



UNIVERSIDAD  
POLITÉCNICA  
DE MADRID



# Desarrollo de una aplicación de gimnasio en Android para personas con discapacidad cognitiva.

Proyecto Fin de Grado

Grado en Sistemas de información

Autor: Jiliang Qiu Wang

Tutor: Bernardo Tabuenca Archilla

Fecha: 10 de Julio 2025

Universidad Politécnica de Madrid  
Escuela Técnica Superior de Ingeniería de  
Sistemas Informáticos



Desarrollo de una aplicación de gimnasio en Android  
para personas con discapacidad cognitiva.

Grado en sistemas de información

Curso académico 2024-2025

Autor:

Jiliang Qiu Wang

Tutores:

Bernardo Tabuenca Archilla

## Resumen

El objetivo de este Proyecto de Final de Grado ha sido diseñar y crear una aplicación móvil para personas con discapacidad cognitiva, con el propósito de ayudarles en sus rutinas de ejercicio indoor y facilitar la gestión de sus actividades físicas diarias. La aplicación también permitirá crear perfiles de usuario personalizados y podrá seleccionar las actividades que vea convenientes.

Se plantea desarrollar una aplicación móvil intuitiva y fácil de usar, diseñada para registrar y guiar las sesiones de ejercicio en tiempo real, utilizando interfaces intuitivas para mejorar la experiencia de usuario.

En el presente documento se explicarán todas las etapas del ciclo de vida del producto software, desde la fase inicial del diseño hasta la posterior publicación para los usuarios finales.

## Palabras clave

Flutter, Android, Dart, discapacidad

## Abstract

The objective of this Final Degree Project has been to design and create a mobile application for people with cognitive disabilities, with the purpose of helping them in their indoor exercise routines and facilitating the management of their daily physical activities.

The application will also allow you to create personalized user profiles and you can select the activities that you see fit. It is proposed to develop an intuitive and easy-to-use mobile application, designed to record and guide exercise sessions in real time, using intuitive interfaces to improve the user experience.

This document will explain all stages of the software product life cycle, from the initial design phase to the subsequent publication for end users.

## Key words

Flutter, Android, Dart, Disable

## Indices

|   |    |
|---|----|
| 1. Introducción.....  | 1  |
| 1.1. Contexto.....  | 1  |
| 1.2. Objetivos.....   | 1  |
| 1.3. Estructura del documento.....  | 2  |
| 2. Estado del arte.....   | 2  |
| 2.1. Aplicaciones más populares.....  | 3  |
| 2.2. Soluciones existentes.....   | 4  |
| 2.2.1. Accessercise.....  | 4  |
| 2.2.2. Wheel With Me Fitness.....   | 4  |
| 2.2.3. Adaptive Yoga Live.....  | 6  |
| 3. Desarrollo del proyecto.....   | 6  |
| 3.1. Desarrollo del proyecto.....   | 7  |
| 3.2. Análisis de requisitos.....  | 8  |
| 3.2.1. RU-1: Gestión de usuarios (selección, login, creación y ocultación)..... | 8  |
| 3.2.2. RU-2: Inicio de sesión con contraseña.....                               | 8  |
| 3.2.3. RU-3: Registro de nuevo usuario.....                                     | 9  |
| 3.2.4. RU-4: Página principal (Home) y navegación.....                          | 9  |
| 3.2.5. RU-5: Gestión de ejercicios.....   | 9  |
| 3.2.6. RU-6: Ejercicios por día (rutina semanal).....                           | 9  |
| 3.2.7. RU-7: Registro de peso.....  | 10 |
| 3.2.8. RU-8: Edición de registros de peso.....                                  | 10 |
| 3.2.9. RU-9: Eliminación de registros de peso.....                              | 10 |
| 3.2.10. RU-10: Visualización gráfica del peso.....                              | 10 |
| 3.2.11. RU-11: Gestión de comidas (detalles).....                               | 10 |
| 3.2.12. RU-12: Visualización de comidas.....                                    | 10 |
| 3.2.13. RU-13: Agregar comidas.....   | 11 |
| 3.2.14. RU-14: Edición de comidas.....  | 11 |
| 3.2.15. RU-15: Eliminación de comidas.....                                      | 11 |
| 3.2.16. RU-16: Asignación y gestión de comidas por día y categoría.....         | 11 |
| 3.2.17. RU-17: Acceso a comidas por día y categoría.....                        | 11 |
| 3.2.18. RU-18: Acceso a la planificación semanal de comidas.....                | 11 |
| 3.2.19. RU-19: Calendario y planificación de entrenamientos.....                | 11 |
| 3.3. Requisitos no funcionales.....   | 12 |
| 3.3.1. Compatibilidad.....  | 12 |

|   |    |
|---|----|
| 3.3.2. Rendimiento y eficiencia.....              | 12 |
| 3.3.3. Extensibilidad y mantenibilidad .....      | 12 |
| 3.3.4. Interoperabilidad .....                    | 12 |
| 3.3.5. Recuperabilidad .....                      | 12 |
| 3.3.6. Confiabilidad .....                        | 13 |
| 3.3.7. Robustez.....                              | 13 |
| 3.3.8. Seguridad .....                            | 13 |
| 3.3.9. Integridad.....                            | 13 |
| 3.3.10. Usabilidad.....                           | 13 |
| 3.3.11. Disponibilidad .....                      | 13 |
| 3.3.12. Accesibilidad .....                       | 13 |
| 3.4. Diseño .....                                 | 14 |
| 3.4.1. Sistema operativo usado.....               | 14 |
| 3.4.2. Prototipo .....                            | 14 |
| 3.4.3. DISEÑO INICIAL .....                       | 18 |
| 3.5. Diagramas .....                              | 32 |
| 3.5.1. Diagrama de casos de uso.....              | 32 |
| 3.5.2. Especificación de casos de uso .....       | 33 |
| 3.5.3. Diagrama de clases .....                   | 46 |
| 3.6. Implementación de la solución.....           | 47 |
| 3.6.1. Herramientas y entorno de desarrollo. .... | 47 |
| 3.6.2. Arquitectura.....                          | 49 |
| 4. Resultado .....                                | 50 |
| 4.1. Resultados obtenidos.....                    | 50 |
| 4.2. Tablas de pruebas.....                       | 55 |
| 5. Conclusión.....                                | 61 |
| 5.1. conclusión .....                             | 61 |
| 5.2. Impacto social y medioambiental.....         | 61 |
| 6. Bibliografía .....                             | 63 |

## Índice de figuras

|  |    |
|--|----|
| Ilustración 1: Ejemplo de introducción de datos (captura de Gym WP Ejercicio de Gimnasio) .....                              | 3  |
| Ilustración 2 Prototipo de la app (Creación propia) con la herramienta Diagrams .....  | 15 |
| Ilustración 3 Prototipo de iniciar sesión Creación propia con diagrams .....   | 16 |
| Ilustración 4 Prototipo de inicio creación propia con diagrams .....   | 16 |
| Ilustración 5 Prototipo de Ejercicio creación propia con Diagrams .....  | 17 |
| Ilustración 6 Prototipo de dieta creación propia con Diagrams .....  | 17 |
| Ilustración 7 Prototipo Calendario Creación propia con Diagrams .....  | 18 |
| Ilustración 8 foto de UISTORE.Design .....   | 19 |
| Ilustración 9 foto de UISTORE.Design login .....   | 19 |
| Ilustración 10 foto de UISTORE.Design rutina .....   | 19 |
| Ilustración 11 foto de UISTORE.Design pagina principal .....   | 19 |
| Ilustración 12 foto de UISTORE.Design Calendario .....   | 20 |
| Ilustración 13 foto de UISTORE.Design peso .....   | 20 |
| Ilustración 14 foto de UISTORE.Design timer .....  | 20 |
| Ilustración 15 foto de UISTORE.Design detalles ejercicio .....   | 20 |
| Ilustración 16 Segundo prototipo creación propia con Android Studio .....  | 22 |
| Ilustración 17 Segundo prototipo creación propia con Android Studio .....  | 22 |
| Ilustración 18 Segundo prototipo creación propia con Android Studio .....  | 22 |
| Ilustración 19 Segundo prototipo creación propia con Android Studio .....  | 23 |
| Ilustración 20 Segundo prototipo creación propia con Android Studio Pantalla inicial .....                                   | 23 |
| Ilustración 21 Segundo prototipo creación propia con Android Studio Iniciar sesión y registrarse .....                       | 24 |
| Ilustración 22 Segundo prototipo creación propia con Android Studio Página principal .....                                   | 25 |
| Ilustración 23 Segundo prototipo creación propia con Android Studio menú de navegación .....                                 | 26 |
| Ilustración 24 Segundo prototipo creación propia con Android Studio menú de navegación Página principal a Ejercicios .....   | 26 |
| Ilustración 25 Segundo prototipo creación propia con Android Studio Menu de navegación Página principal a rutina .....       | 27 |
| Ilustración 26 Segundo prototipo creación propia con Android Studio Menu de navegación Rutina semanal a rutina del día ..... | 28 |
| Ilustración 27 Segundo prototipo creación propia con Android Studio Página principal a logros .....                          | 29 |
| Ilustración 28 Segundo prototipo creación propia con Android Studio Página principal a nutrición ...                         | 30 |
| Ilustración 29 Segundo prototipo creación propia con Android Studio Comidas .....  | 30 |
| Ilustración 30 Segundo prototipo creación propia con Android Studio proceso de añadir comida a dieta .....                   | 31 |
| Ilustración 31 Segundo prototipo creación propia con Android Studio Peso y calendario .....                                  | 31 |
| Ilustración 32: Diagrama de clases .....   | 46 |
| Ilustración 33: Página de inicio de sesión Creación propia .....   | 50 |
| Ilustración 34: pantalla de inicio de sesión y registro creación propia .....  | 51 |
| Ilustración 35: Pantalla principal Creación propia .....   | 51 |
| Ilustración 36: Cambiar foto de perfil .....   | 52 |
| Ilustración 37: Pantalla de ejercicios creación propia .....   | 52 |
| Ilustración 38: creación de ejercicios y edición de ejercicios .....   | 53 |

|   |    |
|---|----|
| Ilustración 39: Inserción de peso ..... | 53 |
| Ilustración 40: Calendario .....        | 54 |

## Índice de tablas

|   |    |
|---|----|
| Tabla 1: Aplicaciones más populares de gym .....          | 4  |
| Tabla 2: Casos de uso extendido “iniciar sesión” .....    | 33 |
| Tabla 3: Casos de uso extendido “Registrar usuario” ..... | 34 |
| Tabla 4: Ejercicios.....                                  | 35 |
| Tabla 5:Añadir ejercicio .....                            | 36 |
| Tabla 6:Editar ejercicio.....                             | 36 |
| Tabla 7: Eliminar ejercicio .....                         | 37 |
| Tabla 8: Ver rutina.....                                  | 38 |
| Tabla 9: Ver comida .....                                 | 38 |
| Tabla 10: Añadir comida .....                             | 39 |
| Tabla 11: Editar comida .....                             | 40 |
| Tabla 12:Eliminar Comida .....                            | 41 |
| Tabla 13: Ver dieta .....                                 | 41 |
| Tabla 14: Ver logros .....                                | 42 |
| Tabla 15: Ejercicio detalle .....                         | 42 |
| Tabla 16:Ver peso .....                                   | 43 |
| Tabla 17: Añadir peso .....                               | 44 |
| Tabla 18: Ver calendario .....                            | 44 |
| Tabla 19: Validar datos .....                             | 45 |
| Tabla 20: Cerrar sesión .....                             | 46 |
| Tabla 21: pruebas sobre la aplicación.....                | 58 |

# 1. Introducción

## 1.1. Contexto

Desde siempre las personas con discapacidad desafortunadamente han tenido que enfrentarse a numerosos problemas y dificultades para poder participar en actividades físicas y acceder a herramientas adaptadas para esas personas. La situación ha ido a peor ya que muchas veces no se consideran estas necesidades importantes ya que este colectivo es una minoría[1], [2], [3].

La falta de accesibilidad y adaptación en las aplicaciones de fitness ha llevado a que muchas personas con discapacidad se vean alejadas de los beneficios del ejercicio físico afectando así negativamente su salud física y mental.

Con conocimiento de este problema el desarrollo de una aplicación de gimnasio adaptada a estas personas se vuelve crucialmente importante. Gracias a esta herramienta no solo se ayudará a su inclusión en las actividades físicas, sino que también contribuirá a la mejora de su autonomía y la salud del individuo [4], [5]. La aplicación tendrá interfaces intuitivas para asegurar que los entrenamientos sean adaptados y personalizados así teniendo una experiencia segura y efectiva.

La Universidad Politécnica de Madrid (UPM), comprometida con la inclusión y la accesibilidad para todo tipo de personas, se ha involucrado ya en diversos proyectos para apoyar a personas con discapacidad. La aplicación tiene varios objetivos clave:

**Facilitar la Inclusión:** Gracias a esta aplicación fomentaremos el hacer ejercicio a este colectivo.

**Mejorar la Autonomía:** Se da una aplicación fácil de usar para que se pueda seguir una rutina.

**Promover la Salud y el Bienestar:** Gracias a la aplicación algunas personas podrán empezar a hacer ejercicio lo cual mejorará tanto como su condición física y mental.

**Apoyo Personalizado:** La aplicación contiene ejercicios expresamente adaptados a este colectivo de personas.

El desarrollo de esta aplicación no solo facilitará el acceso más fácil al ejercicio físico para personas que tengan dificultades, sino también darles confianza para que se vuelvan independientes, mejorando su calidad de vida, confianza y apoyar a una mayor inclusión en actividades saludables. La UPM con este proyecto se compromete de nuevo otra vez con la inclusión y el bienestar de todas las personas sin importar su condición.

## 1.2. Objetivos

En numerosas ocasiones estas personas no tienen recursos adaptados ni accesibles para que puedan facilitar su participación en actividades físicas. La falta de herramientas adaptadas puede producir frustración o desmotivación y así afectar en la productividad de los entrenamientos haciendo que el entrenamiento no sea tan eficiente.

El objetivo de este Proyecto es crear una aplicación móvil que pueda ayudar a las personas discapacitadas en sus actividades físicas indoor, haciendo más fácil la gestión y el seguimiento de sus actividades físicas. La aplicación permitirá la personalización de los entrenamientos con ejercicios adaptados para todo tipo de usuarios.

La plataforma que se ha elegido ha sido el dispositivo móvil porque es uno de los dispositivos que más personas posee y más personas se pueden permitir, lo que lo convierte en la herramienta más accesible.

Se plantea una aplicación móvil que sea fácil de usar y permite personalizar rutinas de ejercicio de manera completa.

Para cumplir dicho propósito se definen estos objetivos:

- 1- Interfaz de usuario amigable, intuitiva y fácil de interpretar.
- 2- Compatible con la mayoría de los dispositivos: Hemos seleccionado Android [6] porque es uno de los sistemas operativos más comunes, más flexibles y asequibles.
- 3- Permitir el almacenamiento de actividades personalizadas.

La implementación de estos objetivos asegurará que la aplicación no solo sea útil y accesible para personas con discapacidad cognitiva, sino que también fomente su participación en actividades físicas, mejorando su calidad de vida y bienestar general.

### 1.3. Estructura del documento

Ya hecho la contextualización y fijado los objetivos en los puntos anteriores, se explicará a continuación qué estructura tendrá.

En el primer bloque se tratará del estado de arte donde se profundizará la información que se basa el proyecto y se hablará de aplicaciones parecidas.

En el segundo bloque se incluirá todos los requisitos funcionales y no funcionales casos de uso y diagramas respectivos.

En el tercer bloque se explicará la metodología usada y la planificación tanto inicial como final del proyecto.

En el cuarto bloque se abordará los problemas encontrados y logros conseguidos para ello.

Por último se expondrán las conclusiones del proyecto y futuras líneas de trabajo. Adicionalmente se incluirá toda la bibliografía visitada durante la creación del proyecto.

## 2. Estado del arte.

Las aplicaciones de gimnasio no es una idea innovadora ni creativa en el momento que vivimos de hecho se han ido evolucionado añadiendo nuevas tecnologías. Todas las aplicaciones tienen numerosas funcionalidades en común lo cual es lógico es tener una rutina de ejercicios que el usuario debe completar.

Estas aplicaciones siempre tienen la funcionalidad de hacer una rutina según el objetivo que tenga el usuario dependiendo qué aspectos quiere enfocarse el usuario. Con esto quiero decir que si un usuario quiere tener "brazos grandes" la aplicación le recomendará ejercicios enfocados que trabaje o ejercite el área de los brazos. Otro de los aspectos más importantes de estar en forma, aunque algunas apps no se enfocan lo suficiente es la dieta. La principal razón de que no se focalizan en este aspecto es debido a que no es fácil de controlar ya que hay mucha variedad de comida incluso con el mismo alimento porque un mismo producto se puede preparar de diferentes maneras.

Las aplicaciones que se ha investigado cuando se inicia como usuario nuevo piden el peso, la altura, sexo y a donde se quiere enfocar el usuario para genera una rutina. Pero en nuestra aplicación no se

va a poder hacer eso ya que cada persona discapacitada tiene un área donde tiene dificultades y dónde querrá enfocarse.

A screenshot of a fitness application's data entry interface. At the top, there is a progress bar. Below it, the 'Años' (Age) field is set to '21 años'. The 'Peso' (Weight) field is set to '71.5' with units 'kg' and 'lbs' visible. The 'Altura' (Height) field is set to '176' with units 'cm' and 'ft/in' visible. Below these fields is a question '¿Cuál es tu meta de peso?' (What is your weight goal?) with a sub-field 'Peso objetivo' (Target weight) set to 'kg'.

Ilustración 1: Ejemplo de introducción de datos (captura de Gym WP Ejercicio de Gimnasio)

De hecho, las actividades generadas normalmente están pre selectas o generadas por IA automáticamente por lo cual nunca o casi nunca estarán adaptadas a personas con discapacidad ya que las rutinas pre selectas o hechas no van a tener rutinas para una minoría de personas y las generadas por IA no habrán podido entrenar con esos datos al haber tan pocos casos.

La principal diferencia de nuestra aplicación frente a las ya existentes es la personalización que ofrece. Mientras que la mayoría de las apps limitan al usuario a rutinas predefinidas o generadas automáticamente por IA, nuestro proyecto permite crear rutinas completamente personalizadas creadas por el usuario mismo. Incluso la aplicación permite añadir ejercicios propios o adaptados que no se encuentran en rutinas convencionales. Esta funcionalidad es especialmente importante para personas con discapacidad, ya que muchas veces requieren movimientos específicos o adaptaciones que no están en las rutinas estándar. De esta forma, el usuario tiene la libertad de adaptar, modificar o registrar ejercicios que se ajusten a sus capacidades, necesidades y preferencias, convirtiendo la aplicación en una lo más flexible e inclusiva.

## 2.1. Aplicaciones más populares.

| Nombre                    | Tipo de ejercicio                    | Ecosistemas                | Indoor/Outdoor | Sensores                                     | Adaptados discapacidad |
|---------------------------|--------------------------------------|----------------------------|----------------|--|------------------------|
| <b>Nike Training Club</b> | yoga, cardio/HIIT, fuerza, movilidad | Nike FuelBand, Apple Watch | indoor/outdoor | acelerómetro, giroscopio, GPS, magnetómetros | No adaptada.           |

|                              |   |                                   |                |                               |   |
|------------------------------|---|-----------------------------------|----------------|-------------------------------|---|
| <b>Fitbit</b>                | cardio/HIIT, fuerza, movilidad              | GPS, móviles, dispositivos Fitbit | indoor/outdoor | acelerómetro, giroscopio, GPS | No adaptada.  |
| <b>MyFitnessPal</b>          | fuerza, cardio, nutrición                   | Android, iOS, wearables           | indoor/outdoor | depende del dispositivo       | No adaptada.  |
| <b>Sweat (Kayla Itsines)</b> | HIIT, fuerza, movilidad                     | Android, iOS, Apple Watch         | indoor/outdoor | acelerómetro, GPS             | No adaptada   |
| <b>Freeletics</b>            | cardio, HIIT, fuerza                        | Android, iOS                      | indoor/outdoor | acelerómetro (con smartwatch) | No adaptada.  |
| <b>JEFIT</b>                 | fuerza, musculación                         | Android, iOS, Web                 | indoor         | acelerómetro, smartwatch      | No adaptada.  |
| <b>Strong</b>                | fuerza, musculación, movilidad              | Android, iOS, Apple Watch         | indoor         | acelerómetro                  | No adaptada   |
| <b>Wheel With Me Fitness</b> | fuerza, cardio, movilidad (silla de ruedas) | Android, iOS                      | indoor         | depende del dispositivo       | Diseñada específicamente para personas en silla de ruedas.        |
| <b>Accessercise</b>          | fuerza, movilidad, HIIT adaptado            | Android, iOS                      | indoor/outdoor | acelerómetro, GPS             | Totalmente accesible; rutinas diseñadas para discapacidad física. |
| <b>Adaptive Yoga Live</b>    | yoga adaptado                               | Web, móvil (navegador)            | indoor         | sin sensores                  | Clases de yoga específicamente adaptadas a capacidades diversas.  |

Tabla 1: Aplicaciones más populares de gym

## 2.2. Soluciones existentes.

### 2.2.1. Accessercise

Accessercise es una aplicación que se enfoca en actividades físicas para personas que tienen una discapacidad física. Ofrece una amplia biblioteca de ejercicios adaptados, grabados en vídeo por entrenadores con discapacidad, y organizados por categoría. La aplicación permite al usuario poner metas personales, registrar su progreso y recibir motivación desde una comunidad activa y que apoya a sus usuarios. Además, ofrece una función de diario de entrenamiento, seguimiento de rutinas y un foro de discusión.

### 2.2.2. Wheel With Me Fitness

Wheel With Me Fitness es una aplicación diseñada para personas con silla de ruedas. La app incluye programas de cardio, fuerza y movilidad que se pueden hacer en silla de ruedas. Cuenta con rutinas

guiadas en video, planes semanales y ejercicios para diferentes niveles de condición física.

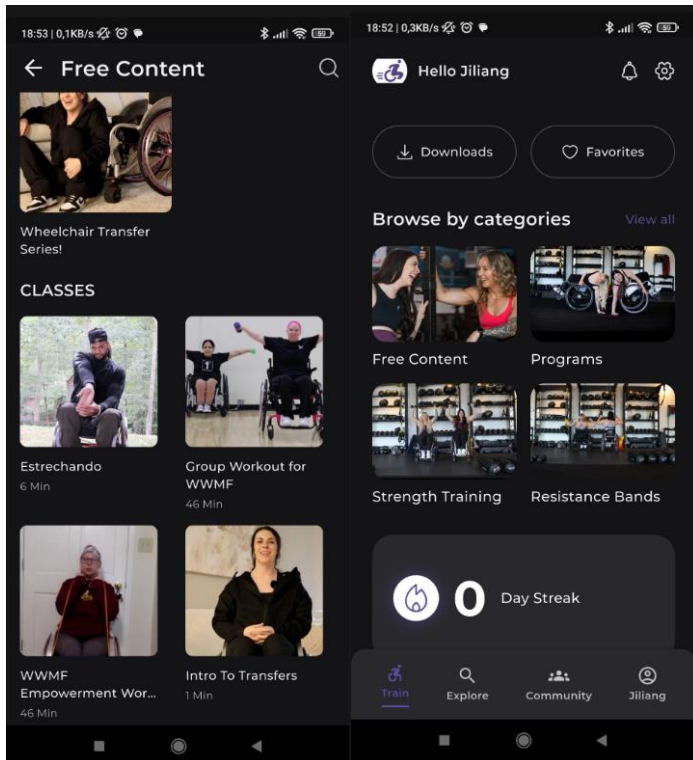


Ilustración 2 captura de pantalla wheel with me

Como se puede apreciar en la Ilustración 2 las aplicaciones no se diferencia entre las aplicaciones de gimnasio convencionales solo que su contenido son orientados a persoans discapitdas en este caso personas en silla de ruedas.

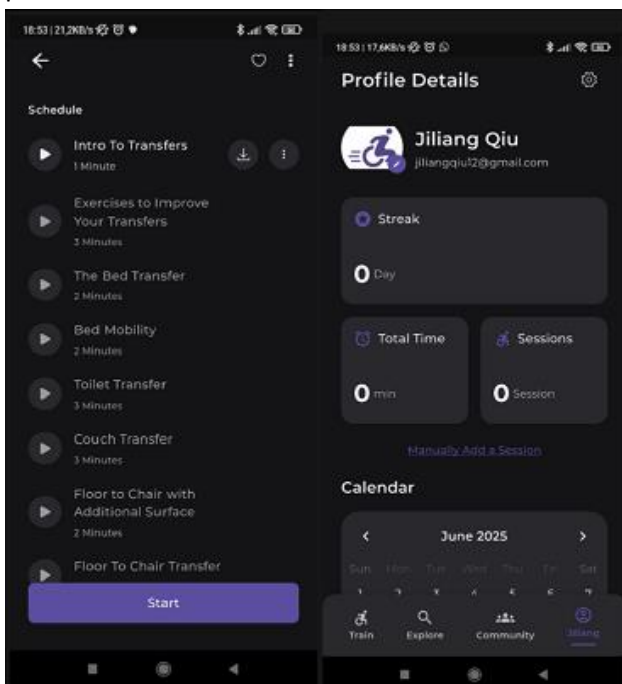


Ilustración 3 Ilustracion rutina perfil de wheel with me

Como se puede apreciar en la ilustración 3 la parte de la izquierda es una rutina que se llega seleccionando una de las tarjetas que se muestran en la ilustración 2 la imagen de la izquierda. Por ultimo se puede apreciar en la ilustracion 3 a la derecha esta el perfil donde hay estadísticas para que el usuario pueda saber su progreso.

### 2.2.3. Adaptive Yoga Live

Adaptive Yoga Live es una plataforma digital orientada a personas con movilidad reducida o personas que padezcan algún tipo de discapacidad. Ofrece clases en vivo y grabadas, dictadas por instructores especializados en yoga accesible.

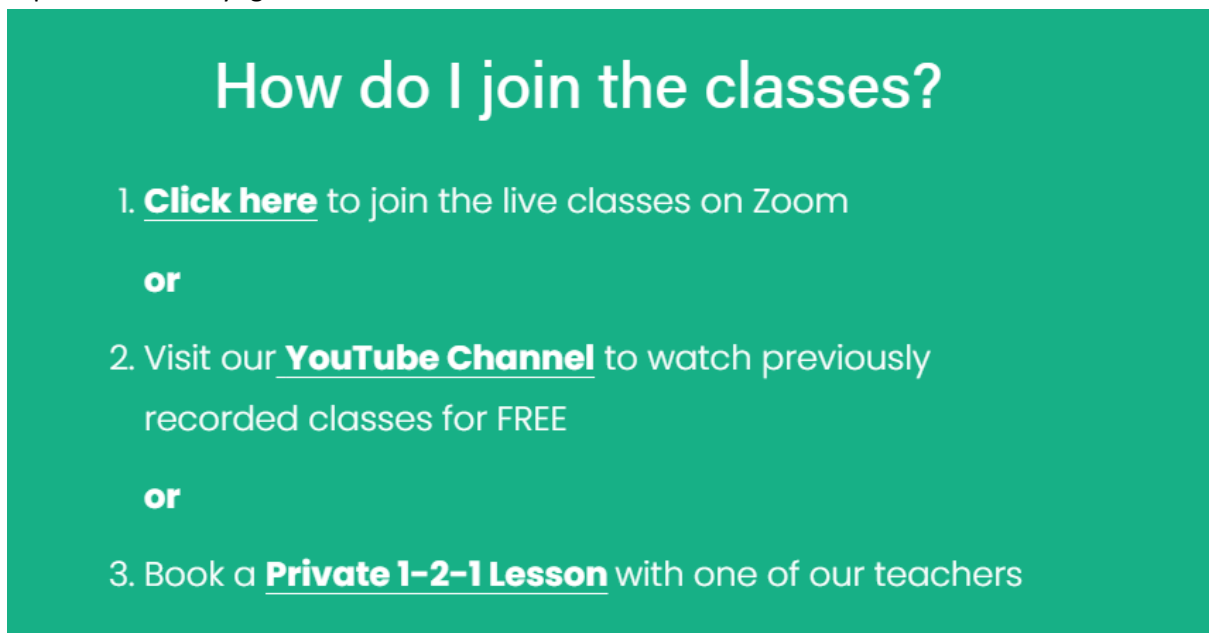


Ilustración 4 Captura de adaptive yoga live

Las sesiones están diseñadas para realizarse completamente sentadas o recostadas, sin tener la necesidad de movimientos complicados. Las sesiones se focalizan en la respiración, la relajación y el fortalecimiento del torso y extremidades superiores.

Las clases se ajustan a diferentes niveles de movilidad y están diseñadas para personas con lesiones medulares, espasticidad, o enfermedades neurodegenerativas.

## 3. Desarrollo del proyecto

En este apartado se describe cómo ha sido el proceso de desarrollo de la aplicación desde la fase de toma de requisitos diseño, implementación y pruebas.

El desarrollo del proyecto consistió en dos etapas: la primera es la descarga e investigación de todas las aplicaciones existentes y posteriormente el desarrollo.

El proyecto inicialmente se empezó con algo básico que es la autenticación del usuario y en un primer momento se pedía el peso, altura y sexo para que generase una serie de ejercicios, pero en el caso de la aplicación que se estaba desarrollando no tenía sentido ya que era para gente limitada. Otra de las ideas descartadas convencional de las Apps de gym fue el GPS ya que esta aplicación se enfoca en el ejercicio indoor de esas personas.

La principal diferencia de nuestra aplicación consistirá en que se podrá recopilar ejercicios especialmente adaptados a esas personas que están limitadas y no pueden realizar algunas actividades porque no pueden hacer ciertas maniobras.

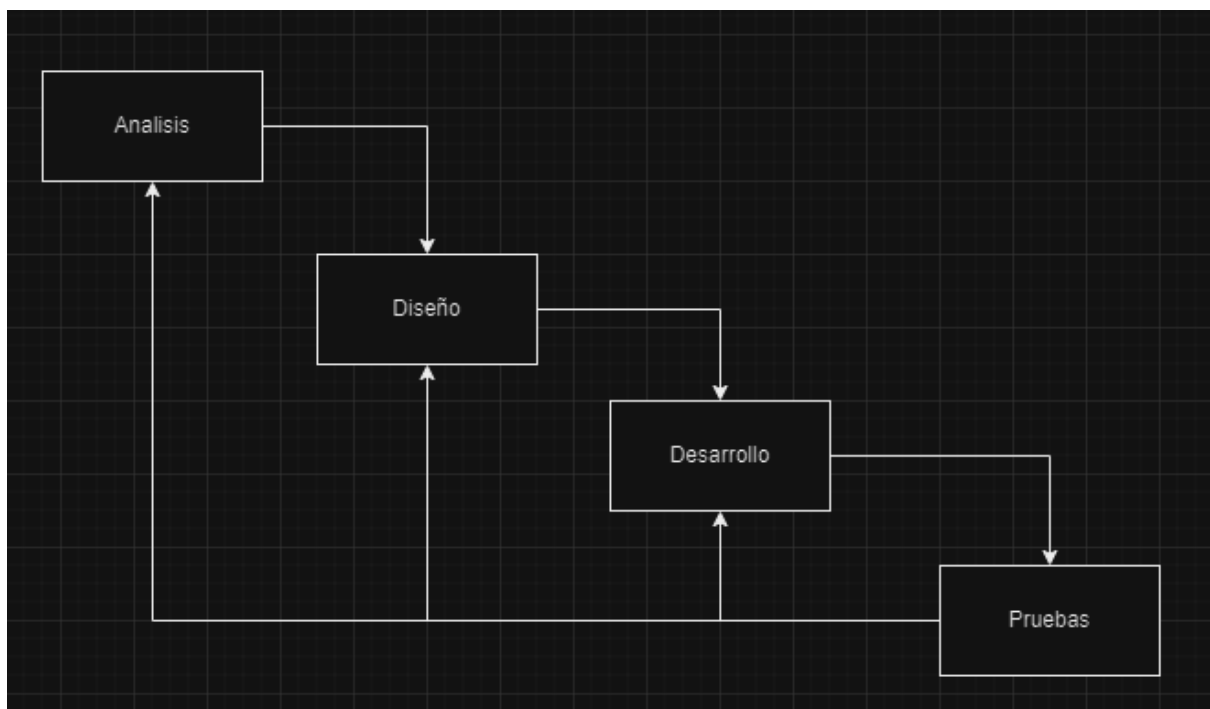
En base a esta información, se empezó a desarrollar la manera de recopilar ejercicios.

La segunda fase del desarrollo comenzó tras dicha investigación y concluyó definiendo las necesidades y requisitos del sistema.

### 3.1. Desarrollo del proyecto

El ciclo de vida que ha utilizado para el desarrollo de la aplicación ha sido el modelo en cascada con retroalimentación. Como se puede apreciar en la imagen 2 este modelo consta de 4 etapas: análisis, diseño implementación y pruebas. En todas las etapas habrá cambios que afecte a las etapas anteriores de este modo produciendo retroalimentación. Y sobre todo poder ir cambiando e ir hacia atrás si ha sugerido ideas nuevas o errores anteriores.

*ilustración 1: Ciclo de vida de modelo en cascada con retroalimentación, (draw.io [7])*



Aquí se hará una descripción de cada fase:

1 **Análisis:** Aquí hizo un estudio sobre qué lenguaje se va a desarrollar la aplicación la cuál fue flutter al ser un lenguaje fácil de manejar. Adicionalmente se ha ido investigando otras aplicaciones del mismo campo.

2. **Diseño:** En la fase de diseño se define la estructura del sistema y cómo están relacionados todos sus componentes. Al principio se hizo un esbozo del prototipo de la aplicación lo que facilitó el desarrollo de la aplicación.

3. **Desarrollo:** Esta parte consiste en implementar todos los esquemas y esbozos hechos anteriormente. Esta fase consistió en 3 fases la primera se desarrolló el software, la segunda consistió en buscar errores y por último en hacer pruebas.

4. **Pruebas:** En esta última fase se realizaron pruebas en su totalidad.

## 3.2. Análisis de requisitos.

Una vez realizado un estudio exhaustivo sobre las aplicaciones existentes en el mercado se procede a establecer los requisitos funcionales y no funcionales del proyecto.

El primer paso para realizar cualquier tipo de proyecto de desarrollo de software es la recopilación de requisitos. Utilizando técnicas de recopilación de información. Esta fase permite al equipo de desarrollo comprender de forma clara y precisa las necesidades de los usuarios, lo cual va a ser esencial para definir posteriormente el diseño, las funcionalidades y el comportamiento del sistema que se va a construir.

### 3.2.1. Requisitos funcionales

#### 3.2.1. RU-1: Gestión de usuarios (selección, login, creación y ocultación)

- **RF-1.1:** El sistema deberá cargar y mostrar la lista de usuarios visibles almacenados en la base de datos o preferencia local.
- **RF-1.2:** Cada usuario deberá mostrarse con su nombre y foto de perfil o un ícono por defecto si no tiene imagen.
- **RF-1.3:** El usuario podrá ocultar un usuario de la lista presionando un botón de "x" junto al usuario, actualizando la lista visible.
- **RF-1.4:** El sistema deberá mostrar un botón para ir a la pantalla de inicio de sesión general.

#### 3.2.2. RU-2: Inicio de sesión con contraseña

- **RF-2.1:** Al seleccionar un usuario de la lista, el sistema deberá mostrar un diálogo solicitando la contraseña para ese usuario.
- **RF-2.2:** El campo de contraseña deberá permitir mostrar u ocultar el texto con un botón de visibilidad.
- **RF-2.3:** Al ingresar la contraseña y confirmar, el sistema deberá validarla.
- **RF-2.4:** Si la contraseña es correcta, el sistema lo redirigirá a la página principal.
- **RF-2.5:** Si la contraseña es incorrecta, el sistema deberá mostrar un mensaje de error visible.

### 3.2.3. RU-3: Registro de nuevo usuario

- **RF-3.1:** El usuario deberá ingresar un nombre de usuario y una contraseña para registrarse.
- **RF-3.2:** El sistema deberá permitir seleccionar una imagen desde la galería como foto de perfil.
- **RF-3.3:** El sistema deberá mostrar la imagen seleccionada en un avatar circular dentro de la pantalla de registro.
- **RF-3.4:** El sistema deberá almacenar el usuario, contraseña y la imagen si se ha seleccionado.
- **RF-3.5:** Al presionar "Registrar", el sistema deberá guardar los datos.
- **RF-3.6:** Si los campos obligatorios están vacíos, se mostrará un mensaje de error.
- **RF-3.7:** Si el registro es exitoso, se mostrará una confirmación y se cerrará la pantalla.
- **RF-3.8:** Si ocurre un error, se deberá mostrar un mensaje de error visible.

### 3.2.4. RU-4: Página principal (Home) y navegación

- **RF-4.1:** El sistema deberá mostrar el nombre de usuario y la foto de perfil en el menú lateral.
- **RF-4.2:** El usuario podrá cambiar su foto de perfil desde el menú lateral.
- **RF-4.3:** Al actualizar la foto de perfil, se mostrará una confirmación.
- **RF-4.4:** El menú lateral deberá permitir la navegación a Ejercicios, Peso, Calendario, Logros, Rutina y Nutrición.
- **RF-4.5:** Al pulsar una sección, el sistema deberá navegar correctamente.
- **RF-4.6:** El sistema deberá permitir cerrar sesión desde el menú lateral.

### 3.2.5. RU-5: Gestión de ejercicios

- **RF-5.1:** El sistema deberá mostrar una lista de ejercicios asociados al usuario.
- **RF-5.2:** Cada ejercicio deberá mostrar título, descripción y miniatura si existe.
- **RF-5.3:** Se podrá eliminar un ejercicio.
- **RF-5.4:** Al tocar un ejercicio, se abrirá una pantalla de detalles editable.
- **RF-5.5:** El botón flotante "+" mostrará un formulario para crear un nuevo ejercicio.
- **RF-5.6:** Se podrá modificar un ejercicio existente.
- **RF-5.7:** Los ejercicios se guardarán en la base de datos.
- **RF-5.8:** La lista se actualizará tras crear, editar o eliminar un ejercicio.
- **RF-5.9:** La foto y video podrán seleccionarse desde archivos locales.
- **RF-5.10:** Imagen y vídeo se mostrarán en la vista de detalle si están disponibles.

### 3.2.6. RU-6: Ejercicios por día (rutina semanal)

- **RF-6.1:** Se deberá cargar la lista de ejercicios asignados al día y usuario actual.
- **RF-6.2:** Se mostrará título, descripción, sets, descanso, peso y foto si está disponible.
- **RF-6.3:** Se podrá marcar/desmarcar un ejercicio como "hecho".
- **RF-6.4:** El estado "hecho" se guardará en la base de datos y se reflejará inmediatamente.
- **RF-6.5:** El usuario podrá abrir los detalles del ejercicio en modo solo lectura.
- **RF-6.6:** Se podrá agregar un ejercicio ya existente al día actual.
- **RF-6.7:** La lista se actualizará automáticamente.
- **RF-6.8:** Si no hay ejercicios asignados, se mostrará un mensaje informativo.

- **RF-6.9:** Se podrá eliminar un ejercicio de un día específico (sin borrarlo de la base).

### 3.2.7. RU-7: Registro de peso

- **RF-7.1:** El usuario podrá ingresar un peso en kilogramos.
- **RF-7.2:** El sistema deberá validar el número ingresado.
- **RF-7.3:** Si el valor no es válido, se mostrará un error.
- **RF-7.4:** Se mostrará una lista de registros con peso y fecha.
- **RF-7.5:** La lista se ordenará mostrando los más recientes primero.

### 3.2.8. RU-8: Edición de registros de peso

- **RF-8.1:** Se podrá editar un peso ya registrado.
- **RF-8.2:** Se validará el número ingresado antes de guardar.
- **RF-8.3:** Si el valor no es válido, se mostrará un error.
- **RF-8.4:** La lista se actualizará con los cambios realizados.

### 3.2.9. RU-9: Eliminación de registros de peso

- **RF-9.1:** Se podrá eliminar un registro de peso.
- **RF-9.2:** La lista y la gráfica se actualizarán tras eliminar un registro.

### 3.2.10. RU-10: Visualización gráfica del peso

- **RF-10.1:** Se mostrará una gráfica de línea con los últimos 10 registros.
- **RF-10.2:** La gráfica solo aparecerá si hay al menos dos registros distintos.
- **RF-10.3:** Si no hay datos suficientes, se mostrará un mensaje correspondiente.

### 3.2.11. RU-11: Gestión de comidas (detalles)

- **RF-11.1:** Se mostrarán título, descripción, peso, puntuación y foto (si existe).
- **RF-11.2:** Si no hay foto, se mostrará un ícono por defecto.
- **RF-11.3:** En modo lectura, los campos serán texto estático.
- **RF-11.4:** En modo edición, se mostrarán campos editables.
- **RF-11.5:** Se validarán peso y puntuación antes de guardar.
- **RF-11.6:** Al guardar, se actualizarán los datos y se volverá a la pantalla anterior.
- **RF-11.7:** El controlador manejará los TextEditingController y persistencia.

### 3.2.12. RU-12: Visualización de comidas

- **RF-12.1:** Se mostrará una lista de comidas del usuario.
- **RF-12.2:** Cada comida mostrará su título, descripción, peso, puntuación y foto.
- **RF-12.3:** Si no hay foto, se mostrará un ícono por defecto.
- **RF-12.4:** Al tocar una comida, se abrirán sus detalles.

### 3.2.13. RU-13: Agregar comidas

- **RF-13.1:** Al pulsar “+”, se mostrará un formulario para nueva comida.
- **RF-13.2:** El formulario incluirá título, descripción, peso, puntuación y foto.
- **RF-13.3:** Se podrá seleccionar una foto desde la galería.
- **RF-13.4:** Al guardar, se almacenarán los datos y se actualizará la lista.
- **RF-13.5:** Si hay campos obligatorios vacíos, se mostrará una alerta.

### 3.2.14. RU-14: Edición de comidas

- **RF-14.1:** Se podrá editar una comida existente.
- **RF-14.2:** Se podrá actualizar la foto desde galería.
- **RF-14.3:** Los cambios se reflejarán en la base de datos y lista.

### 3.2.15. RU-15: Eliminación de comidas

- **RF-15.1:** Se podrá eliminar una comida.
- **RF-15.2:** Antes de eliminar, se deberá mostrar una alerta de confirmación.
- **RF-15.3:** La lista se actualizará tras los cambios.

### 3.2.16. RU-16: Asignación y gestión de comidas por día y categoría

- **RF-16.1:** Se mostrará una lista de comidas por día, categoría y usuario.
- **RF-16.2:** Al tocar una comida, se abrirán sus detalles.
- **RF-16.3:** Se podrá eliminar una comida asignada (con confirmación).
- **RF-16.4:** Se podrán asignar comidas ya creadas.
- **RF-16.5:** El menú de asignación incluirá buscador por título.
- **RF-16.6:** Al asignar o eliminar, la lista se recargará automáticamente.

### 3.2.17. RU-17: Acceso a comidas por día y categoría

- **RF-17.1:** Se mostrará una lista de secciones por día: Desayuno, Comida, Cena y Extra.
- **RF-17.2:** Cada sección se mostrará como una tarjeta con ícono representativo.

### 3.2.18. RU-18: Acceso a la planificación semanal de comidas

- **RF-18.1:** Se mostrará una lista con los 7 días de la semana.
- **RF-18.2:** Cada día se visualizará como una tarjeta con diseño atractivo y color distintivo.

### 3.2.19. RU-19: Calendario y planificación de entrenamientos

- **RF-19.1:** Se mostrará un calendario interactivo.
- **RF-19.2:** Al seleccionar un día, se marcará visualmente.
- **RF-19.3:** Si no hay ejercicios asignados al día, se mostrará un mensaje informativo.

### 3.3. Requisitos no funcionales

#### 3.3.1. Compatibilidad

**RNF-COM-1:** La aplicación deberá funcionar en dispositivos con sistema operativo Android versión 8.0 (Oreo) o superior.

**RNF-COM-2:** La estructura de desarrollo deberá permitir futura adaptación para dispositivos iOS.

#### 3.3.2. Rendimiento y eficiencia

**RNF-EFI-1:** La aplicación deberá cargar listas (usuarios, ejercicios, comidas) en menos de 2 segundos en dispositivos de gama media.

**RNF-EFI-2:** Las operaciones de crear, leer, actualizar o eliminar datos deberán completarse en menos de 1 segundo.

**RNF-EFI-3:** La navegación entre pantallas deberá ser fluida, sin bloqueos ni demoras perceptibles.

#### 3.3.3. Extensibilidad y mantenibilidad

**RNF-EXT-1:** El código deberá estar organizado por capas (modelo, vista, controlador) siguiendo principios de arquitectura limpia o MVC.

**RNF-EXT-2:** El diseño general deberá permitir la adición de nuevas funcionalidades sin necesidad de reescribir la aplicación.

**RNF-EXT-3:** Las funciones reutilizables deberán estar centralizadas para evitar duplicación de código.

**RNF-EXT-4:** El diseño de la base de datos deberá permitir agregar nuevas tablas o campos sin afectar los datos existentes.

#### 3.3.4. Interoperabilidad

**RNF-INT-1:** El sistema deberá poder integrarse con futuras API externas si se desea extender la funcionalidad, como servicios de nutrición o dispositivos de seguimiento físico.

##### **Portabilidad**

**RNF-POR-1:** La aplicación deberá ser funcional en diferentes tamaños de pantalla y resoluciones.

**RNF-POR-2:** La arquitectura deberá permitir su adaptación a plataformas web o escritorio en el futuro.

#### 3.3.5. Recuperabilidad

**RNF-REC-1:** La aplicación deberá manejar errores sin perder los datos del usuario.

**RNF-REC-2:** En caso de cierre inesperado, los datos deberán conservarse al reiniciar la aplicación.

### 3.3.6. Confiabilidad

**RNF-CON-1:** El sistema deberá operar sin fallos graves durante sesiones normales de uso.

**RNF-CON-2:** La aplicación deberá proporcionar una experiencia estable y consistente.

### 3.3.7. Robustez

**RNF-ROB-1:** El sistema deberá manejar excepciones adecuadamente y responder a condiciones inesperadas.

**RNF-ROB-2:** Todas las validaciones de datos deberán evitar entradas erróneas sin bloquear el sistema.

### 3.3.8. Seguridad

**RNF-SEG-1:** Las contraseñas deberán almacenarse utilizando cifrado seguro (no en texto plano).

**RNF-SEG-2:** El acceso a los datos deberá restringirse al usuario con sesión iniciada.

**RNF-SEG-3:** No se permitirá el acceso a funciones privadas sin autenticación.

**RNF-SEG-4:** Acciones críticas como eliminar registros deberán requerir confirmación del usuario.

### 3.3.9. Integridad

**RNF-INT-2:** Sólo usuarios autenticados podrán modificar o agregar datos.

**RNF-INT-3:** El sistema deberá proteger los datos internos cumpliendo principios de protección de datos (como el RGPD si aplica).

### 3.3.10. Usabilidad

**RNF-USA-1:** La interfaz debe ser intuitiva, incluso para usuarios sin experiencia técnica.

**RNF-USA-2:** Todos los botones e íconos deberán tener etiquetas o representaciones claras.

**RNF-USA-3:** El diseño deberá adaptarse correctamente a distintas resoluciones de pantalla.

**RNF-USA-4:** Se deberá proporcionar retroalimentación visual ante las acciones del usuario (mensajes de éxito, error, cargando, etc.).

**RNF-USA-5:** Los formularios deberán prevenir la pérdida de datos al navegar entre pantallas.

**RNF-USA-6:** El sistema deberá proporcionar mensajes de error claros, comprensibles y orientados al usuario.

### 3.3.11. Disponibilidad

**RNF-DIS-1:** La aplicación deberá ser funcional sin conexión a internet, mediante almacenamiento local.

**RNF-DIS-2:** Las funciones principales deberán estar disponibles en todo momento, salvo mantenimiento planificado.

### 3.3.12. Accesibilidad

**RNF-ACC-1:** El diseño deberá garantizar un contraste adecuado entre texto y fondo.

**RNF-ACC-2:** La interfaz deberá seguir buenas prácticas para accesibilidad visual.

**RNF-ACC-3:** Las funciones principales deberán ser accesibles mediante teclado o lector de pantalla si se adapta a otras plataformas.

### 3.4. Diseño

Una vez contextualizado el proyecto y establecidos los requisitos, es momento de definir la solución que se va a desarrollar. En esta sección se presentarán las distintas herramientas que se utilizarán a lo largo del proyecto, así como los dos prototipos de interfaz diseñados para las dos versiones distintas de la aplicación.

#### 3.4.1. Sistema operativo usado

Se desarrollado la aplicación en dispositivos Android por estos motivos:

Las apps móviles permiten interfaz más adaptada, fluida y rápida al entorno del usuario (modo oscuro, gestos, etc.).

Se pueden integrar fácilmente con apps del sistema como calendario, contactos, pagos móviles, etc.

La mayoría de las personas usan smartphones como su principal dispositivo

Android tiene una gran cantidad de documentación y de software libre (open source) que se puede usar como ejemplo o ayuda para implementar funcionalidades específicas.

Los sistemas operativos de Android tienen un entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android llamado Android Studio [8] y que también funciona muy bien con Visual Studio Code donde se ha codificado el proyecto.

#### 3.4.2. Prototipo

El primer paso para empezar es importante crear un prototipo en el que se vea la interfaz y el diseño que tendrá una vez desarrollado, así poder ver las diferencias que ha habido con el resultado final con la idea inicial del desarrollo.

Este método tiene ciertas ventajas

Eficiencia del tiempo: El prototipo es muy sencillo y rápido.

Ayuda a aclarar las ideas: Muchas ocasiones los clientes no tienen conocimientos en este campo y para ello es mucho más fácil enseñar una imagen del resultado que tendrán al final su proyecto. Esto también hace que el cliente al entender la situación mejor pueda darnos críticas o cambios que quiera así dando posible a la retroalimentación.

Hay que dejar claro que no todas las imágenes o pantallas que se presenta al principio será similares a las finales ya que pueden sufrir cambios o modificaciones a lo largo del desarrollo.

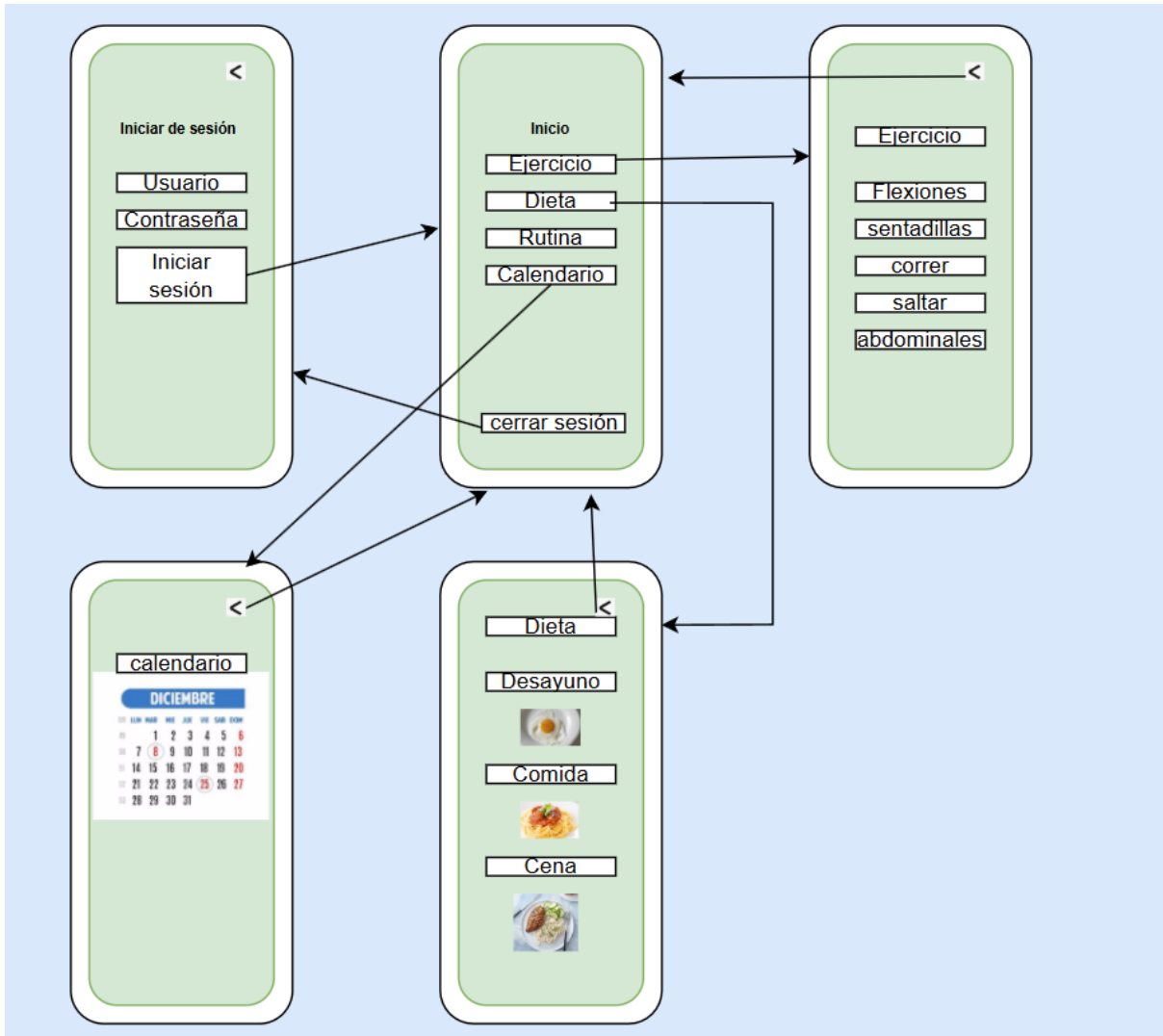


Ilustración 5 Prototipo de la app (Creación propia) con la herramienta Diagrams.

A continuación, se va a explicar la idea que va a contener cada pantalla.

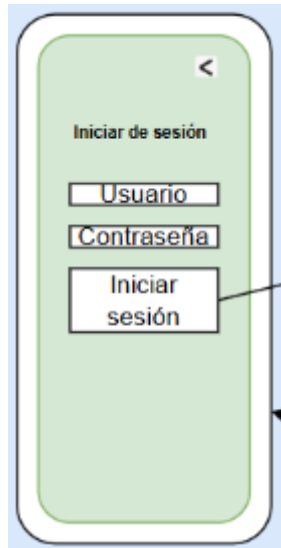


Ilustración 6 Prototipo de iniciar sesión Creación propia con diagrams

Pantalla de inicio: Esta pantalla muestra el inicio de sesión del proyecto con un diseño sencillo y centrado. Tiene un encabezado con el texto "Inicio de sesión". Debajo del título, se encuentran dos campos de entrada: uno para el nombre de usuario y otro para la contraseña. Finalmente, hay un botón con el texto "Iniciar sesión" que permite enviar las credenciales para autenticar al usuario.

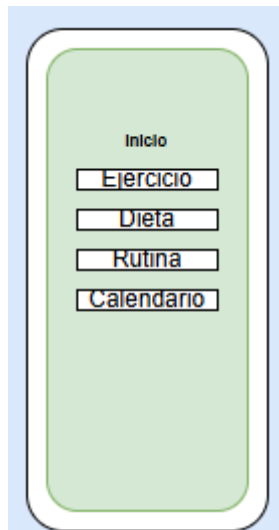


Ilustración 7 Prototipo de inicio creación propia con diagrams

Pantalla home/inicio: Esta pantalla es la principal de una aplicación. Es bastante sencilla. En la parte superior aparece primero el título "Inicio", que te indica que estás en la pantalla principal. Debajo, hay cuatro menús uno debajo del otro, cada uno con una función específica: "Ejercicio", "Dieta", "Rutina" y "Calendario" para llevarte a una sección distinta de la app. Con esto la distribución es clara, intuitiva y pensada para que el usuario pueda acceder rápidamente a lo que desee rápidamente.



Ilustración 8 Prototipo de Ejercicio creación propia con Diagrams

Esta pantalla muestra es una lista simple de ejercicios físicos. Incluye flexiones, sentadillas, correr, saltar y abdominales. Parece ser una rutina básica de entrenamiento o un recordatorio para mantenerse activo. Con un título que dice "Ejercicio", y con un formato minimalista. Por último, tiene una flecha de retroceso que al pulsarla le llevará a la pantalla anterior. Por último, tiene una flecha de retroceso que al pulsarla le llevará a la pantalla anterior.



Ilustración 9 Prototipo de dieta creación propia con Diagrams

Esta pantalla muestra una lista básica de alimentación, con un título "Dieta". Con 3 apartados de las tres comidas principales: "Desayuno", "Comida" y "Cena". El formato es de nuevo sencilla y minimalista. Es un esquema simple con el objetivo de planificar comidas, un recordatorio de horarios de alimentación. Al igual que en la imagen anterior de ejercicios, el diseño es funcional y directo. Por último, tiene una flecha de retroceso que al pulsarla le llevará a la pantalla anterior.

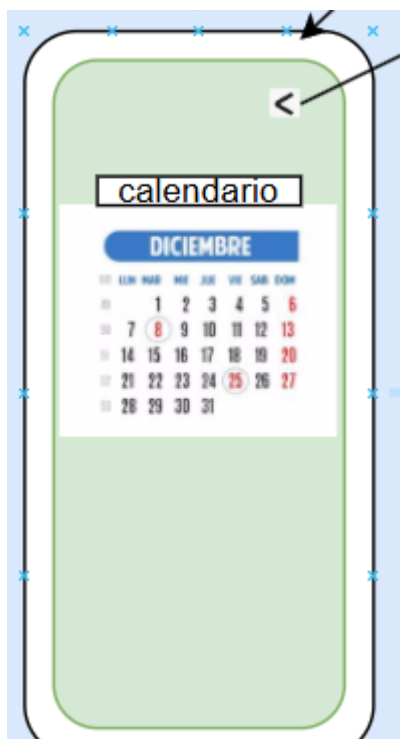


Ilustración 10 Prototipo Calendario Creación propia con Diagrams

Esta pantalla muestra un calendario básico para planificar qué ejercicios hacer que se va a hacer cada día. También tiene una flecha de retroceso que al pulsarla le llevará a la pantalla anterior.

### 3.4.3. DISEÑO INICIAL

Para la primera reunión con el cliente se desarrolló una simulación más detallada que el prototipo para que los clientes tuvieran una mejor idea como va a ser el resultado final del proyecto.



Ilustración 11 foto de UISTORE.Design

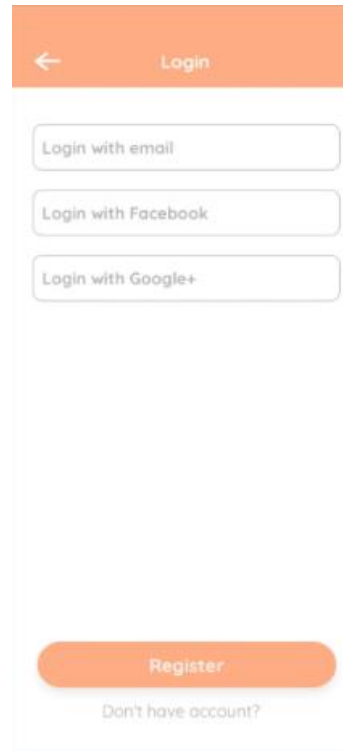


Ilustración 12 foto de UISTORE.Design login

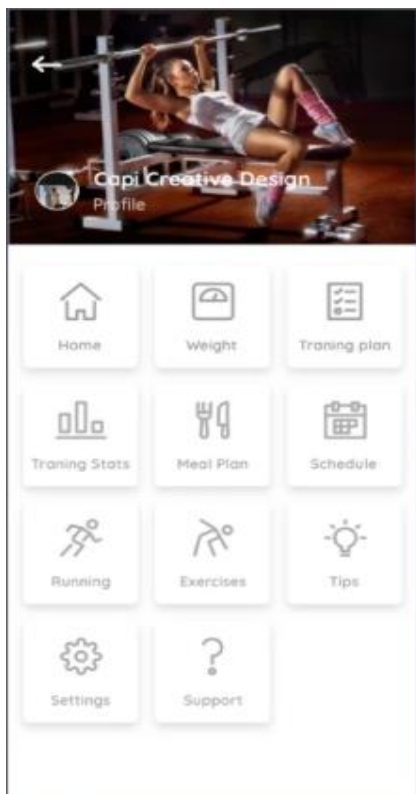


Ilustración 14 foto de UISTORE.Design pagina principal

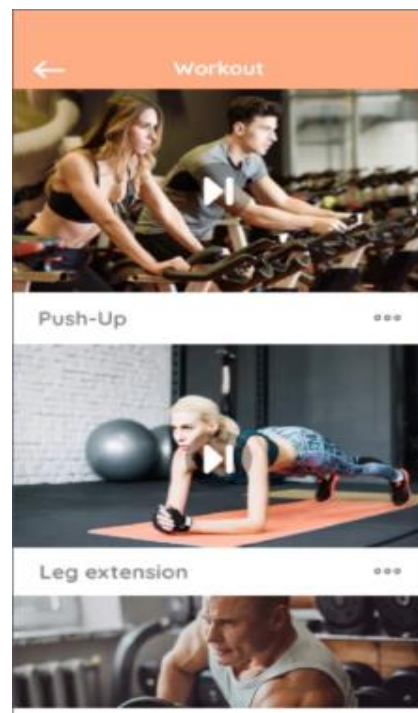


Ilustración 13 foto de UISTORE.Design rutina

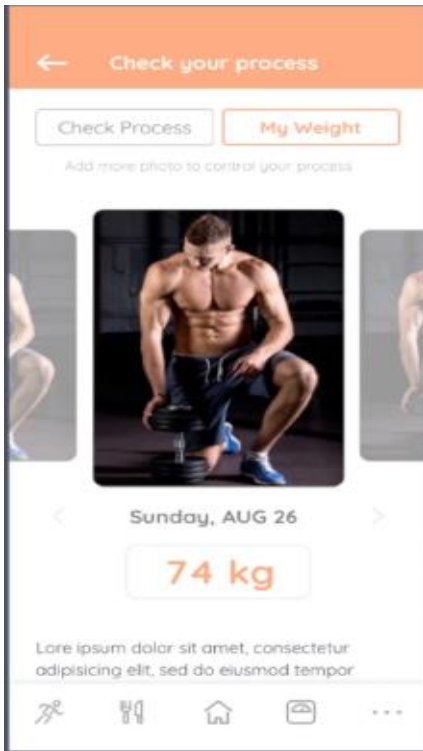


Ilustración 16 foto de UISTORE.Design peso

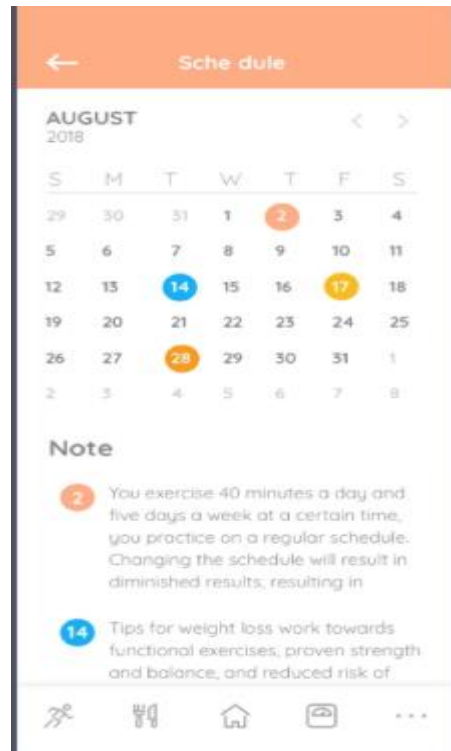


Ilustración 15 foto de UISTORE.Design Calendario

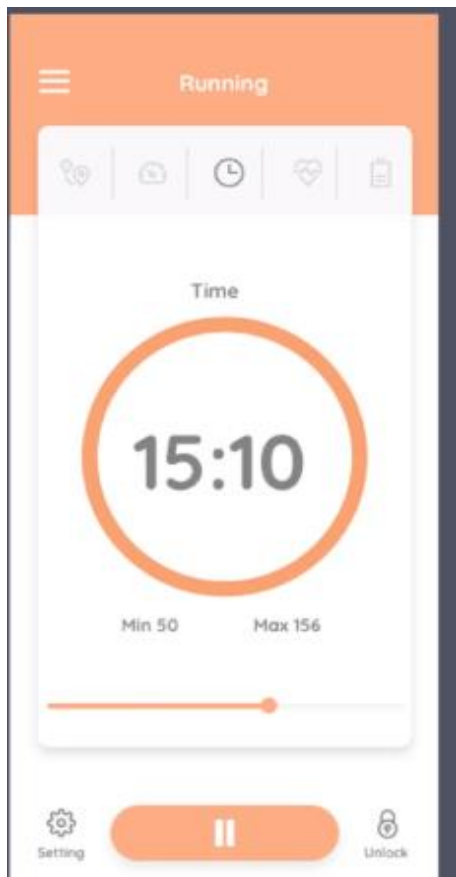


Ilustración 17 foto de UISTORE.Design timer



Ilustración 18 foto de UISTORE.Design detalles ejercicio

#### 3.3.4. Segundo prototipo

Para el desarrollo del segundo prototipo que surgió tras el feedback recibido después de la primera reunión con el cliente. Durante esa entrevista, se recopilaron observaciones, sugerencias y opiniones por parte de los interesados, los cuales sirvieron para la revisión del primer diseño. Gracias a este análisis, se identificaron una serie de problemas tanto funcionales como de usabilidad que causaron un rediseño de la aplicación.

Uno de los problemas más importantes detectados en el primer prototipo fue el exceso de información y menús en la interfaz. La acumulación excesiva de menús, botones y textos e imágenes las cuales generaba una experiencia de usuario poco fluida, especialmente para el público que va dirigido el proyecto. Los cuales pueden tener algún tipo de discapacidad. Este problema no solo afectaba la estética de la aplicación, sino que también afectaba negativamente a su comprensión y uso. Por lo que al final se decidió tener un enfoque de diseño centrado en el usuario, priorizando la simplicidad, la claridad y la accesibilidad de las funciones.

Otro de los aspectos esenciales en esta fase es adaptar la aplicación a distintos grados de discapacidad. Dado que existen múltiples tipos y niveles de discapacidad. Se concluyó que intentar cubrir de forma específica cada caso implicaría un coste de desarrollo muy elevado y una alta dificultad. Por ello, se desarrolló una solución más versátil y escalable: permitir que cada usuario diseñe su propia rutina de ejercicios según sus capacidades, intereses y necesidades personales. Esta personalización incluye la posibilidad de añadir ejercicios específicos, establecer ciertos parámetros como repeticiones o descansos, y realizar un seguimiento de su progreso a lo largo del tiempo.

Además de estas mejoras, el segundo prototipo también tiene una lógica más coherente en la estructura. Así facilitando su mantenimiento y futuras ampliaciones. Se eliminaron funcionalidades que no aportaban un valor o que complicaban la experiencia de uso. De este modo, se logró una interfaz más sencilla, con una navegación más directa y con elementos visuales atractivos.

En resumen, el segundo prototipo representa una mejora respecto a la versión inicial. Gracias a la incorporación de feedback de los interesados.

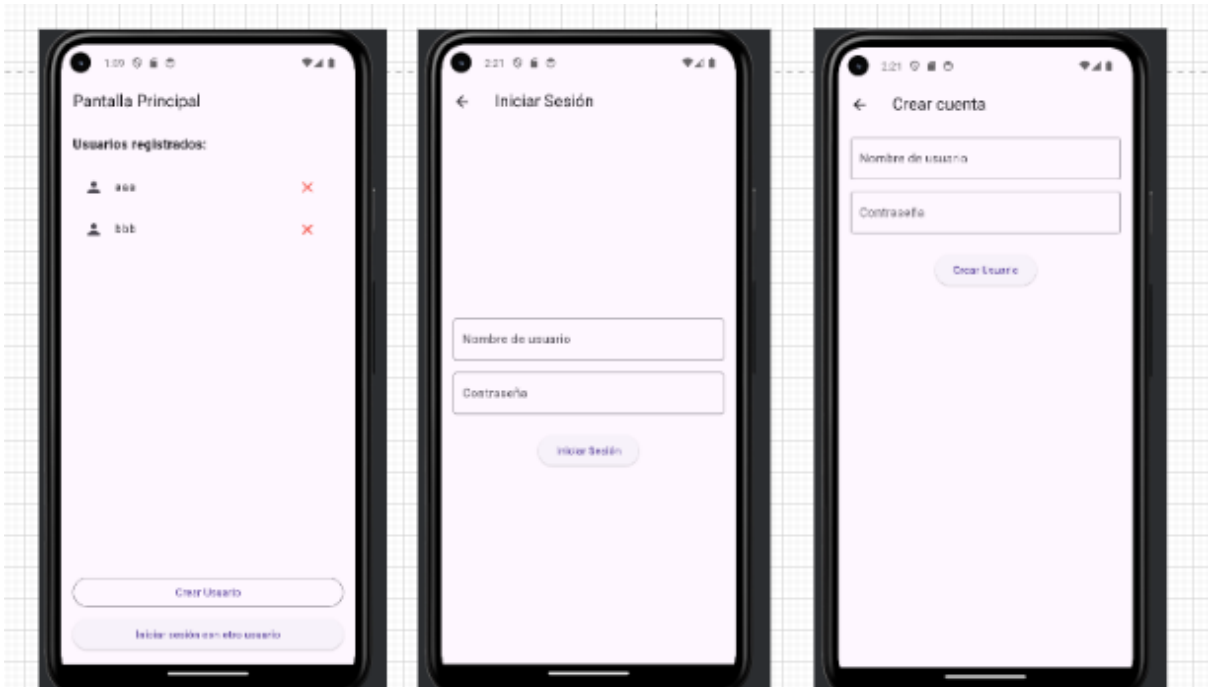


Ilustración 19 Segundo prototipo creación propia con Android Studio

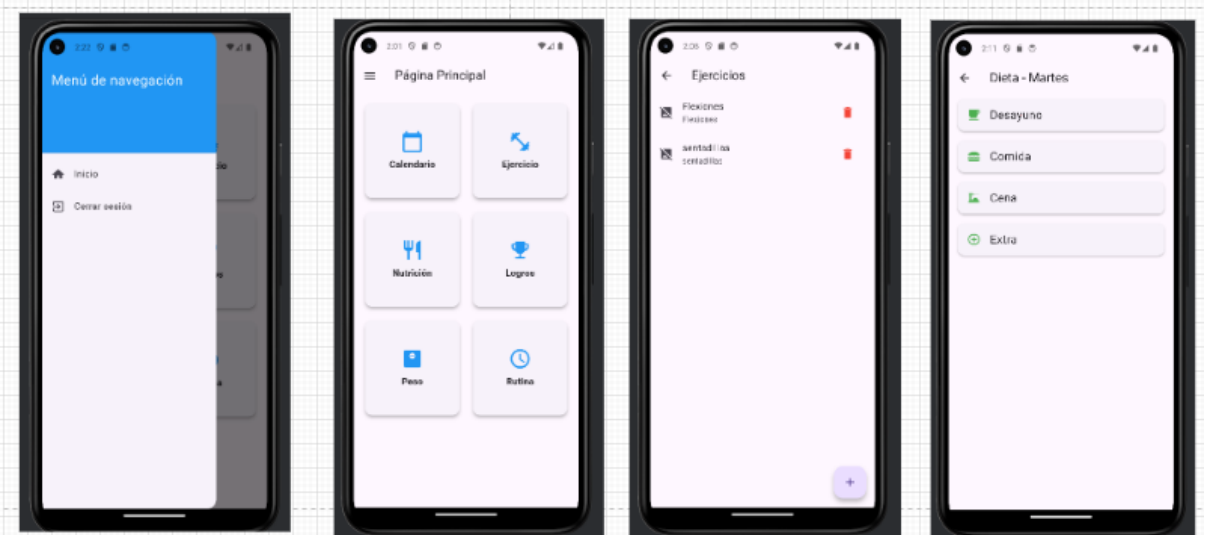


Ilustración 20 Segundo prototipo creación propia con Android Studio

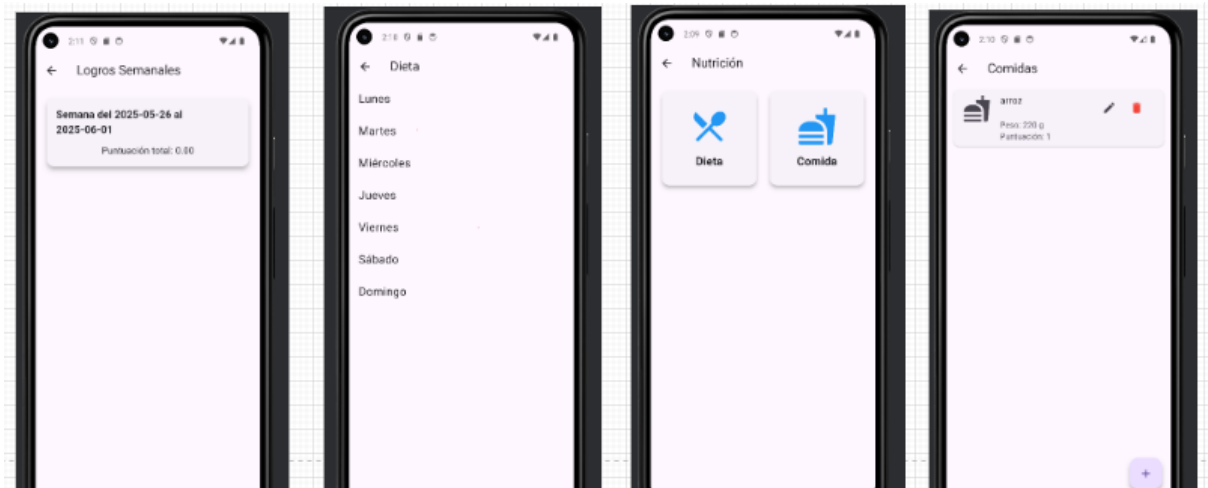


Ilustración 21 Segundo prototipo creación propia con Android Studio

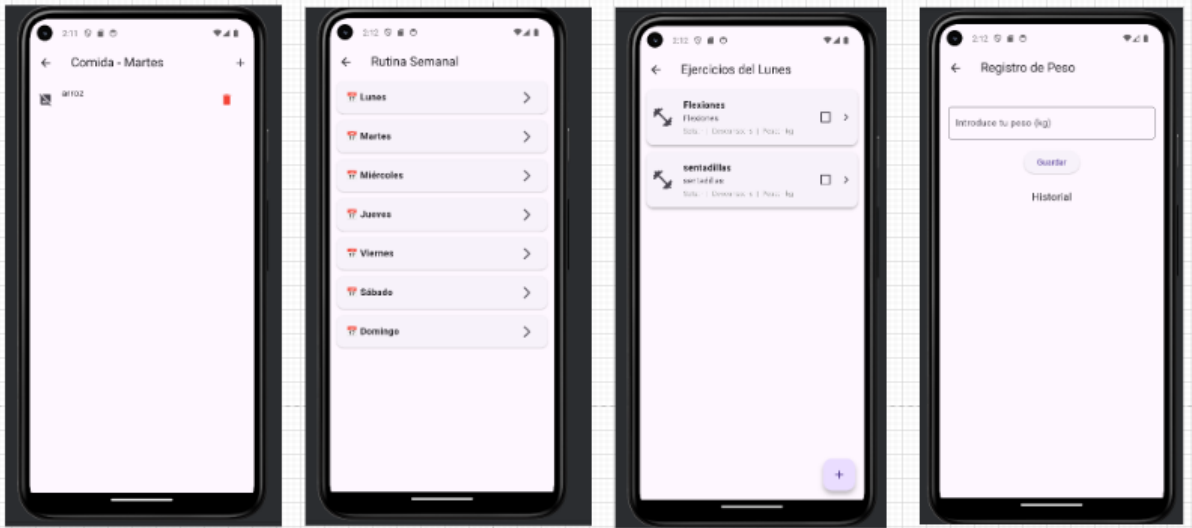


Ilustración 22 Segundo prototipo creación propia con Android Studio

Pantalla "Iniciar sesión":

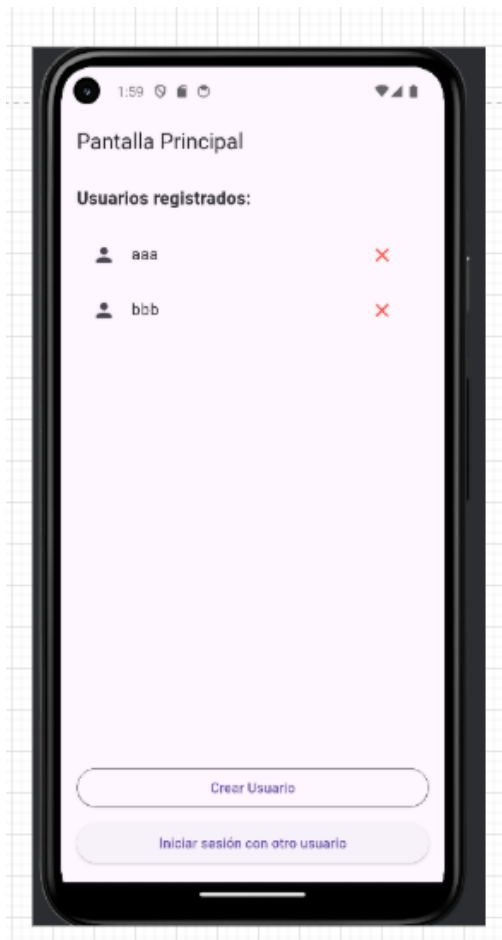
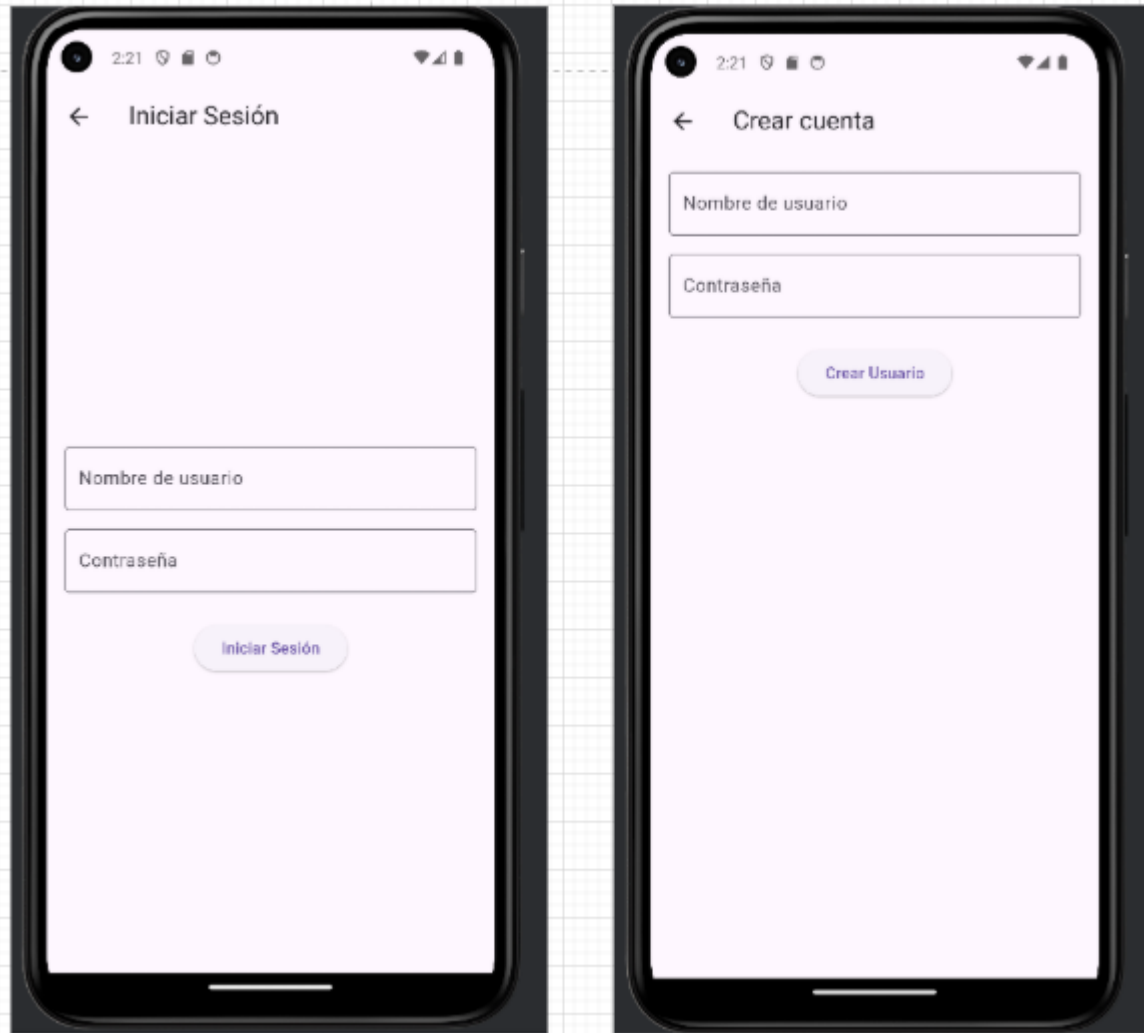


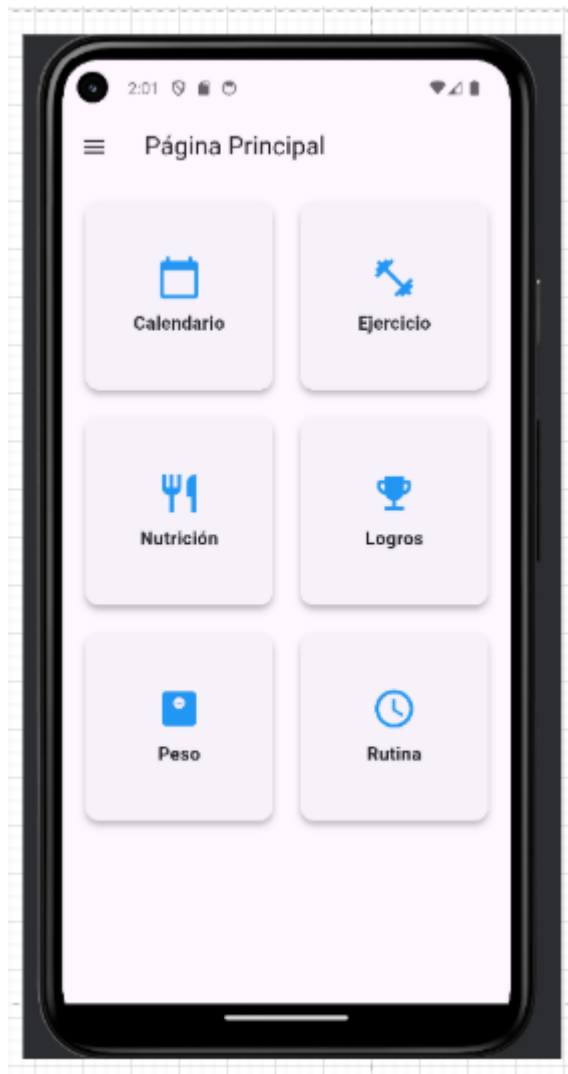
Ilustración 23 Segundo prototipo creación propia con Android Studio  
Pantalla inicial

Para acceder a la aplicación el usuario deberá iniciar sesión. Como se puede ver en la ilustración que se encuentra arriba (20) se muestra dos usuarios que ya han iniciado sesión. Esta función hace que cuando ya se ha iniciado sesión una vez la próxima vez sea más fluido en caso de que el usuario no quiera que se guarde los usuarios se pueden eliminar. En caso de ya tener un usuario, pero el usuario haya sido eliminado se podrá iniciar igualmente con la función “iniciar sesión con otro usuario”. Y por último en el caso de no tener una cuenta se puede crear un usuario desde cero.



*Ilustración 24 Segundo prototipo creación propia con Android Studio Iniciar sesión y registrarse*

Como se puede ver en la ilustración (21) de la izquierda es cuando el usuario haya hecho clic en “iniciar sesión con otro usuario” en la pantalla de iniciar sesión que su función principalmente es iniciar con un usuario que no estaba en los recientes porque el usuario lo haya eliminado o nunca haya iniciado sesión en este dispositivo. La segunda ilustración que se encuentra a la derecha es para crear un usuario que hace falta rellenar las dos casillas.



*Ilustración 25 Segundo prototipo creación propia con Android Studio Página*

### **Pantalla “página principal”**

En esta imagen se puede ver una vez iniciada sesión correctamente se puede contemplar que hay 7 menús 6 imágenes y el último arriba la izquierda representando.

- Calendario: representado por un icono de calendario
- Ejercicio: representado por un icono de pesa / mancuerna
- Nutrición: representado por un icono de tenedor y cuchillo
- Logros: representado por un icono de un trofeo
- Peso: representado por un icono de balanza
- Rutina: representado por un icono de reloj analógico.
- Menú: representado por 3 barras horizontales

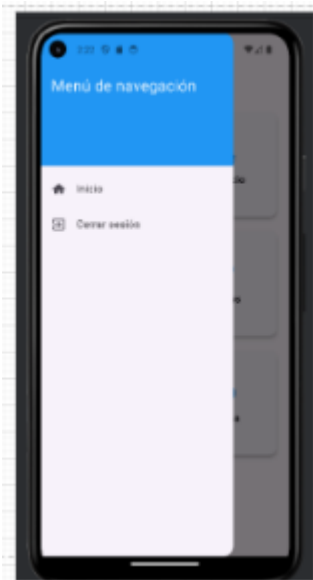


Ilustración 26 Segundo prototipo creación propia con Android Studio menú de navegación

Pantalla menú esta pantalla da la opción de cerrar sesión y se volvería a la pantalla de “Iniciar sesión”.

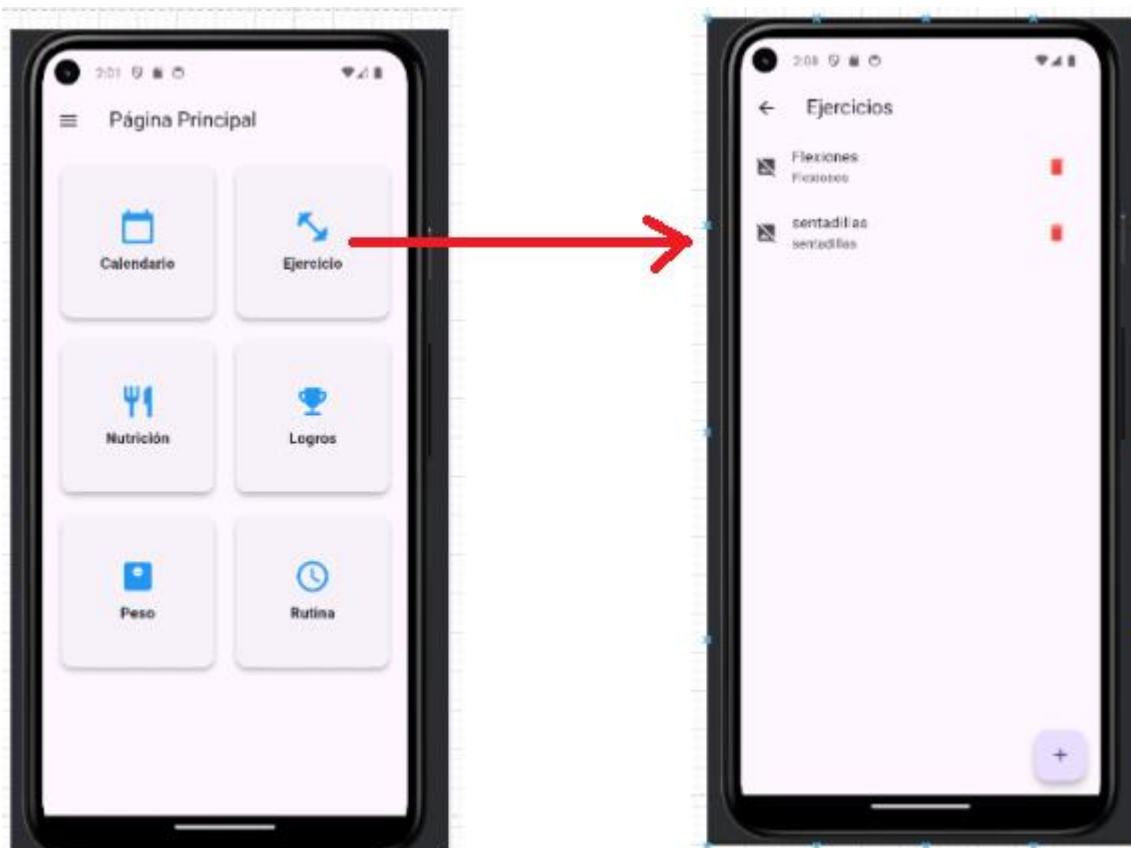
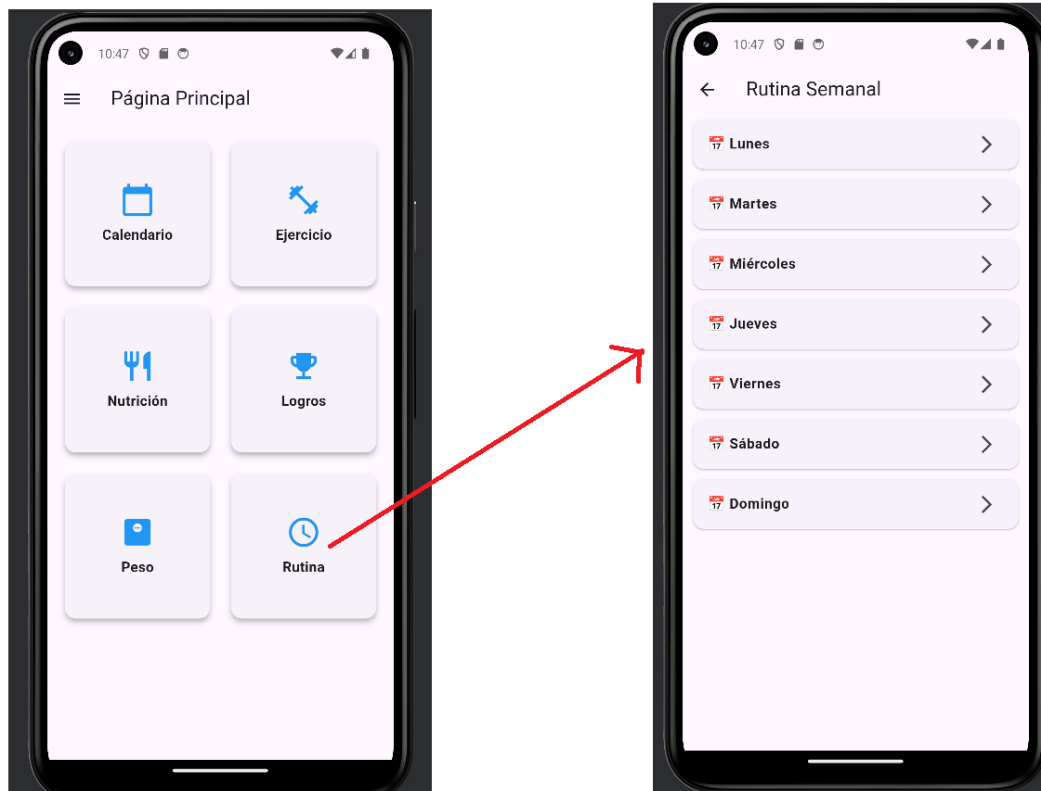


Ilustración 27 Segundo prototipo creación propia con Android Studio menú de navegación Página principal a Ejercicios

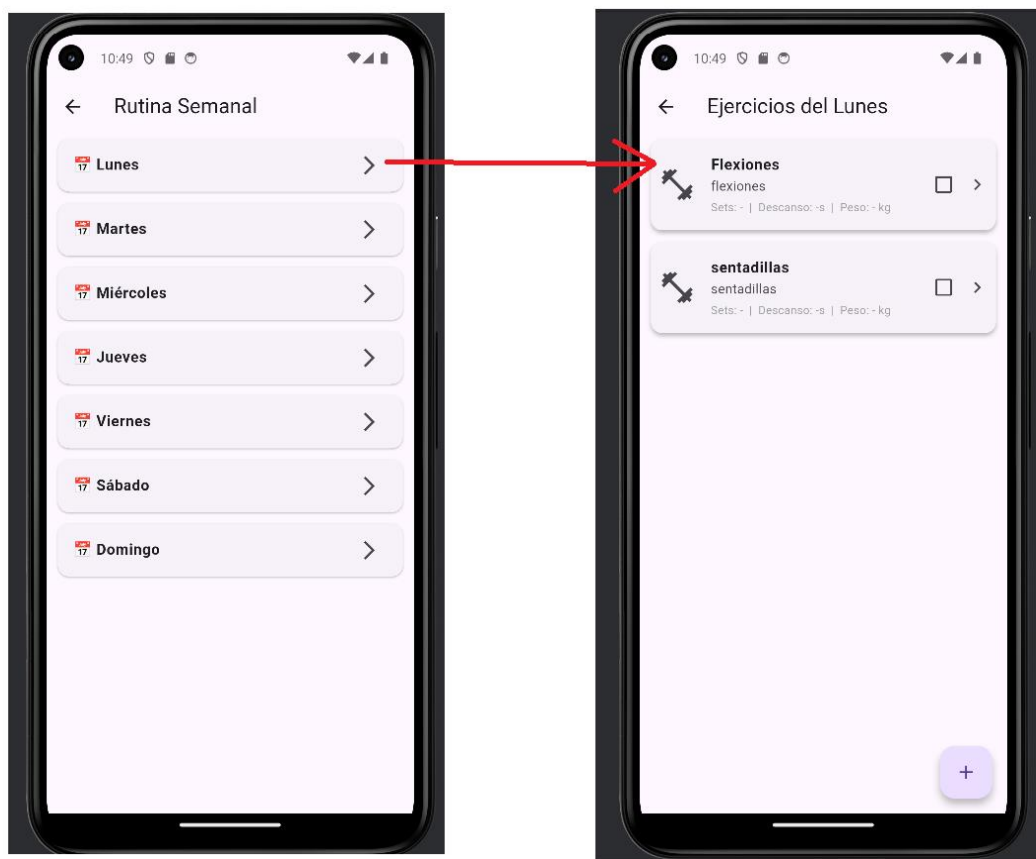
La pantalla de la derecha se accede desde la página principal haciendo click a la opción “ejercicio” que está situado en la primera pantalla. En esta ilustración se muestran todos los ejercicios guardados con

la opción de añadir, modificar y eliminar. Esta pantalla es fundamental ya que en la sección de **Rutina** se tendrá que añadir ejercicios que se hayan creado en el apartado de “Ejercicios”.

Si se hace click en la opción de rutina se dirigirá a una nueva pantalla con los 7 días de la semana en cada una de ellas se podrá añadir los ejercicios que se haya creado en “ejercicios como se ha mencionado en el apartado anterior.



*Ilustración 28 Segundo prototipo creación propia con Android Studio Menu de navegación Página principal a rutina*



*Ilustración 29 Segundo prototipo creación propia con Android Studio Menu de navegación Rutina semanal a rutina del día*

En esta ilustración cuando se accede a cada día de la rutina tendremos un listado de ejercicios y se podrán añadir a la rutina de ese día. En la rutina se podrá marcar como completado en la casilla cuadrada al marcar como completado.

También se podrá acceder a la información del ejercicio por si el usuario se le ha olvidado los detalles del ejercicio como su descripción, el peso, el descanso requerido o un video para ver su correcta ejecución.

Cuando una actividad se marque como completado el sistema sumará una puntuación (del ejercicio) al progreso semanal con el objetivo de hacer un seguimiento y motivar al usuario.

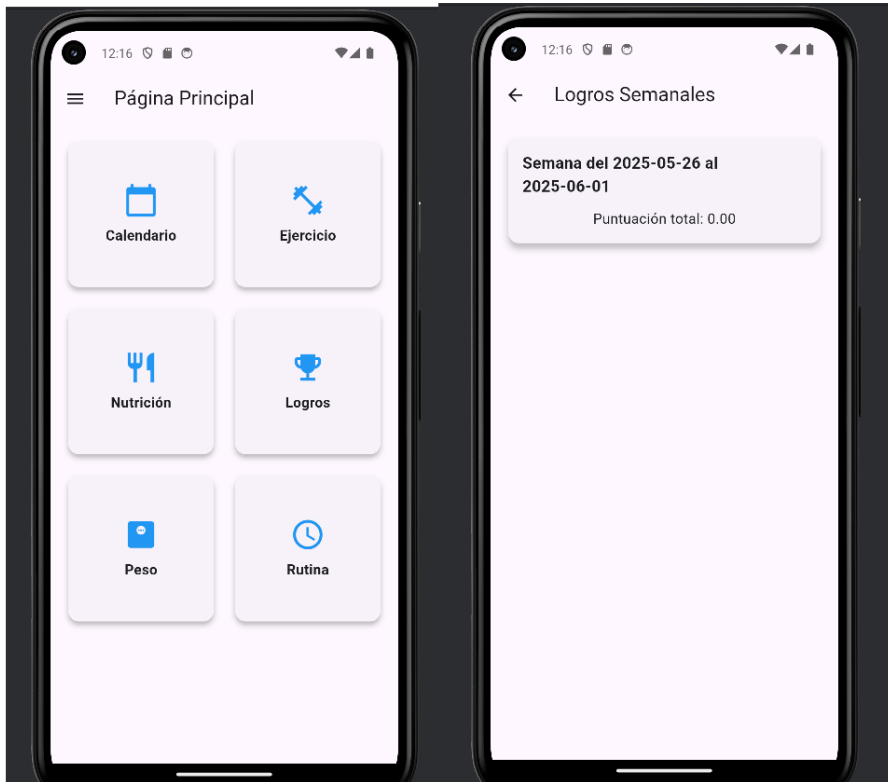
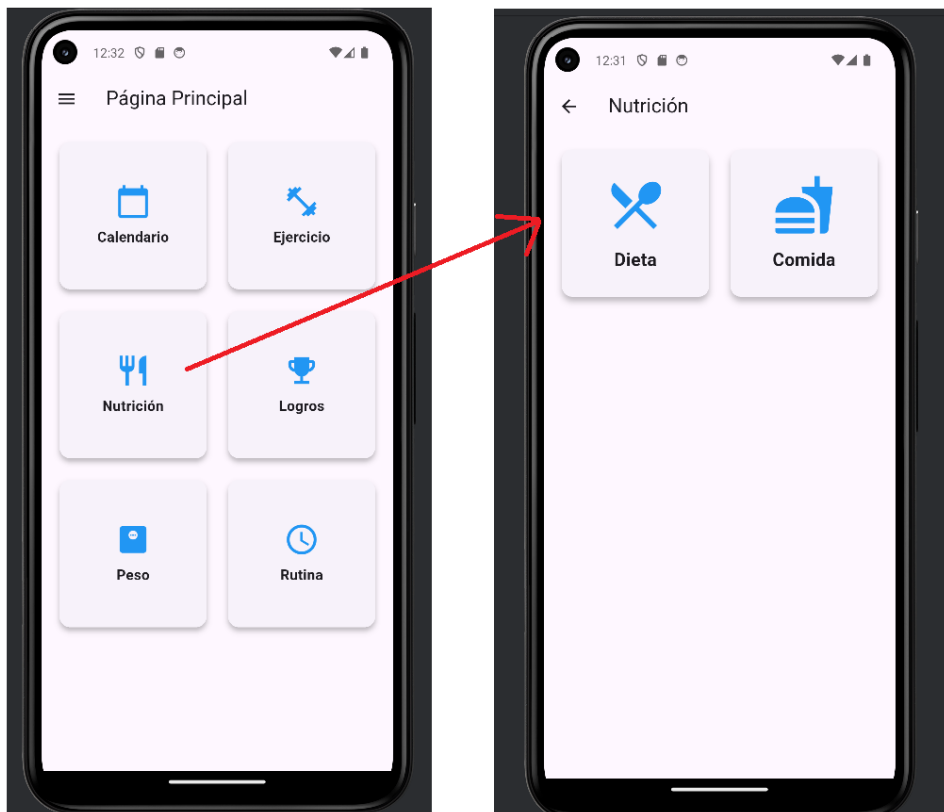


Ilustración 30 Segundo prototipo creación propia con Android Studio Página principal a logros

A la pantalla de logros se accede desde la “**página principal**”. La página de logros tiene un historial de cada semana con una puntuación para que los usuarios puedan ver su progreso. La puntuación se suma cuando en los ejercicios de rutinas se marcan como completados.



En la sección de nutrición funciona tiene la misma base como la pantalla de ejercicios. Se tiene que crear primero las comidas para luego poder añadirlos en la dieta.

Como se puede observar en la siguiente Ilustración 29 abajo derecha cuando se haga click en “+” se podrá añadir comidas. También se puede eliminar y modificar.

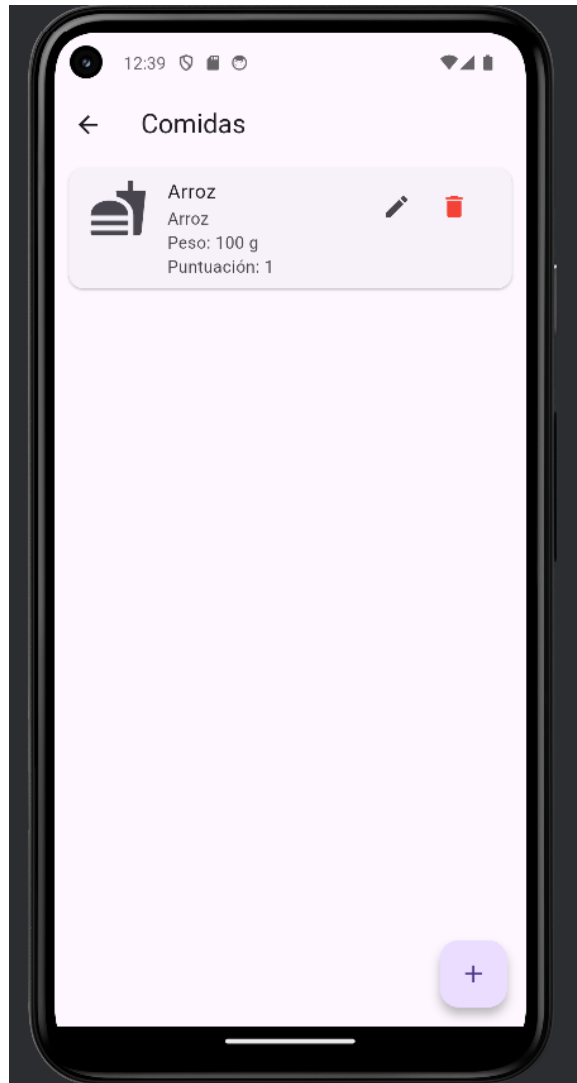


Ilustración 32 Segundo prototipo creación propia con Android Studio Comidas

Una vez creado las comidas se podrán luego añadirlos a cada día para hacer una dieta. Los pasos para seguir es hacer click a “dieta”, elegir un día de la semana, seleccionar una categoría y finalmente añadir la comida. Como se puede ver en la siguiente Ilustración 30:

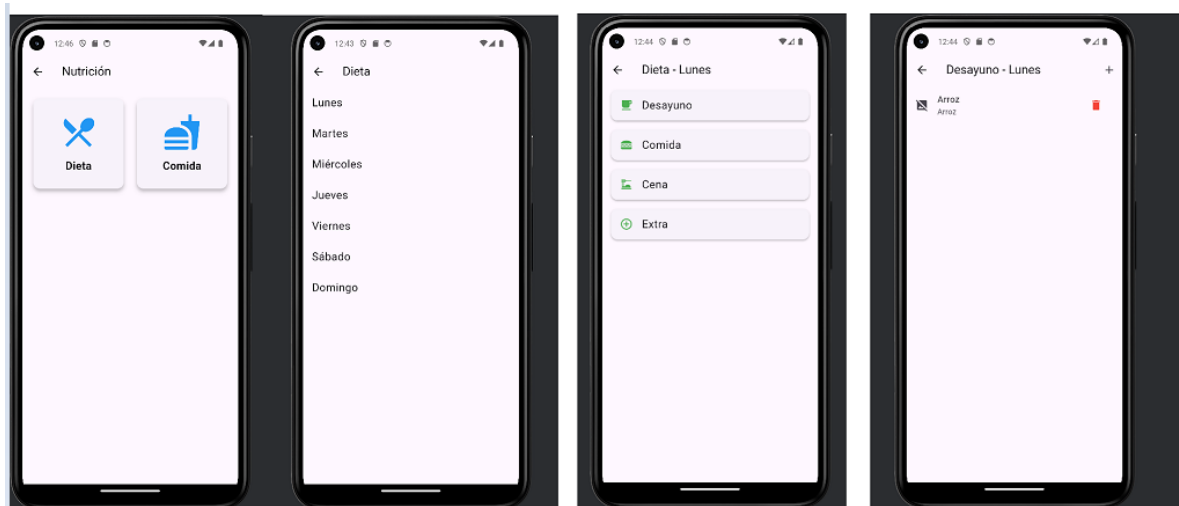


Ilustración 33 Segundo prototipo creación propia con Android Studio proceso de añadir comida a dieta

Por último, tenemos dos opciones en el menú principal que son el peso y el calendario este último para seguir los eventos que sean de interés al usuario y el peso para tener y poder ver una progresión a su cambio físico.

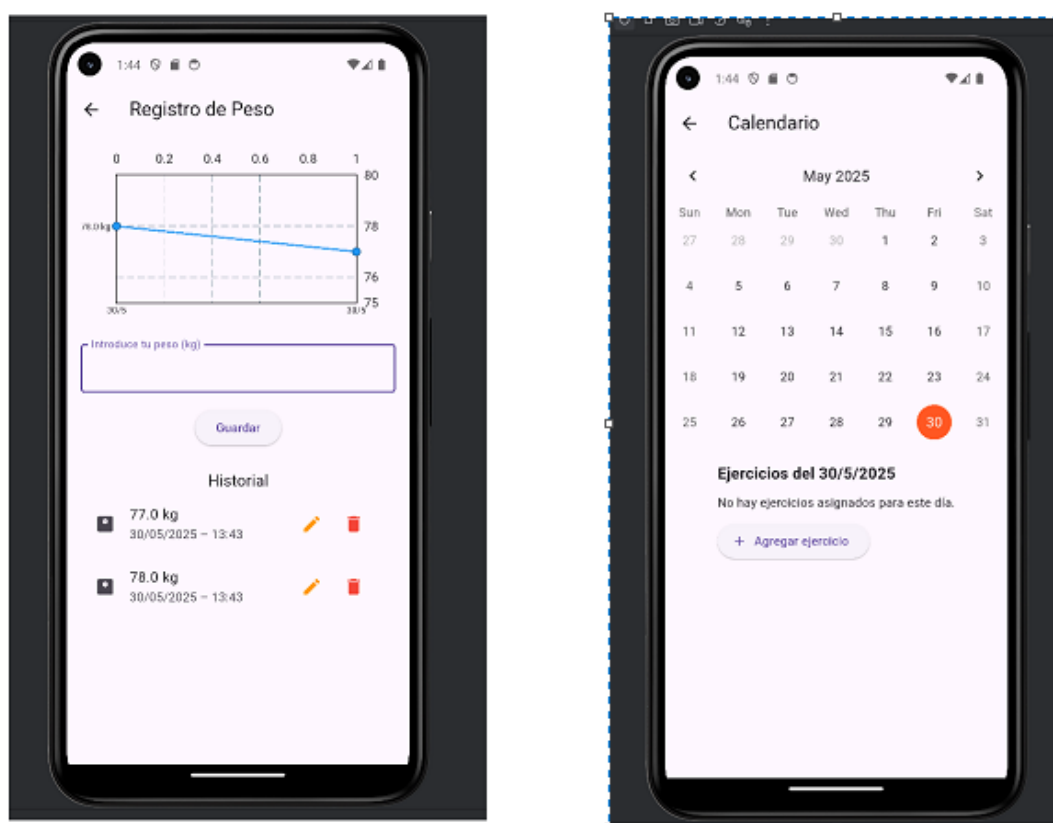


Ilustración 34 Segundo prototipo creación propia con Android Studio Peso y calendario

En la pantalla del peso también tiene incorporada una gráfica para que sea todo más visible.

### 3.5. Diagramas

Para tener una implementación ideal y sin retrasos lo primero es modelar los diagramas del software a implementar. Para llevar estos diagramas se ha cogido en entorno de UML que permite ver de manera gráfica las siguientes 3 perspectivas de la aplicación.

El diagrama de casos de usos explicará cómo funcionará la aplicación, el diagrama de clases se explicará con un diagrama de clases.

#### 3.5.1. Diagrama de casos de uso

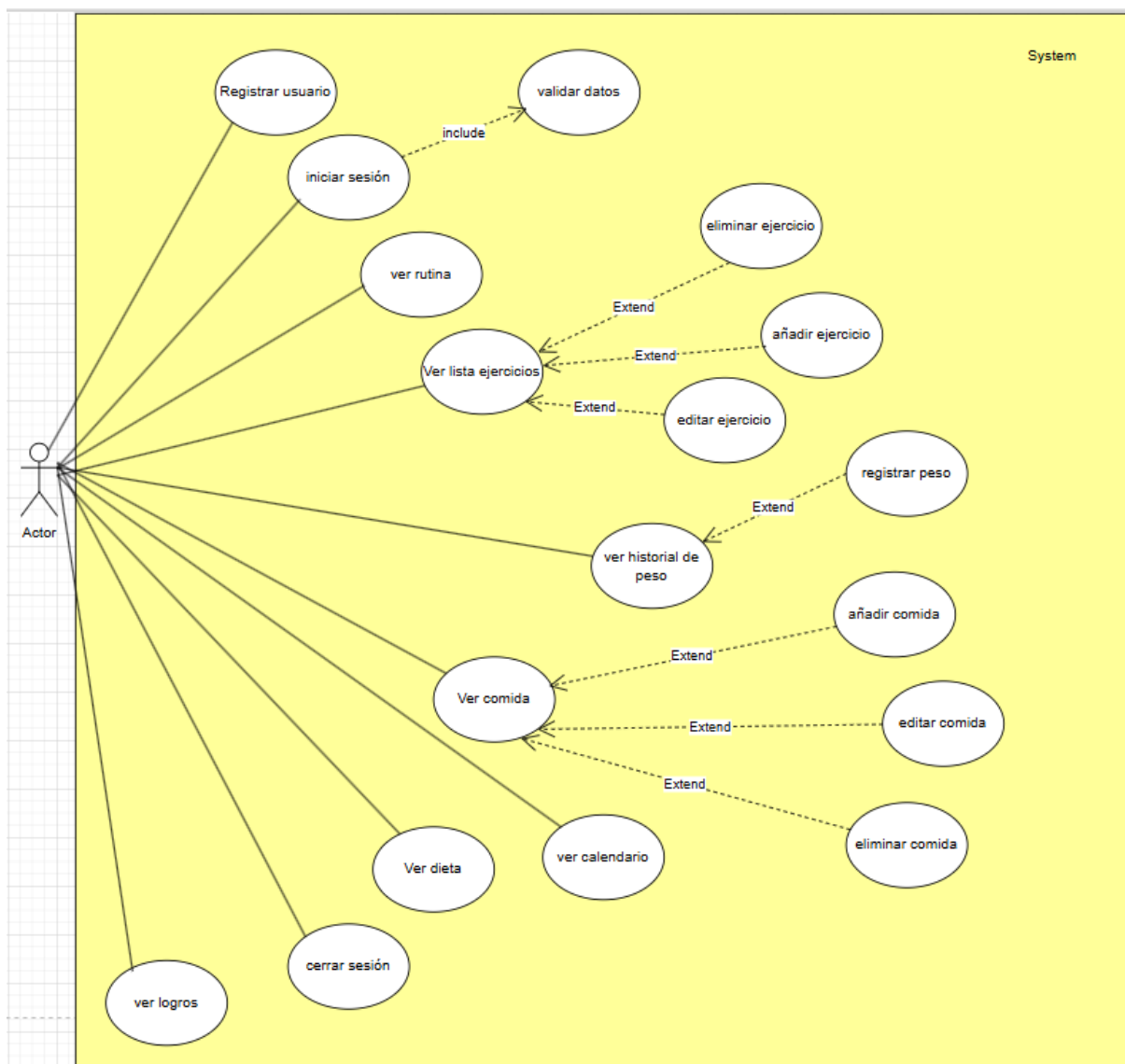


Ilustración 35 Casos de uso

### 3.5.2. Especificación de casos de uso

|                     |  |                     |                |
|---------------------|--|---------------------|----------------|
| ID:01               | CU-1   | Nombre:             | Iniciar sesión |
| Creado por:         | Jiliang Qiu  | Fecha de creación   |                |
| Actor primario:     | Usuario  | Actores secundarios | Ninguno        |
| Descripción:        | Es el proceso por el que el usuario se autentifica para acceder al sistema.                                      |                     |                |
| Evento activador:   | El accede a la aplicación  |                     |                |
| Precondiciones      | El usuario debe tener la aplicación instalada  |                     |                |
| Postcondiciones     | El usuario debe de tener un usuario creado en el sistema   |                     |                |
| Flujo normal        | CU-1-1 El usuario accede a la pantalla para poder iniciar sesión.  |                     |                |
|                     | CU-1-2 El usuario introduce sus credenciales.  |                     |                |
|                     | CU-1-3 El sistema comprueba que todos los campos están rellenos.   |                     |                |
|                     | CU-1-4 El sistema comprueba si el usuario existe.  |                     |                |
|                     | CU-1-5 Una vez el sistema comprobado le da acceso y lo lleva a la página principal de la aplicación.             |                     |                |
| Flujo alternativo 1 | CU-1-3.a.1 El sistema detecta que hay un campo sin rellenar  |                     |                |
|                     | CU-1-3 a.2 El sistema envía un mensaje al usuario informando que hay un campo vacío.                             |                     |                |
|                     | CU-1-3 a 3 El sistema vuelve al CU-1-2.  |                     |                |
| Flujo alternativo 2 | CU-1-4.a.1 El sistema detecta que el usuario no existe por lo que las credenciales introducidas son incorrectas. |                     |                |
| Prioridad:          | Alta   |                     |                |
| Frecuencia de uso:  | Diario   |                     |                |
| Otra información:   | Ninguna  |                     |                |

Tabla 2: Casos de uso extendido "iniciar sesión"

|                 |             |                     |                   |
|-----------------|-------------|---------------------|-------------------|
| ID:02           | CU-2        | Nombre:             | Registrar usuario |
| Creado por:     | Jiliang Qiu | Fecha de creación   |                   |
| Actor primario: | Usuario     | Actores secundarios | Ninguno           |

|                     |  |
|---------------------|--|
| Descripción:        | Es el proceso por el que un usuario se registra en el sistema.               |
| Evento activador:   | El accede a la aplicación  |
| Precondiciones      | El usuario no debe estar registrado previamente.                             |
| Postcondiciones     | Se crea un nuevo usuario en el sistema.                                      |
| Flujo normal        | CU-2-1. El usuario accede a la pantalla de registro.                         |
|                     | CU-2-2. El usuario rellena todos los campos                                  |
|                     | CU-2-3 El sistema comprueba que todos los campos están completos.            |
|                     | CU-2-4 El sistema comprueba si el usuario existe.                            |
|                     | CU-2-5. El sistema guarda los datos y crea el nuevo usuario.                 |
|                     | CU-2-6. El usuario es redirigido a la pantalla principal.                    |
| Flujo alternativo 1 | CU-2-3.a.1. El sistema detecta campos vacíos y muestra un mensaje.           |
|                     | CU-2-3.a.2. Vuelve a CU-2-2.   |
| Flujo alternativo 2 | CU-2-4.a.1. El sistema detecta que ya existe un usuario con el mismo correo. |
|                     | CU-2-4.a.2. Muestra error y vuelve a CU-2-2.                                 |
| Prioridad:          | Alta   |
| Frecuencia de uso:  | Baja   |
| Otra información:   | Ninguna  |

Tabla 3: Casos de uso extendido "Registrar usuario"

|                   |  |                     |                |
|-------------------|--|---------------------|----------------|
| ID:03             | CU-3   | Nombre:             | Ver ejercicios |
| Creado por:       | Jiliang Qiu  | Fecha de creación   |                |
| Actor primario:   | Usuario  | Actores secundarios | Ninguno        |
| Descripción:      | El usuario quiere ver los ejercicios para ello en la página principal hace click en la opción de "ejercicios". |                     |                |
| Evento activador: | El usuario hace click en "Ejercicios"  |                     |                |
| Precondiciones    | ninguna  |                     |                |
| Postcondiciones   | ninguna  |                     |                |

|                   |  |
|-------------------|--|
| Flujo normal      | CU-3-1 El sistema carga todos los ejercicios guardados anteriormente creados por el usuario.         |
|                   | CU-3-2 El sistema muestra todos los ejercicios en forma de etiquetas a los usuarios.                 |
|                   | CU-3-3 El usuario puede pulsar sobre los ejercicios para ver los detalles de ellos                   |
|                   | CU-3-4 El sistema comprueba si el usuario existe.  |
|                   | CU-3-5 Una vez el sistema comprobado le da acceso y lo lleva a la página principal de la aplicación. |
| Flujo alternativo | CU 3-1-a 1 El sistema le da un error al cargar la base de datos.                                     |
| Prioridad         | Alta   |
| Frecuencia de uso | Medio  |

Tabla 4: Ejercicios

|                     |   |                     |                  |
|---------------------|---|---------------------|------------------|
| ID:04               | CU-4  | Nombre:             | Añadir ejercicio |
| Creado por:         | Jiliang Qiu   | Fecha de creación   |                  |
| Actor primario:     | Usuario   | Actores secundarios | Ninguno          |
| Descripción:        | Esta función es para añadir ejercicios  |                     |                  |
| Evento activador:   | El usuario hace click en “+”  |                     |                  |
| Precondiciones      | ninguna   |                     |                  |
| Postcondiciones     | ninguna   |                     |                  |
| Flujo normal        | CU-4-1 El usuario introduce los datos   |                     |                  |
|                     | CU-4-2 El sistema comprueba que se ha rellenado todos los campos obligatorios y cumple las reglas (en peso sólo puede ir números) |                     |                  |
|                     | CU-4-3 El sistema crea su el ejercicio y lo muestra en la lista   |                     |                  |
| Flujo alternativo 1 | CU 4-2-2a1 El sistema detecta que no se han rellenado ciertos campos.   |                     |                  |
|                     | CU 4-2-a2 El sistema manda un mensaje que no se han rellenado ciertos campos.   |                     |                  |
|                     | CU-4-2-a-3 El sistema vuelve al flujo normal CU3.1  |                     |                  |
| Flujo alternativo 2 | CU-4-3 a1 El sistema detecta dos ejercicios con el mismo nombre.  |                     |                  |
|                     | CU-4-3 a2 El sistema da un mensaje de error   |                     |                  |
|                     | CU-4-3 a3 El sistema vuelve al flujo normal CU 3-2  |                     |                  |

|                   |       |
|-------------------|-------|
| Prioridad         | Alta  |
| Frecuencia de uso | Medio |

Tabla 5:Añadir ejercicio

|                     |   |                     |                  |
|---------------------|---|---------------------|------------------|
| ID:05               | CU-5  | Nombre:             | Editar ejercicio |
| Creado por:         | Jiliang Qiu   | Fecha de creación   |                  |
| Actor primario:     | Usuario   | Actores secundarios | Ninguno          |
| Descripción:        | El usuario desea editar ejercicios  |                     |                  |
| Evento activador:   | Estar en ejercicios   |                     |                  |
| Precondiciones      | Tiene que haber ejercicios en la base de datos                                |                     |                  |
| Postcondiciones     | ninguna   |                     |                  |
| Flujo normal        | CU-5-1. El usuario selecciona un ejercicio existente.                         |                     |                  |
|                     | CU-5-2. El usuario pulsa "Editar" (o el icono).                               |                     |                  |
|                     | CU-5-3. Cambia los datos necesarios rellenando los campos.                    |                     |                  |
|                     | CU-5-4. El sistema guarda los cambios.  |                     |                  |
| Flujo alternativo 1 | CU 5-2-2a1 El sistema detecta que no se han rellenado ciertos campos.         |                     |                  |
|                     | CU 5-2-a2 El sistema manda un mensaje que no se han rellenado ciertos campos. |                     |                  |
|                     | CU-5-2-a-3 El sistema vuelve al flujo normal CU5.3                            |                     |                  |
| Flujo alternativo 2 | CU-5-3 a 1 El sistema detecta dos ejercicios con el mismo nombre.             |                     |                  |
|                     | CU-5-3 a2 El sistema da un mensaje de error                                   |                     |                  |
|                     | CU-5-3 a3 El sistema vuelve al flujo normal CU 5-3                            |                     |                  |
| Prioridad           | Alta  |                     |                  |
| Frecuencia de uso   | Medio   |                     |                  |

Tabla 6:Editar ejercicio

|             |             |                   |                    |
|-------------|-------------|-------------------|--------------------|
| ID:06       | CU-6        | Nombre:           | Eliminar ejercicio |
| Creado por: | Jiliang Qiu | Fecha de creación |                    |

|                     |  |                     |         |
|---------------------|--|---------------------|---------|
| Actor primario:     | Usuario  | Actores secundarios | Ninguno |
| Descripción:        | El usuario quiere eliminar ejercicios                                    |                     |         |
| Evento activador:   | Estar en ejercicios  |                     |         |
| Precondiciones      | Tiene que haber ejercicios en la base de datos                           |                     |         |
| Postcondiciones     | ninguna  |                     |         |
| Flujo normal        | CU-6-1. El usuario pulsa "Eliminar".                                     |                     |         |
|                     | CU-6-2. El sistema pide confirmación.                                    |                     |         |
|                     | CU-6-3. Si acepta, elimina el ejercicio.                                 |                     |         |
| Flujo alternativo 1 | CU-6-3.a.1 Cancela   |                     |         |
|                     | CU-6-3.a.2 El sistema vuelve a la pantalla anterior vuelve sin eliminar. |                     |         |
| Prioridad           | Alta   |                     |         |
| Frecuencia de uso   | Medio  |                     |         |

Tabla 7: Eliminar ejercicio

|                   |  |                     |            |
|-------------------|--|---------------------|------------|
| ID:07             | CU-7   | Nombre:             | Ver rutina |
| Creado por:       | Jiliang Qiu  | Fecha de creación   |            |
| Actor primario:   | Usuario  | Actores secundarios | Ninguno    |
| Descripción:      | El usuario quiere acceder a la pantalla de "Rutina".                 |                     |            |
| Evento activador: | El usuario hace click en Rutina                                      |                     |            |
| Precondiciones    | ninguna  |                     |            |
| Postcondiciones   | ninguna  |                     |            |
| Flujo normal      | CU-7-1 El sistema carga todos los días de la semana lunes a domingo. |                     |            |

|                   |   |
|-------------------|---|
|                   | CU-7-2 El usuario hace click en uno de los días |
|                   | CU-7-3 El sistema carga los ejercicios del día  |
| Prioridad         | Alta  |
| Frecuencia de uso | Alta  |

Tabla 8: Ver rutina

|                     |   |                     |            |
|---------------------|---|---------------------|------------|
| ID:08               | CU-8  | Nombre:             | Ver comida |
| Creado por:         | Jiliang Qiu   | Fecha de creación   |            |
| Actor primario:     | Usuario   | Actores secundarios | Ninguno    |
| Descripción:        | El usuario quiere acceder a la pantalla de "Nutrición" y luego seleccionar "comida"                                     |                     |            |
| Evento activador:   | El usuario hace click en comida   |                     |            |
| Precondiciones      | ninguna   |                     |            |
| Postcondiciones     | ninguna   |                     |            |
| Flujo normal        | CU-8-1 El sistema carga todas las comidas agregadas   |                     |            |
|                     | CU-8-2 El sistema muestra todas las comidas en pantalla   |                     |            |
|                     | CU-8-3 El usuario puede añadir una comida pulsando el botón "+" en la pantalla.   |                     |            |
|                     | CU 8-4 El usuario rellena los campos necesarios   |                     |            |
|                     | CU-8-5 El sistema comprueba que todos los campos están rellenos completamente   |                     |            |
|                     | CU-8-6 El sistema añade la comida y actualiza la lista.   |                     |            |
| Flujo Alternativo 1 | CU-8-5-a 1 El sistema detecta que no se han relleno todos los campos necesarios   |                     |            |
|                     | CU-8-5-a2 El sistema muestra un mensaje de error en los campos que no han sido rellenos para que el usuario las rellene |                     |            |
|                     | CU-8-5 a3 El sistema vuelve al flujo normal CU 9-4  |                     |            |
| Flujo Alternativo 2 | CU-8-6a1 El sistema detecta duplicidad y da un error.   |                     |            |
|                     | CU-8-6a2 El sistema vuelve al flujo normal CU 9-4   |                     |            |
| Prioridad           | Alta  |                     |            |
| Frecuencia de uso   | Medio   |                     |            |

Tabla 9: Ver comida

|                     |  |                     |               |
|---------------------|--|---------------------|---------------|
| ID:09               | CU-9   | Nombre:             | Añadir comida |
| Creado por:         | Jiliang Qiu  | Fecha de creación   |               |
| Actor primario:     | Usuario  | Actores secundarios | Ninguno       |
| Descripción:        | El usuario está en la pantalla de rutina con un día de la semana seleccionado.   |                     |               |
| Evento activador:   | El usuario hace click en el "+" de rutina  |                     |               |
| Precondiciones      | Tener ejercicio guardados en la base de datos  |                     |               |
| Postcondiciones     | ninguna  |                     |               |
| Flujo normal        | CU-9-1. El usuario pulsa "+"   |                     |               |
|                     | CU-9-2. El usuario rellena los campos necesarios   |                     |               |
|                     | CU 9-3 sistema comprueba que todos los campos están rellenos completamente   |                     |               |
|                     | CU-9-4 El sistema añade la comida y actualiza la lista.  |                     |               |
| Flujo Alternativo 1 | CU-9-3-a 1 El sistema detecta que no se han relleno todos los campos necesarios  |                     |               |
|                     | CU-9-3-a 2 El sistema muestra un mensaje de error en los campos que no han sido rellenos para que el usuario las rellene |                     |               |
|                     | CU-9-3-a 3 El sistema vuelve al flujo normal CU 9-2  |                     |               |
| Flujo Alternativo 2 | CU-9-4-a1 El sistema detecta duplicidad y da un error.   |                     |               |
|                     | CU-9-4-a2 El sistema vuelve al flujo normal CU 9-2   |                     |               |
| Prioridad           | Alta   |                     |               |
| Frecuencia de uso   | Medio  |                     |               |

Tabla 10: Añadir comida

|                   |  |                     |               |
|-------------------|--|---------------------|---------------|
| ID:10             | CU-10  | Nombre:             | Editar comida |
| Creado por:       | Jiliang Qiu  | Fecha de creación   |               |
| Actor primario:   | Usuario  | Actores secundarios | Ninguno       |
| Descripción:      | El usuario desea editar comidas ya creadas             |                     |               |
| Evento activador: | Estar en comida  |                     |               |
| Precondiciones    | Tiene que haber comidas existentes en la base de datos |                     |               |

|                     |  |
|---------------------|--|
| Postcondiciones     | ninguna  |
| Flujo normal        | CU-10-1. El usuario selecciona una comida existente.                           |
|                     | CU-10-2. El usuario pulsa "Editar" (o el icono) .                              |
|                     | CU-10-3. Cambia los datos necesarios rellenando los campos.                    |
|                     | CU-10-4. El sistema guarda los cambios.  |
| Flujo alternativo 1 | CU 10-2-2a1 El sistema detecta que no se han rellenado ciertos campos.         |
|                     | CU 10-2-a2 El sistema manda un mensaje que no se han rellenado ciertos campos. |
|                     | CU-10-2-a-3 El sistema vuelve al flujo normal CU10-3                           |
| Flujo alternativo 2 | CU-10-3 a 1 El sistema detecta dos comidas con el mismo nombre.                |
|                     | CU-10-3 a2 El sistema da un mensaje de error                                   |
|                     | CU-10-3 a3 El sistema vuelve al flujo normal CU 10-3                           |
| Prioridad           | Alta   |
| Frecuencia de uso   | bajo   |

Tabla 11: Editar comida

|                     |  |                     |                 |
|---------------------|--|---------------------|-----------------|
| ID:11               | CU-11  | Nombre:             | Eliminar comida |
| Creado por:         | Jiliang Qiu  | Fecha de creación   |                 |
| Actor primario:     | Usuario  | Actores secundarios | Ninguno         |
| Descripción:        | El usuario quiere eliminar una comida  |                     |                 |
| Evento activador:   | Estar en nutrición   |                     |                 |
| Precondiciones      | Tiene que haber comidas creadas en la base de datos                              |                     |                 |
| Postcondiciones     | ninguna  |                     |                 |
| Flujo normal        | CU-11-1. El usuario pulsa "Eliminar".  |                     |                 |
|                     | CU-11-2. El sistema pide confirmación.   |                     |                 |
|                     | CU-11-3. Si acepta, elimina la comida seleccionada.                              |                     |                 |
| Flujo alternativo 1 | CU-11-3.a.1 Cancela  |                     |                 |
|                     | CU-11-3.a.2 El sistema vuelve a la pantalla anterior vuelve sin eliminar (11-1). |                     |                 |

|                   |      |
|-------------------|------|
| Prioridad         | Alta |
| Frecuencia de uso | bajo |

Tabla 12: Eliminar Comida

|                   |  |                     |           |
|-------------------|--|---------------------|-----------|
| ID:12             | CU-12  | Nombre:             | Ver dieta |
| Creado por:       | Jiliang Qiu  | Fecha de creación   |           |
| Actor primario:   | Usuario  | Actores secundarios | Ninguno   |
| Descripción:      | El usuario quiere acceder a la pantalla de "Dieta".                                      |                     |           |
| Evento activador: | El usuario hace click en Dieta   |                     |           |
| Precondiciones    | ninguna  |                     |           |
| Postcondiciones   | ninguna  |                     |           |
| Flujo normal      | CU-12-1 El sistema carga todos los días de la semana lunes a domingo.                    |                     |           |
|                   | CU-12-2 El usuario hace clic en uno de los días  |                     |           |
|                   | CU-12-3 El sistema carga las 4 secciones que hay que son desayuno, comida, cena y otros. |                     |           |
|                   | CU-12-3 El usuario pulsa en una de las 4 opciones  |                     |           |
|                   | CU-12-4 El sistema carga todas las comidas de esa sección                                |                     |           |
| Prioridad         | Alta   |                     |           |
| Frecuencia de uso | Alta   |                     |           |

Tabla 13: Ver dieta

|                   |   |                     |            |
|-------------------|---|---------------------|------------|
| ID:13             | CU-13   | Nombre:             | Ver logros |
| Creado por:       | Jiliang Qiu                                   | Fecha de creación   |            |
| Actor primario:   | Usuario                                       | Actores secundarios | Ninguno    |
| Descripción:      | El usuario quiere ver sus logros de la semana |                     |            |
| Evento activador: | El usuario hace click en logros               |                     |            |
| Precondiciones    | ninguna                                       |                     |            |

|                   |   |
|-------------------|---|
| Postcondiciones   | ninguna   |
| Flujo normal      | CU-13-1 El usuario pulsa en puntuación  |
|                   | CU-13-2 El sistema le muestra la puntuación el historial de las puntuaciones. |
| Prioridad         | Baja  |
| Frecuencia de uso | Baja  |

Tabla 14: Ver logros

|                   |   |                     |                       |
|-------------------|---|---------------------|-----------------------|
| ID:14             | CU-14   | Nombre:             | Ver detalle ejercicio |
| Creado por:       | Jiliang Qiu   | Fecha de creación   |                       |
| Actor primario:   | Usuario   | Actores secundarios | Ninguno               |
| Descripción:      | El usuario quiere ver en detalle el ejercicio   |                     |                       |
| Evento activador: | El usuario hace click en la comida  |                     |                       |
| Precondiciones    | Debe haber comida registrado en la base de datos  |                     |                       |
| Postcondiciones   | ninguna   |                     |                       |
| Flujo normal      | CU-14-1 El usuario hace click en una de las comidas existentes en la pantalla de comida |                     |                       |
|                   | CU-14-2 El sistema abre en detalle la comida  |                     |                       |
| Prioridad         | Baja  |                     |                       |
| Frecuencia de uso | Baja  |                     |                       |

Tabla 15: Ejercicio detalle

|                   |   |                     |          |
|-------------------|---|---------------------|----------|
| ID:15             | CU-15   | Nombre:             | Ver Peso |
| Creado por:       | Jiliang Qiu                                   | Fecha de creación   |          |
| Actor primario:   | Usuario                                       | Actores secundarios | Ninguno  |
| Descripción:      | El usuario quiere ver el peso y su historial. |                     |          |
| Evento activador: | El usuario hace click en peso.                |                     |          |
| Precondiciones    | ninguna                                       |                     |          |

|                   |   |
|-------------------|---|
| Postcondiciones   | ninguna   |
| Flujo normal      | 15-1 El usuario hace click en peso              |
|                   | 15-2 El sistema carga todo el historial de peso |
|                   | 15-3 El sistema carga la gráfica de pesos       |
| Prioridad         | baja  |
| Frecuencia de uso | Medio   |

Tabla 16: Ver peso

|                     |  |                     |             |
|---------------------|--|---------------------|-------------|
| ID:16               | CU-16  | Nombre:             | añadir peso |
| Creado por:         | Jiliang Qiu  | Fecha de creación   |             |
| Actor primario:     | Usuario  | Actores secundarios | Ninguno     |
| Descripción:        | El usuario quiere añadir su peso   |                     |             |
| Evento activador:   | El usuario hace click en peso.   |                     |             |
| Precondiciones      | ninguna  |                     |             |
| Postcondiciones     | ninguna  |                     |             |
| Flujo normal        | CU-16-1 El sistema carga los pesos que se han introducido  |                     |             |
|                     | CU-16-2 El usuario introduce un nuevo peso.  |                     |             |
|                     | CU-16-3 El sistema detecta si los campos son numéricos   |                     |             |
|                     | CU 16-4 El sistema añade el peso y actualiza la lista  |                     |             |
|                     | CU-16-5 El sistema actualiza la gráfica.   |                     |             |
| Flujo Alternativo 1 | CU-16-3-a 1 El sistema detecta que no se han rellenado todos los campos necesarios o que no es numérico. |                     |             |
|                     | CU-16-3 a2 El sistema muestra un mensaje de error  |                     |             |
|                     | CU-16-3 a3 El sistema vuelve al flujo normal CU 16-2   |                     |             |

|                     |  |
|---------------------|--|
| Flujo Alternativo 2 | CU-16-6a1 El sistema detecta duplicidad y da un error. |
|                     | CU-16-6a2 El sistema vuelve al flujo normal CU 16-4    |
| Prioridad           | baja   |
| Frecuencia de uso   | Medio  |

Tabla 17: Añadir peso

|                   |   |                     |                |
|-------------------|---|---------------------|----------------|
| ID:17             | CU-17   | Nombre:             | Ver Calendario |
| Creado por:       | Jiliang Qiu   | Fecha de creación   |                |
| Actor primario:   | Usuario   | Actores secundarios | Ninguno        |
| Descripción:      | El usuario quiere ver el calendario y los eventos próximos                              |                     |                |
| Evento activador: | El usuario hace clic en calendario desde la página principal.                           |                     |                |
| Precondiciones    | ninguna   |                     |                |
| Postcondiciones   | ninguna   |                     |                |
| Flujo normal      | CU-17-1 El sistema carga todo el calendario y los eventos si hay previamente guardados. |                     |                |
|                   | CU-17-2 El usuario puede introducir un nuevo evento                                     |                     |                |
|                   | CU-17-3 El sistema añade el evento y se actualiza                                       |                     |                |
| Prioridad         | Baja  |                     |                |
| Frecuencia de uso | Baja  |                     |                |

Tabla 18: Ver calendario

|                   |  |                     |                                  |
|-------------------|--|---------------------|----------------------------------|
| ID:18             | CU-18  | Nombre:             | validar datos                    |
| Creado por:       | Jiliang Qiu  | Fecha de creación   |                                  |
| Actor primario:   | Usuario  | Actores secundarios | Sistema de autenticación interna |
| Descripción:      | El usuario quiere acceder a la pantalla de "Rutina". |                     |                                  |
| Evento activador: | El usuario hace click en Rutina                      |                     |                                  |

|                   |   |
|-------------------|---|
| Precondiciones    | El usuario debe estar en la pantalla de inicio de sesión                              |
| Postcondiciones   | Si los datos son correctos el sistema concederá el acceso                             |
| Flujo normal      | CU-18-1 El usuario introduce sus credenciales   |
|                   | CU-18-2 El sistema valida si el usuario existe  |
|                   | CU-18-3 El sistema compara la contraseña introducida del usuario con la base de datos |
|                   | CU-18-4 Si son correctos el usuario iniciará sesión                                   |
| Flujo alternativo | CU-18-2-a1 El sistema comprueba que los datos son erróneos                            |
|                   | CU-18-2-a1 El sistema muestra un mensaje de error informando al usuario               |
|                   | CU-18-2-a1 Vuelve al flujo normal 18-1  |
| Prioridad         | Alta  |
| Frecuencia de uso | Alta  |

Tabla 19: Validar datos

|                   |   |                     |                                  |
|-------------------|---|---------------------|----------------------------------|
| ID:19             | CU-19   | Nombre:             | Cerrar sesión                    |
| Creado por:       | Jiliang Qiu   | Fecha de creación   |                                  |
| Actor primario:   | Usuario   | Actores secundarios | Sistema de autenticación interna |
| Descripción:      | El usuario quiere cerrar sesión   |                     |                                  |
| Evento activador: | El usuario hace click en Rutina   |                     |                                  |
| Precondiciones    | El usuario debe haber iniciado sesión                                     |                     |                                  |
| Postcondiciones   | La sesión del usuario se cerrará  |                     |                                  |
| Flujo normal      | CU-19-1 El usuario pulsa el botón de cerrar sesión.                       |                     |                                  |
|                   | CU-19-2 El sistema redirige al usuario a la pantalla de inicio de sesión. |                     |                                  |
| Prioridad         | Alta  |                     |                                  |
| Frecuencia de uso | Baja  |                     |                                  |

### 3.5.3. Diagrama de clases

Un diagrama de clases estructura de datos de un sistema con atributos, métodos y relaciones entre ellas.

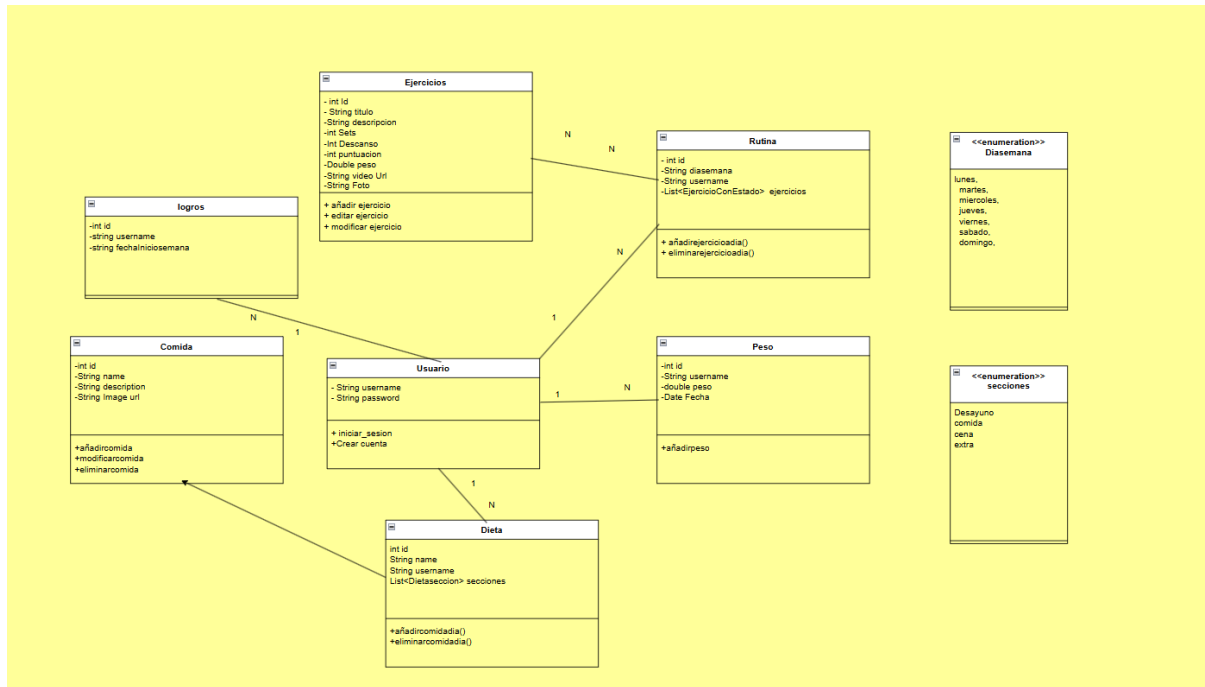


Ilustración 36: Diagrama de clases

A continuación, se va a explicar la ilustración que contiene el diagrama de clases. El diagrama de clases está compuesto por:

**Usuario:** Es una clase que almacena la información del usuario de la aplicación y tiene relación 1 a N con Peso, Rutina, historial (logros), Dieta. Porque cada usuario tiene su propio peso, historial (logros), Dieta.

**Rutina:** Esta clase representa la rutina de ejercicios para un día específico de la semana. Cada rutina contiene una lista de ejercicios con su estado hecho o no.

**Ejercicios:** Esta clase almacena los ejercicios físicos creados por el usuario. Contiene toda la información necesaria de dicho ejercicio y que posteriormente se podrá añadirse en rutinas.

**Logros** Almacena la puntuación de la semana obtenida por completar ejercicios y también guarda el historial de las semanas.

Dietadia: Almacena las comidas dependiendo del día de la semana. Cada día se divide en cuatro secciones: desayuno, comida, cena y extra.

Comida: Esta clase almacena todas las comidas creadas por el usuario que posteriormente se utilizaran en dietas.

Peso: Almacena y tiene un historial del peso del usuario para poder hacer un seguimiento de su progreso físico en caso de interesarle este apartado.

### 3.6. Implementación de la solución.

Continuando el modelo de desarrollo en cascada una vez finalizadas las etapas de análisis y diseño llega el momento del desarrollo final. En este apartado se va a explicar cómo ha sido la implementación y qué arquitectura se ha utilizado y como han sido los pasos del desarrollo y los elementos que lo forman. También se explicará los datos que se manejan, cómo se tratan y cómo se guardan.

La primera base de datos que se creó es la del ejercicio la cual almacena todos los ejercicios creados por el usuario para posteriormente ser utilizado en rutinas. La segunda base de datos es la de la comida en la cual se guardarán todas las comidas que haya añadido el usuario para posteriormente ser utilizado en dietas.

#### 3.6.1. Herramientas y entorno de desarrollo.

**Android** es un sistema operativo diseñado principalmente para teléfonos y relojes inteligentes.

Android, el sistema operativo móvil de Google, representa la mayor parte del mercado mundial de sistemas operativos móviles, con una cuota de alrededor del 72.15%.



Ilustración 33: Arquitectura de Android Keepcoding.io

## Android Studio IDE

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. En esta práctica se ha usado para emular un dispositivo móvil con el sistema operativo de Android.

- Es el IDE **oficial** recomendado por Google.
- Tiene **actualizaciones frecuentes** con nuevas funciones.
- Permite **crear y probar** apps desde cero sin usar otras herramientas externas. Gran comunidad, documentación y compatibilidad con librerías.
- Probar la app en **emuladores** de teléfonos virtuales o directamente en tu dispositivo físico.
- Integración con GitHub en la cual ayuda a compilar funciones de app comunes.

## Visual Studio Code

Es un **editor de código ligero y rápido** desarrollado por **Microsoft**. Y se puede programar en una gran variedad de lenguajes como JavaScript, Python, C++, Java[9], Dart [10] ... A continuación se detalla sus principales ventajas

1. **Ligereza y velocidad** Visual Studio Code es un editor de código ligero que se ejecuta con fluidez, en equipos con recursos limitados.
2. **Soporte para múltiples lenguajes de programación** Visual Studio Code no se limita a un solo lenguaje. Gracias a su sistema de extensiones, es posible trabajar con lenguajes como JavaScript, Python, Java, C++, Dart, Go....
3. **Gran personalización mediante extensiones** Visual Studio Code tiene un ecosistema muy grande de extensiones disponibles en su “marketplace”. Estas extensiones permiten añadir funciones, integraciones externas, soporte para nuevos lenguajes, temas visuales...
4. **Integración con sistemas de control de versiones:** el editor incluye integración nativa con Github esto permite realizar tareas como realizar commits, ver el historial de cambios y gestionar sin salir del entorno de trabajo.

5. **Interfaz sencilla y fácil de aprender** la interfaz de Visual Studio Code es más intuitiva que en otros entornos.
6. **Gran comunidad y soporte activo** es uno de los editores más conocidos, gracias a ello Visual Studio Code tiene una comunidad muy activa que contribuye día a día con documentación, tutoriales, temas y extensiones. Además lo actualizan frecuentemente.
7. **Adecuado para Flutter y desarrollo multiplataforma** Visual studio code es ampliamente utilizado para el desarrollo de aplicaciones móviles multiplataforma, especialmente con Flutter y Dart.

Dart es un lenguaje de programación creado por Google con la finalidad de desarrollar aplicaciones modernas. La que el framework flutter tiene como lenguaje base que permite la creación de aplicaciones móviles, de web o incluso de escritorio desde un código.

Características

- Sintaxis clara y fácil
- Soporte para programación asíncrona
- orientación a objetos
- Actualizaciones frecuentes
- Hot reload: permite hacer cambios en el código y ver resultados casi al instante

**Draw.io** es una herramienta gratuita que se puede acceder desde cualquier navegador que permite crear diagramas de manera fácil y visual. Es comúnmente utilizada para crear diagramas de flujo, esquemas de bases de datos, arquitecturas de software, mapas mentales, diagramas UML. Una de sus principales ventajas es que no requiere instalación y permite guardar los proyectos en la nube como Google Drive o OneDrive. Además, tiene una interfaz intuitiva y su amplia galería de formas y conectores que hace que sea una opción muy sencilla para todo tipo de usuarios.

**SQLite** es un sistema de gestión de bases de datos relacional que se integra directamente en las aplicaciones sin instalar un servidor externo. Se ha escogido este sistema porque es la más utilizada debido a su rapidez, simpleza y eficiencia. Al ser una base de datos de un archivo único esto permite consultar los datos de manera local y hace que nuestro dispositivo no necesite conexión a internet. La estructura está basada en SQL estándar lo que la hace una opción muy accesible además su consumo de recursos es muy bajo lo que es ideal para dispositivos no tan sofisticados.

### 3.6.2. Arquitectura

Este proyecto se ha realizado un patrón de diseño llamada MVC es un patrón muy usado para organizar el código en aplicaciones dividiendo así responsabilidades con el objetivo de que el código sea más ordenado.

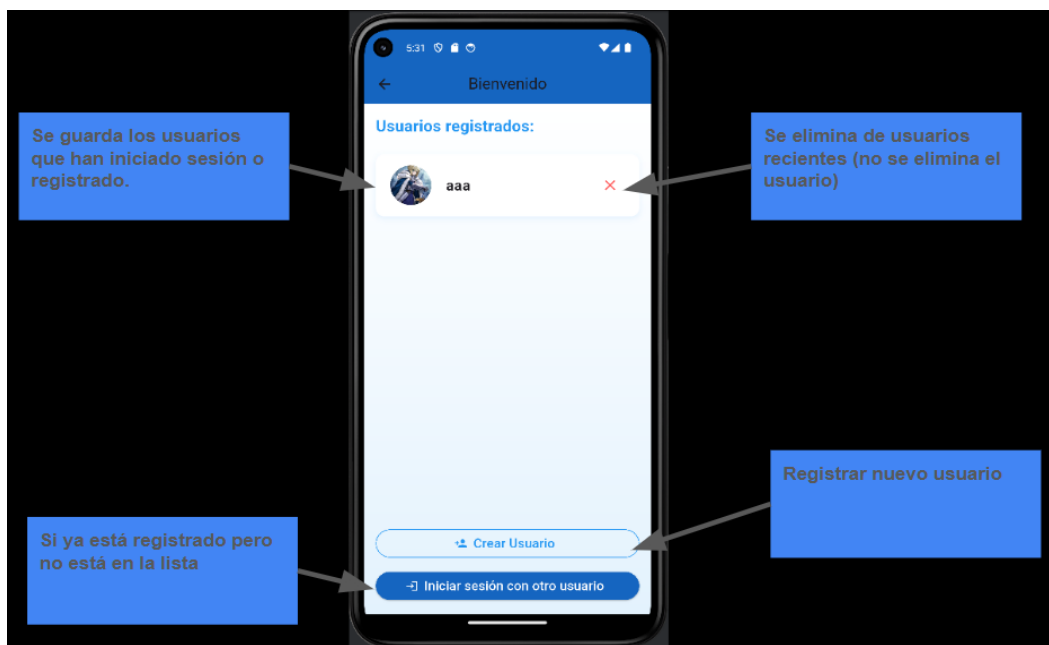
MVC representa Model, View y Controller

- Model(modelo): es la capa donde se manejan los datos y la lógica. En esta capa se definen las estructuras de datos como se guardan y se trabajan con ellos.
- View(vista): Es responsable de lo que ve el usuario, la interfaz. Su principal función es mostrar información al usuario de lo que recibe del modelo y detectar las acciones del usuario.
- Controller(controlador): Es el intermediario recibe las acciones desde la vista procesa esa información y hace cambios en la capa del Modelo o de la Vista dependiendo de la acción realizada.

## 4. Resultado

### 4.1. Resultados obtenidos

En la pantalla de inicio como se puede ver en la ilustración se puede iniciar sesión o crear nuevo usuario. Hay dos formas de iniciar sesión cuando está en la lista o en caso de no estar en la lista usar la opción “iniciar sesión con otro usuario” en los dos métodos hace falta iniciar sesión con contraseña.



La ilustración que se ve a continuación son las pantallas de registro y pantalla de iniciar sesión.

Ilustración 37: Página de inicio de sesión Creación propia

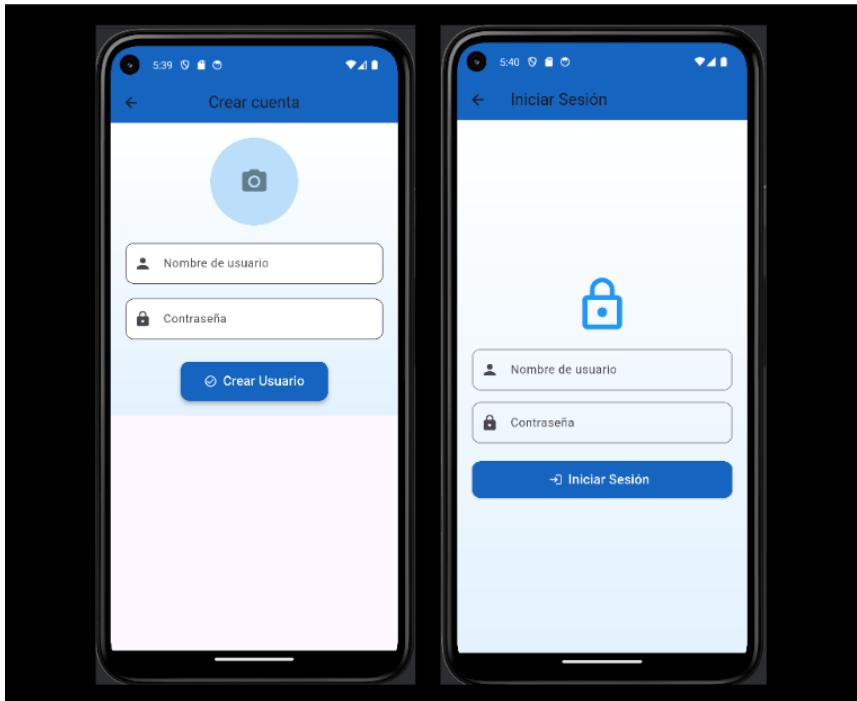


Ilustración 38: pantalla de inicio de sesión y registro creación propia

Pantalla principal

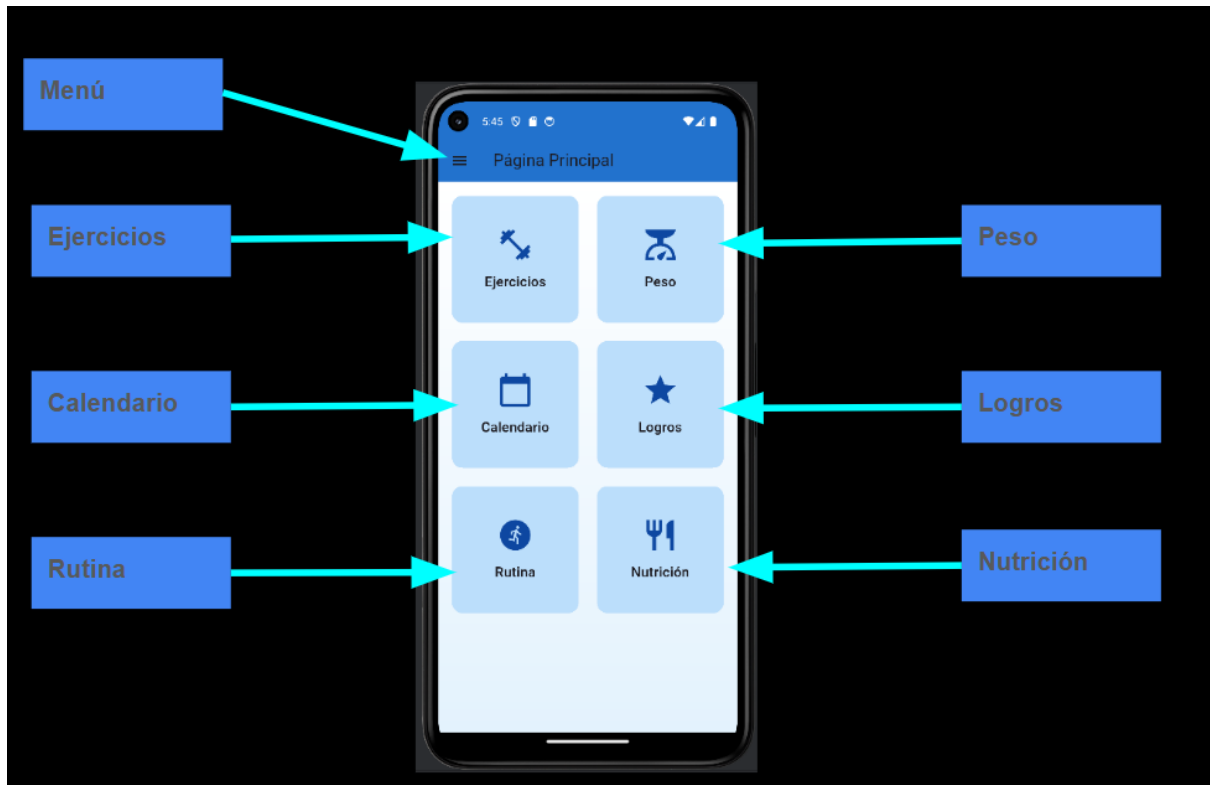


Ilustración 39: Pantalla principal Creación propia

Cambiar foto de perfil.

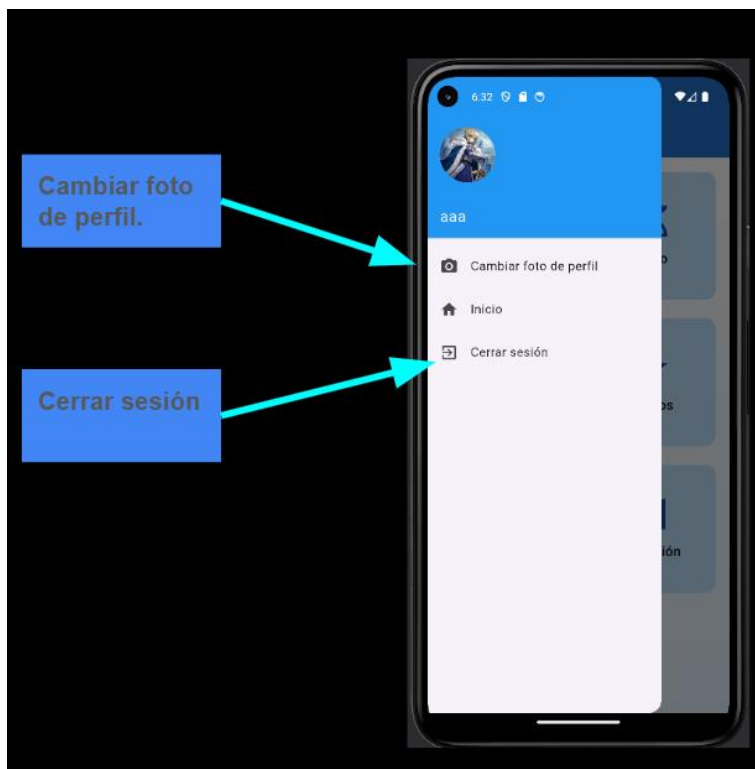


Ilustración 40: Cambiar foto de perfil

Pantalla ejercicios donde se ve la lista de ejercicios creados también se puede crear, modificar o eliminar.

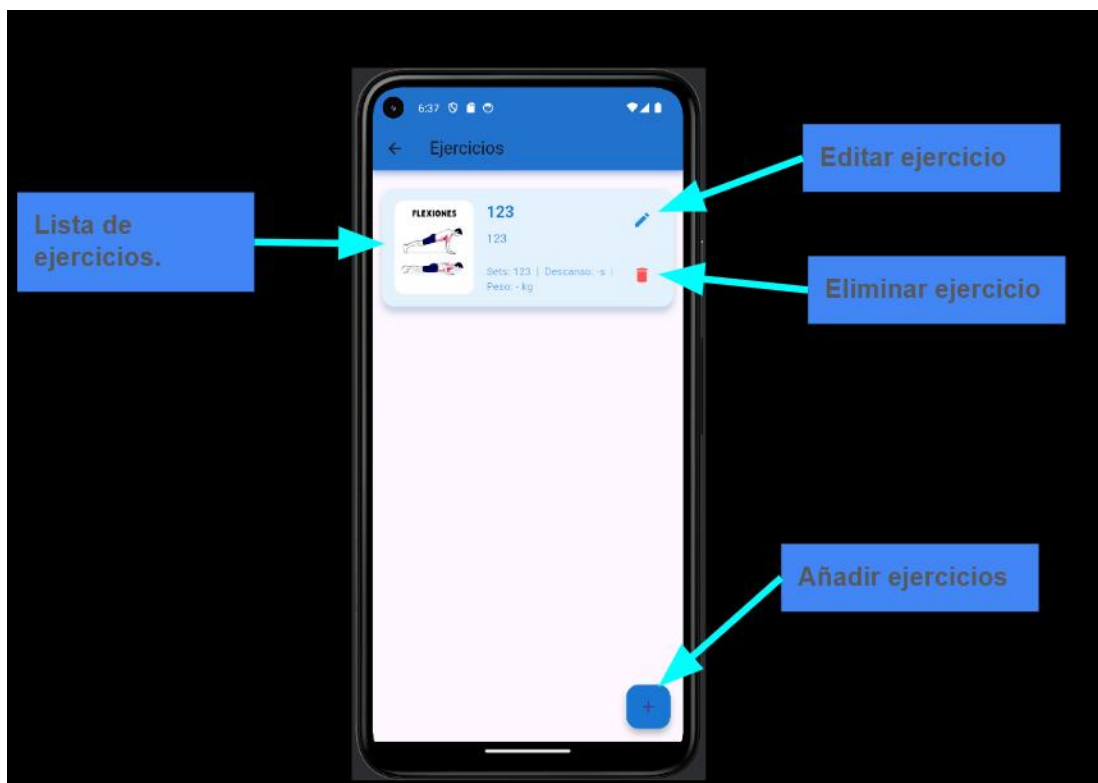


Ilustración 41: Pantalla de ejercicios creación propia

Pantalla de crear ejercicio y editar ejercicios.

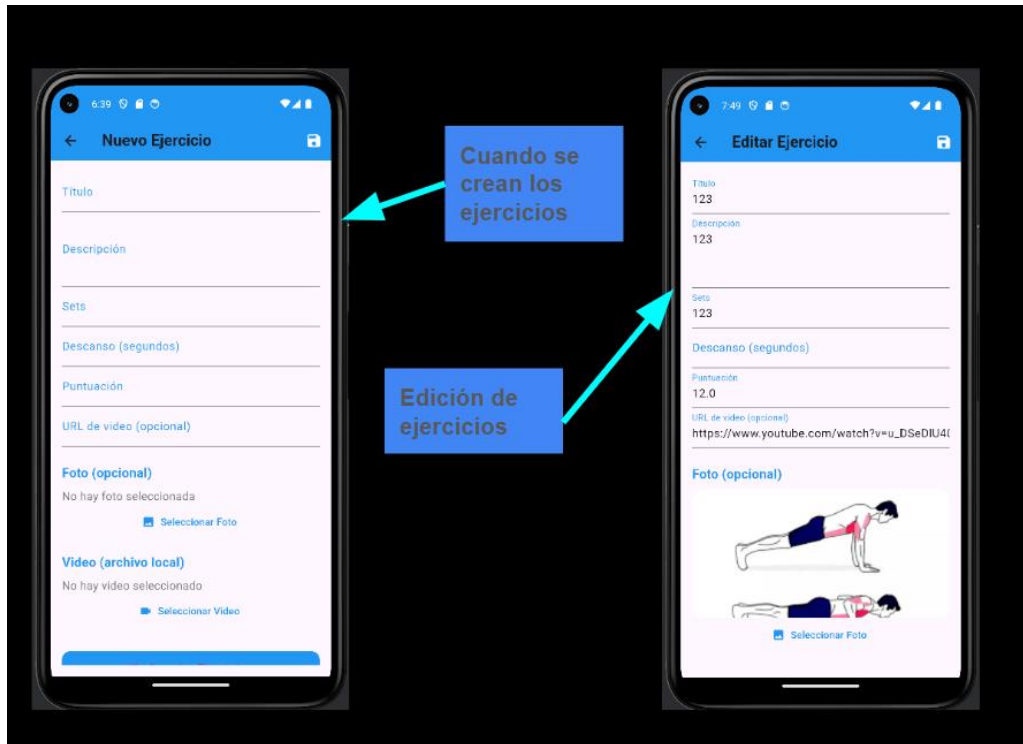


Ilustración 42: creación de ejercicios y edición de ejercicios

Pantalla de peso

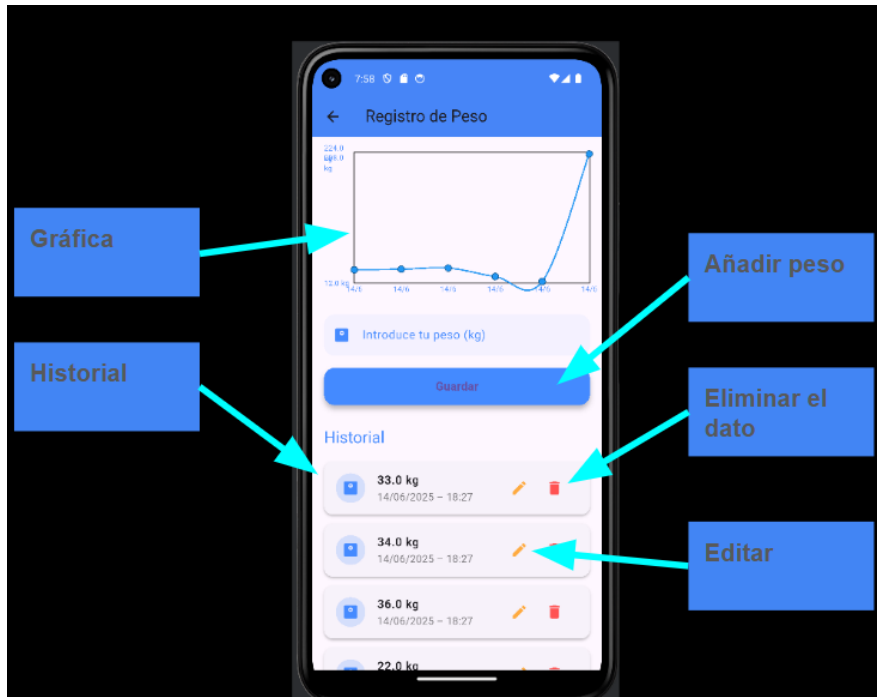


Ilustración 43: Inserción de peso

## Calendario

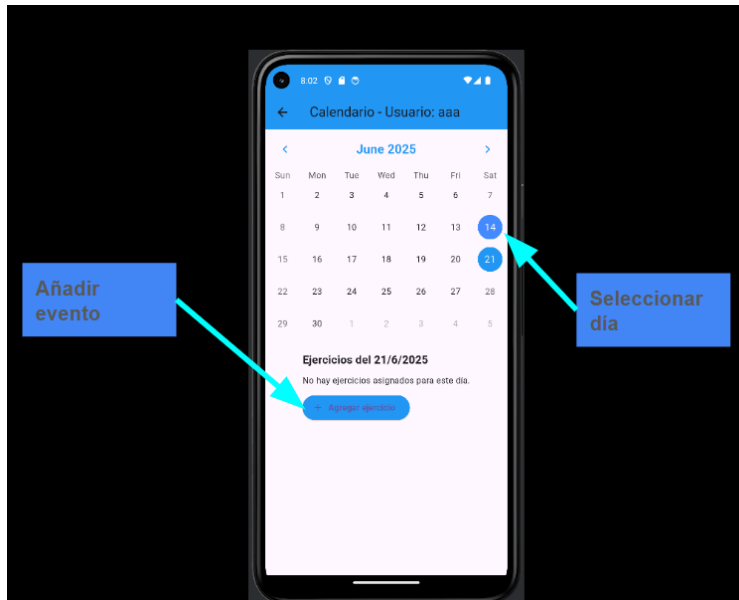


Ilustración 44: Calendario

## 4.2. Tablas de pruebas

| Inicio de sesión |                                  |                           |  |  |
|------------------|----------------------------------|---------------------------|--|--|
| ID               | Prueba                           | Precondición              | Resultado esperado                                   | Resultado obtenido   |
| 1.1              | Registrar usuario                | ninguna                   | Registrar correctamente                              | Aprobada   |
| 1.2              | Iniciar sesión                   | Estar registrado          | Iniciar sesión correctamente                         | Aprobada   |
| 1.3              | Equivocarse al iniciar de sesión | ninguna                   | Enviar un mensaje de error                           | Fallido<br>Se bloqueo la aplicación                              |
| 1.3.1            | Equivocarse al iniciar de sesión | ninguna                   | Enviar un mensaje de error                           | Aprobada   |
| Página principal |                                  |                           |  |  |
| ID               | Prueba                           | Precondición              | Resultado esperado                                   | Resultado obtenido   |
| 2.1              | Cambiar foto de perfil           | Iniciar sesión            | Se refleje el cambio de foto de perfil               | Fallido.<br>Solo se cambiaba en perfil y no en inicio de sesión. |
| 2.1.1            | Cambiar foto de perfil           | Iniciar sesión            | Se refleje el cambio de foto de perfil               | Aprobada   |
| 2.2              | Acceder a ejercicio              |                           | Se accede a la pantalla de ejercicio                 | Aprobada   |
| 2.3              | Acceder a peso                   |                           | Se accede a la pantalla de peso                      | Aprobada   |
| 2.4              | Acceder a calendario             |                           | Se accede a la pantalla de calendario                | Aprobada   |
| 2.5              | Acceder a logros                 |                           | Se accede a la pantalla de logros                    | Aprobada   |
| 2.6              | Acceder a rutina                 |                           | Se accede a la pantalla de rutina                    | Aprobada   |
| 2.7              | Acceder a nutrición              |                           | Se accede a la pantalla de nutrición                 | Aprobada   |
| 2.8              | Cerrar sesión                    |                           | Cerrar sesión y va a la pantalla de inicio de sesión | Aprobada   |
| Ejercicios       |                                  |                           |  |  |
| ID               | Prueba                           | Precondición              | Resultado esperado                                   | Resultado obtenido   |
| 3.1              | Crear ejercicio                  | Ninguna                   | Añadir un ejercicio a la lista                       | Aprobada   |
| 3.2              | Editar ejercicio                 | Tener un ejercicio creado | Modificar dicho ejercicio                            | Aprobada   |
| 3.3              | Eliminar ejercicio               | Tener un ejercicio creado | Borrar ese ejercicio de la base de datos.            | Aprobada   |

| 3.4        | Ver detalles de ejercicio  |                                  | Ver detalles de los ejercicios.  | Aprobada           |
|------------|--|----------------------------------|--|--------------------|
| 3.5        | Crear ejercicio llenando los campos mal                          | Tener un ejercicio creado        | Salta un mensaje de campos vacíos o inválidos                                  | Aprobada           |
| 3.6        | Modificar ejercicios e introducir valores erróneos en los campos | Tener un ejercicio creado        | Salta un mensaje de campos vacíos o inválidos                                  | Aprobada           |
| Peso       |  |                                  |  |                    |
| ID         | Prueba   | Precondición                     | Resultado esperado   | Resultado obtenido |
| 4.1        | Introducir peso  | ninguna                          | Guardar el peso introducido  | Aprobada           |
| 4.2        | Introducir peso con valores no válidos                           |                                  | Saltar un mensaje informando al usuario que no es válido lo que ha introducido | Aprobada           |
| 4.3        | Editar peso  | Tener un peso registrado         | Que los cambios se reflejen  | Aprobada           |
| 4.4        | Introducir peso con valores no válidos en la edición.            | Tener un peso registrado         | Alerta que el valor introducido es incorrecto                                  | Aprobada           |
| 4.5        | Eliminar peso  | Tener un peso registrado         | Se borra el peso   | Aprobada           |
| 4.6        | Mostrar gráfica  | Tener 2 pesos diferentes creados | Mostrar la gráfica con los datos   | Fallo daba error   |
| 4.6.1      | Mostrar gráfica  | Tener 2 pesos diferentes creados | Mostrar la gráfica con los datos   | Aprobada           |
| Calendario |  |                                  |  |                    |
| ID         | Prueba   | Precondición                     | Resultado esperado   | Resultado obtenido |
| 5.1        | Agregar evento   | Ninguna                          | Añadir el evento   | Aprobada           |
| 5.2        | Editar evento  | Tener un evento creado           | Hacer los cambios deseados por el usuario en dicho evento                      | Aprobada           |
| 5.3        | Eliminar evento  | Tener un evento creado           | Borrar ese evento  | Aprobada           |
| Logros     |  |                                  |  |                    |
| ID         | Prueba   | Precondición                     | Resultado esperado   | Resultado obtenido |
| 6.1        | Visualizar logros  |                                  | Poder ver los logros   | Aprobada           |
| Rutina     |  |                                  |  |                    |
| ID         | Prueba   | Precondición                     | Resultado esperado   | Resultado obtenido |
| 7.1        | Seleccionar los días   |                                  | Poder seleccionar los días   | Aprobada           |
| 7.2        | Añadir ejercicios a la rutina                                    |                                  | La lista se actualiza  | Aprobada           |

|           | del día   |   |   |                    |
|-----------|---|---|---|--------------------|
| 7.3       | Quitar ejercicio a la rutina  | Tener ejercicios guardados en la base de datos    |   | Aprobada           |
| 7.4       | Borrar un ejercicio de la base de datos cuando el ejercicio está en una rutina. | Tener el ejercicio creado y asignado a una rutina | Que el ejercicio borrado desaparezca en la rutina y en los ejercicios | Aprobada           |
| 7.5       | Comprobar que el buscador funciona  | Tener ejercicios                                  | El buscador funciona correctamente                                    | Aprobada           |
| Nutrición |   |   |   |                    |
| ID        | Prueba  | Precondición                                      | Resultado esperado  | Resultado obtenido |
| 8.1       | Crear comida  | Ninguna   | Se añade la comida a la base de datos.                                | Aprobada           |
| 8.2       | Editar comida   | Tener una comida                                  | Se modifica dicha comida.   | Aprobada           |
| 8.3       | Eliminar comida   | Tener una comida registrada en la base de datos   | Se elimina dicha comida de la base de datos                           | Aprobada           |
| 8.4       | Ver detalles de comida  |   | Se accede a la pantalla para tener más información de la comida       | Aprobada           |
| 8.5       | Añadir comida en una de las dietas  | Tener comidas creadas.                            | Se añade esa comida a   | Aprobada           |
| 8.6       | Eliminar comida de las dietas   | Tener una comida en dieta                         | Eliminar comida   | Aprobada           |

Tabla 21: pruebas sobre la aplicación

### 4.3. Arquitectura

El proyecto ha sido desarrollado siguiendo el patrón de arquitectura MVC con la finalidad de mantener una estructura clara y escalable. Esta organización hace más fácil la separación de responsabilidades, también hace que el mantenimiento sea eficiente y por último que el código pueda evolucionar coherentemente a lo largo del tiempo.

La organización se divide en 3 secciones como se menciono antes cada sección en el proyecto desarrollado es una carpeta.

#### 4.3.1. Models

En esta carpeta se encuentra los modelos definidos que representan las entidades principales de la aplicación. Cada archivo tiene una clase que define los atributos y sus métodos.

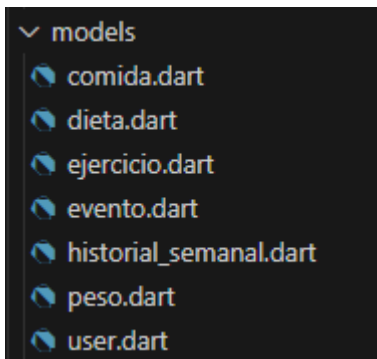


Ilustración 45 modelos creación propia

### 4.3.2. Controllers

En esta carpeta contiene los archivos que gestiona la lógica del negocio y la interacción entre vista y modelo. Los controladores reciben acciones del usuario y con esa información procesan para luego actualizar la vista.

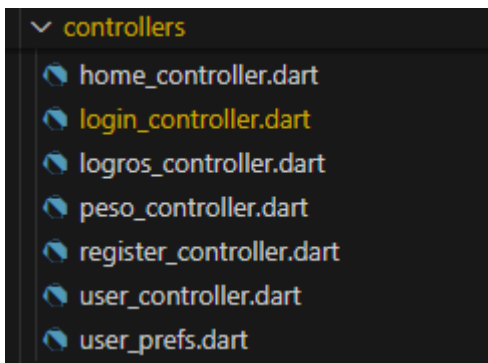


Ilustración 46 Controladores creación propia

### 4.3.3. View

Esta carpeta tiene todas las interfaces o vistas de la aplicación utilizando StatelessWidget o StatefulWidget. StatefulWidget son herramientas que se cambian cuando se interacciona con ellas mientras StatelessWidget nunca cambian.

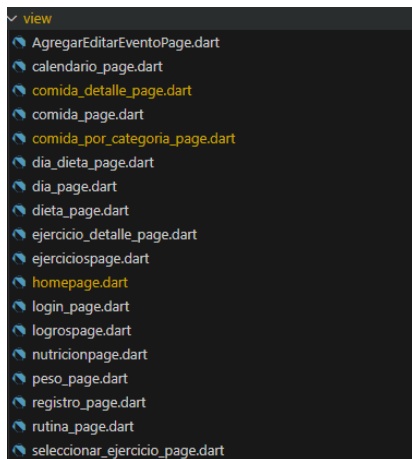


Ilustración 47 Vistas creación propia

#### 4.3.4. Widgets utilizados

El proyecto utiliza una variedad de widgets de Flutter para crear interfaces dinámicas e intuitivas. Algunos de los widgets más empleados son:

Para la estructura general:

- Scaffold, AppBar, Drawer, BottomNavigationBar

Para la interacción y navegación:

- ListView, ListTile, Card, GridView
- TextField, TextFormField, Form
- ElevatedButton, IconButton, FloatingActionButton

Para la gestión del estado:

- StatelessWidget para interfaces estáticas
- StatefulWidget para pantallas que requieren interacción con el usuario

Para funciones específicas:

- Widgets de calendario como TableCalendar o similares
- Widgets gráficos para representar información (por ejemplo, gráficos de peso o logros semanales)

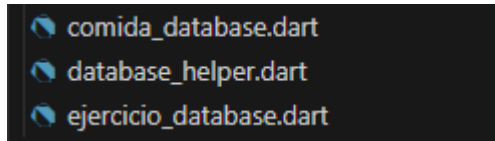
#### 4.3.5. Base de datos

La carpeta db incluye todas las clases necesarias para gestionar el almacenamiento local mediante SQLite. Se encarga de la creación, acceso, actualización y eliminación de datos.

comida\_database.dart, ejercicio\_database.dart: bases de datos específicas para comidas y ejercicios, respectivamente.

Y la database\_helper es para guardar otras cosas como peso.

Las bases de datos tienen diferentes versiones porque es importante guardar las diferentes versiones de la base de datos para realizar migraciones seguras cuando se cambia la estructura.



*Ilustración 48 Base de datos de la app Creación propia*

## 5. Conclusión

### 5.1. conclusión

Este proyecto ha sido una experiencia que me ha sido muy instructiva en muchos aspectos.

Se ha logrado cumplir el objetivo, se ha creado una aplicación móvil que ayude a las personas con discapacidad a seguir y poder realizar sus ejercicios físicos así mejorar su estado de salud y física.

En cuanto a la experiencia me ha servido para mejorar el autoaprendizaje y la resolución de problemas que son fundamentales para la vida laboral. Autoaprendizaje debido a que esta aplicación se usó el lenguaje. dart y Android en la que no se tenía conocimientos sobre ello. Y la parte de resolución de problemas es debido a que en el desarrollo de la aplicación ha aparecido muchos problemas y se ha enfrentado a ellos la mayoría con éxito.

### 5.2. Impacto social y medioambiental

En todo proyecto hay que medir el impacto que genera dicho proyecto. Con este proyecto se ayuda a personas que tienen discapacidad y desean mejorar sus vidas un sitio en gimnasia:

La ONU en 2015 estableció unos Objetivos de desarrollo sostenible que se deberían de cumplir para un futuro más sostenible.

# **OBJETIVOS DE DESARROLLO SOSTENIBLE**



## ODS 3: Salud y bienestar

La aplicación promueve el ejercicio físico accesible, lo que mejora la calidad de vida y la salud tanto física como mental de las personas con discapacidad.

## ODS 10: Reducción de las desigualdades

Esta aplicación hace frente a la exclusión en el ámbito del ejercicio, ofreciendo igualdad de oportunidades a usuarios que tenga discapacidad.

## ODS 11: Ciudades y comunidades sostenibles

Esta aplicación puede incluir rutinas que se pueden hacer en espacios públicos accesibles o en casa, así contribuyendo a crear entornos más inclusivos.

## 5.3. Proyectos futuros.

### **Expandir a otras plataformas o sistemas operativos.**

Actualmente la aplicación solo está desarrollada para dispositivos con sistema operativo Android. Para aumentar los usuarios que pueden acceder a ella.

### **Integración con la red.**

Actualmente esta aplicación funciona sólo localmente sería interesante integrar funcionalidades de red para que los usuarios puedan interactuar entre ellos. Creando así foros o canales, incluso compartir rutinas.

## 6. Bibliografía

- [1] «The effects of different types of physical exercise on physical and cognitive function in frail institutionalized older adults with mild to moderate cognitive impairment. A randomized controlled trial - ScienceDirect». Accedido: 14 de junio de 2025. [En línea]. Disponible en: [https://www.sciencedirect.com/science/article/abs/pii/S0167494319301116?casa\\_token=sX20L3VMY4gAAAAA:1EBw3JVV-EQ8fPOC5P41QeiBQiAyXZ7cNZE5Rh0bi7Z48GBgnxg6D5oddQrkogljuqQaGz1UuV8](https://www.sciencedirect.com/science/article/abs/pii/S0167494319301116?casa_token=sX20L3VMY4gAAAAA:1EBw3JVV-EQ8fPOC5P41QeiBQiAyXZ7cNZE5Rh0bi7Z48GBgnxg6D5oddQrkogljuqQaGz1UuV8)
- [2] «Effects of exercise on the physical fitness level of adults with intellectual disability: a systematic review: Disability and Rehabilitation: Vol 41, No 26». Accedido: 14 de junio de 2025. [En línea]. Disponible en: <https://www.tandfonline.com/doi/abs/10.1080/09638288.2018.1491646>
- [3] «The Influence of Functional Fitness and Cognitive Training of Physical Disabilities of Institutions - Yeh - 2015 - The Scientific World Journal - Wiley Online Library». Accedido: 14 de junio de 2025. [En línea]. Disponible en: <https://onlinelibrary.wiley.com/doi/full/10.1155/2015/686498>
- [4] «Effects of Combined Physical and Cognitive Exercises on Cognition and Mobility in Patients With Mild Cognitive Impairment: A Randomized Clinical Trial - ScienceDirect». Accedido: 14 de junio de 2025. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/abs/pii/S152586101730542X>
- [5] «SciELO Brazil - EFFECTS OF PHYSICAL EXERCISE FOR ADULTS WITH INTELLECTUAL DISABILITIES: A SYSTEMATIC REVIEW EFFECTS OF PHYSICAL EXERCISE FOR ADULTS WITH INTELLECTUAL DISABILITIES: A SYSTEMATIC REVIEW». Accedido: 14 de junio de 2025. [En línea]. Disponible en: <https://www.scielo.br/j/jpe/a/hv8wzTd4w7PFpw6wYVDRQsQ/?lang=en&format=html>
- [6] *Android SO*. Google.
- [7] *Draw.io*. JGraph Ltd, Artisans' House, 7 Queensbridge, NN4 7BF, Northampton, England. [En línea]. Disponible en: <https://www.drawio.com/>
- [8] *Android Studio*. Google. [En línea]. Disponible en: <https://developer.android.com/studio>
- [9] *Java*. Java. [En línea]. Disponible en: <https://www.java.com/en/>
- [10] *Dart*. .dart. Google. Accedido: 14 de junio de 2025. [En línea]. Disponible en: <https://dart.dev/docs>