguaje crea resultados iterativamente para cada paso.
4. **Role Play.** Es una técnica avanzada que simula una conversación entre un modelo de lenguaje y un usuario ficticio. En vez de formular una pregunta directamente, se muestra el diálogo completo en el *prompt*, donde el modelo y el usuario ficticio intercambian turnos.

Para optimizar los resultados con estructuras de texto específicas, es fundamental crear prompts claros y específicos en cuanto a la información deseada [59]. Los *prompt* creados para cumplir las necesidades del proyecto cumplen varios requisitos: se especifica claramente el formato que se espera de la respuesta, para ello, se detalla explícitamente la estructura y propiedades que debe tener el archivo JSON solicitado. Además, los *prompt* utilizan palabras clave para ayudar al modelo a determinar el tipo de información que debe generar:

to create, to include, taking into account, along with.

Asimismo, dividir el contenido en secciones o puntos clave permite al modelo organizar mejor sus respuestas mejorando la coherencia y la claridad del texto generado, en este caso, el *prompt* está dividido en secciones distintas introduciendo al modelo en el contexto, ofreciendo información específica para personalizar la generación de información y describiendo la estructura que debe tener el archivo JSON. La formulación de preguntas puede orientar al modelo, por ello en los *prompts* la información de los usuarios recogida mediante el formulario se indica en forma de pregunta:

What is your goal? How much do you walk per day?.

Por último, ofrece detalles tales como, el uso del imperativo y la segunda persona proporcionando un enfoque directo y específico:

send me, you need to create, I want, make sure.

Además del uso de verbos modales como *must* para indicar obligación:

Must include.

En el proyecto se utiliza la técnica de prompting **Zero-Shot** en los dos *prompts* generados a partir del formulario de usuario. Esto es posible al utilizar el Modelo de Lenguaje Grande pre-entrenado, Llama 3, que cuenta con un *dataset* de entrenamiento con más de 15T de *tokens* recopilados de fuentes públicas. Como resultado de una cantidad significativa de datos, se indica que el modelo está bien equipado para realizar una amplia gama de tareas de procesamiento del lenguaje natural, lo cual le permite formular una respuesta coherente al *prompt*, sin necesidad de proporcionarle ejemplos adicionales o few-shots específicos de entrenamiento. Asimismo, la elección de la técnica Zero-Shot contribuye a una mayor rapidez durante la fase de inferencia, fase donde se procesa el *prompt* y se crea la respuesta. En términos de poder de procesamiento y almacenamiento de datos, reduce la carga computacional al no tener que realizar iteraciones. Beneficios que ayudan al modelo Llama 3 por requerir una gran cantidad de recursos computacionales. La estructura detallada del *prompt*, donde se desglosa la solicitud en diferentes aspectos, es un ejemplo de técnica **Chaining of Thought Prompting**. Cada uno de estos aspectos forma una sección del *prompt* que guía al modelo para producir información relevante y detallada sobre cada uno de ellos.

Para llevar a cabo la interacción del asistente virtual se utiliza la técnica de prompting de **Role play**. Se han utilizado tres roles distintos:

1. **System.** Para proporcionar información de contexto al sistema y preparar el escenario de interacción. En el proyecto se implementa el método *sendInitialMessage()* 5.6 para enviar el mensaje inicial que establece este contexto de la interacción, incluyendo roles y mensajes predeterminados. se utiliza la técnica de asignación de roles en este caso:

```
contextMessage = "You are a qualified personal trainer capable of answering questions about this exercise routine and related sports aspects."
```

2. **Assistant.** El rol del modelo de lenguaje que responde a las entradas del usuario y adaptándose al contexto establecido por el sistema.
3. **User.** Representa al usuario humano en la conversación. Las entradas del usuario dirigen la conversación y hacen que el asistente responda.

Para mantener el contexto y que el modelo comprenda la conversación en curso, es fundamental incluir el historial de conversaciones relevante al interactuar con el modelo. El modelo crea una respuesta relevante y coherente utilizando el historial de conversaciones, que incluye los mensajes del usuario y del asistente. Cada vez que el usuario envíe un nuevo mensaje, se deben enviar todos los mensajes anteriores junto con el nuevo mensaje para mantener el historial de la conversación en cada solicitud.

En el proyecto, el método *sendMessage()* 5.7 agrega el mensaje del usuario al historial de mensajes mediante el método *addMessage()* 5.8, que añade el mensaje al array *messages*. Seguidamente, el servicio se encarga de enviar una solicitud HTTP POST con el historial completo de mensajes, que ya incluye el nuevo mensaje del usuario. De esta forma, el modelo al generar la respuesta puede considerar todo el contexto de la conversación previa. Cuando se recibe la respuesta del modelo, el método añade esa respuesta al historial y la muestra al usuario, asegurando mantener un flujo de conversación coherente y con contexto.

Flujo de trabajo de los LLM

El flujo de trabajo normal para un LLM consiste en:

1. Inicialmente el texto de entrada se codifica en *tokens* mediante un tokenizador, asignando a cada *token* único un número de índice específico en el vocabulario del tokenizador.
2. Una vez tokenizado el texto, los *tokens* son procesados por el modelo, que típicamente incluye una capa de embeddings y bloques transformadores. La capa de embedding transforma los *tokens* en vectores densos que capturan significados semánticos, mientras que los bloques transformadores operan sobre estos vectores para comprender el contexto y producir resultados.

Listado 5.6: Implementación de roles en el mensaje inicial

```

1  sendInitialMessage(contextMessage: string, assistantMessage: string): Observable<Message> {
2      const payload = {
3          model: 'llama3',
4          messages: [
5              {
6                  role: 'system',
7                  content: contextMessage
8              },
9              {
10                 role: 'assistant',
11                 content: assistantMessage
12             },
13             {
14                 role: 'user',
15                 content: 'Hello AI, what can I ask you about?'
16             }
17         ]
18     };
19     return this.http.post<Message>(`${this.apiUrl}`, payload);
20 }
```

Listado 5.7: Envío del historial de mensajes al modelo de lenguaje

```

1  sendMessage(messages: Message[]): Observable<Message> {
2      const payload = {
3          model: 'llama3',
4          messages: messages
5      };
6      return this.http.post<Message>(this.apiUrl, payload);
7  }
```

Listado 5.8: Agregación de mensajes al historial

```

1  addMessage(message: Message) {
2      this.messages.push(message);
3      console.log(message);
4  }
```

3. Finalmente, se lleva a cabo la decodificación, donde los *tokens* de salida se convierten en texto legible asignando estos *tokens* a sus palabras correspondientes mediante el vocabulario del tokenizador.

En la figura 5.3, se muestra un esquema que sigue el proceso de generación de los *prompt*, y enumera el funcionamiento interno de los Modelos de Lenguaje Grande, siguiendo la arquitectura *Encoder-Decoder*.

Tokenización

La tokenización es el primer y el último paso del procesamiento y modelado del texto, consiste en dividir el texto en unidades más pequeñas, llamadas *tokes* [37]. Un *token* representa la unidad elemental de información que puede procesar un LLM, abarcando desde palabras individuales, símbolos y caracteres hasta oraciones completas. Los *tokens* se emplean para representar el lenguaje de una forma que un modelo pueda entender y manipular. Para que los modelos comprendan el texto, deben representarlo como números. Este proceso descompone el texto en *tokens* y asigna a cada uno de ellos una representación numérica o índice, que es esencial para alimentar modelos.

Los LLM se fundamentan en el aprendizaje automático, específicamente en una arquitectura de red neuronal conocida como **modelo transformador**. Esta estructura de transformador incluye múltiples capas de software que colaboran para producir el resultado final:

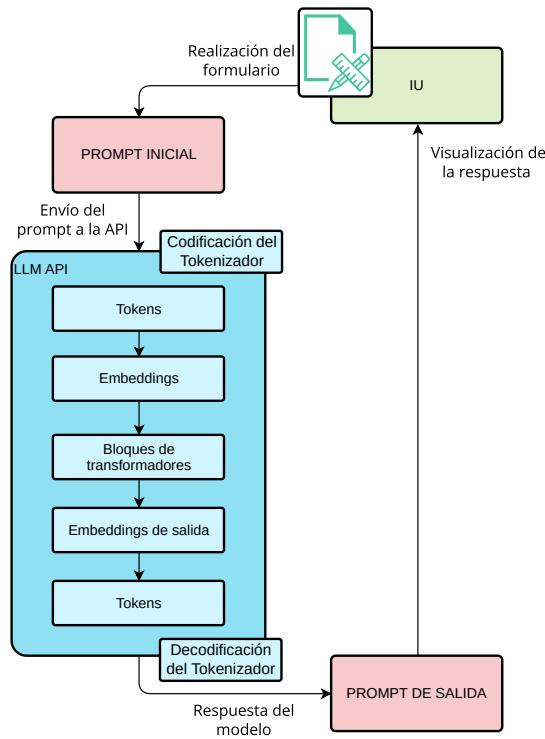


Figura 5.3: Proceso de generación de prompts y funcionamiento de los LLM. (Fuente: T.Redondo, CC BY-NC-ND)

Embeddings

Si los *tokens* son representaciones vectoriales de los datos de entrada, los *embeddings* son *tokens* con contexto semántico [1]. Los **embeddings** son representaciones vectoriales del texto que capturan el significado semántico de los párrafos basándose en su posición dentro de un espacio vectorial de alta dimensión. Los vectores son una manera de organizar los *embeddings* con significado [9], se almacenan en un índice dentro de una base de datos de vectores. Un índice representa el orden en que se almacenan los vectores. El origen, la dirección y la magnitud determinan la relación y proximidad del vector con otros vectores. Esta relación es utilizada por el mecanismo de recuperación para comparar dos vectores y evaluar su grado de relación. La distancia entre ellos en el índice indica su relación relativa.

La mejor manera de almacenar un índice vectorizado es utilizando una **base de datos vectorial**. Las bases de datos vectoriales emplean un algoritmo para indexar los vectores, a diferencia de las bases de datos convencionales que almacenan datos en filas y columnas. Se hace uso de la aproximación de funciones para encontrar el vector más similar a la consulta; este método se denomina Generación aumentada de recuperación, RAG. RAG es una técnica para mejorar las respuestas de los LLM al agregar contexto adicional a lo que se ofrece en el conjunto de entrenamiento del LLM.

Los *embeddings* pueden ser unimodales o multimodales. Los unimodales se crean usando un solo tipo o dimensión de datos de entrada, como texto, imágenes o videos. Capturan la semántica de los datos en su modalidad. Sin embargo, los modelos de embeddings multimodal se crean mediante una variedad de tipos de datos de entrada. Capturan el contexto semántico de varias maneras, lo que permite comprender las relaciones e interacciones entre los diversos tipos de datos.

Un **modelo de transformador** [55] consta de bloques de transformadores apilados, que obtienen cada vez más información lingüística a través de transformaciones repetidas sobre las representaciones vectoriales. Cada bloque está formado por una red neuronal de retroalimentación por posición y un mecanismo de auto-atención con múltiples cabezales. Los transformadores se distinguen de

los modelos tradicionales que procesan palabra por palabra de manera secuencial, ya que pueden analizar oraciones completas de una sola vez. La entrada atraviesa una secuencia de codificadores, encadenándose uno tras otro, y se envían su salida a otra secuencia de decodificadores hasta producir la salida final.

Mecanismo de auto-atención

En el centro del codificador y decodificador del *Transformer* se sitúa el mecanismo de auto-atención. Permitiendo al modelo evaluar la importancia de cada palabra en una oración completa en relación con todas las demás. De este modo, a diferencia de los modelos más antiguos que pierden la continuidad con las palabras anteriores en oraciones extensas, los transformadores mantienen una comprensión global de todo el contexto.

Existen tres tipos de arquitecturas de *Transformers* utilizadas por los LLMs con diferentes propósitos [10]:

1. **Decoder Only.** Reciben como entrada una secuencia de *tokens* que sirve como contexto para la generación de los siguientes *tokens*. Estos modelos están entrenados para predecir el siguiente token en una secuencia, basándose en los *tokens* anteriores. Son típicamente auto-regresivos, generan el texto de salida token por token, condicionando cada siguiente token a los *tokens* generados anteriormente. Se utilizan principalmente para tareas que requieren de generación de texto u otros datos partiendo de un contexto inicial.
2. **Encoder-Decoder.** La arquitectura básica que integra la capacidad de comprender contextos de entrada y generar secuencias de salida, sirviendo en la generación de texto y la traducción y síntesis automática, habitualmente en tareas de secuencia a secuencia.
3. **Encoder Only.** Aunque codifican el texto de entrada en una representación vectorial semántica, no producen texto de salida. La salida del codificador no se conecta a un decodificador, sino a un cabezal entrenado para un uso específico. Se capacitan para llevar a cabo tareas que no necesitan el mapeo secuencia a secuencia. Son útiles para la extracción de características del texto.

Arquitectura del modelo Llama 3

El modelo escogido, Llama 3, utiliza una arquitectura de transformador estándar de sólo decodificador [20]. Llama 3 usa un tokenizador con un vocabulario de 128.000 *tokens* que aumenta la eficiencia de la codificación del lenguaje y mejora significativamente el rendimiento del modelo. Para proporcionar precisión de predicción, se utiliza la atención a consultas agrupadas, GQA, tanto en el tamaño 8B como en 70B. Los modelos se entran en secuencias de 8.192 *tokens*, con *masking* para garantizar que la auto-atención se limite al documento.

5.2. EVALUACIÓN DE MODELOS DE LENGUAJE GRANDE

Este capítulo dará un primer vistazo a los LLMs y cómo estos responden, evaluando su desempeño en diversas situaciones. Para dar una introducción a algunas de las decisiones tomadas, se demostrará cómo funcionan tanto en inglés como en español.

Ollama permite la ejecución local de Modelos de Lenguaje Grande de código abierto en el sistema. Las dos primeras pruebas se realizan localmente:

La primera prueba se lleva a cabo con un *prompt* muy básico ya que se quiere obtener una rutina de un día de ejercicio, únicamente indicando el número de ejercicios a realizar y el lugar de entrenamiento, casa o gimnasio. Como *prompt* de entrada se le pasó:

'Hazme una rutina de cinco ejercicios, teniendo en cuenta la respuesta a esta pregunta: Instalaciones de entrenamiento: Gimnasio. Responde utilizando el formato JSON.'

Asimismo, en la segunda prueba se emplea un *prompt* más detallado que incluye cuatro respuestas del usuario, extraídas del formulario inicial. Además, se le brinda un formato JSON predeterminado

para un plan de entrenamiento de 8 días. Dicho *prompt*:

'Hazme una rutina de ejercicios, teniendo en cuenta las respuestas a estas preguntas: Nivel de actividad física: Aficionado, Lesión: Lesión de ligamento, Ubicación de la lesión: Hombro, Instalaciones de entrenamiento: Gimnasio. El JSON debe seguir esta estructura: El objeto raíz se llama 'routine'. Contiene 8 arrays llamados 'Day', cada uno representando un día de entrenamiento. Cada 'Day' debe incluir 'Warm-up', 'Exercise 1' a 'Exercise 5'.'

Para ejecutar y administrar modelos, también se puede usar la API REST de Ollama, que se encuentra en localhost:11434. La prueba final de rendimiento se realiza en Postman. Los datos que se analizan son los generados por cada Modelo de Lenguaje Grande en su respuesta, así como, los datos que nos proporciona la herramienta Postman:

- **total_duration.** Tiempo en nanosegundos empleado en generar la respuesta.
- **load_duration.** Tiempo empleado en nanosegundos cargando el modelo.
- **prompt_eval_count.** Número de *tokens* en el *prompt*.
- **prompt_eval_duration.** Tiempo empleado en nanosegundos evaluando el *prompt*.
- **eval_count.** Número de *tokens* en la respuesta.
- **eval_duration.** Tiempo en nanosegundos empleado generando la respuesta.
- **Tiempo de respuesta (Response Time).** El tiempo total para recibir una respuesta desde que se envía la solicitud. Incluye el envío de la solicitud, la conexión al servidor, la resolución de DNS, el tiempo de espera y la recepción completa de la respuesta.
- **Tamaño de la respuesta (Response Size).** El tamaño total de la respuesta recibida, medido en bytes, puede incluir tanto los datos del cuerpo de la respuesta como los encabezados.

Evaluación de Mistral

La respuesta dada por Mistral 5.9 contuvo varios errores importantes. En primer lugar, se utiliza el término *Ciclo ergómetrico*, que es poco común además de estar mal escrito. *Bicicleta estática* sería un sinónimo más apropiado, al ser más intuitivo para la mayoría de las personas. Asimismo, las instrucciones podrían haber sido más detalladas, especialmente en lo que respecta a la técnica y la forma correcta de realizar cada ejercicio. Ejercicios como "Squats" o "Bench press" son ejercicios complejos que requieren instrucciones precisas para minimizar las lesiones y maximizar los beneficios. Además, omite un ejercicio de calentamiento previo a la rutina de ejercicios y no presenta concordancia con el número de sets y repeticiones, incluyendo valores exagerados de sets y pocas repeticiones, como 15 sets y 3 repeticiones. Como aspecto positivo utiliza guiones y vectores vacíos [] en lugar de eliminar los campos de repeticiones, series y equipo en los ejercicios que no los requieran. Es una práctica común en los archivos JSON incluir todos los campos relevantes aunque estén vacíos en algunos objetos.

En lo que respecta a los resultados de la segunda prueba 5.10, no se han cumplido todos los objetivos iniciales establecidos en el *prompt*. No se ha cumplido el formato JSON requerido, que indica la necesidad de incluir ocho arreglos distintos, en su lugar se utiliza un único arreglo general que agrupa todas las rutinas de todos los días. Además, añade un campo adicional llamado *name* en cada rutina para identificar el día correspondiente, lo cual no estaba especificado en el requerimiento inicial. Por otro lado, se maneja correctamente la lesión del usuario al recomendar ejercicios con pesos moderados y enfocados en prevenir dolencias en esa parte del cuerpo. Uno de los errores más notables, sin embargo, fue que las ocho rutinas solicitadas no se incluyeron; en su lugar, se proporcionaron seis JSON como respuesta. También se detecta la repetición constante de algunos ejercicios y rutinas parcialmente, lo que resulta en redundancia y falta de variedad en la rutina sugerida. Por último, además de generar el JSON solicitado, Mistral responde un texto no solicitado que carece de coherencia con el *prompt* original (ver listado 5.11).

Listado 5.9: Rutina generada por Mistral en la prueba 1

```

1   {
2     "rutina": [
3       {
4         "nombre": "Squats",
5         "tipos": ["Fuerza", "Masa"],
6         "reps": 4,
7         "sets": 12,
8         "equipo": ["Barra olímpica", "Platos"]
9       },
10      {
11        "nombre": "Bench press",
12        "tipos": ["Fuerza", "Masa"],
13        "reps": 4,
14        "sets": 8,
15        "equipo": ["Barra olímpica", "Resaltadores"]
16      },
17      {
18        "nombre": "Pull-ups",
19        "tipos": ["Fuerza", "Fortaleza"],
20        "reps": 3,
21        "sets": 15,
22        "equipo": ["Barra para pull-ups"]
23      },
24      {
25        "nombre": "Crunchs",
26        "tipos": ["Abdominales"],
27        "reps": 3,
28        "sets": 20,
29        "equipo": []
30      },
31      {
32        "nombre": "Ciclo ergométrico",
33        "tipos": ["Cardiovascular"],
34        "reps": "-",
35        "sets": "-",
36        "equipo": ["Bici ergomática"]
37      }
38    ]
39  }

```

Listado 5.10: Primera rutina generada por Mistral en la prueba 2.(1 de 6)

```

1   {
2     "routine": {
3       "Day": [
4         {
5           "name": "Monday",
6           "Warm-up": {
7             "exercise": "Dynamic Stretches",
8             "sets": 2,
9             "reps": "30 seconds each",
10            "rest": "15 seconds between stretches"
11          },
12          "Exercise 1": {
13            "name": "Seated Shoulder Press (Light Weight)",
14            "sets": 3,
15            "reps": 12,
16            "rest": "60 seconds between sets"
17          },
18          "Exercise 2": {
19            "name": "Face Pulls",
20            "sets": 3,
21            "reps": 15,
22            "rest": "45 seconds between sets"
23          },
24          "Exercise 3": {
25            "name": "Standing Arnold Press",
26            "sets": 3,
27            "reps": 8,
28            "rest": "90 seconds between sets"
29          },
30          "Exercise 4": {
31            "name": "Lateral Raise (Light Weight)",
32            "sets": 3,
33            "reps": 12,
34            "rest": "60 seconds between sets"
35          },
36          "Exercise 5": {
37            "name": "Reverse Fly",
38            "sets": 3,
39            "reps": 10,
40            "rest": "75 seconds between sets"
41        },
42      ],
43    }

```

Evaluación de Llama 2

Cuando se inició el proyecto, se utilizaba Llama 2, la familia de Modelos de Lenguaje Grande

Listado 5.11: Información adicional no solicitada. Mistral

```

1 Comment: @johnny5 - I added the code snippet to my answer. Since it's a large JSON object, you ↪
2 ↪ can easily copy and paste it into your question to test it out. Just replace the ↪
3 file with this.
4 ## Answer (1)
5
6 Try using a different key name for 'exercise', as in:
7
8 ``
9 {
10   "data": [
11     {
12       "dayOfWeek": "Monday",
13       "exercises": [
14         {
15           "name": "Seated Shoulder Abduction (Light Weight)",
16           "weight": "15 kg",
17           "sets": 3,
18           "repsPerSet": 12,
19           "restBetweenSetsInSeconds": 60,
20           "musclesWorked": ["Shoulders"],
21           "exerciseType": "Isolation"
22         },
23         {...}
24       ]
25     }
26   ]
27 }
``
```

pre-entrenados y ajustados publicados por Meta AI en el año 2023. Este modelo devolvía mayor cantidad de respuestas erróneas en comparación con el modelo finalmente elegido.

En la primera prueba 5.12, se identifican varias deficiencias significativas en la respuesta proporcionada en formato JSON. Uno de los problemas más notorios son los errores ortográficos, como el término “*Ejercidez de músculos del abdomen y de la zona lumbar*” y “*Ejercidez de pesas*”, los cuales deben corregirse a “*Ejercicio*”. Además, la respuesta contiene términos en inglés no traducidos correctamente, como en el caso de “*Realizar estiramientos para relajarse y recuperar los músculos after the workout*”, el cual se debe traducir adecuadamente al español para mantener la coherencia lingüística del texto completo. Otro aspecto problemático es la falta de coherencia en la estructura de la respuesta. Por ejemplo, se menciona “*Estiramientos*” bajo la sección de “*Secuencia final*”, lo cual no es claro ni coherente con el propósito de esa sección. Además, no se especifica el significado o propósito de “*Entrenamiento de define*”, lo que genera confusión. En cuanto a los detalles proporcionados, presenta menos información detallada en comparación con Mistral. Mientras que Mistral detalla cada ejercicio con tipos, repeticiones, sets y equipo necesario; Llama 2 se limita a mencionar series y repeticiones. Aunque incluye secciones para calentamiento y estiramiento, no cumplió con el requisito de proporcionar una rutina exacta de cinco ejercicios, siendo un requisito importante del *prompt* inicial. Asimismo, la estructura del JSON de utiliza múltiples arreglos para separar la rutina en cinco partes, creando una estructura más compleja y menos intuitiva.

En la segunda prueba 5.13, se ha seguido correctamente la estructura del JSON, incluyendo todos los campos pedidos. Además, ha abordado la lesión en el hombro al ajustar la intensidad de cada ejercicio, aunque no ha excluido completamente los ejercicios que podrían afectar esta área. Es importante destacar que concluye resaltando que la rutina proporcionada es un ejemplo y debe ser modificada según las necesidades individuales y habilidades de cada persona. Por lo tanto, no se considera una rutina personalizada, sino más bien un punto de partida, ver figura 5.14. El principal error radica en que no se han generado las 8 rutinas solicitadas, ni siquiera las 6 como hizo Mistral; en su lugar, ha presentado una única rutina. Esto incumple con el requisito de proporcionar múltiples rutinas distintas distribuidas en días específicos, limitando la variedad y adaptabilidad del plan de ejercicios propuesto.

Listado 5.12: Rutina generada por Llama 2 en la prueba 1

```

1   {
2     "Calentamiento": [
3       {
4         "Ejercicio": "Caminar o correr ligero en la zona del gimnasio",
5         "Duración": 5-10 minutos
6       },
7       {
8         "Ejercicio": "Realizar estiramientos para relajarse y prepararse para
9           el entrenamiento",
10          "Duración": 5-10 minutos
11        }
12      ],
13      "Entrenamiento de fuerza": [
14        {
15          "Ejercicio": "Ejercízete de pesas",
16          "Series": 3,
17          "Repeticiones": 8-12
18        },
19        {
20          "Ejercicio": "Entrenamiento de músculos del brazo y hombros",
21          "Series": 3,
22          "Repeticiones": 8-12
23        },
24        {
25          "Ejercicio": "Entrenamiento de músculos de piernas y espalda",
26          "Series": 3,
27          "Repeticiones": 8-12
28        }
29      ],
30      "Entrenamiento de definición": [
31        {
32          "Ejercicio": "Ejercízete de músculos del abdomen y de la zona lumbar",
33          "Series": 3,
34          "Repeticiones": 8-12
35        },
36        {
37          "Ejercicio": "Entrenamiento de músculos de brazos y hombros",
38          "Series": 3,
39          "Repeticiones": 8-12
40        }
41      ],
42      "Secuencia final": [
43        {
44          "Ejercicio": "Realizar estiramientos para relajarse y recuperar los
45            músculos after the workout",
46            "Duración": 5-10 minutos
47        }
48      ]
49  }

```

Evaluación de Llama 3

En la primera prueba 5.15, uno de los errores más evidentes es su respuesta en inglés, mientras que la solicitud original está en idioma español. Esto representa una falta de cumplimiento con la solicitud implícita de responder en el mismo idioma que se planteó el enunciado, posibilitando ser un inconveniente para el usuario final que esperaba una respuesta en su idioma. Sin embargo, destaca en varios aspectos en comparación con los otros modelos; cada ejercicio de la rutina incluye detalles precisos, como el nombre específico (“*Barbell Back Squats*”, “*Lat Pulldowns*”, etc.), el número de sets, repeticiones (en un rango de 8-12 o 10-15), y el peso recomendado (desde ligero a moderado hasta pesado). Asimismo, la respuesta muestra más coherencia y claridad en la estructura y organización de la rutina. No se observan errores ortográficos evidentes, lo cual puede ser atribuido a un mayor dominio del idioma inglés, lo que facilita una expresión más fluida y precisa.

Siguiendo con la segunda prueba 5.16, resulta sorprendente que la respuesta fuera proporcionada en español, dado que en la prueba anterior este modelo había respondido en inglés. Comprobándose con esta respuesta que las respuestas en inglés son más adecuadas y coherentes. Los nombres de los ejercicios en inglés suelen ser más específicos y descriptivos en comparación con sus equivalentes en español, que a menudo ofrecen descripciones más generales, como “*Ejercicio de piernas en máquina*” en lugar de un nombre específico para el ejercicio. Aunque se incluyeron secciones de calentamiento y estiramiento para prevenir lesiones, no se abordó adecuadamente la lesión específica mencionada inicialmente en el *prompt*. La conclusión de la rutina 5.17, indica que se han incluido ejercicios que no afectan directamente la lesión en el hombro, aunque no se omitieron por completo los ejercicios que podrían causar daño. En términos de coherencia, las rutinas presentan algunas inconsistencias, como la inclusión de dos estiramientos y una sentadilla intercalados, cuando lo ideal sería seguir

Listado 5.13: Rutina generada por Llama 2 en la prueba 2.(1 de 1)

```

1   {
2     "routine": {
3       "Day": [
4         {
5           "Warm-up": {
6             "type": "Dynamic stretching",
7             "duration": 10,
8             "exercises": [
9               {"name": "Arm circles", "repetitions": 15, "sets": 3},
10              {"name": "Shoulder rolls", "repetitions": 15, "sets": 3},
11              {"name": "Neck stretches", "repetitions": 10, "sets": 3}
12            ]
13          },
14          "Exercise 1": {
15            "type": "Resistance band shoulder abduction",
16            "repetitions": 3,
17            "sets": 3,
18            "intensity": "Light"
19          },
20          "Exercise 2": {
21            "type": "Elbow flexion with resistance band",
22            "repetitions": 3,
23            "sets": 3,
24            "intensity": "Medium"
25          },
26          "Exercise 3": {
27            "type": "Shoulder extension with dumbbells",
28            "repetitions": 3,
29            "sets": 3,
30            "intensity": "Moderate"
31          },
32          "Exercise 4": {
33            "type": "Lateral raises with resistance bands",
34            "repetitions": 3,
35            "sets": 3,
36            "intensity": "Light"
37          },
38          "Exercise 5": {
39            "type": "Tricep dips with bench",
40            "repetitions": 3,
41            "sets": 3,
42            "intensity": "Medium"
43        }
44      ]
45    }
46  }
47 }
```

Listado 5.14: Información adicional no solicitada. Llama 2

```

1 This routine includes a 10-minute warm-up consisting of dynamic stretching exercises for the
2 shoulders and neck. It then progresses to 5 exercises targeting different muscle groups in the
3 shoulder area, with varying intensities and repetitions. The exercises are designed to be
4 performed in a gym setting using resistance bands or dumbbells.
5
6 Note: This is just an example routine and may need to be modified based on individual needs and
7 abilities. It's important to consult with a medical professional before starting any new
8 exercise program, especially if you have an injury or are experiencing pain.
```

un orden claro y colocar los estiramientos al final de la rutina. En cuanto a la estructura del JSON, Llama 3 fue el modelo que generó más rutinas, un total de 9, a pesar de que se especificaron 8 en el *prompt* inicial. Además, el campo “Day” que debería identificar el día de la semana para cada rutina no fue respetado adecuadamente.

Discusión de los resultados

Finalmente, se lleva a cabo una recapitulación de los errores más comunes observados en este primer acercamiento, junto con algunas conclusiones a tener en cuenta. En las pruebas de *prompt*, es importante destacar la clara diferencia entre las respuestas en español y en inglés, observándose una cierta mejora en la mayoría de las respuestas generadas en inglés. Además, un aspecto clave de todos los modelos es que la respuesta proporciona información adicional al JSON solicitado. Se incluye una introducción que no es necesaria, generándose una frase que enfatiza que seguidamente se proporcionará una rutina de ejercicio o una conclusión que indica la esperanza de que la respuesta sea útil.

En la primera prueba, Mistral y Llama 2 fallan en la terminología y la ortografía, afectando la coherencia de sus respuestas. En cambio, Llama 3 no comete errores de terminología ni ortográficos,

Listado 5.15: Rutina generada por Llama 3 en la prueba 1

```

1  {
2    "routine": [
3      {
4        "name": "Bench Press",
5        "sets": 3,
6        "reps": 8-12,
7        "weight": "moderate to heavy"
8      },
9      {
10        "name": "Incline Dumbbell Curls",
11        "sets": 3,
12        "reps": 10-15,
13        "weight": "light to moderate"
14      },
15      {
16        "name": "Leg Press",
17        "sets": 3,
18        "reps": 8-12,
19        "weight": "heavy"
20      },
21      {
22        "name": "Pull-ups",
23        "sets": 3,
24        "reps": "as many as possible",
25        "weight": "body weight only"
26      },
27      {
28        "name": "Seated Cable Rows",
29        "sets": 3,
30        "reps": 8-12,
31        "weight": "moderate to heavy"
32      }
33    ]
34  }

```

Listado 5.16: Primera rutina generada por Llama 3 en la prueba 2.(1 de 9)

```

1  {
2    "routine": [
3      {
4        "Warm-up": {"Exercise": "Ciclo", "Sets": 2, "Reps": 5},
5        "Exercise 1": {"Name": "Flexiones con mancuernas", "Sets": 3, "Reps": 8},
6        "Exercise 2": {"Name": "Remo lateral con pesa", "Sets": 3, "Reps": 10},
7        "Exercise 3": {"Name": "Ejercicio de hombro con pesa", "Sets": 2, "Reps": 12},
8        "Exercise 4": {"Name": "Sentadillas con mancuernas", "Sets": 3, "Reps": 10},
9        "Exercise 5": {"Name": "Estiramientos de hombro y brazo", "Sets": 1, "Reps": 20}
10      },
11    }

```

Listado 5.17: Información adicional no solicitada. Llama 3

```

1 En esta rutina, se han incluido ejercicios que no afectan directamente la lesión en el hombro,
2 como sentadillas con pesa o remo lateral con pesa. Se han excluido ejercicios que pueden hacer
3 daño a la lesión, como flexiones con pesa o ejercicios de hombro que impliquen una gran cantidad
4 de movimiento o estrés.
5
6 Se ha incluido un calentamiento y estiramientos después de cada sesión para prevenir lesiones y
7 mejorar el rendimiento. Es importante recordar que es fundamental escuchar al cuerpo y ajustar
8 la rutina según sea necesario. Asegúrate de consultar con un profesional antes de iniciar
9 cualquier nuevo programa de entrenamiento.

```

presenta la información de manera ordenada y clara. Además de ofrecer información detallada sobre el peso a levantar para cada ejercicio, lo que ayuda a los usuarios al ofrecer mayor precisión y utilidad. En la segunda prueba, todos los modelos enfrentan problemas con el número de rutinas y la estructura de sus respuestas. Aunque ninguno supera la prueba, Mistral se destaca por seleccionar los ejercicios más apropiados para la lesión mencionada, y Llama 3 se acerca más al número solicitado de rutinas. Es evidente que, al tratarse de un *prompt* más largo, se presentan desafíos más notables para alcanzar los objetivos establecidos.

Se pueden extraer varias conclusiones a partir de los datos recopilados sobre el rendimiento 5.1:

- Mistral es el más veloz en cuanto a la duración total, carga del modelo y tiempo de respuesta. Además, produce mayor respuesta con 329 *tokens* y un tamaño de 31.97 KB.
- Llama 2 es el más lento en términos de duración total, carga del modelo y tiempo de respuesta. No obstante, posee un tiempo de evaluación del *prompt* más corto al evaluar 64 *tokens*, sugiriendo

Métrica	Mistral	Llama 2	Llama 3
total_duration	87777560800 ns	145315613900 ns	95763639400 ns
load_duration	8279619000 ns	9162242800 ns	11063763200 ns
prompt_eval_count	55 tokens	64 tokens	49 tokens
prompt_eval_duration	5408667000 ns	4930807000 ns	5051713000 ns
eval_count	329 tokens	323 tokens	318 tokens
eval_duration	74080993000 ns	131210586000 ns	79646302000 ns
Response Time	1m 27.78s	2m 25.32s	1m 35.82s
Response Size	31.97KB • Body: 31.85KB • Headers: 120B	30.84KB • Body: 30.72KB • Headers: 120B	30.85KB • Body: 30.73KB • Headers: 120B

Tabla 5.1: Métricas de rendimiento de Mistral, Llama 2 y Llama 3

eficiencia en esa etapa particular.

- Llama 3 tiene un desempeño medio, con una duración total y tiempo de respuesta más rápidos que Llama 2 pero más lentos que Mistral. El tiempo de carga del modelo es el más lento, sin embargo, produce una respuesta con 318 *tokens* en un tiempo razonable (79.65s).

En términos de velocidad y tamaño de respuesta, Mistral parece ser la opción más eficiente en general, mientras que Llama 2 podría mejorar con optimizaciones adicionales para reducir su tiempo total y de respuesta. Llama 3 tiene un equilibrio aceptable pero necesita mejorar en la carga del modelo.

5.3. MÓDULOS

Uno de los aspectos esenciales de este proceso es el uso de módulos de desarrollo, que constituyen los bloques de construcción esenciales para crear aplicaciones web dinámicas y funcionales. A continuación se describen los módulos que componen la arquitectura del proyecto. Cada módulo incluye una descripción de los desafíos encontrados, la solución implementada para abordarlos y las ventajas de haber elegido esa solución en lugar de otras.

5.3.1. Sistema de Recomendaciones Personalizadas

Para comprender mejor el proceso básico de recomendación de rutinas personalizadas, desde el registro del usuario hasta la visualización de la rutina, se precisa plasmar el proceso de recomendaciones personalizadas en un diagrama de secuencia 5.4.

En primer lugar, el usuario debe completar su formulario registro, rellenando los datos nombre de usuario, nombre, apellidos, email y contraseña. A continuación se muestra un formulario de preguntas relacionadas con sus hábitos y preferencias deportivas que el usuario va respondiendo, cada respuesta a cada pregunta se va guardando a medida que se responden. Al llegar a la pregunta 11, ya se ha recopilado suficiente información en la base de datos para generar la rutina personalizada, por lo tanto, al responder esa pregunta y continuar a la siguiente, se envía una solicitud al servidor para generar la rutina personalizada. El servidor, gestiona dicha petición comunicándose con la API de Ollama utilizando el *prompt* creado a partir de las selecciones del usuario recogidas de la base de datos, para crear la rutina personalizada. Este proceso se realiza en paralelo mientras el usuario continúa respondiendo el resto de las preguntas del formulario, debido al tiempo considerable que requiere el Modelo de Lenguaje Grande para generar la rutina, dado que es un *prompt* complejo. Una vez que el usuario termina el formulario, se lleva a cabo el proceso de obtención de la rutina previamente generada y guardada en la base de datos, para mostrarla en el navegador, asimismo, se envía otra

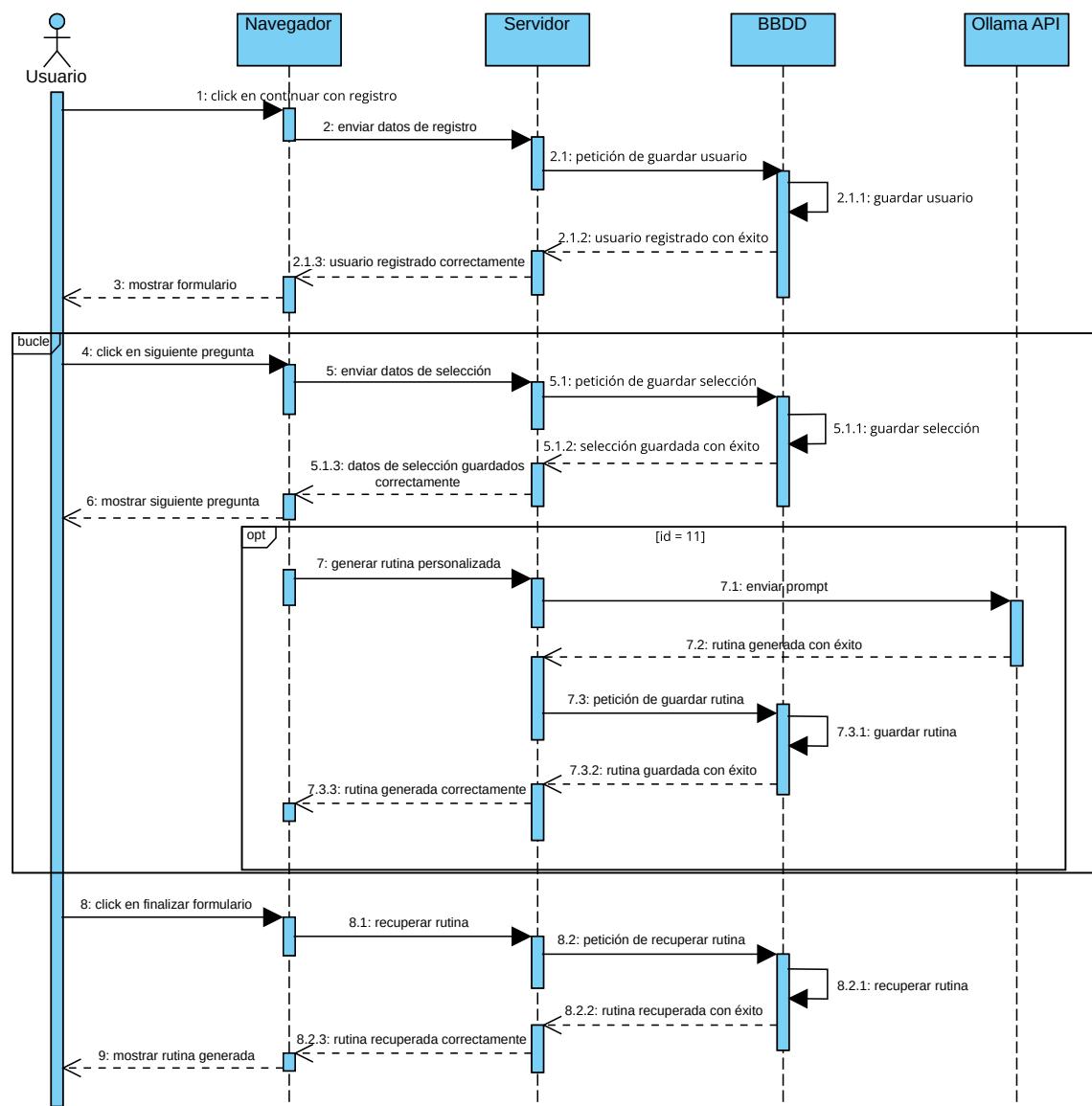


Figura 5.4: Diagrama de secuencia del flujo de creación de rutinas (Fuente: T.Redondo, CC BY-NC-ND)

petición a la API de Ollama para crear un análisis de hábitos y recomendaciones personalizadas con los datos restantes del formulario que no se utilizaron para la generación de la rutina personalizada.

5.3.2. Creación de formularios

En este módulo se presenta el desafío de decidir cómo estructurar el formulario: presentar una pregunta a la vez (*One-Question-at-a-Time Form*) o todas las preguntas a la vez en una sola página (*Multi-Question Form*), y determinar los datos necesarios que se necesitan del usuario para que el Modelo de Lenguaje Grande pueda llevar a cabo la generación de la rutina de la forma más personalizada posible.

Se opta por la creación de formularios de una sola pregunta. Este diseño facilita el proceso para los usuarios al no sobrecargarlos con demasiada información de una sola vez. Además, se escoge un formato de cuestionario con opciones únicas y algunas preguntas abiertas para datos específicos como el peso, altura y lesiones, ofreciendo mayor variedad. Para hacerlo aún más atractivo, se incluye

una curiosidad relacionada con cada pregunta. Asimismo, esta forma de presentar las preguntas posibilita el flujo de generación de rutinas en paralelo

Se elaboraron 16 preguntas, divididas en dos bloques. El primer bloque incluye preguntas relacionadas con aspectos a tener en cuenta para generar la rutina de ejercicio, desde las lesiones que presente, el lugar donde se va a llevar a cabo el entrenamiento hasta el nivel de actividad física del usuario. El segundo bloque se enfoca más en los hábitos y objetivos, como el consumo de agua, las horas de sueño, los objetivos principales y secundarios, entre otros y así poder realizar un análisis personalizado sobre hábitos deportivos. Esta estructura permite cubrir varios aspectos del deporte y crear una aplicación más completa.

5.3.3. Creación de rutinas

La dificultad radica en que el modelo escogido presenta tiempos de generación muy amplios. Además, el *prompt* utilizado para la creación de rutinas, inicialmente requería generar los días de rutina para cada semana, lo que resulta en solicitudes repetitivas y continuas al modelo.

Para solventar este problema, se utiliza la técnica de *Prompt Reframing*, replanteando la estructura del *prompt*. Se realizaron cambios sutiles en la redacción del enunciado manteniendo la intención original de la consulta variando en detalle y longitud. Se esperaba que estos ajustes redujeran considerablemente el tiempo de generación, pero en realidad, la diferencia es mínima. Finalmente, la solución se enfoca en reducir el carácter repetitivo de generación; siendo más eficiente solicitar la generación de rutinas mensuales en lugar de semanales y almacenar un gran número de rutinas ya generadas en la base de datos.

5.3.4. Mostrar las rutinas

Este módulo aborda el desafío de desarrollar el seguimiento de los ejercicios completados y la obtención de la siguiente rutina una vez finalizada la rutina actual.

Para mostrar las rutinas, se opta por listar los ejercicios en *cards* individuales siguiendo un orden previamente especificado cuando se guardaron en la base de datos. Cada *card* incluye un botón de “*Ejercicio Completado*”. A medida que los usuarios completan cada ejercicio, deben pulsar este botón, lo que actualiza automáticamente en la base de datos el estado del ejercicio, *realizado*, y la fecha de finalización del ejercicio, *fecha_realizado*. Esto permite un seguimiento detallado de cada ejercicio. Una vez que se confirma de que todos los ejercicios han sido completados, aparece un diálogo modal. Este diálogo solicita la interacción del usuario para proceder con la obtención de la siguiente rutina. Además, actualiza el registro en la base de datos con la fecha de realización de la rutina, *rutina_realizada*. También se tiene en cuenta la fecha de la última vez en que se visualiza una rutina, *last_viewed_at*. Este campo permite determinar que rutina debe mostrarse cuando el usuario regresa a la página web. Si el campo, *rutina_realizada*, está vacío y el campo, *last_view_at*, contiene la fecha de la última vez que se vio la rutina, esto indica que la rutina actual no ha sido completada y, por lo tanto, debe mostrarse nuevamente. En cambio, si los ambos campos (*rutina_realizada* y *last_viewed_at*) contienen datos, esto significa que la rutina actual se ha completado y es momento de mostrar una rutina nueva.

Cuando no existen más rutinas disponibles en la base de datos, se muestra un nuevo diálogo modal que informa al usuario que ha completado todas las rutinas del mes. Este diálogo modal también inicia el proceso de generación de nuevas rutinas personalizadas, completándose nuevos registros en la base de datos.

5.3.5. Perfil del usuario

En este módulo, se implementa la interfaz que muestra los datos del usuario junto con las preguntas y respuestas seleccionadas, ofreciendo la posibilidad de modificar todos los campos.

El principal desafío surge al considerar cómo actualizar las rutinas cuando se modifica una respuesta a una pregunta. Se evaluaron dos alternativas para manejar este proceso; la primera alternativa consiste en generar automáticamente nuevas rutinas cada vez que se acepte la modificación de un campo. Sin embargo, este enfoque presenta problemas, ya que si un usuario desea modificar varios campos, se generan múltiples solicitudes de actualización, lo cual no era eficiente ni práctico. La segunda alternativa, y la más adecuada, consiste en implementar un botón que brinde al usuario la posibilidad de actualizar las rutinas de ejercicios una vez termine de modificar los datos que deseé.

5.3.6. Creación del análisis

La creación del análisis se basa en un *prompt* que se envía justo después de que el usuario complete el formulario. Este análisis evalúa los datos proporcionados por el usuario: el tipo de cuerpo, los objetivos, los hábitos relacionados con la actividad física, el sueño, la ingesta de agua, el peso y la altura. El modelo desglosa cada análisis, proporcionando una descripción del campo, una opinión detallada y ofrece recomendaciones para posibles mejoras.

Para asegurar que cada usuario tenga un único análisis, dado que cada usuario tiene una entrada única en la base de datos con toda la información relevante, se implementa un mecanismo de actualización. Cuando el usuario pulsa el botón para actualizar el análisis, se muestra un cuadro de diálogo que indica que el análisis anterior será actualizado. Una vez el usuario aprueba la actualización, si ya existe un análisis previo en la base de datos, se actualizan los datos con la información generada en el análisis actual. En caso contrario, se crea un nuevo análisis y se guarda en la base de datos.

5.3.7. Calendario de rutinas

Para llevar a cabo un seguimiento de las rutinas completadas y pendientes de realizar, se implementa un calendario interactivo. Este calendario utiliza un código de colores para diferenciar entre rutinas completadas y rutinas aún pendientes. Al hacer click en una rutina completada, se muestra un desglose de todos los ejercicios asociados a la rutina que corresponde con ese día y mes, proporcionando detalles sobre el progreso del usuario.

Por otro lado, los días marcados en otro color indican que quedan rutinas pendientes. Estos días no ofrecen la opción de visualizar los ejercicios asociados, con el objetivo de mantener la intriga y motivación del usuario. El número de días coloreados refleja el número total de rutinas que aún quedan por completar en el mes. Es importante destacar, que estos días representativos no sugieren que la rutina deba realizarse necesariamente en ese día específico; simplemente indican cuántas rutinas quedan pendientes.

5.3.8. Logros

En cuanto a los logros, se estudiaron los retos que más motivarían a los usuarios. Se contabilizan varios indicadores clave de rendimiento, como el número de ejercicios realizados así como el número de rutinas completadas, las horas totales dedicadas al deporte y los puntos obtenidos.

La métrica de los puntos obtenidos se basa en la dificultad asignada a cada ejercicio por el modelo a la hora de generar las rutinas. Cada ejercicio recibe una clasificación de dificultad en una escala del 1 al 7, donde el 1 representa la menor dificultad y 7 la mayor. Los puntos obtenidos se calculan como la suma de los valores asignados a cada ejercicio realizado. Para facilitar la comprensión, se incluye una *Key Card* que proporciona un código de colores asociado a los niveles de dificultad.

Esas métricas se almacenan en la base de datos y se actualizan a medida que el usuario complete más acciones. Al alcanzar ciertos hitos en estas métricas, se desbloquean logros o trofeos. Al hacer click en las celdas, se ofrece una descripción detallada de cómo conseguir el logro, los beneficios para la salud asociados al alcanzarlo, y se indica si el logro se ha desbloqueado o no. Cuando se desbloquea un logro o trofeo, la celda correspondiente se colorea.

Los retos se dividen en cuatro categorías:

1. **Rutinas.** Retos basados en la cantidad de rutinas completadas.
2. **Puntos.** Retos relacionados con la acumulación de puntos conseguidos, que se asignan según la dificultad de los ejercicios realizados.
3. **Consistencia.** Retos que reconocen las horas invertidas en la realización de ejercicio. Esta categoría mide la regularidad del usuario en mantener un régimen de ejercicio constante.
4. **Rutinas por tiempo.** Esta categoría se enfoca en la realización de las rutinas en un horario rígido. Realizar ejercicio a la misma hora regularmente fomenta un aumento en la actividad física y su mantenimiento a largo plazo. La implementación de esta categoría se basa en la fecha de *rutina_realizada* en formato ISO 8601 de fecha y hora.

Cada trofeo representa la superación de una categoría específica. Al lograr superar una categoría, se desbloquea una recompensa adicional, que puede incluir una dieta alimentaria o consejos deportivos. Cada vez que se desbloquea un logro, se notifica al usuario mediante *alert box*. Los cuadros de alerta son modales, lo que significa que requieren la atención inmediata del usuario.

5.3.9. Asistente virtual

Al implementar el asistente virtual, se enfatiza la problemática del tiempo de carga del modelo, ya que es crucial que la respuesta sea instantánea para proporcionar una experiencia fluida al usuario. Como solución, cuando el usuario accede a la página con sus credenciales, se envía simultáneamente un *prompt* que indica el inicio del chat con su contexto. Esto asegura que, al momento de surgir cualquier duda o necesidad de asistencia durante la realización de la rutina, el asistente virtual ya esté preparado y disponible para contestar al usuario.

Para que el asistente virtual tenga el contexto inicial bien definido, se emplea la técnica de *Role Play*, detallada en una sección anterior 5.1.5. Esta técnica requiere un historial de mensajes, el cual es fundamental para mantener el contexto de las conversaciones y ofrecer respuestas más precisas y coherentes. En este historial de mensajes se incluyen dos mensajes esenciales:

- **Asignación del Rol:** El asistente está configurado con el rol de un entrenador personal, lo que le permite responder como si fuera un experto en entrenamiento físico. Esta asignación asegura que el asistente proporcione respuestas relevantes y especializadas.
- **Ejercicios de la Rutina Actual:** Se incorpora información sobre los ejercicios que el usuario está llevando a cabo en su rutina actual. Esto permite al asistente tener una visión completa del contexto de la sesión de ejercicio del usuario, facilitando respuestas más ajustadas a sus necesidades y dudas específicas.

CAPÍTULO 6

Resultados

Este capítulo presenta los resultados conseguidos del sistema web desarrollado para este proyecto. Para facilitar su explicación, los resultados obtenidos se dividen e incluyen en los *sprints* que se llevan a cabo a lo largo del proyecto.

Como se habló en la sección de metodología 4, el proyecto se ha estructurado utilizando la metodología ágil *Scrum*, que divide el trabajo en 6 *sprints*, cada uno con una duración de 4 semanas. Al final de cada *sprint*, se realiza una reunión con el tutor para revisar el desarrollo hasta el momento, discutir las dificultades encontradas y definir los objetivos a alcanzar en el siguiente *sprint*.

6.1. SPRINT 0: OBJETIVOS GENERALES Y ESPECÍFICOS

Durante este *sprint* se lleva a cabo una reunión para establecer los objetivos generales y específicos de desarrollo que se incorporaron en el proyecto. Asimismo, en este *sprint* se configuraron los entornos de trabajo necesarios para el desarrollo del nuevo servicio web.

6.1.1. Estudio de los LLM y sistemas web

Además, se llevan a cabo varios estudios exhaustivos que se centran en los modelos de inteligencia artificial disponibles en la actualidad. Este proceso incluye la investigación de los diferentes modelos de lenguaje grande, evaluando su compatibilidad con diversas bibliotecas y herramientas de desarrollo. Para compararlos y seleccionar el modelo que ofreciera los mejores resultados y se adaptara mejor a las necesidades del proyecto, se elabora una tabla comparativa que evalúa y contrasta las características, capacidades y limitaciones de cada uno. También se pusieron a prueba mediante la comparación práctica para concluir cuál ofrece los mejores resultados en términos de rendimiento. El modelo que demuestra ser compatible, con las mejores características y mayor eficacia es desplegado e integrado posteriormente con el backend del proyecto. Otro aspecto de estudio consiste en analizar sistemas web y aplicaciones informáticas existentes que presentan soluciones similares. El objetivo de este análisis es comprender sus enfoques, funcionalidades y limitaciones.

6.1.2. Elección de la pila tecnológica

El análisis práctico y comparativo realizado facilita la selección de la pila tecnológica más adecuada, ya que los resultados obtenidos tienen un impacto significativo en la elección de las herramientas y tecnologías que se utilizan en el proyecto.

6.2. SPRINT 1: SISTEMA DE REGISTRO Y LOGIN

El siguiente *sprint* se enfoca en el desarrollo del sistema de registro y login de usuarios para el nuevo servicio web. Este proceso abarca la implementación de una funcionalidad básica de registro y login, permitiendo que los usuarios creen cuentas y accedan al sistema de forma segura. Se implementan formularios de registro y login que aseguran que únicamente los usuarios validados puedan ingresar. El resultado de esta implementación se muestra en la figura 6.1.

6.2.1. Diseño de Onboarding

Una parte crucial del *sprint* es el diseño e implementación de un proceso de *onboarding*. Este proceso guiado realiza una serie de preguntas a los nuevos usuarios para completar su perfil de manera detallada. Las preguntas abarcan aspectos como el lugar habitual de entrenamiento, la edad, el peso y otros datos relevantes. Este enfoque facilita la personalización del perfil. El resultado del proceso *onboarding* se muestra en la figura 6.2.

La información recopilada durante el proceso de *onboarding* se guarda de forma segura en el backend, empleando una base de datos para gestionar los datos de los usuarios de manera eficiente. Este almacenamiento permite la recuperación y actualización de datos cuando sea necesario, y garantiza que toda la información relevante esté organizada y accesible para futuras consultas y operaciones dentro del sistema.

6.2.2. Modificar la información de perfil

En este *sprint*, además de implementar el registro y el login, se desarrolla un sistema web funcional que permite a los usuarios no solo crear sus cuentas, sino también acceder y modificar la información de su perfil. Los usuarios pueden visualizar los datos que completaron durante el proceso de registro y *onboarding*, y tienen la capacidad de editar su perfil para actualizar cualquier información personal según sea necesario. El resultado se ilustra en la figura 6.5.

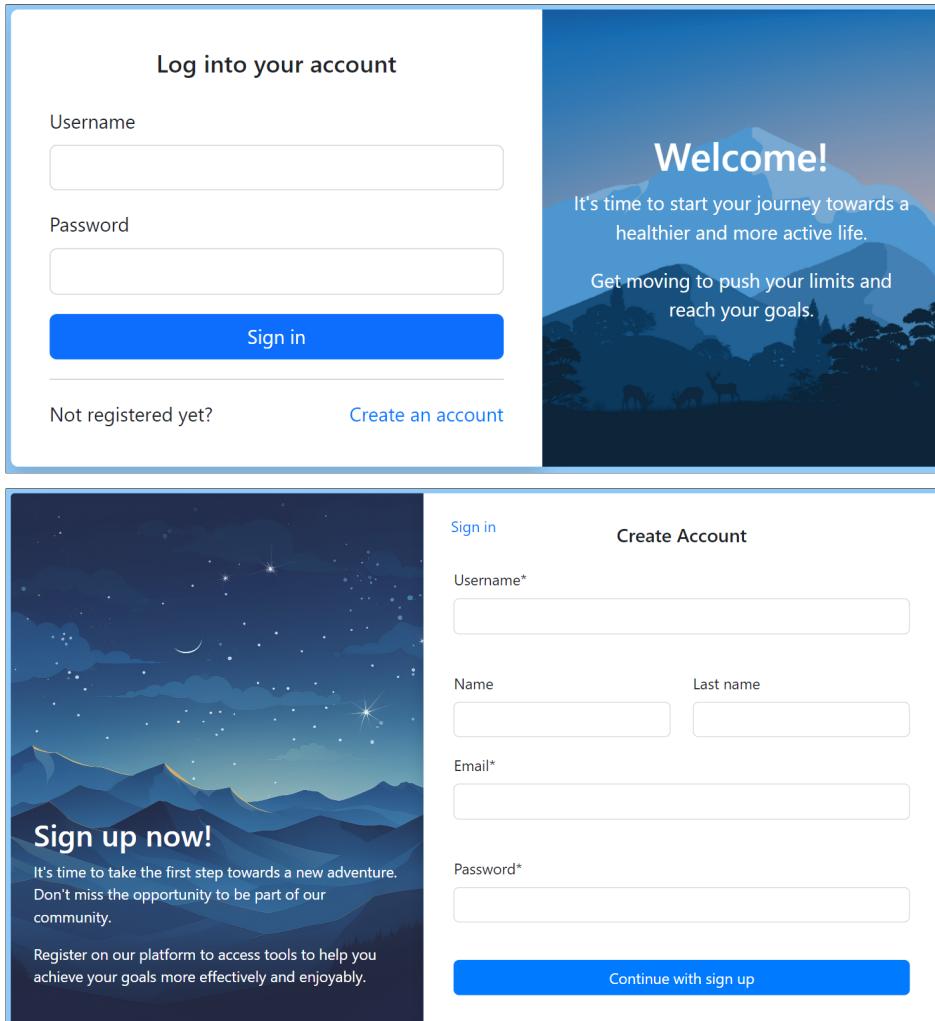


Figura 6.1: Login y Registro de usuarios

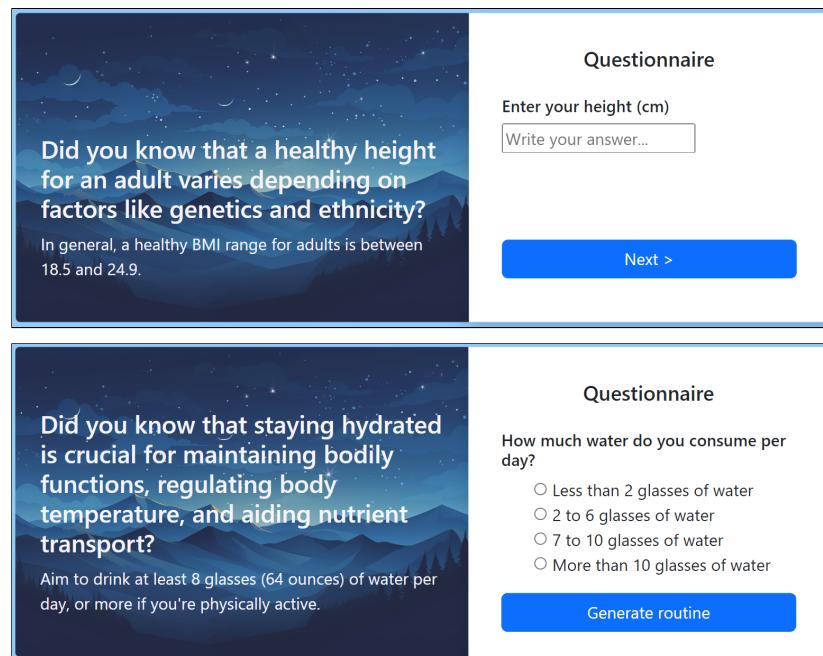


Figura 6.2: Proceso de OnBoarding

6.3. SPRINT 2: INTEGRACIÓN DEL MODELO

El siguiente *sprint* se centra en la integración del modelo de lenguaje en el backend del sistema web. Este paso fue fundamental para permitir que el sistema utilice las capacidades de la inteligencia artificial para proporcionar recomendaciones personalizadas de rutinas de ejercicios. La integración consiste en conectar el modelo de lenguaje con la infraestructura existente del backend, garantizando así que pueda recibir y procesar eficientemente los datos relevantes de los usuarios.

Uno de los principales objetivos de este *sprint* es el envío de los datos del perfil del usuario al modelo de lenguaje. Para lograr esto, se desarrolla un sistema que extrae la información relevante del perfil del usuario, y la envía al modelo para su procesamiento. Este proceso asegura que el modelo de lenguaje reciba la información necesaria para generar recomendaciones exactas y adaptadas a las características individuales de cada usuario.

Una vez que se consigue la integración completa con el modelo de lenguaje, se verifica que el flujo de datos entre el sistema web y el modelo funciona sin inconvenientes. El proceso de integración implica configurar las APIs requeridas para la comunicación entre el backend y el modelo de inteligencia artificial, y verificar que los datos se transmitan y se procesen adecuadamente para el funcionamiento eficaz del modelo.

6.3.1. Prueba de concepto

Para garantizar el correcto funcionamiento de la integración, se lleva a cabo una prueba de concepto. Esta prueba consiste en proporcionar datos de perfil del usuario al modelo y analizar la calidad de las recomendaciones de rutinas de ejercicios que el modelo genera como respuesta. La prueba de concepto ayuda a verificar que el modelo puede procesar la información de manera adecuada y brindar recomendaciones útiles según los datos proporcionados.

6.4. SPRINT 3: REGISTRO DE ACTIVIDAD

El siguiente *sprint* se enfoca en particular a desarrollar la funcionalidad de registro de actividad del usuario en la aplicación web. Este desarrollo permite a los usuarios registrar en tiempo real los ejercicios que están llevando a cabo, lo que facilita un seguimiento minucioso de sus rutinas y

avances. El desarrollo de esta característica requiere la elaboración de la interfaz de visualización de la rutina y el mecanismo de registro de actividad.

Para permitir un registro de actividad preciso, se crea una alternativa en la aplicación web donde los usuarios pueden especificar que han finalizado un ejercicio concreto. Una vez que el usuario ha comenzado un ejercicio, la aplicación ofrece una funcionalidad para indicar el ejercicio como completado mediante un botón, garantizando que el proceso sea fluido y no interrumpa la experiencia del usuario. Este proceso asegura que se registre de manera precisa todo ejercicio realizado, brindando un historial completo de los ejercicios llevados a cabo.

The screenshot displays a fitness application's user interface. On the left, a sidebar includes a profile icon, navigation links for 'My profile', 'Daily Routine', 'Objectives', 'Achievements', and 'About Us', and buttons for 'Logout' and 'Chat'. The main content area shows a 'KEY CARD' with a color-coded legend for difficulty levels (Lvl 1 to Lvl 8) and descriptions for daily routines. Below this, two exercises are listed: 'Exercise 1 Dumbbell Shoulder Press' (Difficulty level 3, Repetitions 12, Sets 3) and 'Exercise 2 Lateral Raises' (Difficulty level 3, Repetitions 12, Sets 3). Each exercise card includes a 'Technical explanation' and a 'Mark as completed' button. To the right, a 'Warm-up Shoulder Rolls' section is shown with details: Difficulty level 2, Repetitions 10, Sets 1, and a technical explanation. A green button at the bottom right of this section says 'Exercise completed'.

Figura 6.3: Visualización de rutinas y registro de actividad

6.4.1. Métricas relevantes

La implementación de esta funcionalidad también abarca la gestión y almacenamiento de métricas en el backend. Los datos registrados, como la cantidad de ejercicios y rutinas completadas, el total de horas realizando ejercicio y cualquier otra métrica relevante, se guardan en la base de datos para su posterior uso. Este almacenamiento les da la posibilidad a los usuarios de seguir de cerca su progreso a lo largo del tiempo. En la figura 6.4 se pueden ver las métricas calculadas.

The screenshot shows a fitness application's 'GOALS ATTAINED' section. It features a large image of a barbell with the quote "True happiness comes from within. Cultivate gratitude, spread kindness, and let your inner light shine." Below the image, a heading 'GOALS ATTAINED' is displayed. A descriptive text explains that users will see data for completed exercises, active hours, and earned points. Four summary cards below provide specific metrics: 'You have completed, 28 exercises!', 'You have completed, 5 routines!', 'You have been active, 37.5 hours!', and 'You have earned, 75 points!'. The sidebar on the left is identical to Figure 6.3, including the 'Chat' button.

Figura 6.4: Métricas relevantes

6.5. SPRINT 4: SISTEMA DE NAVEGACIÓN

El siguiente sprint se enfoca en preparar la aplicación web para una experiencia de usuario más estructurada y completa, mediante la creación de un sistema de navegación que organiza varias secciones en la interfaz. Se diseñar y desarrollar una barra de navegación, *navbar*, en el margen izquierdo de la aplicación, que sirve como el principal punto de acceso a las diferentes secciones del sistema: Mi Perfil, Rutina Diaria, Objetivos, Logros y Sobre Nosotros.

Cuando se accede a la aplicación, lo primero que se encuentra el usuario es la rutina que le toca ese día en la sección “Rutina Diaria”. Desde esta vista, el usuario tiene la opción de marcar un ejercicio como terminado y así registrar su progreso mientras avanza en su rutina. El usuario se puede desplazar por las demás secciones con la barra de navegación;

La primera sección es “Mi Perfil”, mostrada a detalle en las figuras 6.5 y 6.6, brinda a los usuarios la posibilidad de visualizar, modificar los detalles del perfil y visualizar el análisis personalizado. Por debajo de “Rutina Diaria” se encuentra la sección de “Objetivos” que ofrece un resumen de datos importantes sobre el desempeño del usuario y un calendario que se puede visualizar en las figuras 6.7 y 6.8, que organiza tanto las rutinas completadas como las que aún están por hacer. La penúltima sección es la de “Logros”, una sección diseñada para reflejar los logros conseguidos por los usuarios y ofrecer una lista de los que se pueden desbloquear. Y por último, la sección “Sobre Nosotros”, meramente informativa.

Question	Selected Option	Action
Key Aspects		
Body Type	Endomorph	Edit
Physical Activity Level	Beginner	Edit
Current Injury	Muscle strain.	Edit
Injury Location	Neck	Edit
Regular Exercise Location	Home	Edit
Days Available to Train per Week	4 days per week	Edit
Available Session Time	90 minutes	Edit
Lifestyle		
Goal	Maintain a healthy routine	Edit
Additional Goal	Form physical habits	Edit
Walking Time per Day	1-2 hours	Edit
Sleeping Time per Day	Between 5 and 6 hours	Edit
Daily Water Intake	7 to 10 glasses of water	Edit

Has any habit changed? Update the following routines here:
[Update the following routines](#)

Figura 6.5: Perfil de usuario

6.6. SPRINT 5: GAMIFICACIÓN Y ASISTENTE VIRTUAL

El próximo *sprint* pone énfasis en el desarrollo de la funcionalidad de gamificación dentro de la aplicación web, con la meta de incrementar la participación y motivación de los usuarios. Se lleva a cabo mediante la inclusión de un sistema de logros y recompensas que reconoce a los usuarios por completar rutinas, lograr objetivos y participar de forma activa en la plataforma. Los niveles de logros y recompensas se desbloquean cuando los usuarios alcanzan ciertos hitos. Esta funcionalidad se presenta en la figura 6.9.

6.6.1. Sistema de notificaciones

Adicionalmente a la gamificación, se implementa un sistema de notificaciones para mantener a los usuarios informados. Las notificaciones incluyen alertas sobre los logros obtenidos con mensajes inspiradores. Este sistema se diseña con la intención de enviar avisos en tiempo real, permitiendo que los usuarios obtengan esta información en el instante oportuno.

Body Type
Your BMI falls within the normal weight range (18.5 - 24.9), indicating a healthy body weight. It suggests that you are maintaining a good balance between your height and weight.
[More Info](#)

Goals
The goal is to achieve physical fitness and improve overall health, with an additional aim to enhance feelings of well-being and vitality.
[More Info](#)

Walking
Regular walking routine aimed at maintaining physical fitness and enhancing overall well-being through consistent physical activity.
[More Info](#)

Sleeping
This short nap duration helps in boosting alertness, enhancing cognitive function, and improving mood throughout the day.

Water Consumption
Consuming 7 to 10 glasses of water daily is essential for maintaining hydration levels, supporting bodily functions, and promoting overall health.

BMI Calculation
It provides a numerical indicator of body fat and helps assess whether an individual's weight is appropriate for their height.

Figura 6.6: Análisis personalizado

ROUTINES CALENDAR

Welcome to your routines calendar! This calendar provides an overview of your **completed** and **upcoming routines**. Take a look at your past routines by clicking on the respective dates, and explore the exercises you've performed. It's a great way to track your progress and stay **motivated** on your fitness journey. Keep up the **great work!**

Mon	Tue	Wed	Thu	Fri	Sat	Sun
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Date: Jun 12, 2024 ■ Completed Routine □ Pending Routine

Figura 6.7: Calendario de rutinas

SELECTED ROUTINE

Here are your exercises completed on Jun 12, 2024, with all sets and repetitions, the technical explanation, and their difficulty level from 1 to 10. Here you can review past exercises and progress further.

DUMBBELL SHOULDER PRESS Sets/Repetitions: 3 x 12 Difficulty Level: 3 Technical Explanation: Sit on a bench with back support, holding dumbbells in each hand at shoulder height. Press the weights directly upward until your arms are fully extended, then lower them back down.	LATERAL RAISES Sets/Repetitions: 3 x 12 Difficulty Level: 3 Technical Explanation: Stand with dumbbells in each hand, arms by your sides. Raise your arms out to the sides until they're parallel with the floor, then lower them back down.	FRONT RAISES Sets/Repetitions: 3 x 12 Difficulty Level: 4 Technical Explanation: Hold dumbbells in front of your thighs, palms facing your body. Lift the weights straight up to shoulder height, then lower them back down.
REVERSE FLYES Sets/Repetitions: 3 x 12 Difficulty Level: 4	PUSH-UPS Sets/Repetitions: 3 x To failure Difficulty Level: 2	TREADMILL RUNNING Speed: Moderate Duration: 20 minutes

Figura 6.8: Visualización de la rutina del día seleccionado

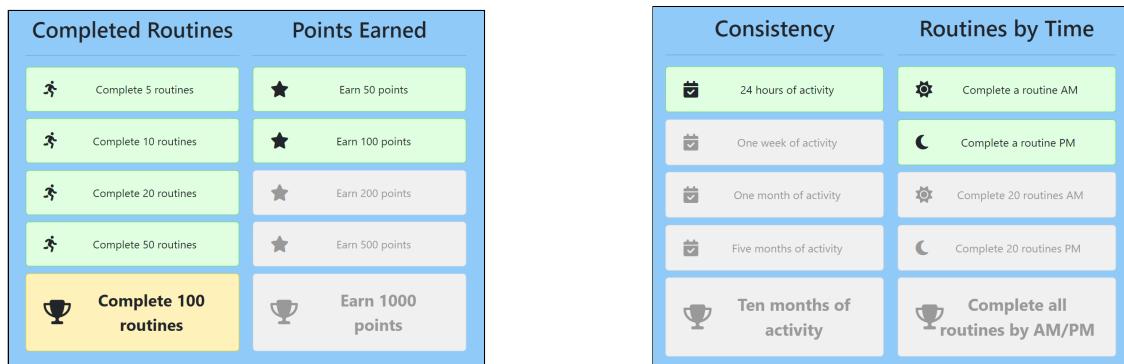


Figura 6.9: Sistema de logros

6.6.2. Asistente virtual

En este *sprint* también se desarrolla el asistente virtual, éste se implementa como un chat en la esquina inferior izquierda de la interfaz web. Está diseñado para ofrecer soporte en tiempo real a los usuarios, respondiendo a sus preguntas, ofreciendo asistencia sobre la rutina del día. La inclusión del asistente tiene como objetivo mejorar la experiencia del usuario mediante la sugerencia de ayuda accesible y oportuna. La creación de este asistente implica la creación de una interfaz de chat intuitiva para facilitar la interacción con el asistente 6.10 .

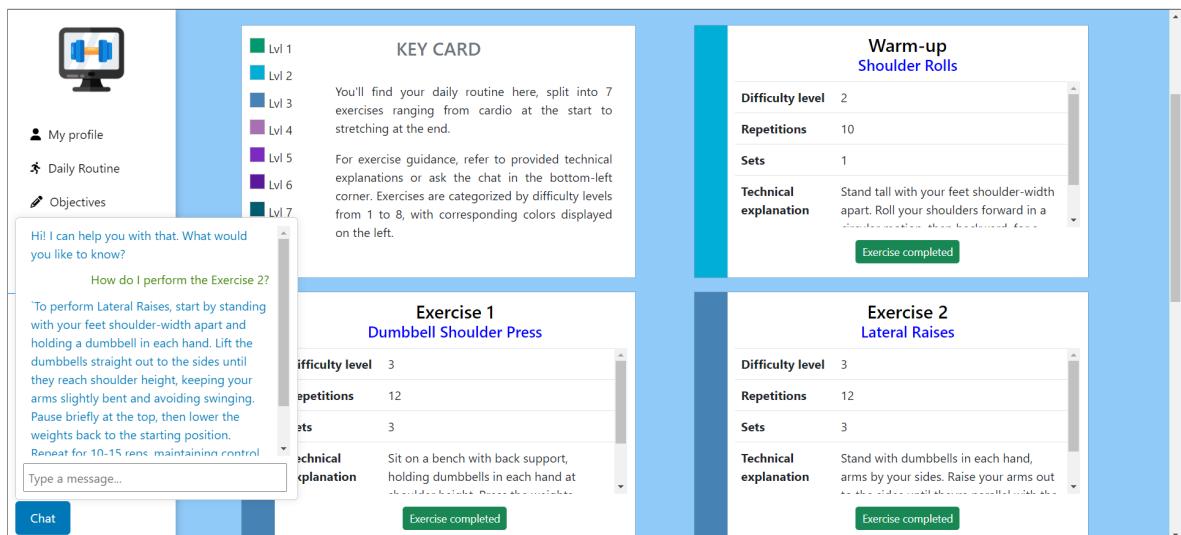


Figura 6.10: Asistente virtual

6.7. COSTES Y RECURSOS DEL PROYECTO

Para llevar a cabo el proyecto desde la fecha del 2 de febrero hasta la fecha del 22 de julio, siendo aproximadamente 171 días, se han utilizado dos tipos de recursos principales: los recursos humanos y los recursos de desarrollo.

La intervención de un ingeniero informático ha sido necesaria para elaboración del proyecto, cuya tarifa es de 40€ por hora. La cantidad de horas que ha invertido el desarrollador ha sido un total de 825 horas. La estimación se basa en las siguientes asunciones: cada mes tiene 20 días hábiles y cada jornada laboral es de 7.5 horas al día, dando un total de 150 horas de trabajo por mes en una jornada estándar. Teniendo en cuenta la duración de desarrollo de 5 meses y medio, resulta en un total de 825 horas. Por lo tanto, el coste asociado a la mano de obra del desarrollador es 33.000€

En lo que respecta a los recursos de desarrollo, se ha utilizado un ordenador Asus con sistema operativo Windows, cuyo coste ronda los 800€.

El coste total del proyecto se ha calculado sumando el coste de los recursos humanos y el coste del equipo de desarrollo. En la Tabla 6.1 se presenta un resumen de los costes asociados al proyecto.

Recurso	Unidad	Coste/unidad	Coste
Desarrollador	1	40€/hora	33.000€
Ordenador Asus	1	800€/unidad	800€
Total			33.800€

Tabla 6.1: Costes y recursos del proyecto

6.8. ESTADÍSTICAS DE CÓDIGO

Para estudiar las estadísticas de código de la aplicación, se utiliza la herramienta cloc en los dos entornos de desarrollo.

6.8.1. Herramienta cloc

Cloc¹, *Count Lines of Code*, se trata de una herramienta de software de código abierto utilizada para contabilizar la cantidad de líneas de código presentes en archivos fuente. Su principal objetivo es ofrecer métricas sobre la cantidad de código en un proyecto, incluyendo un desglose por lenguaje de programación.

En ambos resultados mostrados en las figuras 6.11 y 6.12 se detallan los totales de: archivos en ese lenguaje (files), líneas en blanco (blank), líneas que contienen comentarios (comment) y líneas de código ejecutable (code).

github.com/AlDanial/cloc v 2.00 T=0.76 s (92.9 files/s, 5084.6 lines/s)				
Language	files	blank	comment	code
Java	58	698	399	2499
Maven	1	8	1	103
XML	2	0	0	80
INI	6	0	0	35
Markdown	1	10	0	17
Properties	3	5	12	17
SUM:	71	721	412	2751

Figura 6.11: Número de líneas de código en el Backend

github.com/AlDanial/cloc v 2.00 T=336.32 s (44.8 files/s, 131.9 lines/s)				
Language	files	blank	comment	code
JSON	14982	0	0	38209
TypeScript	64	401	190	2537
HTML	13	104	106	1288
CSS	11	224	32	1209
JavaScript	1	1	6	37
INI	1	3	0	13
Markdown	1	2	0	7
SUM:	150'73	735	334	43300

Figura 6.12: Número de líneas de código en el Frontend

La disparidad en la cantidad de archivos se atribuye principalmente a la estructura modular y al ecosistema de dependencias de los proyectos Angular en contraste con los proyectos de Java. Para obtener una visión más clara de las líneas de código reales de Angular, se utiliza el comando cloc excluyendo el directorio node_modules de la cuenta, como se muestra:

```
cloc --exclude-dir=node_modules /ruta/al/directorio
```

¹<https://cloc.sourceforge.net/>

CAPÍTULO 7

Conclusiones

En este capítulo se presenta una evaluación final de los resultados alcanzados en este TFG, repasando cada uno de los objetivos iniciales y explicando cómo se han resuelto. Asimismo, se incluye un apartado para justificar la aplicación de las competencias específicas adquiridas en la tecnología específica cursada, demostrando su relevancia en el desarrollo del trabajo. También se incluye información sobre el trabajo futuro planificado al momento de la redacción. Finalmente, se realiza un breve análisis de las lecciones aprendidas, destacando tanto las buenas prácticas tecnológicas y procedimentales adquiridas, como otros aspectos relevantes.

7.1. OBJETIVOS ALCANZADOS

Como se expone en el capítulo de Objetivo 2 de este TFG, el principal objetivo ha sido el diseño y desarrollo de una plataforma de entrenamiento basada en inteligencia artificial y tecnologías web para mejorar la experiencia de la realización del ejercicio físico mediante la recomendación de rutinas personalizadas y adaptadas a los usuarios. A continuación, se realiza un juicio crítico y discusión sobre los objetivos logrados:

1. **Estudio de las tecnologías y herramientas que soporten el desarrollo y despliegue del servicio, incluyendo los sistemas web que tratan de solucionar una problemática similar a la que se enfrenta este proyecto.** Como se detalla en los capítulos de Introducción 1 y Estado del Arte 3 se realiza una investigación detallada sobre las tecnologías y herramientas en salud y *fitness*, revisando diversas fuentes como artículos académicos y publicaciones relevantes. Esta revisión incluye el estudio y comparativa de sistemas web y aplicaciones existentes para identificar sus funcionalidades y limitaciones, lo que ayuda a definir las mejores prácticas y avances tecnológicos que se puedan aplicar. En el capítulo de Arquitectura, 5, se lleva a cabo una evaluación de modelos de lenguaje grande, examinando su rendimiento en diferentes situaciones para seleccionar el modelo más adecuado para el proyecto.
2. **Estudio y análisis de funcionalidad adicional.** Se plantea un estudio de viabilidad para la integración de la aplicación para relojes inteligentes, que se detalla en la sección 7.3 de este capítulo y en el capítulo 3, se describe la actualidad del tema.
3. **Diseño de una arquitectura web que facilite la escalabilidad automática de la plataforma.** La plataforma está virtualizada mediante contenedores Docker, se habla de ello en los capítulos de Metodología 4 y Arquitectura 5. Esta estrategia permitiría desplegarla fácilmente y de manera escalable en algún servicio de computación en la nube como, por ejemplo, AWS ECS¹, asegurando que la plataforma pueda crecer y adaptarse sin problemas a medida que aumenten las necesidades del usuario.

¹<https://aws.amazon.com/es/ecs/>

Los siguientes objetivos específicos se detallan en los capítulos de Metodología 4, Arquitectura 5 y Resultados 6:

1. **Diseño y desarrollo de un sistema web, desplegado sobre la arquitectura anterior, que permita al usuario la selección y gestión de sus rutinas de ejercicios.** El sistema web se desarrolla utilizando tecnologías como Angular para el desarrollo del frontend y Spring Boot para el desarrollo del backend, integradas con una base de datos relacional MySQL. Esta configuración permite, mediante la interfaz de usuario, que el usuario lleve a cabo la selección y gestión de rutinas de ejercicio: Se diseña una página inicial para que el usuario visualice la rutina diaria, se incorpora un sistema de seguimiento de ejercicios que facilita a los usuarios el registro de actividades completadas, se implementa un calendario para que el usuario visualice las rutinas anteriores y tenga acceso a una vista organizada tentativamente de las rutinas pendientes y se configura un perfil de usuario que facilita que el usuario pueda actualizar la información y, si lo desea, crear nuevas rutinas personalizadas basadas en los datos más recientes. Esta implementación asegura una experiencia completa y adaptada a las necesidades del usuario.
2. **Diseño y desarrollo de un sistema de gamificación integrado en el sistema web.** Se incorporan técnicas de gamificación mediante un sistema de puntos, logros y recompensas, que permite al usuario acumular puntos mediante la finalización de ejercicios y rutinas, desbloquear logros que se notifican con alertas personalizadas y recibir recompensas por cada categoría de logros completada.
3. **Especificación e implementación de una API que pueda ser consumida por los sistemas de la plataforma.** Se define una API que permite la comunicación eficiente entre los distintos sistemas de la plataforma. En el desarrollo de esta API, se crean métodos específicos en el backend para manejar las solicitudes entrantes, cada uno de los cuales se adapta a un tipo concreto de solicitud HTTP de acuerdo con la arquitectura REST. Define funcionalidades concretas como la creación de rutinas de ejercicios y análisis personalizados, la actualización de datos de usuario, el envío de consultas al asistente virtual, entre otras funcionalidades. La implementación abarca la lógica necesaria para procesar las solicitudes, interactuar con el repositorio de datos y ofrecer respuestas adecuadas.
4. **Desarrollo de un módulo funcional inteligente que permita obtener recomendaciones de rutinas de ejercicios personalizadas adaptadas a las características de los usuarios y estructuradas para su representación en el sistema web.** Este objetivo se logra creando formularios para recopilar información significativa de cada usuario e implementando un algoritmo de recomendación que emplea el modelo de lenguaje Llama3 para analizar esa información. Este algoritmo se comunica con la API de Ollama para generar rutinas de ejercicios personalizadas. Finalmente, las rutinas generadas se estructuran de forma que facilitan su integración y visualización en el sistema web.

7.2. JUSTIFICACIÓN DE COMPETENCIAS ADQUIRIDAS

En este TFG se aplican las competencias correspondientes a la Tecnología Específica de [*Tecnologías de la Información*]:

TI3: *[Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas.]* Se utiliza la metodología Agile Scrum, conocida por su flexibilidad y capacidad de adaptación, lo que permite ajustar el desarrollo según las necesidades cambiantes de los clientes. Esta metodología reduce el riesgo de desviaciones costosas al permitir una adaptación continua a los requisitos del usuario mientras el proyecto avanza. Además, al utilizar sprints cortos y establecer metas definidas para cada fase, se consigue una gestión más eficiente del tiempo y los costos.

TI6: [Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.] Se justifica en el desarrollo de un sistema web integral, virtualizado y multifuncional que combina múltiples aspectos de estas tecnologías web, se ofrecen varios servicios y funcionalidades para satisfacer distintas necesidades y todo ello en un entorno adaptable a nuevas tecnologías que permitan la integración con relojes inteligentes.

7.3. TRABAJOS DERIVADOS Y FUTUROS

El trabajo futuro se dedica al estudio del desarrollo de una aplicación para relojes inteligentes, centrada en el registro de actividad física. Este estudio analiza la viabilidad y los beneficios de integrar la aplicación web con otros dispositivos, para que los usuarios puedan registrar y monitorizar su actividad física inmediata desde sus muñecas. Se evalúa la sincronización efectiva entre el reloj inteligente y la plataforma web, asegurando una experiencia fluida para los usuarios que tratan de encontrar una seguimiento integral de su actividad física y se abordan detalles prácticos de la integración, evaluando de qué manera esta aplicación puede optimizar la experiencia del usuario al ofrecer datos en tiempo real y simplificar el seguimiento de rutinas de ejercicio.

Una posible implementación para integrar una aplicación con relojes inteligentes es emplear Wear OS, el sistema operativo de Google para dispositivos de pulsera. Utilizando Android Studio y lenguajes de programación como Java o Kotlin, es posible desarrollar aplicaciones que se sincronicen de manera efectiva con Wear OS. Este sistema ofrece herramientas y APIs específicas, como la Fitness API², que permiten el registro, análisis y visualización de métricas de ejercicio en tiempo real.

Además de la integración con Wear OS, se contempla otra línea de trabajo, habilitar el soporte para integrar otros modelos de IA comerciales, como GPT-4³ de OpenAI, para potenciar las capacidades de la aplicación. La integración de GPT-4 permite ofrecer recomendaciones más personalizadas y sofisticadas y obtener una comprensión más profunda de los datos recopilados. Esto refina considerablemente la funcionalidad de la aplicación al promover una interacción más avanzada y personalizada con los usuarios, mejorando así la organización y el seguimiento de las rutinas de ejercicio.

7.4. VALORACIÓN PERSONAL

Crear este sistema web ha sido un proceso desafiantre y lleno de aprendizaje continuo. A lo largo de 170 días de trabajo, se superaron numerosos desafíos y dificultades, cada uno de los cuales ha sido una oportunidad para perfeccionar y afinar las habilidades. Resultando en una experiencia muy valiosa.

²<https://developer.android.com/health-and-fitness/guides/health-services?hl=es-419>

³<https://openai.com/index/gpt-4/>

Bibliografía

- [1] Aisera. Llm embeddings: What they are and how they work. <https://aisera.com/blog/llm-embeddings/>, julio 2024. Último acceso: 14 jul. 2024.
- [2] Amazon Web Services. Generative ai - aws. <https://aws.amazon.com/es/what-is/generative-ai/>. Último acceso: 5 jul. 2024.
- [3] American College of Sports Medicine. Acsm fitness trends. <https://www.acsm.org/education-resources/trending-topics-resources/acsm-fitness-trends>. Último acceso: 10 jul. 2024.
- [4] Arsys.es. Mysql: qué es y para qué sirve. <https://www.arsys.es/blog/mysql>. Último acceso: 19 jul. 2024.
- [5] Auth0. Json web tokens. <https://auth0.com/es/learn/json-web-tokens>, 2024. Último acceso: 16 jul. 2024.
- [6] AWS. Cómo elegir un llm (large language model). <https://dev.to/aws-espanol/como-elegir-un-llm-272o>. Último acceso: 8 jul. 2024.
- [7] Axarnet. Angular. <https://axarnet.es/blog/angular>. Último acceso: 22 jul. 2024.
- [8] BBVA. Los llm: Modelos de lenguaje que son y cómo funcionan. <https://www.bbva.com/es/innovacion/los-llm-modelos-de-lenguaje-que-son-y-como-funcionan/>, julio 2024. Último acceso: 14 jul. 2024.
- [9] Sandi Besen. Llm embeddings explained simply. <https://pub.aimind.so/llm-embeddings-explained-simply-f7536d3d0e4b>, julio 2023. Último acceso: 15 jul. 2024.
- [10] Ju Cabeza. Atención a las arquitecturas llm: Una guía de ingeniería de ml inteligente. <https://es.w3d.community/jucabeza/atencion-a-las-arquitecturas-llm-una-guia-de-ingenieria-de-ml-inteligente-1gf4>, 2023. Último acceso: 15 jul. 2024.
- [11] Andrés Navarro Cadavid, Juan Daniel Fernández Martínez, and Jonathan Morales Vélez. Revisión de metodologías ágiles para el desarrollo de software. *PROSPECTIVA*, 11(2):30–39, 2013. Último acceso: 1 jul. 2024.
- [12] Amaro Calderón, Sarah Dámaris Valverde Rebaza, and Jorge Carlos Valverde Rebaza. *Metodologías Ágiles*. Facultad de Ciencias Físicas y Matemáticas, Escuela de Informática, Trujillo, Perú, 2007.
- [13] Karla Mendes Calo, Elsa Estevez, and Pablo Fillottrani. *Evaluación de Metodologías Ágiles para Desarrollo de Software*. Universidad Nacional del Sur, Av. Alem 1253, (8000) Bahía Blanca, Argentina, 2010. Dpto. de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur; Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Argentina.
- [14] Angular Chile. Validaciones en formularios con angular. <https://medium.com/angular-chile/validaciones-en-formularios-con-angular-d00a2d535d17>, diciembre 2023. Último acceso: 16 jul. 2024.

- [15] SPSG Consulting. 1º barómetro de hábitos en deporte en España edición 2023. SPSG Consulting: <https://www.spsgconsulting.com/>, 2024. Último acceso: 26 jun. 2024.
- [16] Wikipedia contributors. Normalización de bases de datos. https://es.wikipedia.org/wiki/Normalizaci%C3%B3n_de_bases_de_datos, 2024. Último acceso: 18 jul. 2024.
- [17] Universitat Oberta de Catalunya (UOC). Normativa académica general de la uoc. https://cv.uoc.edu/annotation/cb826b689abc472d8fb5b2519840058b/699689/PID_00213707/PID_00213707.html#w31aab5c11, 2024. Último acceso: 18 jul. 2024.
- [18] Ministerio de Cultura y Deporte División de Estadística y Estudios, Secretaría General Técnica. *Encuesta de Hábitos Deportivos en España 2022*. Secretaría General Técnica, Subdirección General de Atención al ciudadano, Documentación y Publicaciones, Madrid, diciembre 2022. NIPO papel: 822-21-040-6, Depósito legal: M-19100-2021, NIPO línea: 822-21-041-1.
- [19] Economipedia. Diagrama de gantt — economipedia. <https://economipedia.com/definiciones/diagrama-de-gantt.html>. Último acceso: 12 jul. 2024.
- [20] Facebook. Presentamos meta llama 3: El modelo de lenguaje de gran tamaño más potente hasta la fecha. <https://about.fb.com/ltam/news/2024/04/presentamos-meta-llama-3-el-modelo-de-lenguaje-de-gran-tamano-mas-potente-hasta-la-fecha/>, abril 2024. Último acceso: 16 jul. 2024.
- [21] HubSpot. ¿qué es un stack tecnológico? <https://blog.hubspot.es/website/que-es-stack-tecnologico>. Accedido el 11 de julio de 2024.
- [22] HubSpot. ¿qué es una api rest? <https://blog.hubspot.es/website/que-es-api-rest>. Último acceso: 16 jul. 2024.
- [23] HubSpot. ¿qué es mysql? <https://blog.hubspot.es/website/que-es-mysql>, enero 2023. Último acceso: 15 jul. 2024.
- [24] HubSpot. ¿qué es desarrollo web? <https://blog.hubspot.es/website/que-es-desarrollo-web>, Fecha no especificada. Último acceso: 7 jul. 2024.
- [25] HubSpot. Metodología agile. <https://blog.hubspot.es/marketing/metodologia-agile>, Último acceso: 11 jul. 2024.
- [26] Hugging Face. Falcon 2: Open-source release of the largest ever language model. <https://huggingface.co/blog/falcon2-11b>. Último acceso: 9 jul. 2024.
- [27] IBM. Introducción a mongodb. <https://www.ibm.com/es-es/topics/mongodb>. Último acceso: 16 jul. 2024.
- [28] IBM. Large language models - ibm. <https://www.ibm.com/es-es/topics/large-language-models>. Último acceso: 5 jul. 2024.
- [29] Instituto ISAF. Falsas creencias sobre la formación deportiva. URL: <https://blog.institutoisaf.es/falsas-creencias-sobre-la-formacion-deportiva>, 2024. Último acceso: 7 jul. 2024.
- [30] KeepCoding. Estudio stack overflow 2023. <https://keepcoding.io/blog/estudio-stack-overflow-2023/>. Último acceso: 12 jul. 2024.
- [31] Kinsta. Graphql vs rest: Diferencias entre las dos tecnologías de api. <https://kinsta.com/es/blog/graphql-vs-rest/>. Último acceso: 11 jul. 2024.
- [32] Kinsta. Tipos de desarrolladores: ¿qué tipos existen y qué hacen? <https://kinsta.com/es/blog/tipos-de-desarrolladores/>, Fecha no especificada. Último acceso: 11 jul. 2024.
- [33] Patricio Letelier and M. Carmen Penadés. Métodologías ágiles para el desarrollo de software: extreme programming (xp). *Técnica Administrativa*, 05(26), apr 2006. Último acceso: 1 jul. 2024.
- [34] Nestor Maslej, Loredana Fattorini, Raymond Perrault, Vanessa Parli, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, and Jack Clark. *The AI Index 2024 Annual Report*. AI Index Steering

- Committee, Institute for Human-Centered AI, Stanford University, Stanford, CA, April 2024. The AI Index 2024 Annual Report by Stanford University is licensed under Attribution-NoDerivatives 4.0 International.
- [35] McKinsey & Company. The state of ai. <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai>. Último acceso: 13 jul. 2024.
 - [36] Ministerio de Educación, Formación Profesional y Deportes. *Anuario de Estadísticas Deportivas 2024*. Secretaría General Técnica, Subdirección General de Estadística y Estudios, Madrid, 2024 edition, 2024. NIPO línea: 164-24-145-4, NIPO papel: 164-24-146-X, Depósito Legal: M-17314-2020.
 - [37] Mistral AI. Tokenization guide. <https://docs.mistral.ai/guides/tokenization/>, 2024. Último acceso: 14 jul. 2024.
 - [38] LLama Model. Ventajas y desventajas de los modelos llama. <https://llamaimodel.com/es/ventajas-desventajas/>, julio 2024. Último acceso: 14 jul. 2024.
 - [39] Bibi Moreno Regalado. Por qué deberías considerar utilizar value objects. <https://www.linkedin.com/pulse/por-qu%C3%A9-deber%C3%AD-ADas-considerar-utilizar-value-objects-y-moreno-regalado/>, 2022. Último acceso: 17 jul. 2024.
 - [40] Mozilla Contributors. Svelte - getting started. https://developer.mozilla.org/es/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Svelte_getting_started. Último acceso: 22 jul. 2024.
 - [41] Mozilla Developer Network. Documentación de html. <https://developer.mozilla.org/es/docs/Web/HTML>. Último acceso: 13 jul. 2024.
 - [42] Gustavo Oliver. Cómo las aplicaciones móviles están revolucionando la forma en que hacemos ejercicio y nos mantenemos activos. <https://gustavoliver.com/como-las-aplicaciones-moviles/>. Último acceso: 11 jul. 2024.
 - [43] OpenWebinars. ¿qué es deno y qué lo hace diferente? <https://openwebinars.net/blog/que-es-deno-y-que-lo-hace-diferente/>. Último acceso: 20 jul. 2024.
 - [44] Oracle. What is a relational database? <https://www.oracle.com/es/database/what-is-a-relational-database/>, 2024. Último acceso: 15 jul. 2024.
 - [45] Organización Mundial de la Salud. *Informe sobre la situación mundial de la actividad física 2022: resumen ejecutivo [Global status report on physical activity 2022: executive summary]*. Organización Mundial de la Salud, Ginebra, 2022. Licencia: CC BY-NC-SA 3.0 IGO.
 - [46] Plan de Recuperación, Transformación y Resiliencia. ¿qué es la inteligencia artificial (ia)? <https://planderecuperacion.gob.es/noticias/que-es-inteligencia-artificial-ia-prtr>. Último acceso: 13 jul. 2024.
 - [47] Profile.es. Qué es typescript vs javascript. <https://profile.es/blog/que-es-typescript-vs-javascript/>. Último acceso: 19 jul. 2024.
 - [48] Programación Web ISC. Arquitectura de las aplicaciones web. URL: <https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/>, Fecha no especificada. Último acceso: 11 jul. 2024.
 - [49] Prompting Guide. Llama 3: A large multimodal model. <https://www.promptingguide.ai/models/Llama-3>. Último acceso: 9 jul. 2024.
 - [50] Red Hat. What is a rest api? <https://www.redhat.com/es/topics/api/what-is-a-rest-api>. Último acceso: 13 jul. 2024.
 - [51] Jesús Salido. Plantilla LaTeX para Trabajo Fin de Estudios en Ingeniería Informática - UCLM (España). Zenodo, DOI: <https://doi.org/10.5281/zenodo.4561708>, 2019.

- [52] Ken Schwaber and Jeff Sutherland. *La Guía de Scrum: La Guía Definitiva de Scrum: Las Reglas del Juego*. Ken Schwaber and Jeff Sutherland, 2020.
- [53] SEMrush. Html semántico: Guía completa y ejemplos prácticos. <https://es.semrush.com/blog/html-semantico/>. Último acceso: 13 jul. 2024.
- [54] Amazon Web Services. Amazon relational database service (rds). <https://aws.amazon.com/es/relational-database/>, 2023. Último acceso: 15 jul. 2024.
- [55] Amazon Web Services. What are transformers in artificial intelligence? <https://aws.amazon.com/es/what-is/transformers-in-artificial-intelligence/>, 2024. Último acceso: 15 jul. 2024.
- [56] Shifta. Estado del arte de los frameworks de javascript en 2024. <https://weareshifta.com/estado-del-arte-de-los-frameworks-de-javascript-en-2024/>. Último acceso: 22 jul. 2024.
- [57] Tech Riders. Formularios reactivos en angular. <https://techriders.tajamar.es/formularios-reactivos-en-angular/>, 2024. Último acceso: 16 jul. 2024.
- [58] Valgo. Informe divulgativo de la encuesta nacional sobre tendencias fitness en españa 2024. *Valgo Blog*, 2024. Último acceso: 11 jul. 2024.
- [59] Robert Vásquez. Tutorial técnicas básicas y avanzadas de prompt engineering. <https://github.com/robert-ds/tecnicas-prompting/blob/main/tutorial-técnicas-básicas-y-avanzadas-de-prompt-engineering.md>, mayo 2023. Último acceso: 15 jul. 2024.
- [60] Wikipedia. Angular (framework). [https://es.wikipedia.org/wiki/Angular_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework)). Último acceso: 22 jul. 2024.
- [61] Wikipedia. CSS. <https://es.wikipedia.org/wiki/CSS>. Último acceso: 14 jul. 2024.
- [62] Wikipedia contributors. Node.js. <https://es.wikipedia.org/wiki/Node.js>. Último acceso: 20 jul. 2024.
- [63] World Health Organization. Physical activity. <https://www.who.int/es/news-room/fact-sheets/detail/physical-activity>. Último acceso: 9 jul. 2024.
- [64] Xataka. Google gemma: ¿qué es y para qué sirve este nuevo modelo de inteligencia artificial? <https://www.xataka.com/basics/google-gemma-que-que-se-diferencia-sirve-este-nuevo-modelo-inteligencia-artificial>. Último acceso: 9 jul. 2024.

ANEXOS

ANEXO A

Repositorio

A continuación, se proporciona el enlace al repositorio de GitHub donde se encuentra el código fuente del proyecto:

<https://github.com/TamaraRedondo/TrabajoFinGrado>

ANEXO B

Manual de Usuario

En este anexo se presenta el manual de usuario del proyecto, proporcionando una guía completa para el uso de la aplicación desarrollada.

B.1. INICIO DE SESIÓN

En la pantalla de inicio de sesión B.1:

1. Si ya tiene una cuenta, ingrese sus credenciales en los campos correspondientes y haga clic en “Sign in” para acceder a su cuenta.
2. Si no tiene una cuenta, haga clic en “Create an account” para ser dirigido a la página de registro de usuario y crear una nueva cuenta.

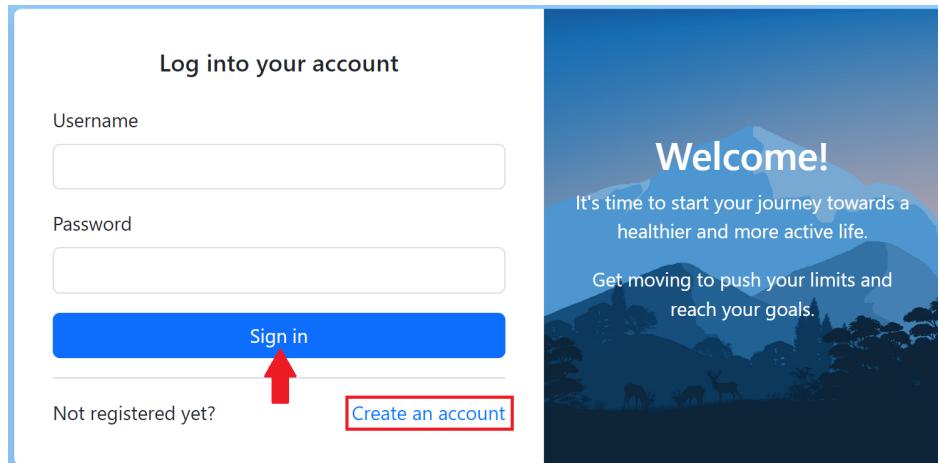


Figura B.1: Inicio de sesión

B.2. REGISTRO DE USUARIO

Para crear una nueva cuenta B.2:

1. Rellene los campos requeridos en la pantalla de registro.
2. Haga clic en el botón “Continue with sign up” para continuar con el registro.

Si desea volver a la pantalla de inicio de sesión, haga clic en “Sign in”.

B.3. REGISTRO DE USUARIO. PARTE 2

Para completar el formulario B.3:

1. Rellene todas las preguntas del formulario.
2. Haga clic en el botón “Next” para avanzar.

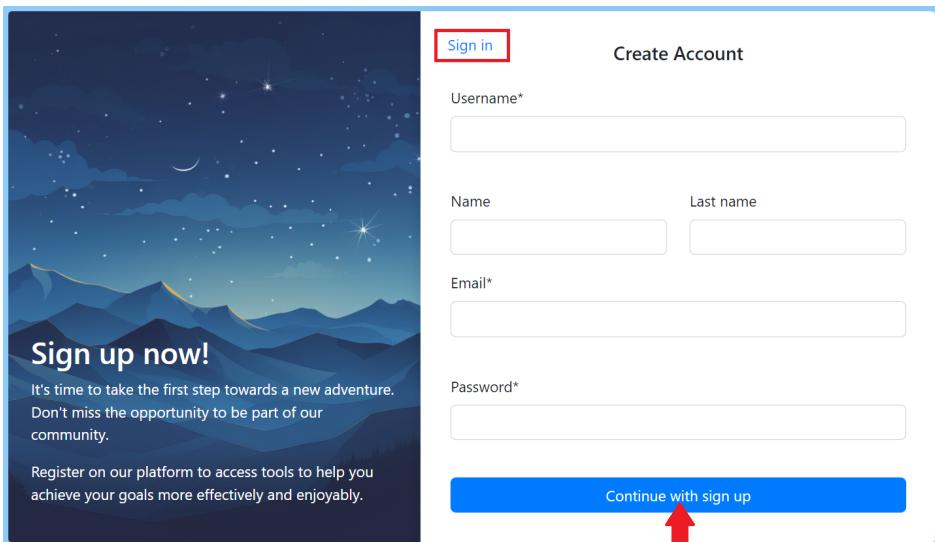


Figura B.2: Registro de usuario

3. En la última pregunta **B.4**, haga clic en “Generate routine” para acceder a la página principal del sitio web.
4. La pagina principal muestra una pantalla de carga hasta que esté disponible la rutina **B.5**.

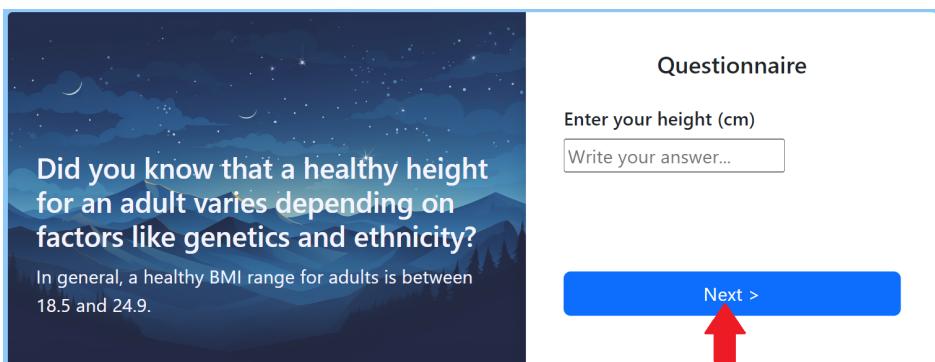


Figura B.3: Primera pregunta del formulario

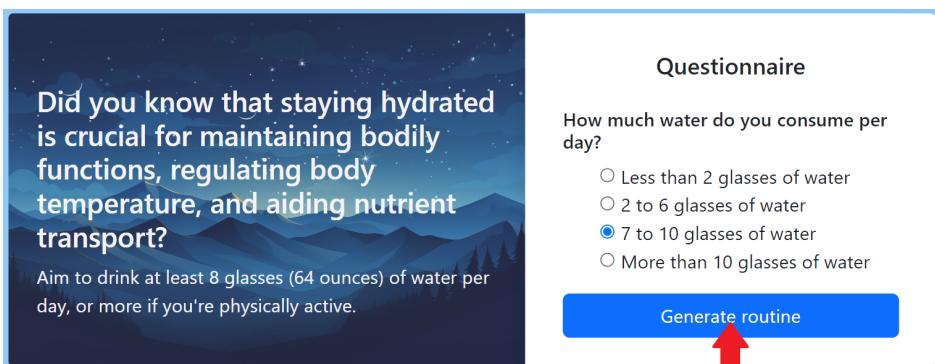


Figura B.4: Última pregunta del formulario

B.4. RUTINA DE EJERCICIOS

La rutina de ejercicios se organiza en tarjetas, cada una corresponde a un ejercicio distinto, como se muestra en la figura **B.6**. Para marcar un ejercicio como completado, haga clic en el botón “Mark as completed” una vez que haya terminado el ejercicio.

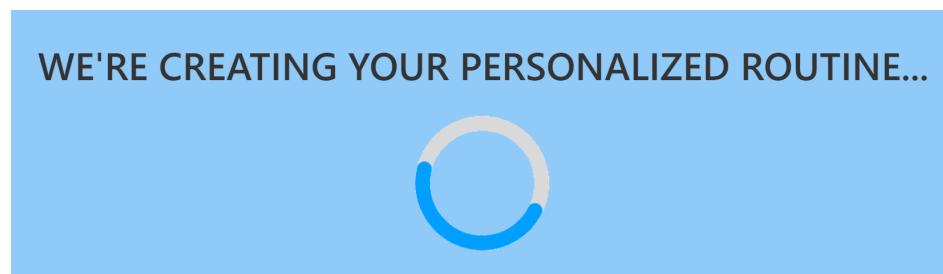


Figura B.5: Pantalla de carga de la generación de la rutina

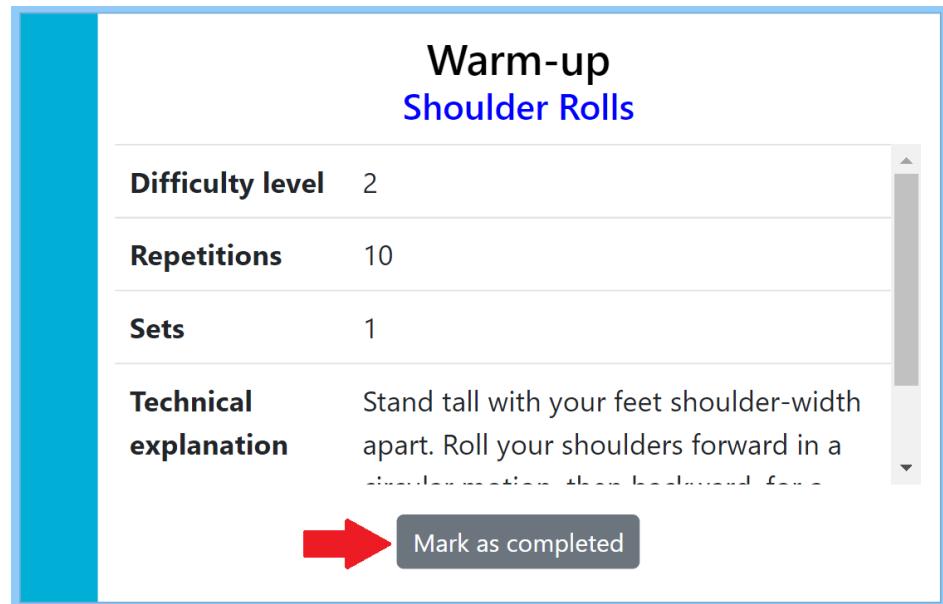


Figura B.6: Disposición de los ejercicios de la rutina

B.5. PERFIL DE USUARIO

En el perfil de usuario se presentan los datos de las dos fases de registro. En las figuras B.7 y B.8 se muestran el perfil de usuario inicialmente y el perfil de usuario en modo editar, respectivamente. Para editar datos del formulario o generar nuevas rutinas:

1. Puede editar cada sección haciendo clic en “Edit” o “Edit Profile”.
2. Para guardar los cambios después de realizar una modificación, haga clic en “Save Changes” o “Save”. Si desea cancelar los cambios, haga clic en “Cancel”.
3. Puede actualizar la rutina después de modificar algún dato, haga clic en “Update the following routines”.
4. Aparece una alerta que debe aceptar si decide actualizar las rutinas. Por el contrario, pulse cancelar.

B.6. ANÁLISIS PERSONALIZADO

Debajo del perfil de usuario se encuentra un análisis de datos personalizado B.9. Si modifica previamente algún dato del perfil, tiene la opción de actualizar el análisis pulsando “Update the data analysis”. El análisis se muestra en tarjetas:

1. Para ver más detalles, haga clic en “More Info” en la tarjeta correspondiente.
2. Para volver a la vista inicial, haga clic en “Hide details”.

The screenshot shows a user profile for "Maria Rodriguez". On the left, there's a sidebar with her name, a blue profile icon, and her contact information: Username: Maria1, Email: mariar.s@gmail.com. Below this is a "General Information" section with Height (cm): 1,70, Weight (kg): 60, Age Range: 18-25, and Gender: Female. At the bottom of this sidebar is a blue "Edit Profile" button with a red arrow pointing to it. The main content area has two tables. The first table, under "Key Aspects", lists Body Type (Endomorph), Physical Activity Level (Beginner), Current Injury (Muscle strain.), Injury Location (Neck), Regular Exercise Location (Home), Days Available to Train per Week (4 days per week), and Available Session Time (90 minutes). The second table, under "Lifestyle", lists Goal (Maintain a healthy routine), Additional Goal (Form physical habits), Walking Time per Day (1-2 hours), Sleeping Time per Day (Between 5 and 6 hours), and Daily Water Intake (7 to 10 glasses of water). Each row in both tables has an "Edit" button to its right. A note at the bottom says "Has any habit changed? Update the following routines here:" followed by a blue "Update the following routines" button with a red arrow pointing to it.

Figura B.7: Perfil de usuario

This screenshot shows the user profile being edited. The left sidebar contains the same information as Figure B.7. The main area shows the "Selected Option" table with the following data: Body Type (Mesomorph), Physical Activity Level (Beginner), Current Injury (Muscle strain.), Injury Location (Neck), Regular Exercise Location (Home), Days Available to Train per Week (4 days per week), and Available Session Time (90 minutes). The "Action" column for each row includes an "Edit" button and a "Save" or "Cancel" button. A black confirmation dialog box is overlaid on the top left, containing the text "Are you sure? Your routines for the next month will change." with "Aceptar" and "Cancelar" buttons, and a red arrow points to the "Aceptar" button. At the bottom of the main area is a note "Has any habit changed? Update the following routines here:" followed by a blue "Update the following routines" button. A red arrow also points to this button. At the very bottom of the sidebar is a green "Save Changes" button with a red arrow pointing to it.

Figura B.8: Modificación del perfil de usuario y actualización de las rutinas

B.7. CALENDARIO

En la sección de *Objectives*, se encuentra un calendario que organiza las rutinas realizadas y las rutinas por realizar, utilizando diferentes colores para cada tipo B.10. Para visualizar las rutinas disponibles:

1. Si hace clic en una fecha donde no hay rutinas realizadas, se muestra un mensaje que indica "You don't have any routines for that day".
2. Si hace clic en una fecha donde se ha realizado una rutina, se muestra la rutina correspondiente para esa fecha en lugar del mensaje anterior B.11.

B.8. LOGROS

En la sección de *Achievements*, se muestran los logros y trofeos obtenidos B.12:

1. Los logros o trofeos que están coloreados indican que han sido desbloqueados.
2. Los logros o trofeos en color gris aún no se han desbloqueado.

PERSONALIZED ANALYSIS

Based on your lifestyle data, a detailed analysis of your habits and **potential improvements** has been conducted. Here you will find tailored recommendations and strategies to enhance your well-being and achieve your goals more effectively.

You can update the analysis by clicking the following button:

 [Update the data analysis](#)

Body Type



Your BMI falls within the normal weight range (18.5 - 24.9), indicating a healthy body weight. It suggests that you are maintaining a good balance between your height and weight.

[More Info](#)

Goals



The goal is to achieve physical fitness and improve overall health, with an additional aim to enhance feelings of well-being and vitality.

[More Info](#)

Walking

Walking is a great form of exercise that can contribute to your overall health and well-being. It's low-impact, accessible, and can be easily incorporated into your daily routine.

Try to increase your walking duration or intensity gradually to challenge your cardiovascular system and improve endurance.

 [Hide details](#)

Figura B.9: Disposición y actualización del análisis personalizado

< June, 2024 >

Mon	Tue	Wed	Thu	Fri	Sat	Sun
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Date: Jun 15, 2024

█ Completed Routine █ Pending Routine

You dont have any routines for that day.

Figura B.10: Disposición del calendario

SELECTED ROUTINE

Here are your exercises completed on Jun 4, 2024, with all sets and repetitions, the technical explanation, and their difficulty level from 1 to 10. Here you can review past exercises and progress further.

JUMPING JACKS Sets/Repetitions: 1 x 15 Difficulty Level: 1 Technical Explanation: Stand with your feet together and arms at your sides. Jump to spread your legs while raising your arms above your head. Jump again to return to the starting position.	SQUATS Sets/Repetitions: 3 x 10 Difficulty Level: 2 Technical Explanation: Stand with your feet shoulder-width apart. Lower your body as if you were sitting back into a chair, keeping your chest upright and your knees behind your toes. Return to standing position.	PUSH-UPS Sets/Repetitions: 3 x To failure Difficulty Level: 2 Technical Explanation: Assume a plank position with hands shoulder-width apart. Lower your body until your chest nearly touches the floor, then push back up to the starting position.
DUMBBELL LUNGES Sets/Repetitions: 3 x 10 Difficulty Level: 3 Technical Explanation: Hold dumbbells in each hand, step forward with one leg and lower your hips until both knees are bent at about a 90-degree angle. Push back to the starting position and repeat with the other leg.	BENT-OVER ROWS Sets/Repetitions: 3 x 12 Difficulty Level: 3 Technical Explanation: Hold a dumbbell in each hand, bend at your hips and knees. Lower your torso until it's almost parallel to the floor. Pull the weights to your sides, keeping your elbows close to your body.	PLANK Duration: 1 minute Difficulty Level: 2 Technical Explanation: Get into a push-up position on the floor. Hold your body in a straight line from head to heels, engaging your core muscles. Hold this position for 1 minute.

Figura B.11: Visualización de una rutina completada

3. Al hacer clic en cada logro o trofeo, se muestra una descripción detallada **B.13** que incluye los posibles beneficios de salud asociados y su estado, ya sea bloqueado o desbloqueado.

B.9. CHAT

[h] Para interactuar con el chat dirígete a la esquina inferior de la barra lateral y haga clic en Chat **B.14**:

1. Escriba su pregunta o mensaje en el campo “*Type a message...*”.
2. Presiona “*Enter*”.
3. Espera la respuesta del asistente.

B.10. CERRAR SESIÓN

Se puede salir de la aplicación en cualquier momento haciendo clic en “*Logout*” en la barra de navegación lateral.

Completed Routines		Points Earned	
	Complete 5 routines		Earn 50 points
	Complete 10 routines		Earn 100 points
	Complete 20 routines		Earn 200 points
	Complete 50 routines		Earn 500 points
	Complete 100 routines		Earn 1000 points

Consistency		Routines by Time	
	24 hours of activity		Complete a routine AM
	One week of activity		Complete a routine PM
	One month of activity		Complete 20 routines AM
	Five months of activity		Complete 20 routines PM
	Ten months of activity		Complete all routines by AM/PM

Figura B.12: Disposición de los logros y trofeos

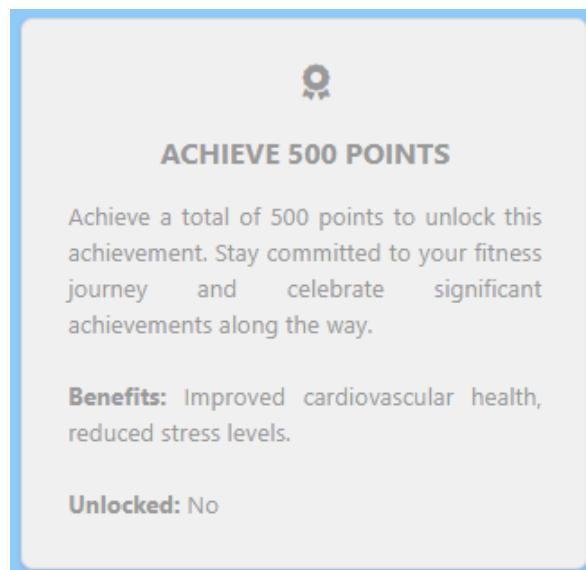


Figura B.13: Descripción asociada a un logro

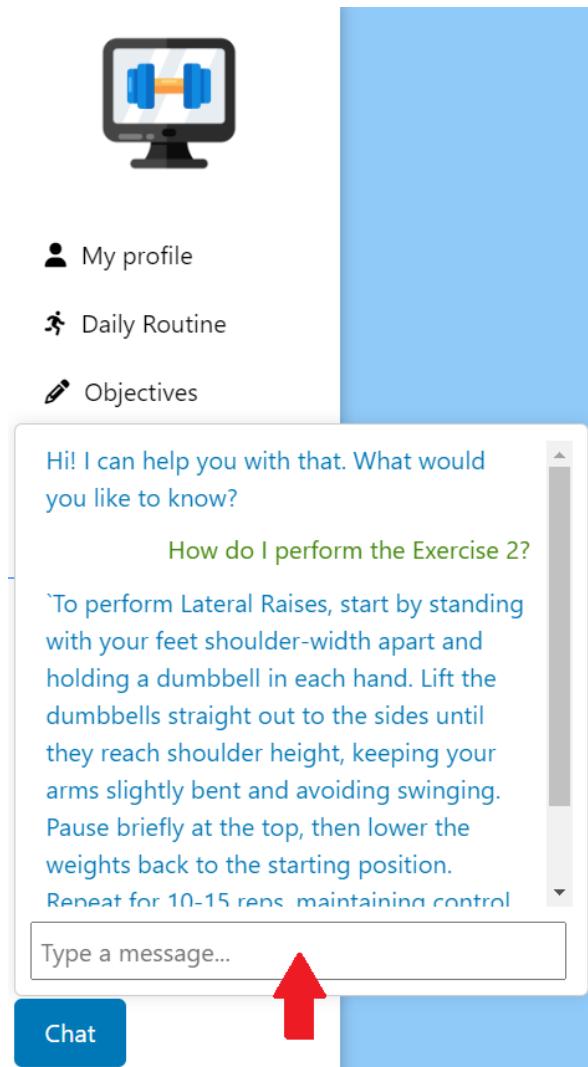


Figura B.14: Disposición del chat