

Desarrollo de aplicación móvil Android: IntelliFit

Diego Esteban Guaña Logroño

Grado de Ingeniería Informática

Desarrollo de aplicaciones para dispositivos móviles Android (DADMA)

David Escuer Latorre

Jordi Almirall López

Carles Garrigues Olivella



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-

SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Desarrollo de aplicación móvil Android: IntelliFit
Nombre del autor:	Diego Esteban Guaña Logroño
Nombre del consultor/a:	David Escuer Latorre Jordi Almirall López
Nombre del PRA:	Carles Garrigues Olivella
Fecha de entrega (mm/aaaa):	06/2023
Titulación:	Grado de Ingeniería Informática
Área del Trabajo Final:	Desarrollo de aplicaciones para dispositivos móviles Android (DADMA)
Idioma del trabajo:	Castellano
Palabras clave	Android, ejercicio, inteligencia artificial.
Resumen del Trabajo:	<p>El presente trabajo tiene como objetivo desarrollar una aplicación móvil para guiar a los usuarios en la correcta ejecución de ejercicios físicos utilizando técnicas de inteligencia artificial, específicamente redes neuronales convolucionales (CNN). Esta aplicación permite a los usuarios seleccionar entre una variedad de entrenamientos y, mediante el uso de la cámara del dispositivo móvil, compara en tiempo real la ejecución del usuario con la forma correcta de realizar los ejercicios, detectando posibles movimientos incorrectos para prevenir lesiones.</p> <p>La aplicación se dirige a usuarios interesados en hacer ejercicio físico de manera segura y efectiva. Se ha realizado entrevistas con usuarios con diferentes niveles de experiencia en el ejercicio físico, así como con un</p>

entrenador personal, para comprender sus necesidades y expectativas. Se han identificado diferentes perfiles de usuario y se han creado escenarios de uso para garantizar que la aplicación cumpla con sus requerimientos.

La metodología utilizada se basa en el enfoque de Diseño Centrado en el Usuario, que involucra la etapa de indagación para recopilar información sobre los usuarios y sus necesidades, la fase de diseño conceptual para generar escenarios de uso y flujos de interacción, y la etapa de prototipado para desarrollar prototipos de alta fidelidad y evaluar su usabilidad.

Como resultado del trabajo, se ha desarrollado un prototipo funcional de la aplicación que incluye las funcionalidades clave identificadas durante la fase de indagación. Se ha evaluado el prototipo con usuarios y se han realizado mejoras basadas en el *feedback* recibido. Aunque no se ha podido llevar a cabo una evaluación exhaustiva con múltiples usuarios debido a limitaciones de tiempo, se han obtenido conclusiones importantes sobre la viabilidad y usabilidad de la aplicación.

En resumen, este trabajo ha permitido diseñar y desarrollar una aplicación móvil innovadora que utiliza inteligencia artificial. Si bien quedan aspectos pendientes de explorar en futuras versiones, este trabajo sienta las bases para el desarrollo de una aplicación móvil efectiva en el ámbito del ejercicio físico.

Abstract:

The objective of this work is to develop a mobile application to guide users in the correct execution of physical exercises using artificial intelligence techniques, specifically convolutional neural networks (CNN). This application allows users to select from a variety of workouts and, using the mobile device's camera, compares the user's execution in real-time with the correct form of the exercises, detecting possible incorrect movements to prevent injuries. The application is targeted towards users interested in safely and effectively exercising. Interviews have been conducted with users with different levels of exercise experience, as well as with a personal trainer, to understand their needs and expectations. Different user profiles have been identified, and usage

scenarios have been created to ensure that the application meets their requirements. The methodology used is based on the User-Centered Design approach, which involves the inquiry stage to gather information about users and their needs, the conceptual design phase to generate usage scenarios and interaction flows, and the prototyping stage to develop high-fidelity prototypes and evaluate their usability. As a result of the work, a functional prototype of the application has been developed, including the key functionalities identified during the inquiry phase. The prototype has been evaluated with users, and improvements have been made based on the feedback received. While a comprehensive evaluation with multiple users could not be conducted due to time constraints, important conclusions have been drawn regarding the feasibility and usability of the application. In summary, this work has allowed for the design and development of an innovative mobile application that utilizes artificial intelligence. While there are aspects that remain to be explored in future versions, this work lays the foundation for the development of a effective mobile application in the field of physical exercise.

Índice

1. INTRODUCCIÓN	1
1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO	1
1.2 OBJETIVOS DEL TRABAJO.....	3
1.3 ENFOQUE Y MÉTODO SEGUIDO.....	5
1.4 PLANIFICACIÓN DEL TRABAJO.....	6
1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS	8
1.6 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA	9
2. DISEÑO Y ARQUITECTURA.....	10
2.1 DISEÑO CENTRADO EN EL USUARIO.	10
2.1.1. USUARIOS Y CONTEXTOS DE USO	10
2.1.1.1 Análisis Competitivo.....	12
2.1.1.2 Entrevistas en profundidad.....	15
2.2 DISEÑO CONCEPTUAL	18
2.2.1 Escenarios de uso	18
2.2.2 Flujo de interacción	19
2.3 PROTOTIPADO	20
2.3.1 Sketches.....	21
2.3.2 Prototipo horizontal de alta fidelidad	24
2.4 EVALUACIÓN	28
2.4.1 Preguntas referentes al usuario.....	28
2.4.2 Tareas a realizar por el usuario.....	29
2.4.3 Mejoras luego de la evaluación.....	29
3. DISEÑO TÉCNICO	32
3.1 DEFINICIÓN DE CASOS DE USO.....	32
3.1.1 DIAGRAMA DE CASOS DE USO	32
3.1.2 Listado DE CASOS DE USO	33
3.2 ARQUITECTURA.....	40
3.2.1 MODELO DE DATOS	42
4. IMPLEMENTACIÓN.....	43
4.1 TECNOLOGIA USADA.....	43
4.1.2. lenguaje de programación	43
4.1.1. Entorno de desarrollo integrado (IDE).....	43
4.1.3. Librerías	43
4.1.4. Versiones de android.....	44

4.1.5. Variantes de compilación	46
4.1.6. Sistema de control de versiones	47
4.2. PATRONES Y BUENAS PRACTICAS	48
4.3 ENTRENAMIENTO DE MODELO	49
4.4 ESTADO ACTUAL DEL PROYECTO.....	51
5. PRUEBAS.....	52
5.1 PLAN DE PRUEBAS.....	52
5.2 PRUEBAS UNITARIAS.....	52
6. CONCLUSIONES	55
4. GLOSARIO	56
5. BIBLIOGRAFÍA.....	58
6. ANEXOS	59
ANEXO 1. ENTREVISTAS.....	59

Lista de figuras

Ilustración 1: Captura Artifit	2
Ilustración 2. Home Workout	12
Ilustración 3 : Virtuagym Fitness	13
Ilustración 4: ArtiFit	14
Ilustración 5: Gym Workout Plan	14
Ilustración 6. Flujo de interacción	20
Ilustración 7. Sketch pantalla de carga	21
Ilustración 8. Sketch pantalla de inicio de sesión	22
Ilustración 9. Sketch pantalla crear usuario	22
Ilustración 10. Sketch pantalla principal	23
Ilustración 11. Sketch pantalla detalles del entrenamiento	23
Ilustración 12. Sketch pantalla de ejercicio	24
Ilustración 13. Pantalla de carga	25
Ilustración 14. Pantalla inicio de sesión	25
Ilustración 15. Pantalla de registro	26
Ilustración 16. Pantalla de entrenamientos	26
Ilustración 17. Pantalla de entrenamientos	27
Ilustración 18. Pantalla de ejercicio	27
Ilustración 19. Pantalla menú de navegación	30
Ilustración 20. Pantalla de ajustes	31
Ilustración 21. Diagrama de casos de uso	32
Ilustración 22. Diagrama de una arquitectura	41
Ilustración 23. Distribución de versiones API	45
Ilustración 24. Testing SaveCapture	53
Ilustración 25. Testing deleteCapture	53
Ilustración 26. Testing getCaptures	54
Ilustración 27. Resultado testing	54

1. INTRODUCCIÓN

1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO

Realizar ejercicio físico regularmente es uno de los pilares fundamentales para mantener una buena salud y prevenir enfermedades crónicas como la obesidad, la diabetes, enfermedades cardiovasculares, entre otras. El sedentarismo y la falta de actividad física son factores de riesgo para la salud que se han convertido en un problema global, especialmente en los países desarrollados. Además, la pandemia mundial ha aumentado aún más el sedentarismo debido a las restricciones de movilidad y el cierre de instalaciones deportivas y gimnasios.

Para muchos, hacer ejercicio no es una tarea fácil, especialmente cuando no tienen conocimientos suficientes sobre la realización de estos o tienen dificultades para motivarse. En este sentido, la tecnología puede jugar un papel fundamental para realizar el ejercicio físico de manera correcta, segura y efectiva. La implementación de aplicaciones móviles y dispositivos electrónicos que monitorizan el ejercicio se ha vuelto cada vez más popular en los últimos años, lo que ha llevado a una mayor accesibilidad y comodidad para los usuarios que quieren mejorar su condición física.

Existe un gran catálogo de aplicaciones con rutinas que permiten al usuario hacer ejercicio en cualquier lugar y momento del día, sin necesidad de acudir a un gimnasio o contratar a un entrenador personal. Sin embargo, muchas de estas aplicaciones se limitan a ofrecer videos o imágenes de los ejercicios a realizar, sin prestar atención a la correcta ejecución.

Actualmente solo existe una aplicación que realiza un seguimiento de la técnica del entrenamiento llamado ArtiFit, sin embargo, esta aplicación solo está disponible para dispositivos con sistema operativo iOS.

ArtiFit: Es un entrenador personal en tu teléfono. Utilizando tu cámara, AI monitoriza tu entrenamiento, te da información sobre la técnica del ejercicio y realiza un seguimiento de tu progreso.

[Artifit – AI Personal Trainer on the App Store \(apple.com\)](https://apple.com/artifit)

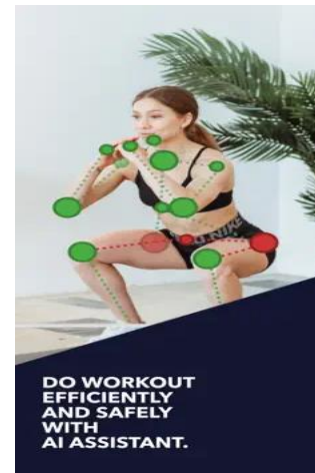


Ilustración 1: Captura Artifit

Es importante destacar que la plataforma Android cuenta con una gran cantidad de usuarios en todo el mundo, superando incluso a iOS en número de dispositivos activos. Por lo tanto, desarrollar una aplicación similar para Android sería fundamental para llegar a un público más amplio y tener un mayor impacto en el público objetivo. Además, existen muchas personas que prefieren utilizar dispositivos Android por diversos motivos, como, por ejemplo, su precio más accesible o su mayor variedad de opciones. Por lo tanto, desarrollar esta aplicación en la plataforma Android ampliará el alcance y accesibilidad de la tecnología a un grupo importante de usuarios.

A continuación, se presentan algunas similitudes y diferencias entre la aplicación ArtiFit y nuestra propuesta:

Similitudes:

- Ambas aplicaciones tienen como objetivo guiar a los usuarios con la correcta ejecución de los ejercicios utilizando la inteligencia artificial.
- Ambas proporcionan retroalimentación en tiempo real al usuario sobre la postura y técnica durante el ejercicio.

Diferencias:

- IntelliFit permitirá a los usuarios crear y guardar rutinas personalizadas a partir de los ejercicios disponibles.
- Nuestra propuesta implementa un sistema de notificaciones para motivar a los usuarios y recordarles cuando deben realizar sus ejercicios, mientras que la aplicación solo disponible en iOS parece no tener esta funcionalidad.

En definitiva, este proyecto tiene como objetivo desarrollar una aplicación para guiar a los usuarios a realizar ejercicios de forma correcta utilizando la cámara del móvil y la visión por computadora para detectar si el usuario realiza el ejercicio de manera adecuada. Esto no solo ayudará a mejorar la motivación y confianza de los usuarios, sino que también mejorará la seguridad al evitar posibles lesiones.

1.2 OBJETIVOS DEL TRABAJO

El objetivo general de este trabajo es desarrollar una aplicación Android para guiar a los usuarios en la correcta ejecución de los ejercicios físicos mediante la utilización de técnicas de inteligencia artificial, específicamente de redes neuronales convolucionales (CNN). La aplicación permitirá a los usuarios seleccionar entre una variedad de ejercicios, y mediante el uso de la cámara de su dispositivo móvil, la aplicación comparará en tiempo real la ejecución del usuario con la correcta realización del ejercicio, detectando si se están realizando movimientos inadecuados para evitar posibles lesiones.

Objetivos funcionales:

1. Desarrollar una aplicación móvil para guiar a los usuarios a realizar ejercicios de forma correcta utilizando la inteligencia artificial.
2. Integrar la funcionalidad de detección de postura en la aplicación móvil para proporcionar retroalimentación en tiempo real al usuario.
3. Permitir a los usuarios crear y guardar rutinas personalizadas de ejercicios y hacer un seguimiento del progreso de su entrenamiento.
4. Implementar un sistema de notificaciones para motivar a los usuarios y recordarles cuándo deben realizar sus ejercicios.

5. Realizar pruebas exhaustivas en diferentes dispositivos Android y con diferentes usuarios para asegurar el correcto funcionamiento de la aplicación.

Objetivos no funcionales:

1. Garantizar la privacidad y seguridad de los datos de los usuarios, asegurando la confidencialidad de la información personal.
2. Ser capaz de manejar una gran cantidad de usuarios simultáneamente, sin comprometer su rendimiento o estabilidad.
3. La aplicación debe ser intuitiva y fácil de usar para todos los usuarios, independientemente de su nivel de experiencia con aplicaciones móviles.
4. La aplicación debe ser fácil de mantener y actualizar, permitiendo la incorporación de nuevas funcionalidades y corrección de errores de manera ágil y eficiente.
5. La aplicación debe ser rápida y eficiente en la ejecución de las tareas requeridas, sin tiempos de espera prolongados o retrasos en la respuesta a las interacciones del usuario.
6. La aplicación debe ser adaptable a diferentes contextos y situaciones de uso, como por ejemplo con conexión a redes móviles, redes inalámbricas, y sin ningún tipo de conexión a red, funcionando de forma correcta en todos los casos.

1.3 ENFOQUE Y MÉTODO SEGUIDO

Para llevar a cabo el proyecto, se seguirá la estrategia de desarrollar un producto nuevo ya que se considera la más apropiada porque se busca ofrecer una solución innovadora y específica a la necesidad identificada. Además, desarrollar un producto nuevo permitirá adaptar la aplicación a las necesidades y preferencias de los usuarios, teniendo en cuenta aspectos como la usabilidad, la interfaz gráfica, etc.

El enfoque metodológico seguido será el modelo clásico en cascada, que permitirá llevar a cabo una planificación detallada y secuencial de las diferentes fases del proyecto, desde la investigación inicial hasta la puesta en marcha y evaluación final del producto. Esto garantizará un enfoque estructurado y ordenado en el desarrollo de la aplicación, asegurando la calidad y eficiencia del proceso.

El entorno de desarrollo elegido para la creación de la aplicación móvil es Android Studio, ya que es el IDE oficial de desarrollo de aplicaciones móviles para Android, proporcionado por Google. Esta herramienta incluye un editor de código, un depurador, un emulador de dispositivos móviles y muchas otras características útiles para el desarrollo de aplicaciones móviles.

Además, la aplicación se desarrollará utilizando el lenguaje de programación Kotlin, ya que es un lenguaje moderno, conciso, seguro y que en la actualidad se ha convertido en el lenguaje de programación preferido para el desarrollo de aplicaciones móviles en Android.

Para facilitar el proceso de desarrollo, se utilizarán las librerías Jetpack de Android, estas librerías proporcionan un conjunto de componentes y herramientas para simplificar el desarrollo de aplicaciones móviles, como el manejo de fragmentos, la navegación, el manejo de datos, la inyección de dependencias, entre otros.

Para conseguir el objetivo principal de la aplicación utilizaremos la biblioteca Movenet de TensorFlow para obtener las características de la postura del usuario durante el ejercicio y con estos datos entrenar un modelo de red neuronal convolucional (CNN) para detectar si el usuario está realizando el ejercicio de manera adecuada.

En cuanto a la gestión de usuarios, base de datos, notificaciones, se creará un *backend* en Python utilizando FastAPI, este es un *framework* web moderno y rápido para crear API REST en Python, y permite construir aplicaciones escalables y de alto rendimiento en poco tiempo. Con esta API se podrán realizar todas las operaciones de gestión de usuarios y almacenamiento de datos necesarias para el correcto funcionamiento de la aplicación.

Para almacenar y distribuir los modelos personalizados de TensorFlow se utilizará Firebase, ya que permite alojar modelos personalizados de aprendizaje automático en la nube y procesarlos en dispositivos móviles de forma dinámica, evitando la gestión y descarga manual de los mismos, lo cual nos permite actualizar y mejorar continuamente los modelos sin tener que preocuparse por actualizar manualmente las aplicaciones de los usuarios.

1.4 PLANIFICACIÓN DEL TRABAJO

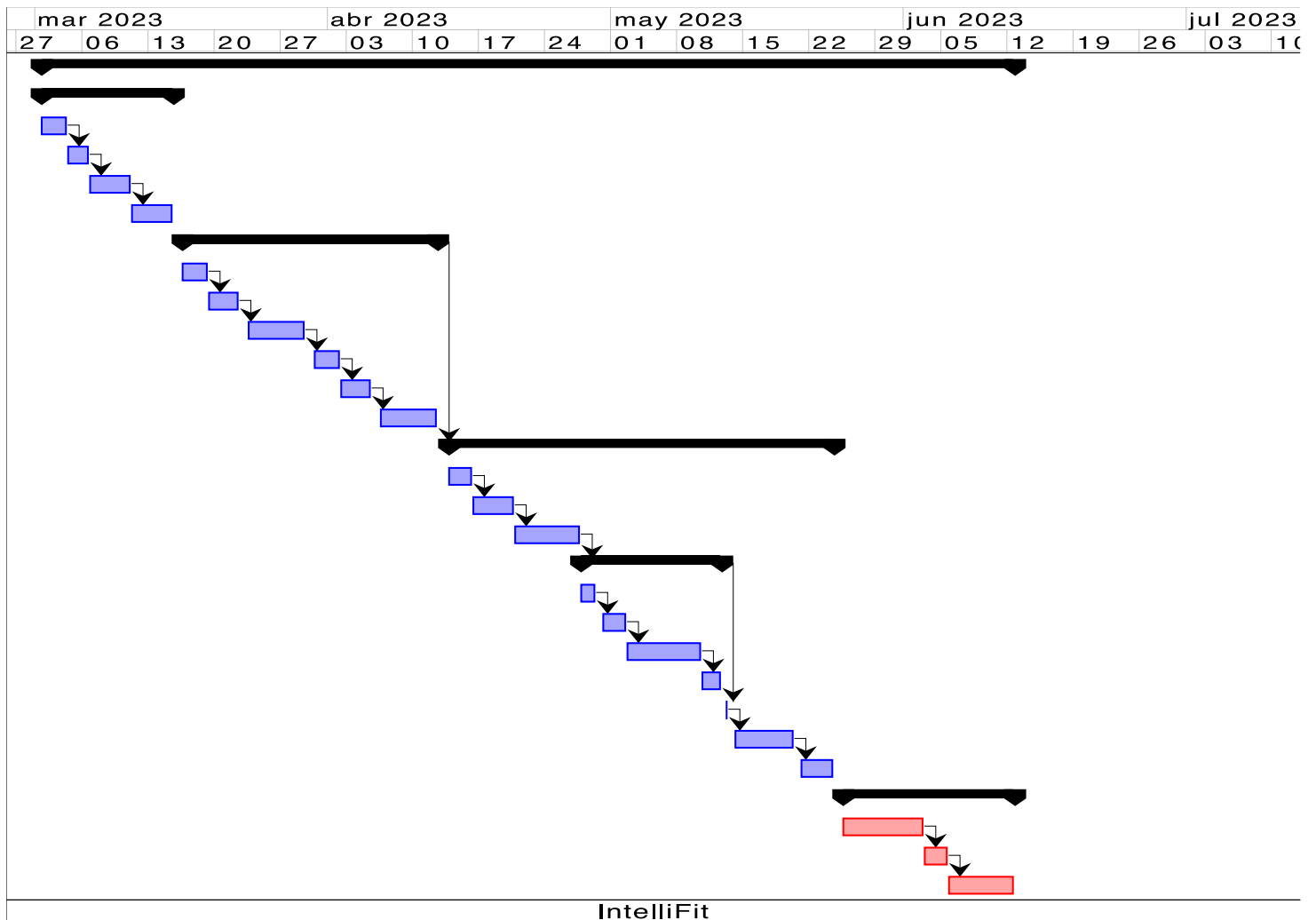
Para establecer la planificación temporal se pretende hacer una correcta estimación y un seguimiento posterior. El proyecto se enmarca en la asignatura de trabajo final de grado, la cual tiene asignados 12 ECTS, lo que se corresponde con una carga lectiva de aproximadamente 300 horas. Cabe mencionar que se está compaginando este proyecto con mi trabajo actual de desarrollador de lunes a viernes de 8:00 a 17:30, y como la duración del proyecto será desde el 1 de marzo hasta el 16 de junio de 2023, se estima una dedicación semanal de 20 horas con la siguiente carga de trabajo:

- 2 horas diarias de lunes a viernes.
- 5 horas diarias los sábados y domingos

La planificación temporal se ha representado mediante un diagrama de Gantt, en el que se han establecido las fechas de inicio y fin de cada tarea, y se han identificado los hitos parciales correspondientes a las fechas de entrega de las PECs. Esto permitirá una correcta estimación y seguimiento del proyecto, identificando posibles desviaciones en el logro de los objetivos parciales. Este diagrama se ha elaborado utilizando el software gratuita Project Libre, haciendo uso de esta herramienta nos ha permitido realizar una planificación efectiva y un seguimiento riguroso del proyecto.

	Nombre	Trabajo	Inicio	Terminado
1	Proyecto IntelliFit	299 horas	1/03/23 18:00	12/06/23 19:00
2	PEC1: Plan de Trabajo	41 horas	1/03/23 18:00	15/03/23 20:00
3	Contexto y justificación	8 horas	1/03/23 18:00	4/03/23 10:00
4	Objetivos	8 horas	4/03/23 11:00	6/03/23 19:00
5	Enfoque y método utilizado	10 horas	6/03/23 19:00	11/03/23 9:00
6	Planificación y riesgos	15 horas	11/03/23 9:00	15/03/23 20:00
7	PEC2: Diseño y arquitectura	85 horas	16/03/23 18:00	12/04/23 19:00
8	Usuarios y contexto de uso	10 horas	16/03/23 18:00	19/03/23 9:00
9	Diseño conceptual	10 horas	19/03/23 9:00	22/03/23 20:00
10	Prototipado	20 horas	23/03/23 18:00	29/03/23 20:00
11	Evaluación	10 horas	30/03/23 18:00	2/04/23 9:00
12	Definición de los casos de uso	10 horas	2/04/23 9:00	5/04/23 20:00
13	Diseño de la arquitectura	25 horas	6/04/23 18:00	12/04/23 19:00
14	PEC3: Implementación	118 horas	13/04/23 19:00	24/05/23 19:00
15	Estructura básica	10 horas	13/04/23 19:00	16/04/23 10:00
16	Inicialización y aceptación de permisos	10 horas	16/04/23 10:00	20/04/23 19:00
17	Inicio de sesión de usuario	20 horas	20/04/23 19:00	27/04/23 19:00
18	Casos de uso y operaciones	43 horas	27/04/23 19:00	12/05/23 20:00
19	Listado de rutinas	8 horas	27/04/23 19:00	29/04/23 13:00
20	Detalle de las rutinas	8 horas	30/04/23 8:00	2/05/23 19:00
21	Ejecución del ejercicio	22 horas	2/05/23 19:00	10/05/23 19:00
22	Perfil del usuario	5 horas	10/05/23 19:00	12/05/23 20:00
23	Tareas en segundo plano	5 horas	13/05/23 8:00	13/05/23 13:00
24	Desarrollo del backend	20 horas	14/05/23 8:00	20/05/23 13:00
25	Testing	10 horas	21/05/23 8:00	24/05/23 19:00
26	Entrega final	55 horas	25/05/23 18:00	12/06/23 19:00
27	Memoria	25 horas	25/05/23 18:00	3/06/23 9:00
28	Manual de usuario	10 horas	3/06/23 9:00	5/06/23 19:00
29	Presentación	20 horas	5/06/23 19:00	12/06/23 19:00

IntelliFit



Es importante mencionar que, debido a la naturaleza del proyecto, el número de ciclos y funcionalidades desarrolladas en cada ciclo puede variar, por lo que se deberá realizar una revisión y ajuste periódico de la planificación temporal.

1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS

Al finalizar el proyecto, se generarán los siguientes entregables:

1. Una aplicación móvil compilada para su instalación con extensión apk.
2. El código fuente de la aplicación.
3. Un breve manual de usuario que incluirá las principales funcionalidades de la aplicación.
4. La memoria de proyecto que detalla todo el proceso de desarrollo.
5. Una presentación en formato de vídeo.

1.6 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA

La memoria del proyecto está estructurada en varios capítulos que describen el trabajo realizado en cada etapa del proyecto:

- **Diseño:** En este capítulo se describen las decisiones tomadas en la etapa de diseño de la aplicación móvil, incluyendo la identificación de los requisitos, el diseño conceptual, la arquitectura y los casos de uso. También se presentan los mockups y prototipos desarrollados y se explica cómo se han aplicado los principios de diseño centrado en el usuario.
- **Implementación:** En este apartado se describe la implementación de la aplicación móvil, detallando las tecnologías y herramientas utilizadas, la estructura de la aplicación y el desarrollo de las diferentes funcionalidades.
- **Pruebas:** En este capítulo se describe el proceso de pruebas realizadas para asegurar la calidad de la aplicación móvil. Estas pueden ser pruebas unitarias, de integración y de aceptación, y se presentan los resultados obtenidos.
- **Conclusiones:** En este apartado se presentan las conclusiones obtenidas tras la realización del proyecto, incluyendo una valoración de los resultados obtenidos, los retos a los que se ha enfrentado, las lecciones aprendidas y las posibles líneas de trabajo futuro.
- **Glosario:** En este capítulo se recogen los términos y conceptos clave utilizados en la memoria y se proporciona una breve definición de cada uno de ellos.
- **Bibliografía:** Se recopilan las referencias bibliográficas consultadas durante la realización del proyecto.

- **Anexos:** En este apartado se incluyen los documentos adicionales relevantes para el proyecto, como por ejemplo el manual de usuario.

2. DISEÑO Y ARQUITECTURA

2.1 DISEÑO CENTRADO EN EL USUARIO.

El diseño centrado en el usuario (DCU) es un enfoque de diseño que se centra en las necesidades, deseos y habilidades de los usuarios finales del producto. El objetivo principal del DCU es crear productos que sean fáciles de usar y satisfagan las necesidades de los usuarios. El proceso de diseño centrado en el usuario se divide en cuatro etapas:

- Usuarios y contexto de uso (Análisis)
- Diseño conceptual (Diseño)
- Prototipado (Diseño)
- Evaluación (Evaluación)
-

2.1.1. USUARIOS Y CONTEXTOS DE USO

El diseño centrado en el usuario se basa en la idea de que para que un producto o servicio tenga éxito, se deben considerar las necesidades, deseos y limitaciones del usuario final. Por lo que, en la fase inicial de investigación del proceso de diseño, es importante identificar los usuarios y los contextos de uso. Los usuarios son las personas que interactuarán con el producto o servicio, y los contextos de uso se refieren a las situaciones en las que los usuarios utilizarán el producto o servicio.

Para identificar a los usuarios, se pueden utilizar diferentes métodos de investigación como la observación, entrevistas, encuestas y análisis competitivo.

El análisis competitivo consiste en evaluar productos o servicios similares a los que se están desarrollando y analizar sus fortalezas y debilidades. De esta manera, se pueden identificar características o funcionalidades que son

importantes para los usuarios y asegurar que el producto o servicio desarrollado cumpla con sus necesidades y expectativas.

Otro método de investigación útil es el método de diario, en el que se pide a los usuarios que registren sus experiencias y pensamientos sobre el uso de un producto o servicio en su vida cotidiana. Las entrevistas y las encuestas también son útiles para recopilar información sobre los usuarios, como sus necesidades, hábitos, preferencias y requisitos.

El perfilado de usuarios es una técnica útil para crear una imagen detallada de los usuarios. Se trata de una descripción detallada de los usuarios que incluye información sobre sus antecedentes, características demográficas, necesidades y objetivos. Esto puede ayudar a asegurar que el producto o servicio sea relevante y útil para los usuarios.

Para esta fase del proyecto, hemos elegido dos métodos de indagación: el análisis competitivo y las entrevistas en profundidad. La razón por la que hemos elegido estos dos métodos es porque creemos que son los más adecuados para obtener información detallada sobre los usuarios y su contexto de uso.

En cuanto al análisis competitivo, realizaremos una investigación exhaustiva sobre las aplicaciones similares en el mercado, analizando sus características, funcionalidades, fortalezas y debilidades. Esto nos permitirá identificar las necesidades y expectativas de los usuarios y desarrollar una aplicación que supere a la competencia y satisfaga las necesidades de los usuarios de manera más efectiva.

Por otro lado, las entrevistas en profundidad nos permitirán obtener información directa de los usuarios potenciales de nuestra aplicación, lo que nos ayudará a comprender mejor sus necesidades, motivaciones, intereses y expectativas en cuanto a la aplicación. Además, podremos conocer el contexto de uso de la aplicación y las tareas específicas que los usuarios necesitan realizar para lograr sus objetivos.

2.1.1.1 ANÁLISIS COMPETITIVO

Se realiza un análisis competitivo para conocer las fortalezas y debilidades de las aplicaciones móviles similares a la que se quiere desarrollar. De esta forma, se podrá diseñar una aplicación que se diferencie de las existentes y que ofrezca un valor añadido a los usuarios.

Para llevar a cabo el análisis competitivo, se pueden utilizar diferentes herramientas, como el benchmarking o la evaluación heurística. En este caso, se ha decidido utilizar el benchmarking, ya que permite una comparación directa entre la aplicación que se quiere desarrollar y las existentes en el mercado.

Se han identificado varias aplicaciones similares a la que se quiere desarrollar, basándose en las funcionalidades que ofrecen. En concreto, se han identificado cuatro aplicaciones: Home Workout, Virtuagym Fitness, ArtiFit, Gym Workout Plan.

- **Home Workout:** Ofrece una variedad de ejercicios para diferentes grupos musculares, con seguimiento de progreso y rutinas de entrenamiento personalizadas. También permite crear rutinas de entrenamiento personalizadas y cuenta con la guía de entrenadores virtuales. Cuenta con una interfaz sencilla y clara, con un diseño moderno y elegante. Los menús son intuitivos y fáciles de usar.

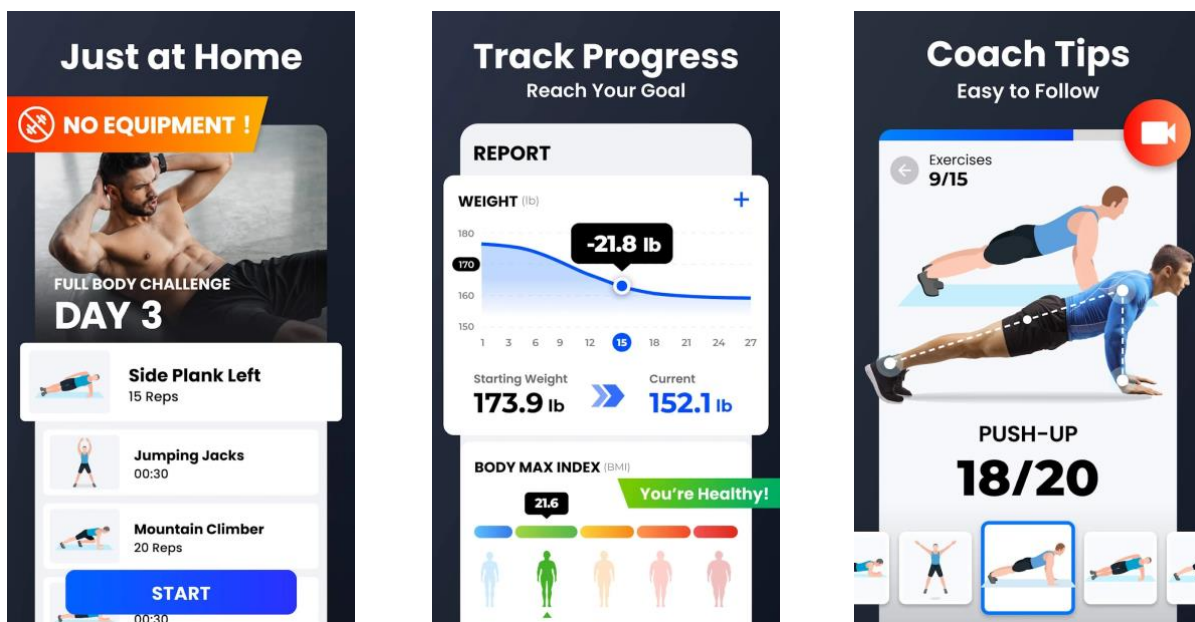


Ilustración 2. Home Workout

- **Virtuagym Fitness:** Ofrece una variedad de ejercicios para diferentes grupos musculares, con seguimiento de progreso y un plan de entrenamiento semanal. También ofrece la posibilidad de crear rutinas personalizadas y contar con la guía de entrenadores virtuales. Utiliza una tecnología avanzada de seguimiento de progreso. Ofrece soporte a través de correo electrónico y chat en vivo.

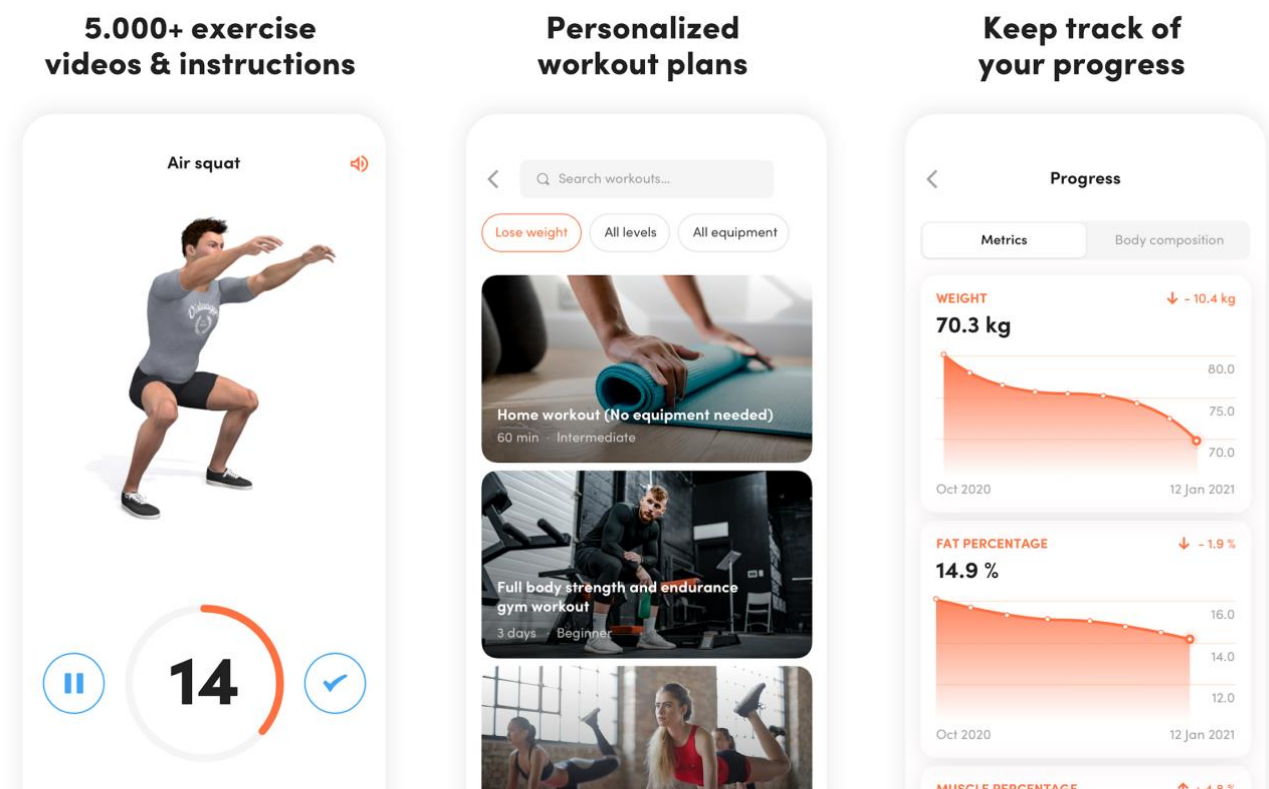


Ilustración 3 : Virtuagym Fitness

- **ArtiFit:** Ofrece una variedad de ejercicios para diferentes grupos musculares, con seguimiento de progreso y corrección de técnica a través de la cámara del dispositivo móvil. También ofrece rutinas de entrenamiento personalizadas y un plan de entrenamiento semanal. Solo está disponible para iOS

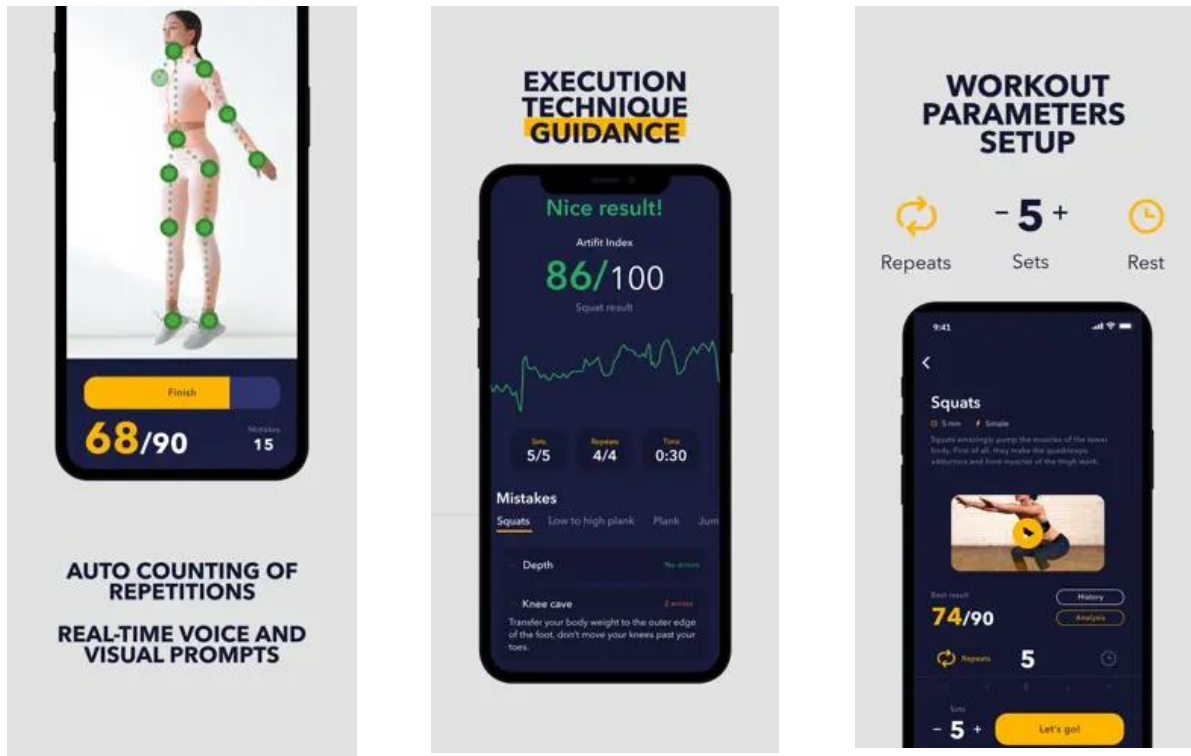


Ilustración 4: ArtiFit

- **Gym Workout Plan:**

Permite a los usuarios crear rutinas personalizadas a partir de los ejercicios disponibles. Ofrece una amplia variedad de ejercicios y planes de entrenamiento, por otro lado, la interfaz puede resultar algo anticuada y poco intuitiva para algunos usuarios.

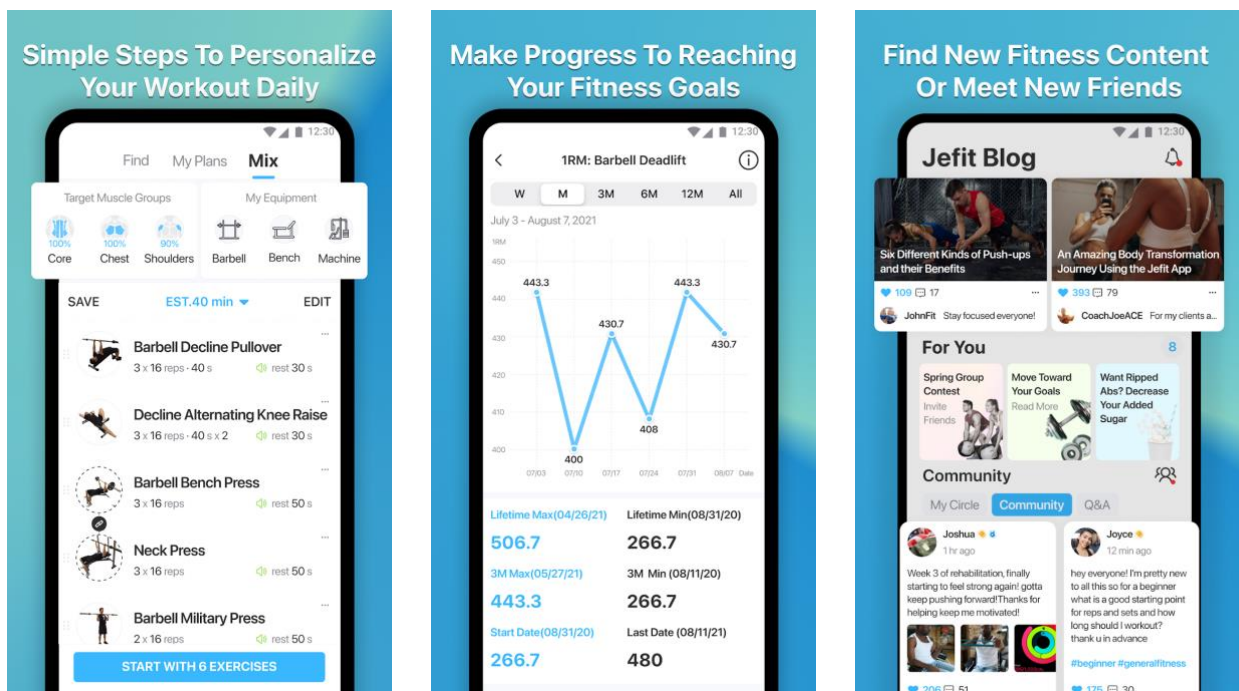


Ilustración 5: Gym Workout Plan

En conclusión, se puede observar que todas las aplicaciones tienen sus características. La aplicación que se quiere desarrollar se diferencia principalmente por su sistema de monitorización de la técnica del ejercicio mediante el uso de la cámara y la inteligencia artificial, lo que puede ser un factor de seguridad importante para los usuarios. Además, contará con un sistema de notificaciones para motivar y recordar a los usuarios la realización de sus ejercicios.

2.1.1.2 ENTREVISTAS EN PROFUNDIDAD

En el método de las entrevistas en profundidad, se busca obtener información detallada y significativa de los participantes a través de una conversación estructurada y dirigida por el entrevistador. A continuación, se presentará un guion para una entrevista en profundidad, enfocada en conocer las necesidades y expectativas de los usuarios potenciales de nuestra aplicación de ejercicio físico.

Guion de la entrevista

Es importante tener en cuenta que el guion de la entrevista debe ser flexible y adaptable a la situación y a las respuestas del entrevistado, permitiendo una conversación natural y espontánea.

- ¿Cuáles son tus hábitos de ejercicio físico?
- ¿Qué te motiva a hacer ejercicio?
- ¿Has utilizado alguna aplicación móvil para hacer ejercicio?
- ¿Qué funcionalidades te gustaría ver en una aplicación para hacer ejercicio?
- ¿Te gustaría que la aplicación tenga alguna función adicional?
- ¿Como utilizarías esta aplicación?

Entrevistas

Se pueden consultar las entrevistas en el apartado de anexos del capítulo 6,

Perfil de usuarios:

Perfil 1: Usuario principiante

Características:

- Edad: 25 años
- Género: Femenino
- Nivel de experiencia en el ejercicio físico: bajo
- Motivaciones: mejorar la salud, perder peso
- Intereses: la música, la moda, la tecnología
- Experiencia con el uso de la tecnología móvil: regular

Contextos de uso:

- En casa, en un espacio limitado
- Después del trabajo o los fines de semana

Análisis de tareas:

- Crear una rutina de ejercicio personalizada
- Ver y seguir una demostración de técnica de ejercicios
- Ver el progreso y la evolución de la actividad física

Características o elementos necesarios en la interfaz:

- Una guía visual de la técnica de cada ejercicio
- Un sistema de seguimiento de progreso y estadísticas claras y fáciles de entender

Perfil 2: Usuario con experiencia previa realizando ejercicio

Características:

- Edad: 35 años
- Género: Masculino
- Nivel de experiencia en el ejercicio físico: alto
- Motivaciones: mejorar su propio rendimiento
- Intereses: la nutrición, la ciencia del deporte, la tecnología

- Experiencia con el uso de la tecnología móvil: alta

Contextos de uso:

- En el gimnasio o en el exterior
- Luego del trabajo o los fines de semana

Análisis de tareas:

- Crear y planificar rutinas de entrenamiento personalizadas
- Hacer un seguimiento de su progreso y modificar las rutinas según sea necesario
- Acceder a una amplia variedad de ejercicios y demostraciones de técnica

Características o elementos necesarios en la interfaz:

- Un sistema fácil de usar para crear y planificar rutinas de entrenamiento personalizadas
- Una amplia variedad de ejercicios y demostraciones de técnica
- Un sistema de seguimiento del progreso y estadísticas claras y fáciles de entender

A continuación, se presenta un listado de características o elementos que se han descubierto como importantes durante la fase de indagación para la interfaz de la aplicación:

1. Personalización de rutinas de ejercicio: Los usuarios desean tener la capacidad de personalizar sus propias rutinas de ejercicios de acuerdo con sus necesidades y objetivos específicos.
2. Guía de técnica para ejercicios: Los usuarios buscan una guía de técnica clara y detallada para cada ejercicio, incluyendo videos o animaciones.
3. Seguimiento de progreso: Los usuarios valoran la capacidad de realizar un seguimiento de su progreso a lo largo del tiempo, para ver cómo están mejorando y establecer objetivos realistas para su entrenamiento.

4. Notificaciones y recordatorios: Los usuarios encuentran útil recibir notificaciones y recordatorios para motivarlos a hacer ejercicio y recordarles cuándo es el momento de hacerlo.

2.2 DISEÑO CONCEPTUAL

En la fase de diseño conceptual, se establecen las bases y los requisitos de la aplicación, se definen los objetivos y las funcionalidades que se deben implementar. Es un paso fundamental antes de comenzar con el diseño detallado y la implementación. Para este proyecto se realizó una investigación previa para determinar las necesidades y problemas comunes que los usuarios enfrentan al hacer ejercicios sin la ayuda de un entrenador personal.

2.2.1 ESCENARIOS DE USO

Escenario 1: Explorar la lista de entrenamientos

Perfil de usuario	Usuario sin experiencia previa
Contexto	En casa, después de la jornada laboral.
Objetivos	Mantenerse en forma y mejorar su salud.
Tareas realizadas	<ul style="list-style-type: none"> • Navegar por la lista de entrenamientos.
Necesidades de información	<ul style="list-style-type: none"> • Estar registrado previamente • Descripción detallada de cada entrenamiento y sus objetivos.
Funcionalidades necesarias	<ul style="list-style-type: none"> • Registro de usuario • Listas los entrenamientos disponibles
Desarrollo de las Tareas	<ul style="list-style-type: none"> • Ingresar a la aplicación y dirigirse a la pantalla principal. • Navegar por la lista de entrenamientos. • Leer la descripción de cada entrenamiento y seleccionar aquel que se ajuste a sus objetivos de entrenamiento.

Escenario 1: Buscar un entrenamiento

Perfil de usuario	Usuario con experiencia previa
Contexto	El usuario ya ha utilizado la aplicación anteriormente y desea buscar un entrenamiento específico para su rutina de entrenamiento actual.
Objetivos	Encontrar un entrenamiento específico para su rutina de entrenamiento actual.
Tareas realizadas	<ul style="list-style-type: none">• Introducir el nombre del entrenamiento en el cuadro de búsqueda
Necesidades de información	<ul style="list-style-type: none">• Estar registrado previamente• Nombres específicos de entrenamientos y descripciones detalladas.
Funcionalidades necesarias	<ul style="list-style-type: none">• Registro de usuario• Función de búsqueda
Desarrollo de las tareas	<ul style="list-style-type: none">• Ingresar a la aplicación y dirigirse a la pantalla principal.• Utilizar la función de búsqueda para encontrar el entrenamiento específico deseado.• Leer la descripción detallada del entrenamiento y determinar si es adecuado para su rutina de entrenamiento actual.

2.2.2 FLUJO DE INTERACCIÓN

En base a los datos obtenidos en los apartados anteriores se gráfica cómo el usuario interactúa con la aplicación, desde la pantalla de inicio hasta la finalización de una tarea o actividad. En este caso, el flujo de interacción de la aplicación de entrenamiento físico con inteligencia artificial consta de varios pasos.

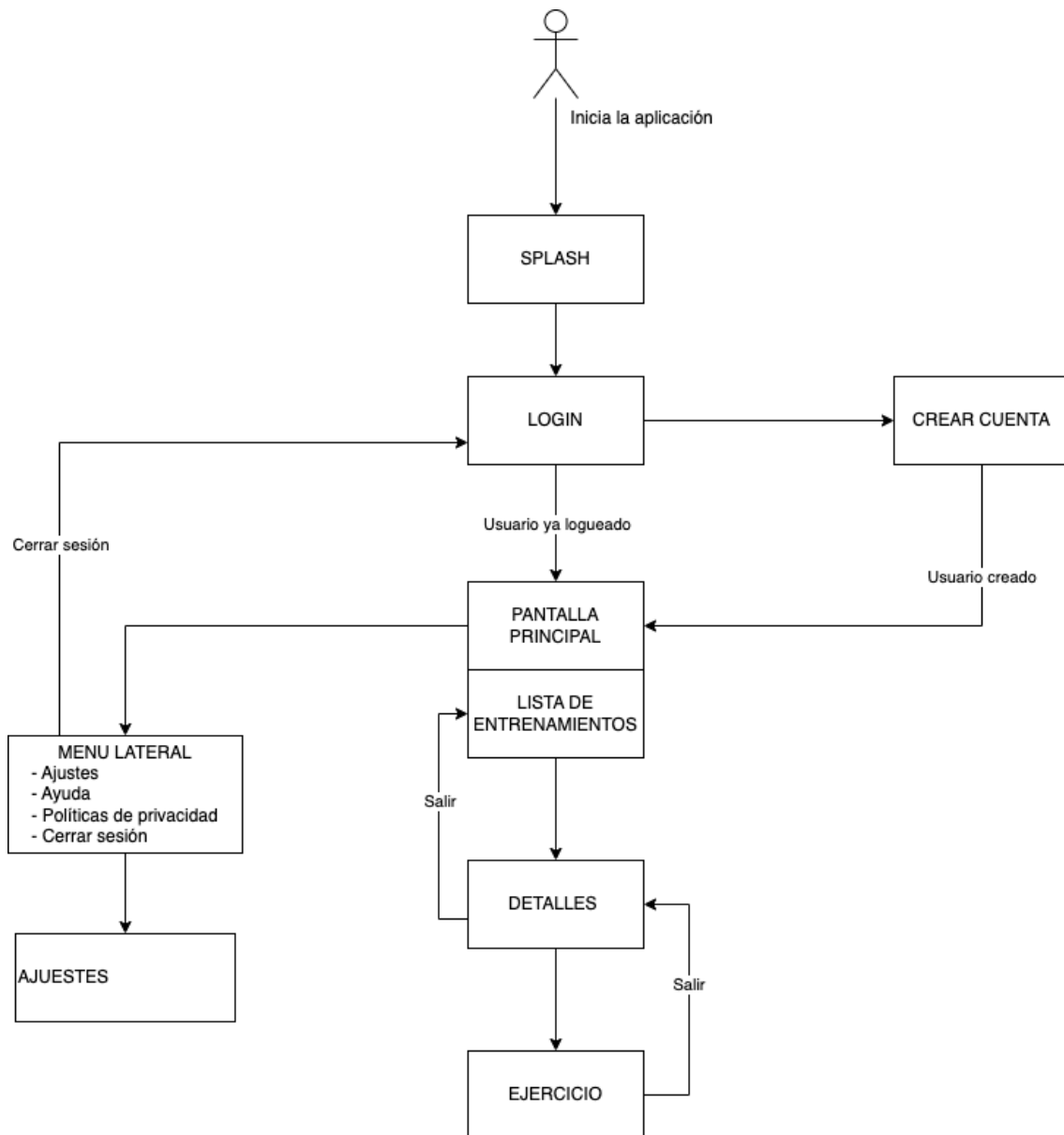


Ilustración 6. Flujo de interacción

2.3 PROTOTIPADO

En la fase de prototipado, se busca crear una representación visual de la aplicación que se está diseñando, de manera que sea posible evaluar su funcionamiento antes de llevarla a cabo de manera definitiva. El objetivo de esta fase es ofrecer una idea clara de cómo se verá y cómo funcionará la aplicación, de manera que se puedan realizar modificaciones en el diseño de manera temprana, antes de que se realice la implementación.

Comenzaremos con un prototipado de baja fidelidad para proporcionar una primera idea de cómo será la interfaz, la disposición de sus elementos y su visibilidad.

2.3.1 SKETCHES

El siguiente boceto a mano alzada representa una propuesta de diseño, sin preocuparnos por la precisión o los detalles nos ayudara a visualizar de manera rápida las posibles soluciones de diseño, así como explorar diferentes opciones.



Ilustración 7. Sketch pantalla de carga

Pantalla de carga

La pantalla de carga es la primera pantalla que los usuarios ven cuando abren la aplicación. Su función es mostrar una animación que indique que la aplicación está cargando los elementos necesarios para su correcto funcionamiento.

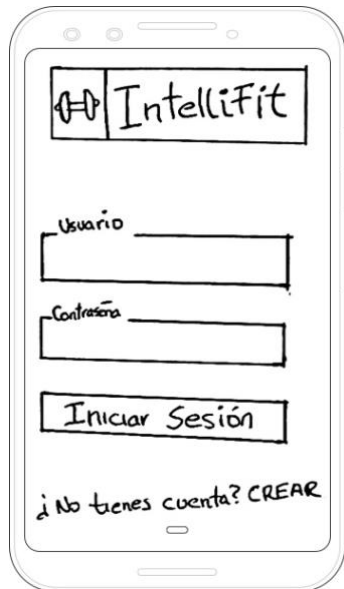


Ilustración 8. Sketch pantalla de inicio de sesión

Pantalla de inicio de sesión

La pantalla de inicio de sesión es la primera pantalla que el usuario verá después de que se haya cargado la aplicación. En esta pantalla, el usuario debe ingresar sus credenciales para acceder a la aplicación si es la primera vez que utiliza la aplicación, también tendrá la opción de crear una nueva cuenta.

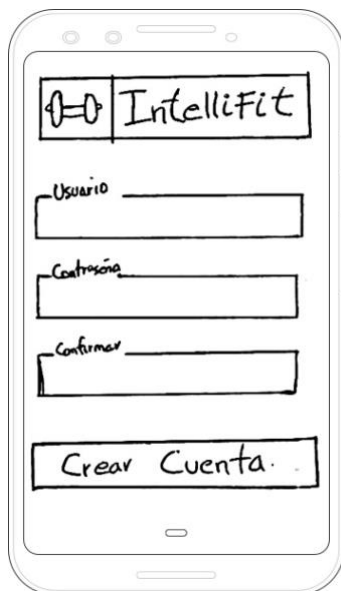


Ilustración 9. Sketch pantalla crear usuario

Pantalla de registro

La pantalla de crear usuario permite que los nuevos usuarios puedan registrarse para utilizar la aplicación. Esta pantalla contiene campos en los que los usuarios pueden ingresar información como el usuario y contraseña.

Después de que el usuario ha completado los campos obligatorios deben hacer clic en el botón crear cuenta para enviar su información. Si hay algún error o campo faltante, la pantalla muestra un mensaje de error para informar al usuario que debe completar el campo faltante o corregir el error antes de continuar.



Ilustración 10. Sketch pantalla principal

Pantalla principal

Esta pantalla muestra al usuario una lista de los entrenamientos disponibles en la aplicación. En esta pantalla, el usuario puede seleccionar el entrenamiento que desee para empezar a realizarlo.

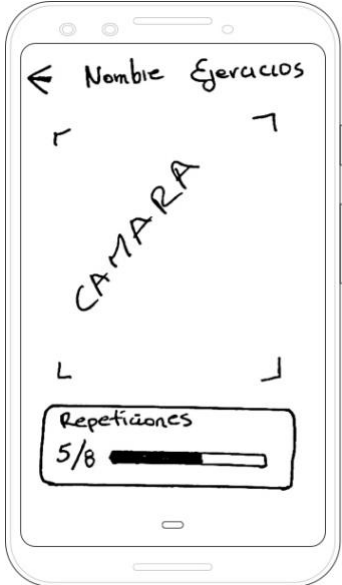
La lista incluye el nombre de cada entrenamiento y una imagen representativa del mismo.



Ilustración 11. Sketch pantalla detalles del entrenamiento

Pantalla detalles del entrenamiento

Esta pantalla muestra toda la información relacionada con el entrenamiento que ha sido seleccionado por el usuario. En la parte superior de la pantalla aparece una imagen, seguido por el nombre del entrenamiento con el tiempo estimado. A continuación, se encuentran los detalles específicos del entrenamiento, como el nombre de cada ejercicio y número de repeticiones. Por último, un botón de "iniciar" que permita al usuario comenzar el entrenamiento seleccionado.

 <p data-bbox="325 833 667 896">Ilustración 12. Sketch pantalla de ejercicio</p>	<p data-bbox="804 246 1114 282">Pantalla de ejercicio</p> <p data-bbox="804 304 1361 833">Esta pantalla proporciona al usuario una retroalimentación en tiempo real sobre la ejecución del ejercicio haciendo uso de la cámara delantera del dispositivo móvil. Además, la pantalla cuenta con una barra de progreso en la parte inferior que muestra el avance del usuario en la realización del ejercicio y el número de repeticiones que ha completado.</p>
---	---

2.3.2 PROTOTIPO HORIZONTAL DE ALTA FIDELIDAD

El prototipo horizontal de alta fidelidad de la aplicación se construye mostrando una interfaz de usuario detallada y precisa. Para la creación del prototipo se utiliza la herramienta de diseño Figma.

Para asegurar la coherencia en el diseño y la implementación real, se utilizará el [kit de diseño material de Android para Figma](#). Este kit de diseño incluye una amplia variedad de componentes visuales y funcionales de Android, como botones, menús, tarjetas y otros elementos de interfaz de usuario.

Además, se realiza un prototipo interactivo que permite al equipo de desarrollo y a los usuarios potenciales probar la aplicación antes de su implementación y hacer sugerencias para mejorar su usabilidad y funcionalidad, el prototipo interactivo está disponible en el siguiente enlace:

<https://www.figma.com/proto/IC8z8321CqZDh4YgVTmJ2D/IntelliFit?page-id=53798%3A27459>



Ilustración 13. Pantalla de carga

Pantalla inicial de carga, respecto al sketch solo se agrega una animación de progreso para indicar que se está inicializando la aplicación.

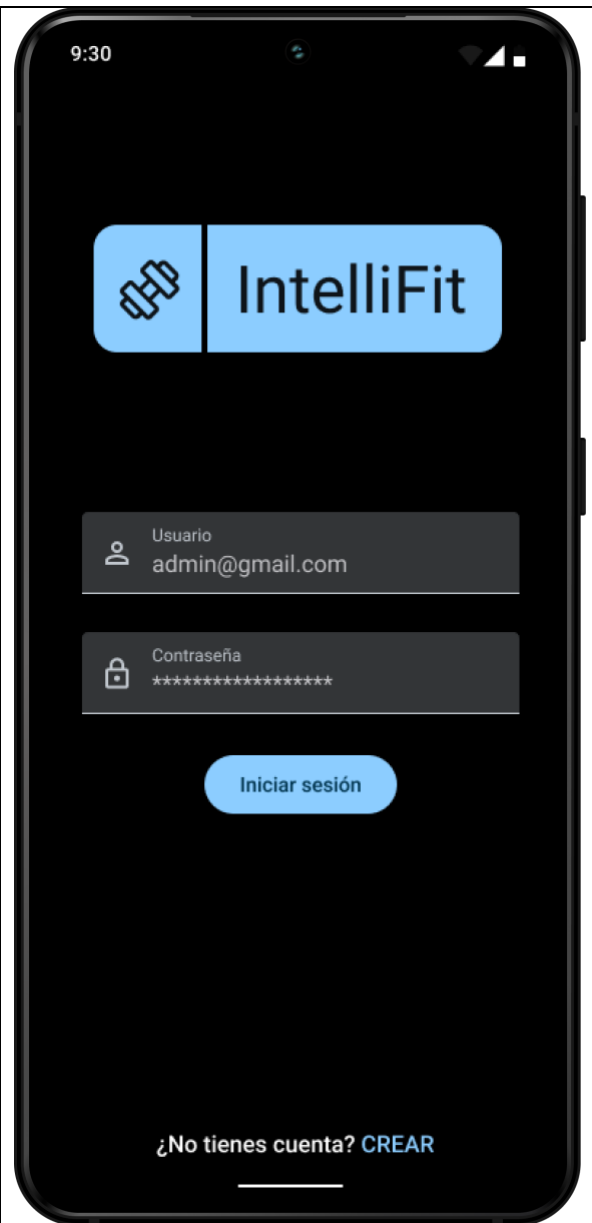


Ilustración 14. Pantalla inicio de sesión

Pantalla de inicio de sesión, no hay cambios destacables respecto al sketch

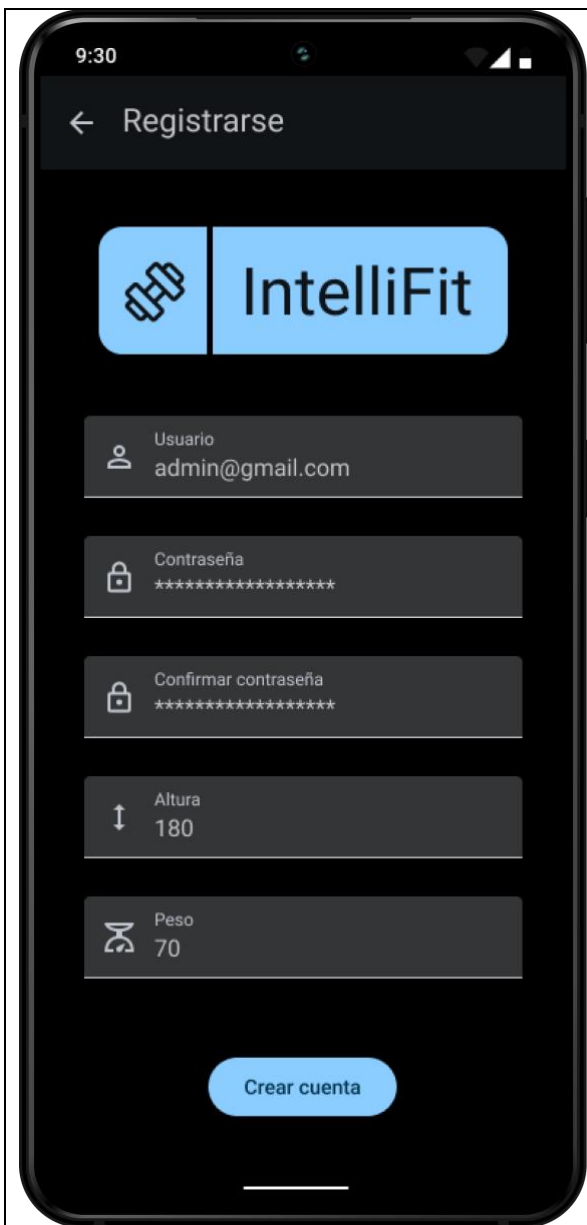


Ilustración 15. Pantalla de registro

En la pantalla de registro respecto al sketch se agregó las entradas de la altura y el peso, que servirá para en versiones posteriores agregar un registro del peso y cambios de este en el tiempo.

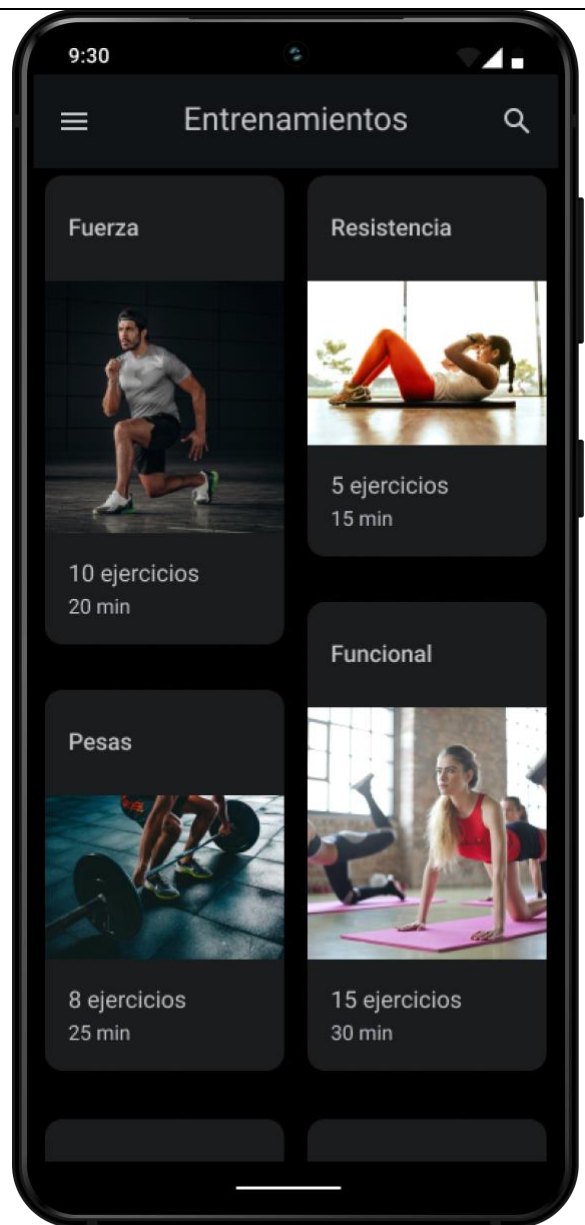


Ilustración 16. Pantalla de entrenamientos

En la pantalla principal de la aplicación luego de la evaluación se detectó la necesidad de agregar algunas opciones por lo que se agregó un botón de menú con diferentes opciones y un botón para buscar en la lista de entrenamientos disponibles.

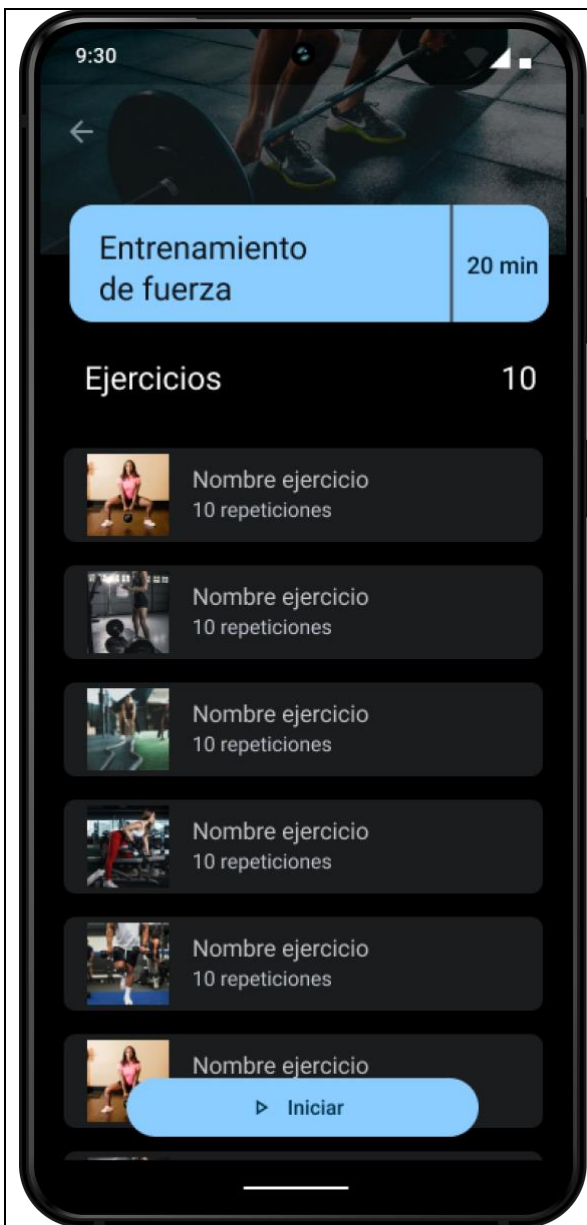


Ilustración 17. Pantalla de entrenamientos

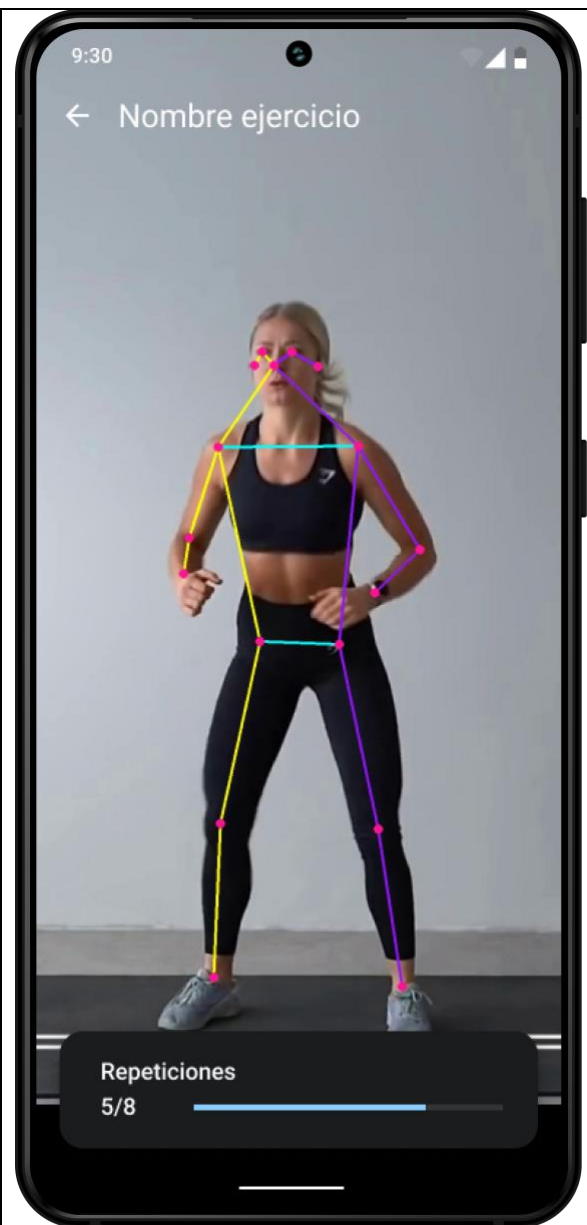


Ilustración 18. Pantalla de ejercicio

Pantalla de detalle del entrenamiento, sin cambios respecto al sketch.

Pantalla de ejercicio. Sin cambios respecto al sketch, se agrega una imagen real de la detección de pose que fue obtenida utilizando Movenet de TensorFlow .

2.4 EVALUACIÓN

En este proyecto, aunque no ha sido posible realizar una evaluación exhaustiva con varios usuarios y en múltiples iteraciones, se ha llevado a cabo una evaluación con al menos un usuario y se han incorporado algunas mejoras en base al *feedback* obtenido.

La evaluación se realizó de forma iterativa y continua a lo largo del proceso de diseño, lo que nos permitió ir refinando el prototipo y asegurando que se cumplan los objetivos establecidos y se ofrezca una experiencia de usuario satisfactoria.

Se han realizado preguntas a los usuarios para identificar posibles problemas y mejoras en el prototipo. Es importante tener en cuenta que la evaluación no solo se centra en la usabilidad de la aplicación, sino también en otros aspectos como la seguridad de los datos de los usuarios, la adaptabilidad a diferentes dispositivos y situaciones de uso, entre otros. Todos estos aspectos deben ser evaluados de forma sistemática y rigurosa para asegurar un producto final de calidad.

2.4.1 PREGUNTAS REFERENTES AL USUARIO

Para obtener información relevante acerca del usuario que realizará el test, se plantean las siguientes preguntas las cuales permiten conocer su perfil y sus características:

1. ¿Qué edad tienes?
2. ¿Cuál es su nivel de experiencia con aplicaciones móviles?
3. ¿Tienes experiencia previa realizando ejercicio?
4. ¿Cuál es tu objetivo en cuanto a la práctica de ejercicio?
5. ¿Tienes algunas limitaciones físicas o de salud?
6. ¿Cuál es el modelo de su dispositivo móvil?

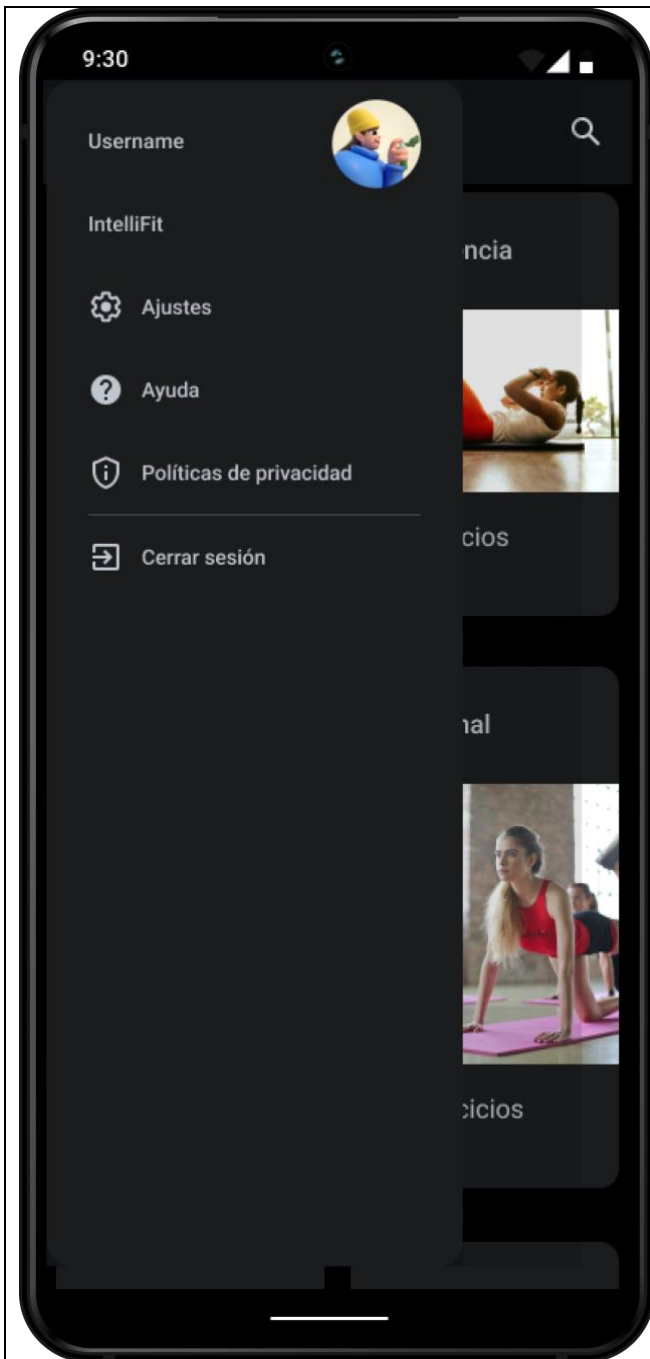
2.4.2 TAREAS A REALIZAR POR EL USUARIO

Tras la definición de las preguntas que realizaremos al usuario, se fijan una serie de tareas a realizar por el usuario, estas están diseñadas de manera clara y concisa para que los usuarios puedan entenderlas fácilmente y realizarlas sin problemas. Además, son representativas de las funcionalidades y flujos más importantes de la aplicación, para que se pueda obtener conclusiones relevantes sobre la experiencia de usuario.

1. Registrarse en la aplicación
2. Iniciar sesión
3. Seleccionar una rutina de entrenamiento de la lista
4. Realizar los ejercicios propuestos siguiendo las indicaciones de la aplicación
5. Realizar un ejercicio y verificar que la detección de postura funciona correctamente.

2.4.3 MEJORAS LUEGO DE LA EVALUACIÓN

Tras la evaluación realizada con el usuario, se identificaron algunas mejoras que podrían implementarse en el diseño de la aplicación. Estas mejoras se enfocaron en aspectos como la facilidad de uso de la interfaz de usuario, así como en la funcionalidad de la aplicación.



Durante la evaluación se ha detectado la necesidad de agregar un cajón de navegación en la pantalla principal, con distintas opciones como; ajustes, enviar un correo al soporte, revisar las políticas de privacidad y cerrar sesión.

Ilustración 19. Pantalla menú de navegación

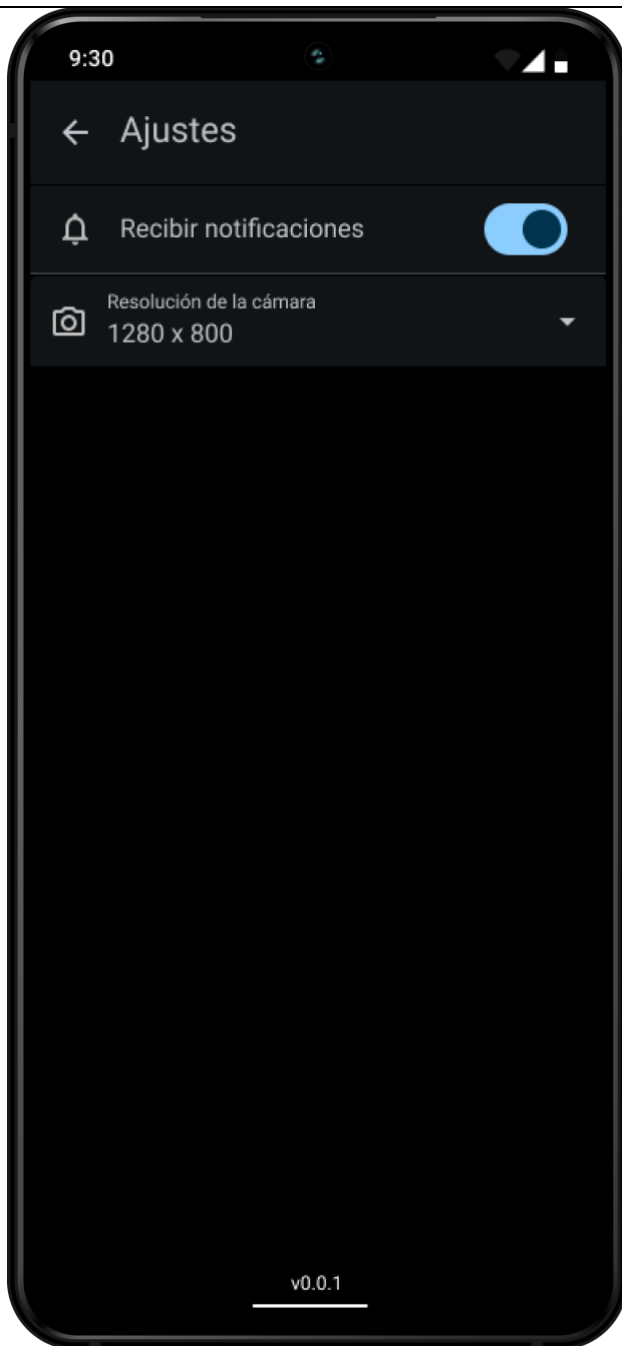


Ilustración 20. Pantalla de ajustes

Durante la evaluación se ha detectado la necesidad de agregar una pantalla de ajustes. La pantalla de ajustes de la aplicación incluye varias opciones para que el usuario pueda personalizar su experiencia de uso. En primer lugar, se encuentra la opción para habilitar o deshabilitar las notificaciones, también se incluye la opción para cambiar la resolución de la cámara, lo que permite al usuario ajustar la calidad de la cámara delantera que se utiliza durante el ejercicio.

En versiones futuras de la aplicación, se podrían agregar nuevas opciones de ajustes, como por ejemplo la posibilidad de cambiar el idioma de la aplicación, ajustar el volumen de los sonidos de la aplicación o personalizar la interfaz de usuario con diferentes temas. Estas opciones de personalización podrían mejorar la experiencia de usuario y hacer que la aplicación sea más atractiva y fácil de usar para un público más amplio.

3. DISEÑO TÉCNICO

3.1 DEFINICIÓN DE CASOS DE USO

3.1.1 DIAGRAMA DE CASOS DE USO

Se realiza una representación gráfica que muestra los diferentes casos de uso de la aplicación y las relaciones entre ellos.

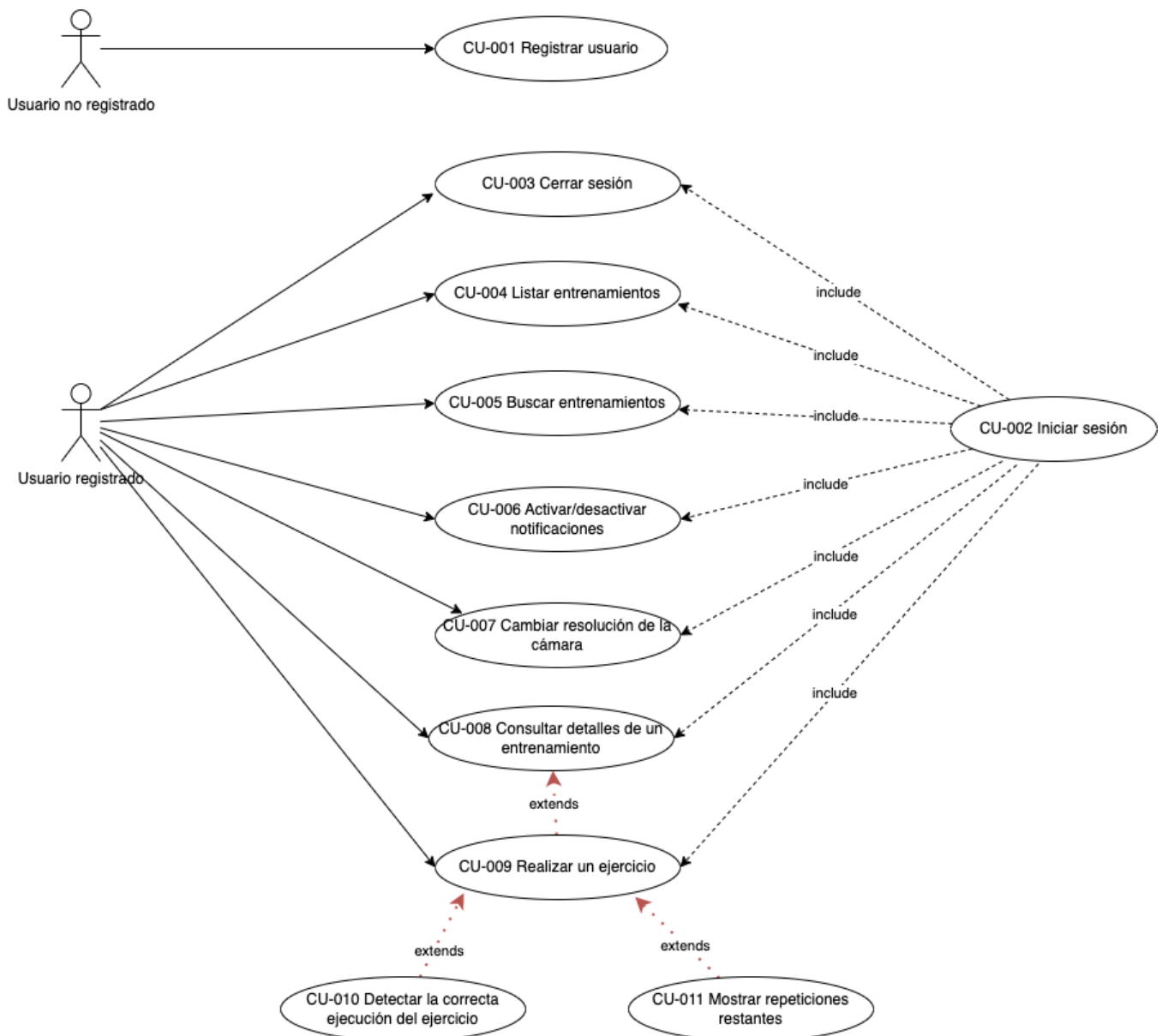


Ilustración 21. Diagrama de casos de uso

3.1.2 LISTADO DE CASOS DE USO

Identificador	CU-001
Nombre	Registrar usuario
Prioridad	Alta
Descripción	Un usuario nuevo debe registrarse en la aplicación para poder acceder a las funcionalidades.
Actores	Usuario no registrado
Precondiciones	Ninguna
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none">1. El usuario accede a la pantalla de registro.2. El usuario introduce su información personal y credenciales de acceso.3. La aplicación verifica la información y crea la cuenta de usuario.4. La aplicación redirige al usuario a la pantalla de inicio de sesión.
Postcondiciones	El usuario tiene una cuenta creada y puede iniciar sesión en la aplicación
Notas	

Identificador	CU-002
Nombre	Iniciar sesión
Prioridad	Alta
Descripción	Permite al usuario acceder a la aplicación con sus credenciales de acceso.
Actores	Usuario registrado
Precondiciones	El usuario debe tener una cuenta creada en la aplicación.
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none">1. El usuario accede a la pantalla de inicio de sesión.2. El usuario introduce sus credenciales de acceso.

	<ol style="list-style-type: none"> 3. La aplicación verifica la información y permite el acceso del usuario. 4. La aplicación redirige al usuario a la pantalla de lista de entrenamientos.
Postcondiciones	El usuario tiene acceso a su cuenta y a las funcionalidades de la aplicación
Notas	

Identificador	CU-003
Nombre	Cerrar sesión
Prioridad	Alta
Descripción	Permite al usuario cerrar sesión en la aplicación.
Actores	Usuario registrado
Precondiciones	El usuario debe haber iniciado sesión en la aplicación.
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. El usuario accede el menú lateral. 2. El usuario selecciona la opción de cerrar sesión. 3. La aplicación cierra la sesión del usuario y redirige a la pantalla de inicio de sesión.
Postcondiciones	El usuario ha cerrado sesión y no tiene acceso a las funcionalidades de la aplicación.
Notas	

Identificador	CU-004
Nombre	Listar entrenamientos
Prioridad	Alta
Descripción	Permite al usuario ver la lista de entrenamientos disponibles en la aplicación
Actores	Usuario registrado
Precondiciones	El usuario debe haber iniciado sesión en la aplicación.

Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de lista de entrenamientos. 2. La aplicación muestra la lista de entrenamientos disponibles. 3. El usuario puede desplazarse por la lista y seleccionar un entrenamiento para ver más detalles.
Postcondiciones	El usuario ha visto la lista de entrenamientos disponibles en la aplicación.
Notas	

Identificador	CU-005
Nombre	Buscar entrenamientos
Prioridad	Baja
Descripción	Permite al usuario buscar un entrenamiento específico en la aplicación.
Actores	Usuario registrado
Precondiciones	El usuario debe haber iniciado sesión en la aplicación.
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de búsqueda en la pantalla principal. 2. El usuario ingresa un término de búsqueda en la barra de búsqueda. 3. La aplicación muestra los resultados de la búsqueda.
Postcondiciones	La aplicación muestra el entrenamiento deseado.
Notas	

Identificador	CU-006
Nombre	Activar/desactivar notificaciones
Prioridad	Media
Descripción	Permite a los usuarios activar o desactivar las

	notificaciones de la aplicación.
Actores	Usuario registrado
Precondiciones	El usuario debe haber iniciado sesión en la aplicación.
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de ajustes de la aplicación. 2. El usuario activa o desactiva la opción de notificaciones. 3. La aplicación guarda la configuración del usuario.
Postcondiciones	Las notificaciones se han activado o desactivado según la preferencia del usuario.
Notas	

Identificador	CU-007
Nombre	Cambiar resolución de la cámara
Prioridad	Baja
Descripción	Permite al usuario cambiar la resolución de la cámara utilizada por la aplicación para la detección de la correcta ejecución del ejercicio.
Actores	Usuario registrado
Precondiciones	El usuario debe haber iniciado sesión en la aplicación.
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de ajustes de la aplicación. 2. El usuario selecciona la opción de cambiar la resolución de la cámara. 3. La aplicación muestra las diferentes opciones de resolución disponibles. 4. El usuario selecciona la resolución deseada. 5. La aplicación guarda la configuración del usuario.
Postcondiciones	La resolución de la cámara ha sido cambiada según la preferencia del usuario.
Notas	

Identificador	CU-008
Nombre	Consultar detalles de un entrenamiento
Prioridad	Alta
Descripción	Permite a los usuarios ver los detalles de un entrenamiento específico.
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión en la aplicación. • El usuario debe haber seleccionado un entrenamiento
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona un entrenamiento de la lista de entrenamientos. 2. La aplicación muestra los detalles del entrenamiento, incluyendo el nombre, la descripción, la duración y los ejercicios que lo componen.
Postcondiciones	El usuario ha visto los detalles de un entrenamiento
Notas	

Identificador	CU-009
Nombre	Realizar un ejercicio
Prioridad	Alta
Descripción	Permite al usuario pueda realizar un ejercicio seleccionado previamente.
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión en la aplicación. • El usuario debe haber seleccionado un entrenamiento
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona iniciar en la pantalla de detalle. 2. La aplicación muestra al usuario una vista previa de

	<p>su cámara.</p> <p>3. El usuario se posiciona frente a la cámara y comienza a realizar el ejercicio.</p> <p>4. La aplicación utiliza técnicas de inteligencia artificial para detectar la correcta ejecución del ejercicio.</p>
Postcondiciones	Se registra en el historial del usuario la realización del ejercicio.
Notas	

Identificador	CU-010
Nombre	Detectar la correcta ejecución del ejercicio
Prioridad	Alta
Descripción	La aplicación debe ser capaz de analizar la posición del usuario y compararla con la posición correcta del ejercicio para detectar posibles errores.
Actores	Usuario autenticado
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión en la aplicación. • El usuario debe haber seleccionado un entrenamiento • EL usuario debe haber iniciado un entrenamiento • El usuario debe estar realizando un ejercicio.
Iniciado por	Sistema tras realizar una repetición
Flujo	<ol style="list-style-type: none"> 1. La aplicación utiliza la cámara del dispositivo para capturar la imagen del usuario realizando el ejercicio. 2. La aplicación utiliza técnicas de inteligencia artificial para comparar la posición del usuario con la posición correcta del ejercicio. 3. Si la posición del usuario es correcta, la aplicación muestra el progreso en la barra de repeticiones. 4. Si la posición del usuario es incorrecta, la aplicación

	muestra una notificación para corregir su postura.
Postcondiciones	La aplicación registra en el historial del usuario la realización del ejercicio y los posibles errores detectados durante la ejecución.
Notas	

Identificador	CU-011
Nombre	Mostrar repeticiones restantes
Prioridad	Media
Descripción	La aplicación muestra las repeticiones restantes de un ejercicio.
Actores	Usuario autenticado
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión en la aplicación. • El usuario debe haber seleccionado un entrenamiento • EL usuario debe haber iniciado un entrenamiento • El usuario debe estar realizando un ejercicio.
Iniciado por	Sistema
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona un ejercicio para realizar. 2. En la pantalla se muestra el número total de repeticiones del ejercicio. 3. A medida que el usuario va realizando repeticiones, la aplicación actualiza y muestra las repeticiones restantes.
Postcondiciones	El usuario ha visualizado el número de repeticiones restantes del ejercicio.
Notas	

3.2 ARQUITECTURA

La arquitectura de la aplicación será basada en el patrón MVVM (Model-View-ViewModel) de Android, el cual permite separar la lógica de la vista y facilita la creación de aplicaciones escalables y mantenibles.

En este patrón, la vista representa la capa de presentación, la cual se encarga de interactuar con el usuario a través de la interfaz gráfica de la aplicación. El modelo representa los datos y la lógica de negocio, mientras que el ViewModel es el encargado de conectar la vista con el modelo y manejar la lógica específica de la presentación.

La arquitectura se dividirá en capas, lo que permitirá una fácil gestión y mantenimiento de la aplicación. Las capas serán las siguientes:

- Capa de presentación: esta capa estará compuesta por las vistas y los ViewModels. Los ViewModels contendrán la lógica de presentación y se comunicarán con la capa de datos. La vista estará encargada de mostrar la información al usuario a través de la interfaz ya sea en la actividad o fragmento.
- Capa de dominio: esta capa contendrá las reglas de negocio y la lógica de aplicación. Las entidades y las interfaces de los repositorios estarán en esta capa.
- Capa de datos: esta capa será la encargada de manejar la persistencia de los datos y la comunicación con los servicios externos, a través de la implementación de los repositorios y las fuentes de datos, que en este caso serán dos, la comunicación de la API REST y el cache que se guardara en local.

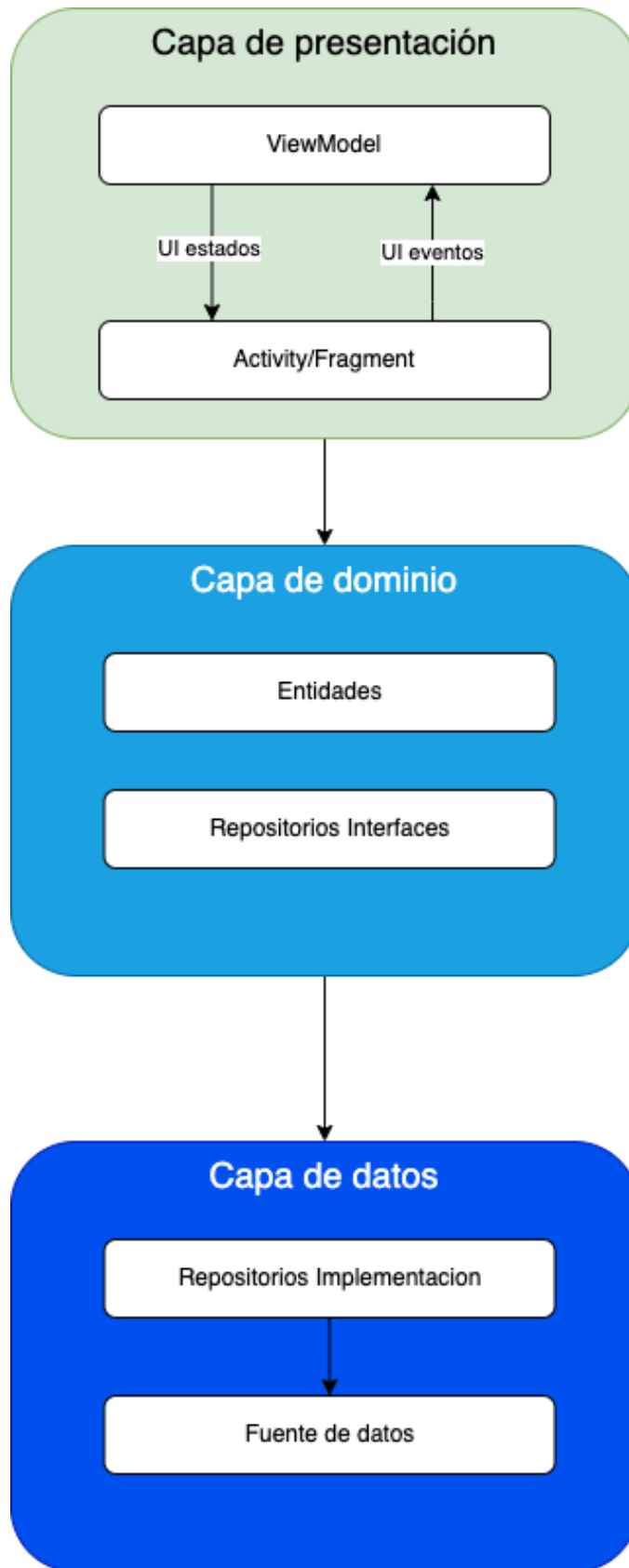


Ilustración 22. Diagrama de una arquitectura

3.2.1 MODELO DE DATOS

El modelo de datos de la aplicación consta de cinco tablas

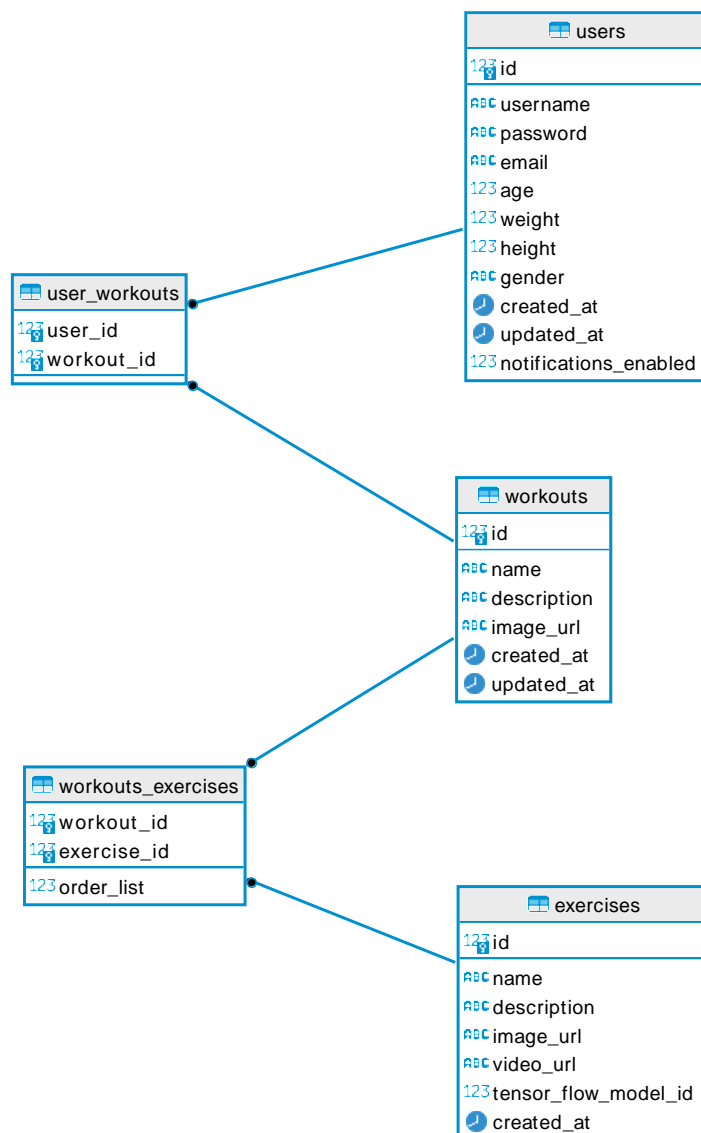
users: Almacena la información de los usuarios registrados en la aplicación

workouts: Almacena la información de los entrenamientos disponibles

exercises: Almacena la información de los ejercicios disponibles

workouts_exercises: Es una tabla intermedia que relaciona los entrenamientos con los ejercicios que los componen.

user_workouts: Almacena la información de los entrenamientos que cada usuario ha asignado para sí mismo.



4. IMPLEMENTACIÓN

4.1 TECNOLOGIA USADA

4.1.2. LENGUAJE DE PROGRAMACIÓN

El lenguaje de programación utilizado para desarrollar la aplicación ha sido Kotlin, ya que es un lenguaje moderno, orientado a objetos y estáticamente tipado, que en la actualidad se ha convertido en el lenguaje de elección para muchos desarrolladores de Android gracias a sus características de seguridad, concisión e interoperabilidad con Java. Kotlin también tiene soporte de primera clase en Android Studio, lo que facilita aún más su uso en el desarrollo de aplicaciones Android.

4.1.1. ENTORNO DE DESARROLLO INTEGRADO (IDE)

Para el desarrollo de la aplicación se ha utilizado Android Studio, el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android. Este IDE está basado en IntelliJ IDEA de JetBrains, y proporciona una serie de características que facilitan el desarrollo de aplicaciones.

Android Studio ofrece herramientas especializadas para el desarrollo de Android, como un diseño de interfaz de usuario visual con soporte para arrastrar y soltar, también incluye soporte integrado para Google Cloud Platform, y emuladores rápidos para pruebas.

4.1.3. LIBRERIAS

A continuación, se describen las librerías empleadas en el desarrollo de la aplicación:

Navigation: Utilizada para gestionar la navegación compleja en la aplicación.

Lifecycle: Proporciona clases y métodos para gestionar el ciclo de vida de la aplicación y sus componentes.

Fragment: Facilita la creación de fragmentos en la aplicación.

Room: Biblioteca de persistencia que proporciona una capa de abstracción sobre SQLite para facilitar el uso de bases de datos en las aplicaciones.

JUnit y Espresso: Para realizar pruebas unitarias y de interfaz de usuario, respectivamente.

Kotlin Coroutines: Para simplificar el código que se ejecuta de forma asíncrona y proporcionar un manejo más fácil de las operaciones de hilos.

TensorFlow Lite: Se emplea para ejecutar modelos de aprendizaje automático en dispositivos móviles y sistemas embebidos.

Timber: Una biblioteca de registro

Koin: Para la inyección de dependencias para Kotlin.

Gson: Para convertir objetos Java en su representación JSON y viceversa.

CameraX: Facilita el uso de la cámara en aplicaciones de Android.

Retrofit: Para las llamadas de red REST.

Glide: Biblioteca de manipulación y carga de imágenes, ampliamente usada por el manejo eficiente de imágenes y la caché de las mismas.

CoreSplashScreen: Proporciona una forma estandarizada y sencilla de implementar la pantalla de inicio en las aplicaciones de Android.

Security Crypto: Es una biblioteca de Jetpack para facilitar el cifrado y descifrado en Android.

Java JWT : Se utiliza para decodificar, verificar, generar y manipular JWTs.

4.1.4. VERSIONES DE ANDROID

La elección de la versión mínima de SDK (Software Development Kit) es un aspecto crucial en el desarrollo de una aplicación Android. Establece la versión

más baja de Android que la aplicación admite y determina la base de usuarios potenciales de la aplicación.

La versión mínima del SDK fue establecida en 28, lo que significa que la aplicación es compatible con Android 9.0 (Pie) y versiones superiores. Esta decisión se tomó por varias razones:

Acceso a características modernas: Las versiones más recientes de Android ofrecen funcionalidades y APIs que no están disponibles en las versiones más antiguas. Esto permite a la aplicación aprovechar las capacidades modernas del sistema operativo.

Seguridad mejorada: Android ha mejorado significativamente la seguridad y la privacidad del sistema operativo con cada nueva versión. Al admitir versiones más recientes, la aplicación puede aprovechar estas mejoras y proporcionar una experiencia más segura a los usuarios.

Adopción de dispositivos: Según las estadísticas de adopción de versiones de Android proporcionadas por Google, la mayoría de los usuarios de Android utilizan versiones recientes del sistema operativo. Esto significa que elegir una versión mínima de SDK más reciente no debería excluir a una gran parte de la base de usuarios potenciales.

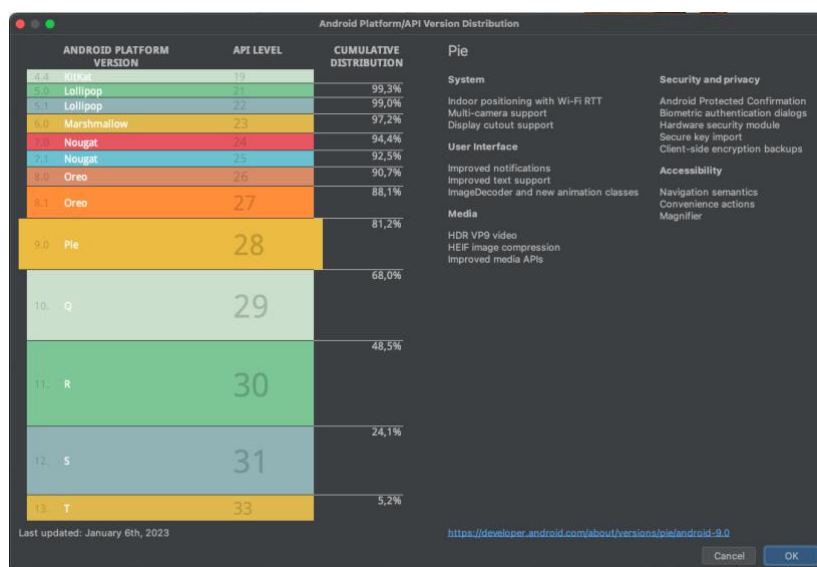


Ilustración 23. Distribución de versiones API

Mantenimiento y soporte: Es más fácil y eficiente mantener y ofrecer soporte a una aplicación si no se tiene que preocupar por la compatibilidad con versiones muy antiguas de Android.

4.1.5. VARIANTES DE COMPILACIÓN

El proceso de definir variantes de compilación en el gradle permite configurar diferentes versiones de la aplicación con diversas características y ajustes. En el caso de este proyecto, se han definido dos variantes: **debug** y **release**, y adicionalmente se ha establecido una distinción a nivel de flavors para diferenciar la versión de producción y la versión de demostración.

La versión debug es utilizada durante el desarrollo y la fase de pruebas. Está configurada para facilitar la detección y corrección de errores, permitiendo la depuración del código y desactivando la minimización y ofuscación del código. Por otro lado, la versión release está destinada a la distribución al usuario final. En esta versión, se activa la minimización y ofuscación del código para mejorar el rendimiento y la seguridad de la aplicación.

Adicionalmente, se ha establecido la distinción entre flavors **demo** y **production**. La versión demo se utiliza para la fase de desarrollo y pruebas, mientras que la versión production es la que se distribuirá a los usuarios finales.

En el proceso de desarrollo de una aplicación, a menudo intervienen diferentes equipos, desde desarrolladores de frontend y backend hasta equipos de calidad y diseño. Para facilitar la colaboración entre estos equipos y mantener un flujo de trabajo eficiente, se ha creado la variante demo. Esta variante demo se ha diseñado específicamente para simular todas las conexiones al backend de la aplicación. El objetivo de esta estrategia es permitir el desarrollo y la prueba independiente de las funcionalidades que pueden aún no estar disponibles en el backend.

Por ejemplo, si el equipo de frontend está trabajando en una característica que depende de una API aún en desarrollo por el equipo de backend, la variante demo puede simular la respuesta de esta API para que el equipo de frontend pueda continuar su trabajo.

Esto ayuda a acelerar el proceso de desarrollo, ya que permite que diferentes partes de la aplicación se construyan en paralelo, sin que un equipo tenga que esperar a que el otro termine su trabajo antes de poder avanzar. Esta estrategia también beneficia al equipo de pruebas, ya que pueden empezar a probar las nuevas funcionalidades en la variante demo en cuanto estén disponibles, en lugar de tener que esperar a que la API correspondiente esté completa en el backend.

4.1.6. SISTEMA DE CONTROL DE VERSIONES

Para el control de versiones del proyecto se ha utilizado Git, un sistema de control de versiones distribuido que permite a múltiples desarrolladores trabajar en un proyecto al mismo tiempo sin interferir en el trabajo de los demás.

Además de Git, el proyecto también ha utilizado GitFlow, un flujo de trabajo basado en Git que proporciona una estructura muy robusta para el desarrollo y mantenimiento de software.

GitFlow define un modelo de ramas que facilita la organización del trabajo, especialmente en proyectos con múltiples desarrolladores. Según este modelo, el desarrollo se lleva a cabo en distintas ramas según el tipo de trabajo: features para nuevas funcionalidades, hotfixes para correcciones urgentes, release para preparar nuevas versiones de lanzamiento, y develop y master para la integración y lanzamiento de las versiones estables.

Cada desarrollador puede trabajar en su propia rama sin afectar al código principal del proyecto. Una vez que el trabajo en una rama está completo, puede ser fusionado con la rama develop o master según corresponda. Esto

permite un flujo de trabajo eficiente y ordenado, evitando conflictos y facilitando la colaboración entre los desarrolladores.

4.2. PATRONES Y BUENAS PRACTICAS

Para el desarrollo de la aplicación, se han seguido una serie de patrones de diseño y buenas prácticas con el fin de asegurar la calidad, eficiencia y mantenibilidad del código:

Patrón MVVM (Model-View-ViewModel): Promueve la separación de la lógica de negocio y la lógica de presentación, lo que facilita el desarrollo y la prueba de la aplicación.

Principio SOLID: Los principios SOLID son un conjunto de cinco principios de programación orientada a objetos que, cuando se aplican juntos, hacen que un sistema sea más fácil de mantener y de expandir a lo largo del tiempo. Estos principios son:

- **Single Responsibility Principle (SRP):** Cada clase debe tener una única responsabilidad.
- **Open-Closed Principle (OCP):** Las entidades de software deben estar abiertas para su extensión, pero cerradas para su modificación.
- **Liskov Substitution Principle (LSP):** Los objetos de un programa deben ser reemplazables por instancias de sus subtipos sin afectar la corrección del programa.
- **Dependency Inversion Principle (DIP):** Dependier de abstracciones, no de implementaciones.

Pruebas unitarias y de integración: Es esencial realizar pruebas para verificar la funcionalidad del software.

Refactorización de código: La refactorización se ha llevado a cabo regularmente para mejorar la estructura y la calidad del código sin cambiar su comportamiento externo.

La aplicación de estos principios ha contribuido a que el código sea limpio, legible, reutilizable y fácil de mantener.

4.3 ENTRENAMIENTO DE MODELO

Después de obtener los datos de las poses con MoveNet para cada ejercicio, los procesé de manera específica para prepararlos para el entrenamiento y se guardaron en un archivo CSV, cada fila de estos archivos contiene $17 * 3 = 51$ valores, que representan las coordenadas x, y y la puntuación de confianza para cada una de las 17 partes del cuerpo detectadas por MoveNet. Los archivos también incluyen etiquetas que indican si el ejercicio correspondiente es correcto o incorrecto.

A continuación, normalicé los datos de las poses para eliminar cualquier variación debido a la posición y el tamaño de la persona en la imagen. Esto se hace calculando el centro de la pose (el punto medio entre las caderas) y trasladando este punto al origen (0,0). También calculé el tamaño de la pose como la distancia máxima de cualquier parte del cuerpo al centro de la pose, y escalé todas las coordenadas de las partes del cuerpo por este tamaño. Este paso es crucial para que el modelo pueda reconocer las poses independientemente de dónde se encuentre la persona en la imagen o cuán cerca o lejos esté de la cámara.

Finalmente, construí el modelo de clasificación usando TensorFlow, este modelo consta de dos capas ocultas, cada una con una función de activación ReLU, y una capa de salida con una función de activación softmax para producir las probabilidades de las dos clases.

Para uno de los ejercicios entrenados tenemos los siguientes resultados:

Historial de entrenamiento

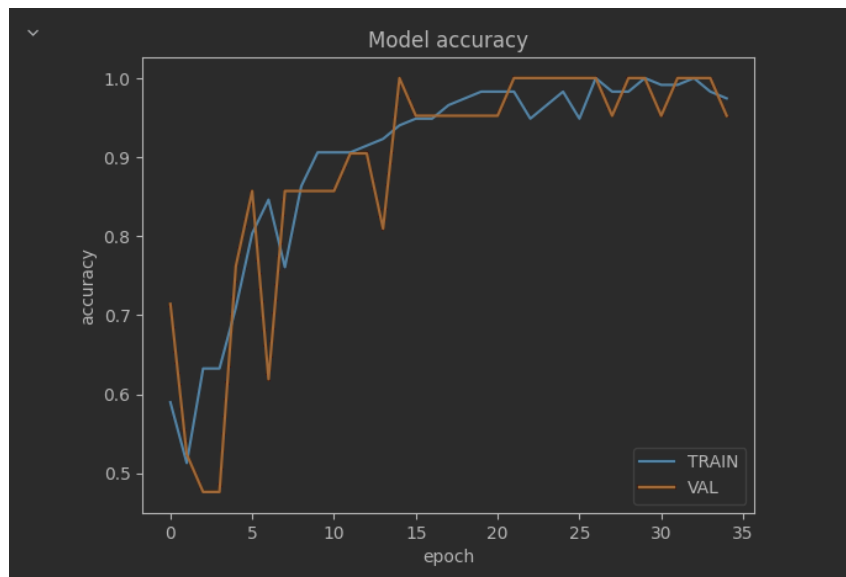


Ilustración 24. Historial de entrenamiento

Matriz de confusión

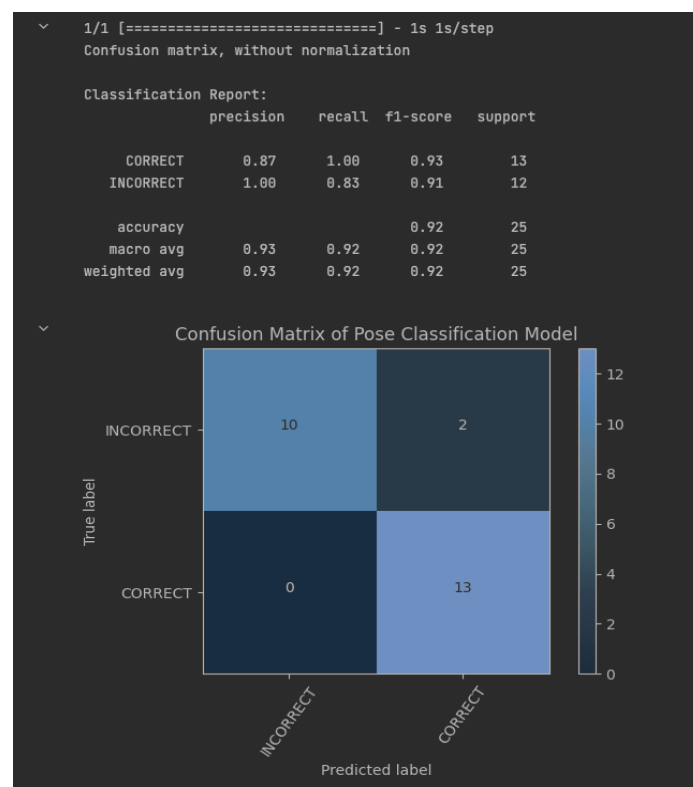


Ilustración 25. Matriz de confusión

Estos resultados indican que el modelo ha tenido un rendimiento bastante bueno al clasificar el ejercicio en el conjunto de datos de prueba. La precisión es la proporción de identificaciones positivas que fueron correctas.

En este caso, la precisión del modelo es del **92%**, lo que significa que el 92% de las veces el modelo predijo la clase correcta. El modelo parece estar haciendo un buen trabajo al clasificar los ejercicios como correctos o incorrectos.

4.4 ESTADO ACTUAL DEL PROYECTO

En el inicio del proyecto, se propuso un plan detallado que definía varias etapas de desarrollo, que iban desde el diseño inicial y la definición de requisitos hasta la implementación de las funcionalidades y las pruebas.

Hasta ahora, la mayor parte del proyecto ha seguido la planificación propuesta. Sin embargo, se han identificado algunas desviaciones con respecto a esta planificación que se detallan a continuación:

Retraso en la integración del modelo de inteligencia artificial: La integración del modelo de inteligencia artificial se retrasó debido a que la captura de datos se realizó de forma manual en esta fase del proyecto. Esto resultó ser un proceso más lento y laborioso de lo que se había anticipado originalmente.

Para abordar este retraso, se decidió dedicar más tiempo a la fase de captura de datos. Adicionalmente se ha propuesto para versiones posteriores la implementación de un sistema automático para la captura de datos. Este sistema debería facilitar y agilizar el proceso de entrenamiento de los modelos, lo que podría ayudar a evitar retrasos similares en el futuro.

Complejidad en el desarrollo de algunas funcionalidades: Algunas funcionalidades resultaron ser más complejas de lo anticipado inicialmente, lo cual llevó a un retraso en su implementación.

Para lidiar con esta situación se ha llevado a cabo una revisión de estas funcionalidades para entender la causa de su complejidad y buscar soluciones alternativas. En algunos casos se ha optado por simplificar las funcionalidades para asegurar su implementación en el tiempo previsto.

5. PRUEBAS

5.1 PLAN DE PRUEBAS

Nuestra estrategia de pruebas para la aplicación Intellifit se centra en garantizar la máxima calidad y fiabilidad del software, validando tanto la funcionalidad individual como la integración de los componentes del sistema. Para lograr esto, emplearemos una combinación de pruebas unitarias y de integración.

Las pruebas unitarias se llevarán a cabo con el fin de verificar la funcionalidad de los componentes individuales de la aplicación. Este tipo de pruebas ayuda a detectar errores a nivel de función y asegura que cada parte del código funcione como se espera.

Las pruebas de integración nos permitirán validar la correcta interacción entre los diferentes componentes del sistema. Al considerar conjuntos de componentes que interactúan entre sí, estas pruebas pueden identificar problemas que no se detectarían en las pruebas unitarias.

Para implementar este plan de pruebas, utilizaremos un conjunto de herramientas y marcos de trabajo que faciliten la tarea. Para las pruebas unitarias e integración utilizaremos JUnit un framework de pruebas para Java ampliamente utilizado que ofrece un conjunto de anotaciones y aserciones para estructurar y realizar las pruebas.

5.2 PRUEBAS UNITARIAS

Este conjunto de pruebas unitarias se está ejecutando para la clase `LocalLocalDataSourceImpITest` que es una implementación de la interfaz

LocalDataSource en el proyecto. Esta interfaz es responsable de las operaciones de los datos locales en la aplicación, como guardar, recuperar y eliminar capturas.

Prueba de guardado capturas (saveCapture): Esta prueba verifica que el método `saveCapture` de la fuente de datos local está funcionando correctamente. Crea y guarda 10 capturas y luego verifica que se hayan guardado correctamente comprobando que su recuento es correcto y que los datos guardados coinciden con los datos originales.

```
diegolog
@Test
fun saveCapture() = runTest {
    (0 until 10).forEach { _ ->
        localDataSource.saveCapture(sampleCapture())
    }
    val saveCaptures= localDataSource.getCaptures()

    assertEquals( expected: 10, saveCaptures.size)
    assertEquals(MoveType.CORRECT, saveCaptures[0].moveType)
    assertEquals( expected: 12, saveCaptures[0].samples.size)
    assertEquals( expected: 17, saveCaptures[0].samples[0].keyPoints.size)
}
```

Ilustración 26. Testing SaveCapture

Prueba de eliminar de Capturas (deleteCapture): Esta prueba verifica que el método `deleteCapture` de la fuente de datos local está funcionando correctamente. Crea y guarda una captura, luego elimina esa captura y verifica que la captura se ha eliminado correctamente comprobando que la cantidad de capturas es cero.

```
diegolog
@Test
fun deleteCapture() = runTest {
    localDataSource.saveCapture(sampleCapture())
    var saveCapture = localDataSource.getCaptures()
    assertEquals( expected: 1, saveCapture.size)
    localDataSource.deleteCapture(saveCapture[0].id)
    saveCapture = localDataSource.getCaptures()
    assertEquals( expected: 0, saveCapture.size)
}
```

Ilustración 27. Testing deleteCapture

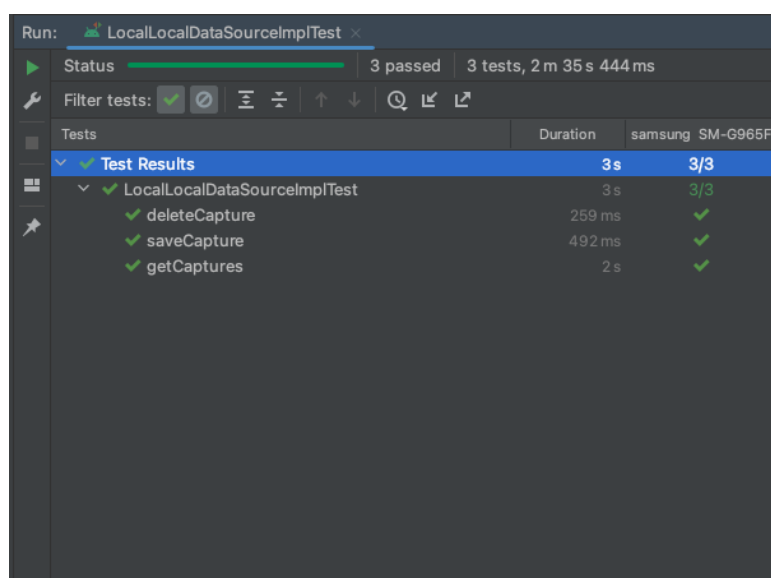
Prueba obtener capturas (getCaptures): Esta prueba verifica que el método getCaptures de la fuente de datos local está funcionando correctamente. Crea y guarda 100 capturas y luego verifica que todas las capturas se hayan recuperado correctamente comprobando que su recuento es igual al número original de capturas guardadas.

```
diegolog
@Test
fun getCaptures() = runTest {
    val size = 100
    (0..size-1).forEach { _ ->
        localDataSource.saveCapture(sampleCapture())
    }

    val saveCapture = localDataSource.getCaptures()
    assertEquals(size, saveCapture.size)
}
```

Ilustración 28 Testing getCaptures

Estas pruebas se ejecutan con el uso de corutinas de Kotlin para garantizar que las operaciones asíncronas se ejecuten de manera sincrónica durante las pruebas, lo que permite un control más preciso del flujo de ejecución durante las pruebas. Después de cada prueba, se ejecuta el método tearDown para limpiar cualquier cambio realizado por la prueba y así garantizar que cada prueba se ejecute en un entorno aislado.



Test Results	Duration	Result
LocalLocalDataSourceImplTest	3 s	3/3
deleteCapture	259 ms	✓
saveCapture	492 ms	✓
getCaptures	2 s	✓

Ilustración 29. Resultado testing

6. CONCLUSIONES

Durante el desarrollo de este proyecto, he adquirido conocimientos y habilidades en el diseño y desarrollo de una aplicación móvil para guiar a los usuarios en la ejecución correcta de ejercicios físicos. También experiencia en la utilización de técnicas de inteligencia artificial, específicamente redes neuronales convolucionales, para detectar la postura y movimiento del usuario durante los ejercicios. Esta experiencia me ha permitido comprender la importancia de brindar retroalimentación en tiempo real y personalizada a los usuarios.

En general, se ha logrado satisfactoriamente la mayoría de los objetivos planteados inicialmente. Se desarrolló una aplicación móvil funcional que permite a los usuarios explorar y realizar una variedad de entrenamientos, recibir retroalimentación en tiempo real sobre la ejecución de los ejercicios. Sin embargo, algunos objetivos específicos podrían haber sido abordados de manera más completa y algunas funcionalidades podrían mejorarse en futuras iteraciones del proyecto.

En lo referente al seguimiento de la planificación, es importante destacar que se encontró con desafíos adicionales durante el desarrollo del proyecto. Uno de los principales desafíos fue la captura manual de datos para entrenar los modelos de TensorFlow. Este proceso resultó más laborioso y requirió más tiempo del inicialmente estimado. Debido a la complejidad de capturar y etiquetar correctamente los datos, se requirió un esfuerzo adicional para asegurar la calidad de los modelos de detección de ejercicios.

Esta desviación en el tiempo invertido en la captura de datos tuvo un impacto en la planificación, ya que repercutió en el tiempo disponible para otras tareas. Aunque se intentó mitigar este impacto realizando ajustes en la planificación, se reconoce que la falta de experiencia en la estimación de tiempo y en la complejidad de la captura de datos fue un factor que influyó en esta desviación.

Se plantea líneas de trabajo futuro en áreas que no se pudo explorar completamente en este trabajo y que quedan pendientes para futuras versiones. Estas áreas incluyen la implementación de funcionalidades sociales para fomentar la interacción entre los usuarios, y la incorporación de un sistema de recomendación personalizado basado en las preferencias y objetivos de cada usuario. Estas líneas de trabajo futuro tienen el potencial de mejorar aún más la experiencia de los usuarios y hacer que la aplicación sea más completa y personalizada.

En resumen, este proyecto me ha brindado una valiosa experiencia en el diseño y desarrollo de una aplicación móvil.

4. GLOSARIO

Android: Sistema operativo para dispositivos móviles desarrollado por Google.

iOS: Sistema operativo para dispositivos móviles desarrollado por Apple.

Inteligencia artificial: Área de la informática que se ocupa del diseño de algoritmos y sistemas que imitan la inteligencia humana, y que son capaces de aprender y mejorar a partir de la experiencia.

Visión por computadora: Área de la inteligencia artificial que se encarga del procesamiento de imágenes y videos, para que las máquinas sean capaces de interpretar la información visual y tomar decisiones a partir de ella.

TensorFlow: Biblioteca de software de código abierto para realizar cálculos numéricos, especialmente en el ámbito del aprendizaje automático y la inteligencia artificial.

Redes neuronales convolucionales: Tipo de red neuronal artificial que se utiliza en el aprendizaje automático y la visión por computadora, especialmente

para el reconocimiento de imágenes. Está compuesta por capas que se encargan de procesar la información visual de forma progresiva y jerárquica.

Figma: Es una aplicación web colaborativa para el diseño de interfaces, con funciones adicionales fuera de línea habilitadas por aplicaciones de escritorio para macOS y Windows.

ViewModels: Son componentes componentes de la arquitectura MVVM que se utilizan en Android para separar la lógica empresarial de la interfaz de usuario.

API: Conjunto de rutinas, protocolos y herramientas para la construcción de software y aplicaciones. Una API especifica cómo deben interactuar los componentes de software.

SOLID: Acrónimo de los cinco principios de diseño orientados a objetos y de programación para hacer que el software sea más comprensible, flexible y mantenible.

5. BIBLIOGRAFÍA

- Implementing a CNN in TensorFlow. Consultada el 4 de marzo de 2023 en: [Implementing a CNN in TensorFlow & Keras \(learnopencv.com\)](https://www.learnopencv.com/implementing-a-cnn-in-tensorflow-and-keras/)
- Android para desarrolladores. Consultada el 5 de marzo de 2023 en: [Desarrolladores de Android | Android Developers](https://developer.android.com/)
- MoveNet: modelo de detección de pose ultrarrápido y preciso. Consultada el 5 de marzo de 2023 en: [MoveNet: modelo de detección de pose ultrarrápido y preciso. | TensorFlow Hub](https://www.tensorflow.org/research/movenet)
- Unsplash: Más de 2 millones de imágenes de alta definición gratuitas. Consultado el 23 de marzo del 2023: [Hermosas imágenes y fotos gratuitas | Unsplash](https://unsplash.com/)
- Design and build beautiful, usable products with Material 3. Consultado el 24 de marzo del 2023: [Material Design](https://material.io/design/)
- Diseño Centrado en el Usuario para dispositivos móviles. Consultado el 23 de marzo del 2023: [Diseño Centrado en el Usuario para dispositivos móviles - XWiki \(uoc.edu\)](https://xwiki.uoc.edu/wiki/xwiki/bin/view/Main/UXDesign)
- Guía de arquitectura de apps: Consultado el jueves 6 de abril del 2023: [Guía de arquitectura de apps | Desarrolladores de Android | Android Developers](https://developer.android.com/guide/architecture)
- Cómo guardar contenido en una base de datos local con Room: Consultado el 1 de Mayo del 2023: [Cómo guardar contenido en una base de datos local con Room | Android Developers](https://developer.android.com/topic/libraries/architecture/room)
- TensorFlow Lite para Android: Consultado el 5 de mayo de 2023: [TensorFlow Lite para Android](https://www.tensorflow.org/lite)
- Descripción general de ViewModel. Consultado el 1 de mayo de 2023: [Descripción general de ViewModel | Desarrolladores de Android | Android Developers](https://developer.android.com/topic/libraries/architecture/viewmodel)
- Flujo de trabajo de Gitflow. Consultado el 2 de mayo del 2023: [Flujo de trabajo de Gitflow | Atlassian Git Tutorial](https://www.atlassian.com/git/tutorials/setting-up-a-work-flow)

- Cómo configurar variantes de compilación. Consultado el 10 de mayo del 2023: [Cómo configurar variantes de compilación | Desarrolladores de Android | Android Developers](#)

6. ANEXOS

ANEXO 1. ENTREVISTAS

Entrevista a un usuario con poca experiencia.

- ¿Cuáles son tus hábitos de ejercicio físico?
Suelo hacer ejercicio unas tres veces por semana. Me gusta correr y hacer algunas rutinas de entrenamiento de fuerza en casa.
- ¿Qué te motiva a hacer ejercicio?
Principalmente, me gusta mantenerme en forma y saludable. Además, me ayuda a despejar mi mente y liberar estrés.
- ¿Has utilizado alguna aplicación móvil para hacer ejercicio?
Sí, he utilizado algunas aplicaciones en el pasado, pero no me han gustado mucho. Siento que no son muy personalizadas y no se adaptan a mis necesidades.
- ¿Qué funcionalidades te gustaría ver en una aplicación para hacer ejercicio?
Me gustaría que la aplicación me permita crear mis propias rutinas de ejercicio, de acuerdo con mis necesidades y objetivos. También me gustaría que tenga una guía de técnica para asegurarme de estar realizando los ejercicios de manera correcta y segura. Por último, me gustaría que la aplicación tenga un sistema de seguimiento de progreso, para poder ver cómo estoy mejorando.
- ¿Te gustaría que la aplicación tenga alguna función adicional?

Sería útil si la aplicación pudiera recordarme cuándo es el momento de hacer ejercicio y enviarme notificaciones para motivarme.

- ¿Como utilizarías esta aplicación?

Utilizaría esta aplicación para ayudarme a mantener una rutina de ejercicios y asegurarme de que estoy haciendo los movimientos de manera correcta. Me gusta la idea de tener una retroalimentación en tiempo real para poder corregir mi técnica y evitar lesiones. También me gustaría poder crear mis propias rutinas personalizadas y realizar un seguimiento de mi progreso a lo largo del tiempo.

Entrevista a un entrenador personal.

- ¿Cuáles son tus hábitos de ejercicio físico?

Soy un entrenador personal, así que suelo hacer ejercicio de forma regular, al menos cinco veces por semana. Me gusta combinar diferentes tipos de entrenamiento, como el levantamiento de pesas, el cardio y el entrenamiento de alta intensidad.

- ¿Qué te motiva a hacer ejercicio?

Me encanta el ejercicio y los beneficios que aporta a mi salud y bienestar. También es importante para mí mantenerme en forma para poder ayudar a mis clientes a lograr sus objetivos.

- ¿Has utilizado alguna aplicación móvil para hacer ejercicio?

Sí, he utilizado varias aplicaciones en el pasado para hacer ejercicio y también para ayudar a mis clientes a mantenerse en forma.

- ¿Qué funcionalidades te gustaría ver en una aplicación para hacer ejercicio?

Me gustaría que la aplicación incluya una amplia variedad de ejercicios, desde los más básicos hasta los más avanzados. También sería útil si la aplicación pudiera adaptarse a diferentes niveles de condición física, y ofrecer recomendaciones personalizadas de ejercicios en función de los objetivos y necesidades de cada usuario.

Además, sería importante que la aplicación incluya una guía de técnica para cada ejercicio, así como una demostración en video, para asegurarnos de que los usuarios están realizando los movimientos de manera correcta y segura.

Por último, me gustaría que la aplicación incluya un sistema de seguimiento de progreso, para que los usuarios puedan ver cómo están progresando y establecer objetivos realistas para su entrenamiento.

- ¿Te gustaría que la aplicación tenga alguna función adicional?

Sí, sería útil si la aplicación pudiera integrarse con dispositivos de seguimiento de actividad, como relojes inteligentes o bandas de fitness, para que los usuarios puedan tener una idea clara de su nivel de actividad diaria. También sería interesante si la aplicación pudiera incluir una función de nutrición para ayudar a los usuarios a mantener una dieta saludable.

- ¿Como utilizarías esta aplicación?

Utilizaría esta aplicación para planificar las rutinas de entrenamiento de mis clientes y hacer un seguimiento de su progreso. Me gustaría poder personalizar las rutinas para cada cliente y tener una guía de técnica para asegurarme de que los ejercicios se realicen correctamente. También me gustaría poder ver su progreso y modificar sus rutinas si es necesario.