



UNIVERSITAT DE  
BARCELONA

Trabajo final de grado

**GRADO DE INGENIERÍA INFORMÁTICA**

**Facultad de Matemáticas e Informática**

**Universidad de Barcelona**

---

# **GymStrat: Una aplicación de gimnasio para iPhone**

---

Mark Rivera Reoyo

Director/a: Polyxeni Gkontra

Realizado en: Departamento de Matemáticas e Ingeniería informática

Barcelona, 12 de enero de 2023

## **Agradecimientos**

Quiero agradecer a mi tutora del trabajo, por aceptar mi idea de proyecto y por hacerme un buen seguimiento durante el desarrollo, preocupándose de las horas a dedicar y de resolverme las dudas que han ido surgiendo.

Agradecer también a mi familia, especialmente a mi madre, por el apoyo y por el constante hincapié en mis estudios.

Y finalmente, agradecer también a mis amigos de la universidad por haber estado siempre ahí durante mi periodo universitario y por haberme ayudado siempre en todos mis problemas.

## Abstract

GymStrat is a mobile application developed for iPhone and oriented to people who go to the gym. The application helps you to organize your workouts, day by day, or make plans about your workouts, weekly.

GymStrat is adapted for people who already have a minimum of knowledge about and can organize by themselves their routines and exercises. But, in case of new users in the world of fitness, it implements a trained Bot with frequently asked questions for beginners, which will help in the process of creating routines and schedules.

In this way, getting a minimalist and intuitive application to the eye, we can leave behind the paper and pen to make a more sustainable planet.

## Resum

GymStrat és una aplicació mòbil desenvolupada per iPhone i orientada a persones que van al gimnàs. L'aplicació t'ajuda a organitzar els teus entrenaments, dia a dia, o fer plans sobre els teus entrenaments, setmanalment.

GymStrat està adaptat per a persones que ja tenen un mínim de coneixements i poden organitzar per si mateixes les seves rutines i exercicis. Però, en cas de nous usuaris en el món del fitness, implementa un Bot entrenat amb preguntes freqüents per a principiants, que ajudarà en el procés de creació de rutines i horaris.

D'aquesta manera, aconseguint una aplicació minimalista i intuïtiva a la vista, podem deixar enrere el paper i el bolígraf per fer un planeta més sostenible.

## Resumen

GymStrat es una aplicación móvil desarrollada para iPhone y orientada a las personas que van al gimnasio. La aplicación te ayuda a organizar tus entrenamientos, día a día, o hacer planes sobre tus entrenamientos, semanalmente.

GymStrat está adaptado para personas que ya tienen un mínimo de conocimientos y pueden organizar por sí mismos sus rutinas y ejercicios. Pero, en caso de nuevos usuarios en el mundo del fitness, implementa un Bot entrenado con preguntas frecuentes para principiantes, que ayudará en el proceso de creación de rutinas y horarios.

De esta forma, consiguiendo una aplicación minimalista e intuitiva a la vista, podemos dejar atrás el papel y el bolígrafo para hacer un planeta más sostenible.

## Contenido

1	Introducción .....	5
1.1	Contexto .....	5
1.2	Motivación .....	5
1.3	Objetivo principal .....	5
1.3.1	Problema principal .....	6
1.4	Objetivos específicos.....	6
1.5	Planificación temporal del trabajo .....	7
1.6	Estructura del documento.....	7
2	Conceptos previos.....	9
2.1	Rutina .....	9
2.2	Planificación .....	9
2.2.1	Planificación PIP .....	9
2.2.2	Planificación PIE .....	10
2.3	Ejercicio básico .....	11
3	Análisis de requisitos.....	13
3.1	Definición de usuarios.....	13
3.2	Requisitos de los usuarios .....	13
3.3	Requisitos funcionales.....	14
3.3.1	Casos de uso .....	14
3.3.2	Diagrama de casos de uso.....	19
3.4	Requisitos no funcionales .....	21
3.5	Restricciones .....	21
4	Diseño.....	22
4.1	Prototipo de la aplicación .....	22
4.2	Metodología de trabajo .....	24
5	Implementación y resultado .....	26
5.1	Tecnologías aplicadas.....	26
5.1.1	Git y GitHub .....	26
5.1.2	Visual Studio Code y React Native.....	26
5.1.3	Expo / Expo Go .....	26
5.1.4	Firebase .....	26
5.1.5	Google Cloud: Dialogflow .....	26
5.1.6	Postman.....	26
5.1.7	Trello.....	26
5.2	Detalles de la implementación.....	26

5.2.1	Estructura del código .....	26
5.2.2	Estructura de la base de datos .....	28
5.2.3	Obtención de datos de la API .....	31
5.3	Resultado.....	33
5.3.1	Pantallas .....	33
5.3.2	Flujo de la aplicación .....	44
6	Conclusiones.....	46
6.1	Conclusiones.....	46
6.1	Futuro trabajo .....	46
7	Referencias.....	48
Anexos	.....	51

# 1 Introducción

## 1.1 Contexto

Actualmente, la gran mayoría de personas que deciden llevar un registro de sus avances o rutina de ejercicios apuntado, opta por llevarlo escrito en un papel, cosa que depende de la persona, podría resultarle incómodo.

Cada vez que se realizaba un ejercicio, este/a apuntaba en la hoja anotaciones según le había ido el ejercicio, si había completado bien una serie o si era demasiado peso para levantar en ese día. Entonces, la pregunta principal que se formula es: ya que todo el mundo lleva siempre en su bolsillo un teléfono móvil, ¿por qué no, en vez de llevarse el papel y el bolígrafo para anotar sobre su progreso o acordarse sobre él, se descarga una aplicación para su dispositivo el cual le permita hacer lo mismo de forma muy dinámica y sencilla?

Durante el grado de ingeniería informática, se han impartido diferentes asignaturas que están brevemente relacionadas con el tema que estoy tratando. Sobre todo, en las asignaturas de software donde se ha tenido que desarrollar una web/aplicación tratando usuarios, gestionar bases de datos o desarrollar tanto *frontend* como *backend*.

## 1.2 Motivación

Desde siempre he querido crear aplicación para móvil relacionada con lo que me gusta hacer hoy en día, e ir al gimnasio es una de las grandes aficiones que tengo actualmente.

Durante mis años en el gimnasio, me he interesado bastante por el *powerlifting*, y uno de los puntos esenciales de este tipo de deporte era hacerte una buena planificación del ejercicio a realizar, calculando el peso a levantar cada día del plan, incluidas las repeticiones y series, para así tener una progresión eficiente. En su momento le vi mucho potencial al cálculo de la planificación, ya que necesitas calcular porcentajes, días y organización del ejercicio, por lo tanto, decidí que hacer una aplicación con este tipo de contenido podría ser muy provechoso si conseguía automatizarlo.

También durante este año “2022 – 2023”, se ha hecho muy famoso un chat programado con inteligencia artificial llamado ChatGPT [1], que interactúa de manera conversacional. Investigando sobre ChatGPT decidí entonces que añadir una versión de chat por inteligencia artificial a una aplicación fitness, pero de una versión muy reducida, la cual solo podría responder preguntas que yo habría programado y entrenado, podría instruir a usuarios novatos tanto a saber más acerca de las planificaciones como a planificar rutinas.

## 1.3 Objetivo principal

El objetivo principal de este proyecto es entonces crear una aplicación móvil para iPhone que permita a los usuarios poder evadir el papel y el bolígrafo para tomar anotaciones de sus entrenamientos de una manera más sostenible.

Más detalladamente, crear una aplicación móvil dónde puedas anotar tus ejercicios formando rutinas para cada día de la semana y donde puedas también crear planificaciones para los ejercicios básicos del gimnasio. La característica principal será que podrás anotar tus avances por escrito.

A más, incorporará un pequeño *ChatBot* con el cual podrás interactuar para que te cree rutinas y puedas preguntarle casi cualquier duda respecto a términos del gimnasio, resolviendo así las dudas de un usuario que acaba de empezar en el mundo del fitness.

### 1.3.1 Problema principal

Antes de empezar con la implementación de la aplicación, he realizado una búsqueda en profundidad de aplicaciones similares existentes. Aunque he encontrado varias en la *App Store*, ninguna ofrecía lo que de verdad se quería implementar.

Entre las aplicaciones similares más relevantes, hay que destacar:

- Strong Workout Tracker Gym Log: Aplicación la cual permite al usuario organizarse las rutinas de la semana, añadiendo ejercicios con sus respectivas series y repeticiones. Además, ofrece algún gráfico en función de tu avance, un apartado donde puedes apuntar tus récords, un temporizador de descanso e indicación de cómo realizar el ejercicio. [2]
- Hevy – Gym Log Workout Tracker: Aplicación con la que puedes organizar tus ejercicios semanalmente, analizar tus entrenamientos, hacer un seguimiento de tus medidas corporales y permite comparar tu fuerza con la comunidad. Además de ofrecer una biblioteca de ejercicios. [3]
- JEFIT Workout Planner Gym Log: Aplicación con la que puedes organizar tus entrenamientos y sobre todo alimentarte de información respecto a la comunidad del gimnasio, ejercicios que realizar, etc. [4]

En conclusión, ninguna de estas aplicaciones hace cálculos de planificaciones para los ejercicios, no puedes añadir notas específicas para cada ejercicio realizado ni ofrecen un sistema de ayuda para la aplicación, como puede ser un *ChatBot*, el cual te haga rutinas y te resuelva dudas importantes.

### 1.4 Objetivos específicos

- Crear una aplicación móvil con React Native [5], así definiendo la arquitectura de la aplicación y las tecnologías a utilizar.
- Diseñar e implementar un método de inicio de sesión y registro de usuarios, así como de modificación y eliminación de este.
- Crear las ventanas donde el usuario necesitará interactuar y ver sus ejercicios.
- Desarrollar un sistema de creación de ejercicios intuitivo y sencillo para el usuario, así como de planificaciones.
- Diseñar una base de datos donde administrar todos los usuarios y guardar los datos creados desde la aplicación.
- Adquirir conocimientos sobre los *ChatBots*, diseñar e implementar Bot.
- Crear nuevos apartados en la aplicación con nuevas funcionalidades adicionales.
- Documentación y memoria.

## 1.5 Planificación temporal del trabajo

	Periodo 1	Periodo 2	Periodo 3	Periodo 4	Periodo 5	Periodo 6
Tarea 1						
Tarea 2						
Tarea 3						
Tarea 4						
Tarea 5						
Tarea 6						
Tarea 7						
Tarea 8						
Tarea 9						
Tarea 10						

Tabla 1: Diagrama GANTT de la organización del trabajo

### Tareas

- ❖ Tarea 1: Pensar en el tema de mi proyecto y definir la arquitectura de la aplicación.
- ❖ Tarea 2: Creación del proyecto con su lenguaje y dependencias iniciales.
- ❖ Tarea 3: Desarrollo del planificador de ejercicios.
- ❖ Tarea 4: Desarrollo de la página principal y visualizador de planificaciones.
- ❖ Tarea 5: Implementación de una base de datos y desarrollo del perfil de usuario.
- ❖ Tarea 6: Inicio de sesión/registro de un usuario y persistencia de inicio de sesión.
- ❖ Tarea 7: Documentarme sobre DialogFlow [6] y desarrollar el *ChatBot*.
- ❖ Tarea 8: Añadir funcionalidades extras como las notas o modificar ejercicios.
- ❖ Tarea 9: Test con usuarios.
- ❖ Tarea 10: Redactar la memoria documentando todo acerca del desarrollo.

### Franjas de tiempo

- ❖ Periodo 1: del 20 de octubre de 2022 al 3 de noviembre de 2022.
- ❖ Periodo 2: del 4 de noviembre de 2022 al 27 de noviembre de 2022.
- ❖ Periodo 3: del 28 de noviembre de 2022 al 13 de diciembre de 2022.
- ❖ Periodo 4: del 14 de diciembre de 2022 al 31 de diciembre de 2022.
- ❖ Periodo 5: del 1 de enero de 2023 al 6 de enero de 2023.
- ❖ Periodo 6: del 6 de enero 2023 al 24 de enero de 2023.

## 1.6 Estructura del documento

Breve descripción del contenido de cada uno de los capítulos de la memoria.

- **Capítulo 1, Introducción:** En este capítulo se explica cuál es el problema que solucionar y como se pretende solucionar, mostrando también la planificación del



trabajo que se ha seguido y una orientación al lector sobre la estructura del documento.

- **Capítulo 2, Conceptos previos:** En este capítulo se dejan claros algunos conceptos del gimnasio básicos utilizados a lo largo del presente trabajo.
- **Capítulo 3, Análisis de requisitos:** En este capítulo se realiza un exhausto análisis de requisitos para la aplicación, detallando las necesidades y expectativas de los usuarios, así como los objetivos y funcionalidades de la aplicación.
- **Capítulo 4, Diseño:** En este capítulo se muestra una primera versión de la aplicación, el prototipo, y se detalla la metodología de trabajo seguida.
- **Capítulo 5, Implementación y resultado:** En este capítulo se muestran las tecnologías que han sido aplicadas para el desarrollo de la aplicación, se documenta como ha estado implementada la aplicación y se muestra el resultado final de las pantallas que han sido diseñadas.
- **Capítulo 6, Conclusiones:** En este capítulo se explican las conclusiones del trabajo y se hace una estimación del trabajo futuro en función de los resultados del proyecto.

## 2 Conceptos previos

### 2.1 Rutina

Una rutina es un conjunto de ejercicios organizados que se realizan en un mismo día de la semana. Como por ejemplo de lunes hasta domingo.

Existen diferentes tipos de organización de rutina:

- *PPL (Push – Pull – Legs)*: Se dividen los entrenamientos semanales según entrenamientos de empuje, entrenamientos de tirar y entrenamientos de pierna.
- *Upper – Lower*: Se dividen los entrenamientos semanales según entrenamientos de torso y entrenamientos de piernas o parte baja.
- *Fullbody*: Se dividen los entrenamientos semanales según entrenamientos en los que ejerces todos los músculos en cada día.
- *Muscle per day*: Se dividen los entrenamientos semanales según entrenamientos en los que solo se ejercita un tipo de músculo por día.

### 2.2 Planificación

Una planificación de un ejercicio se basa en realizar ese ejercicio durante un periodo de tiempo, modificando o no, una progresión o disminución en las series, pesos o repeticiones.

- Repeticiones: Número de veces que se realiza el ejercicio.
- Series: Es un conjunto de repeticiones separadas por descansos.

Ejemplo (mes de enero de 2023):

Día	Serie	Reps.	Peso	Serie	Reps.	Peso	Serie	Reps.	Peso
1	1	10	20 kg	2	8	30 kg	3	6	40 kg
10	1	6	40 kg	2	8	30 kg	3	10	20 kg
15	1	10	20 kg	2	10	20 kg	3	10	20 kg
22	1	10	20 kg	2	6	40 kg	3	8	30 kg

Tabla 2: Tabla de ejemplo de una planificación (Reps. -> Repeticiones)

Las planificaciones por defecto que he decidido añadir a la aplicación son: Planificación PIP o programación progresiva de la intensidad con una reducción progresiva de la repetición por serie, y la Planificación PIE o programación de un aumento progresivo de la intensidad absoluta con volumen e intensidad relativa estables.

#### 2.2.1 Planificación PIP

También conocida por Programación progresiva de la intensidad con una reducción progresiva de la repetición por serie [7]. Este tipo de planificación es la más clásica dónde el volumen va a ir disminuyendo en función de las semanas y la intensidad aumenta, suele tener una duración aproximada de 1 mes en total.

En cuanto al volumen nos referimos a la cantidad de repeticiones a realizar por serie, en cambio, la intensidad significa el porcentaje de la carga final que vamos a alcanzar.

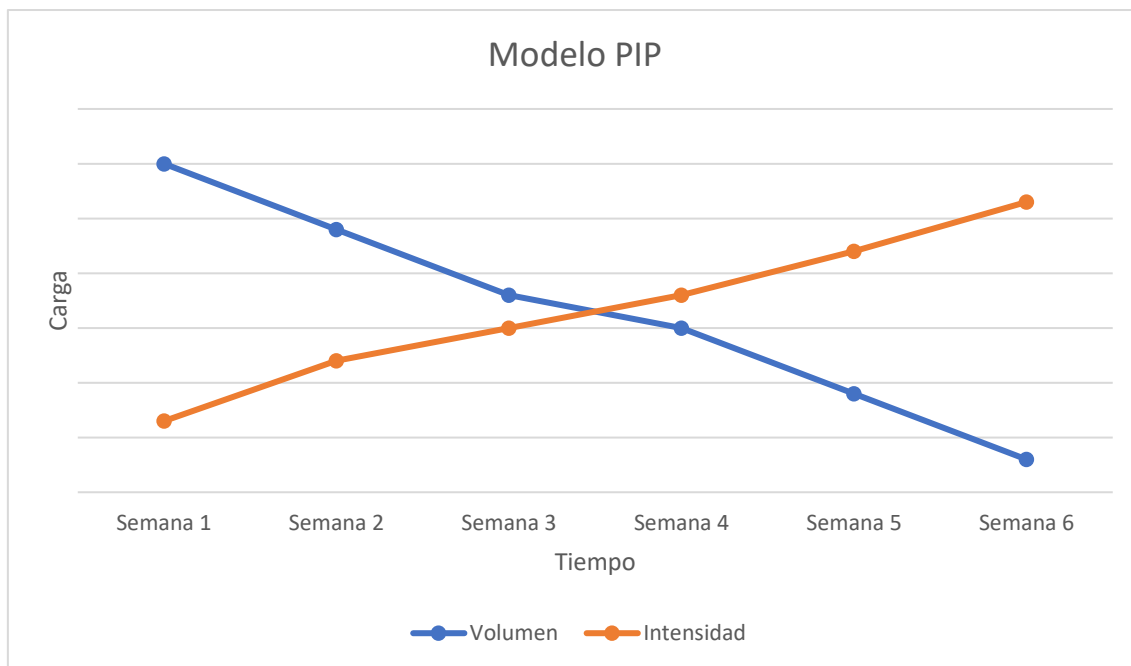


Figura 1: Gráfico del modelo PIP

Ejemplo práctico: Sabemos que nuestro récord personal (PR) de X ejercicio, gira en torno a los 95 kilogramos. Para calcular nuestra planificación PIP vamos a aumentar la carga en 5kg o 2.5kg, estableciendo así el peso final a levantar (100kg o 97.5kg).

Reglas de la planificación:

- Semana 1: Volumen, 3 series \* 10 repeticiones | Intensidad, 65% del peso final.
- Semana 2: Volumen, 3 series \* 8 repeticiones | Intensidad, 70% del peso final.
- Semana 3: Volumen, 3 series \* 6 repeticiones | Intensidad, 75% del peso final.
- Semana 4: Volumen, 3 series \* 5 repeticiones | Intensidad, 80% del peso final.
- Semana 5: Volumen, 3 series \* 3 repeticiones | Intensidad, 85% del peso final.
- Semana 6: Volumen, 3 series \* 1 repeticiones | Intensidad, 90% del peso final.

Finalmente, la Semana 7, realizaríamos la prueba de 1 serie \* 1 repetición con el peso final para comprobar la eficiencia del método.

### 2.2.2 Planificación PIE

También conocida como Programación de un aumento progresivo de la intensidad absoluta con volumen e intensidad relativa estables [8]. En este tipo de planificación no hay ni aumento ni disminuyo del volumen. El único valor que va aumentando durante el curso de las semanas es el de la intensidad. Para esta planificación se recomiendan periodos cortos para una mayor efectividad.

En este caso no vamos a calcular el peso final a levantar, sino que vamos a calcular un peso con el que podamos hacer 5 repeticiones.

Reglas de la planificación:

- Semana 1: Volumen, 3 series \* 5 repeticiones | Intensidad, 93% del peso final.

- Semana 2: Volumen, 3 series \* 5 repeticiones | Intensidad, 98% del peso final.
- Semana 3: Volumen, 3 series \* 5 repeticiones | Intensidad, 98% del peso final.
- Semana 4: Volumen, 3 series \* 5 repeticiones | Intensidad, 100% del peso final.

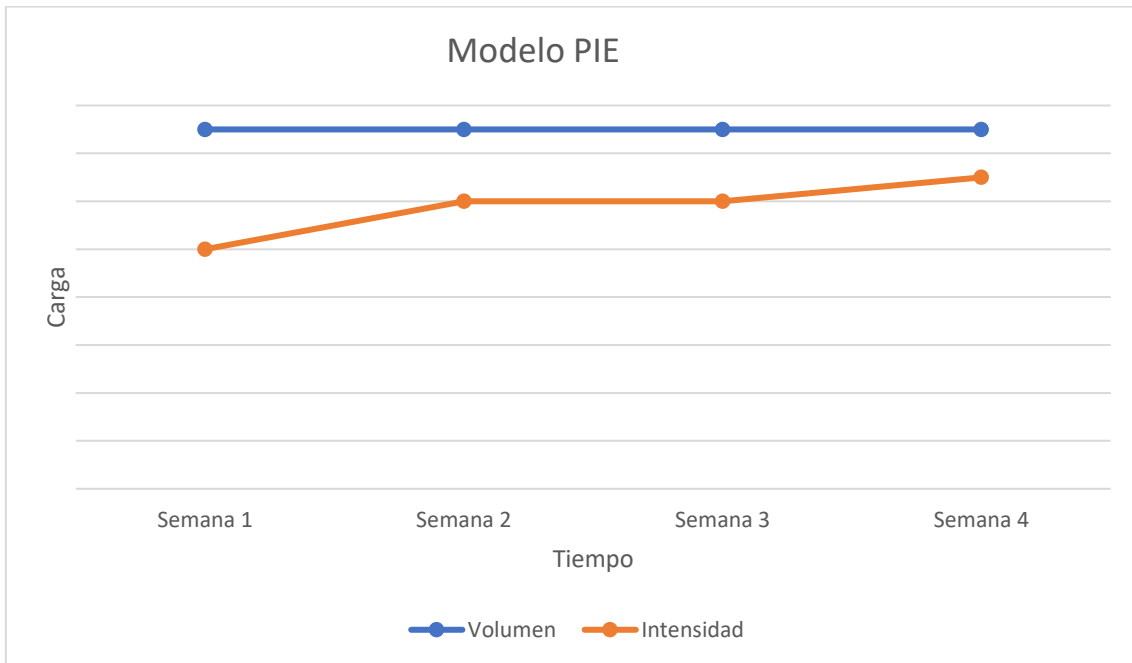


Figura 2: Gráfico del modelo PIE

### 2.3 Ejercicio básico

Los ejercicios básicos del gimnasio se refieren a aquellos ejercicios que implican el movimiento simultáneo de varias articulaciones del cuerpo, implicando así una variedad de músculos. A estos ejercicios también se les llama multiarticulares. [9]

Ejercicios básicos más comunes:



Figura 3: Press de Banca [10]

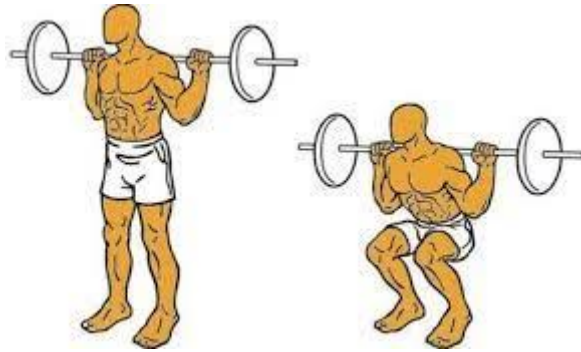


Figura 4: Sentadilla [11]



Figura 5: Peso Muerto [12]

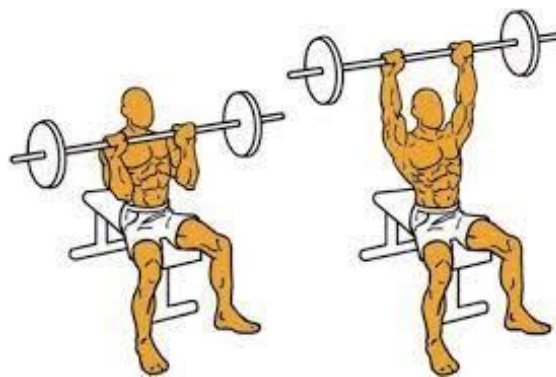


Figura 6: Press Militar [13]

Como podemos observar, realizar estos 4 ejercicios suponen ejercitar todo nuestro cuerpo en conjunto, tanto el tren superior como el tren inferior, por lo que se consideran los ejercicios base sobre los cuales crear una rutina de entrenamiento.

También podemos encontrar variaciones de estos ejercicios. En todo caso, el concepto del ejercicio es el mismo, pero con alguna modificación, como por ejemplo el Press de banca con parada, o el Peso muerto sumo, entre otros.

Un ejercicio no-básico serían todos los alternativos a los mencionados.

## 3 Análisis de requisitos

### 3.1 Definición de usuarios

La aplicación móvil principalmente va a estar orientada para usuarios que se dirigen a un gimnasio expresa y únicamente para realizar actividades físicas. Por lo tanto, el usuario principal y único de la aplicación va a ser el cliente del gimnasio, es decir, el usuario que va a realizar los entrenamientos.

Los usuarios, tanto veteranos como novatos, van a calificarse por igual, ya que el uso de la aplicación no va a variar y realizarán los dos las mismas acciones disponibles.

### 3.2 Requisitos de los usuarios

Se identificarán las necesidades y expectativas de los usuarios, incluyendo sus objetivos y tareas, características deseadas y restricciones.

En una primera iteración, basándome en el principal problema a resolver y teniendo en cuenta que yo mismo era un potencial usuario de la aplicación, he sacado esta primera lista de requisitos:

- La aplicación deberá tener un registro de usuarios para poder obtener tus avances en cualquier dispositivo iPhone, así cumpliendo con un sistema de inicio de sesión y registro.
- Una vez iniciada la sesión, tendrá un apartado de perfil dónde se podrán modificar valores del usuario como por ejemplo el correo, la contraseña y el nombre del usuario.
- En el perfil también se podrán anotar récords personales y cerrar sesión.
- La aplicación tendrá una ventana dónde se podrán crear ejercicios, anotando el número de repeticiones, series, etc. Y se podrán crear planificaciones para los ejercicios, así obteniendo una completa guía de su realización durante las semanas.
- Los ejercicios creados se visualizarán de forma sencilla e intuitiva en la ventana principal de la aplicación, dónde estarán separados en días de la semana.
- Para no entrar en dudas, conjuntamente a la ventana principal de la aplicación, habrá una ventana similar, dónde, en vez de visualizar los ejercicios, se visualizarán las planificaciones de los ejercicios.
- Cada ejercicio tendrá varias opciones, entre ellas se encontrará: eliminar el ejercicio, renombrar el ejercicio, cambiar el orden del ejercicio, añadir una serie y cambiar el valor de peso o repeticiones de una serie del ejercicio.
- La aplicación tendrá un *ChatBot* con el cual un usuario novato podrá hacerle preguntas, crear rutinas y orientar al usuario.

De los requisitos que no va a contener la aplicación, he creído que no son el punto fuerte para desarrollar y he decidido descartarlos.

- No dispondrá de un sistema de verificación del correo electrónico.

- No dispondrá de un sistema de notificaciones.
- La aplicación no dispondrá de una sección para la alimentación inicialmente, ya que la idea está en hacer un seguimiento de ejercicios únicamente.

También quiero comentar que, en caso de cumplir con todos los requisitos con éxito una vez finalizada la primera versión de la aplicación, se procedería a realizar una segunda iteración de obtención de nuevos requisitos, esta vez con usuarios alternativos a mí, e implementarlos para una segunda versión.

### 3.3 Requisitos funcionales

Se detallarán las funcionalidades y características que la aplicación debe tener para satisfacer las necesidades de los usuarios mediante casos de uso.

He definido unos 13 casos de uso que definen las funcionalidades principales y más relevantes de la aplicación.

#### 3.3.1 Casos de uso

Nombre del Caso de Uso	UC1 – Registro.
Descripción	El usuario se da de alta en la aplicación.
Actor	Usuario, Usuario no registrado.
Precondición	La aplicación esta iniciada por primera vez, no hay ninguna sesión abierta.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario presiona el botón “Registrarte”.</li> <li>2. Una vez en la ventana del registro, el usuario rellena los campos del formulario.</li> <li>3. El usuario presiona el botón “Registrarse”.</li> <li>4. El sistema comprueba que los campos son correctos.</li> <li>5. El sistema redirige al usuario a la ventana principal de la aplicación concluyendo con el registro.</li> </ol>
Flujo alternativo	<p>4a. El sistema comunica el error al usuario.</p> <p>4b. El usuario modifica los campos erróneos. (<i>vuelve al paso 3</i>)</p>
Postcondición	El usuario es registrado en la base de datos y redirigido a la ventana principal de la aplicación.

Nombre del Caso de Uso	UC2 – Iniciar sesión.
Descripción	El usuario entra a la aplicación.
Actor	Usuario.
Precondición	La aplicación esta iniciada, no hay ninguna sesión abierta y el usuario debe de estar registrado.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario rellena los campos del formulario.</li> <li>2. El usuario presiona el botón “Iniciar sesión”.</li> <li>3. El sistema comprueba que los campos son correctos.</li> <li>4. El sistema redirige al usuario a la ventana principal de la aplicación.</li> </ol>
Flujo alternativo	<p>3a. El sistema comunica el error al usuario.</p> <p>3b. El usuario modifica los campos erróneos. (<i>vuelve al paso 2</i>)</p>
Postcondición	El usuario es redirigido a la ventana principal de la aplicación.

Nombre del Caso de Uso	UC3 – Visualizar perfil
Descripción	El usuario visualiza su perfil para comprobar sus datos.
Actor	Usuario
Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario presiona el botón con silueta humana que se encuentra en la barra de navegación en la parte inferior de la aplicación.</li> <li>2. El sistema muestra la ventana de “Perfil” del usuario.</li> </ol>
Flujo alternativo	No hay.
Postcondición	Se visualiza el perfil.

Nombre del Caso de Uso	UC4 – Cerrar sesión.
Descripción	El usuario cierra la sesión iniciada.
Actor	Usuario.
Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta y el usuario debe de encontrarse en el apartado de “Perfil”.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario hace clic en el botón de cerrar sesión.</li> <li>2. El sistema muestra un mensaje de confirmación del cierre de sesión.</li> <li>3. El usuario presiona el botón “Salir”.</li> <li>4. El sistema gestiona el cierre de sesión del usuario y lo redirige a la ventana de inicio de sesión.</li> </ol>
Flujo alternativo	<ol style="list-style-type: none"> <li>3a. El usuario presiona el botón “Cancelar”</li> <li>3b. El sistema cierra la alerta y no cierra la sesión.</li> </ol>
Postcondición	El sistema se encarga de cerrar la sesión del usuario y lo redirige a la ventana de inicio de sesión.

Nombre del Caso de Uso	UC5 – Eliminar cuenta de la aplicación.
Descripción	El usuario elimina su cuenta de la aplicación.
Actor	Usuario.
Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta y el usuario debe de encontrarse en el apartado donde se modifican los datos del “Perfil”.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario presiona el botón “Elimina tu cuenta”.</li> <li>2. El sistema muestra un mensaje de confirmación.</li> <li>3. El usuario presiona el botón “Eliminar”.</li> <li>4. El sistema despliega un formulario para eliminar permanentemente la cuenta.</li> <li>5. El usuario rellena los campos del formulario.</li> <li>6. El usuario presiona el botón “Confirmar”.</li> <li>7. El sistema comprueba que los campos son correctos.</li> <li>8. El sistema muestra un mensaje de confirmación.</li> <li>9. El usuario presiona el botón “Continuar”.</li> <li>10. El sistema elimina permanentemente la cuenta.</li> </ol>
Flujo alternativo	<ol style="list-style-type: none"> <li>3a. El usuario presiona el botón “Cancelar”.</li> <li>3b. El sistema cierra la alerta.</li> <li>6a. El usuario presiona “Cancelar”.</li> </ol>



	6b. El sistema cierra el desplegable. 7a. El sistema comunica el error al usuario. 7b. El usuario modifica los campos erróneos. ( <i>vuelve al paso 6</i> ) 9a. El usuario presiona el botón "Cancelar". 9b. El sistema cierra la alerta.
Postcondición	La cuenta del usuario se elimina permanentemente de la base de datos.

Nombre del Caso de Uso	UC6 – Añadir una marca personal.
Descripción	El usuario añade una marca de levantamiento en su perfil.
Actor	Usuario.
Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta y el usuario debe de encontrarse en el apartado de "Perfil".
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario presiona el botón de "+" que se encuentra al lado del texto: TUS MARCAS PERSONALES ( PR ).</li> <li>2. El sistema muestra un mensaje de confirmación para añadir una nueva marca.</li> <li>3. El usuario presiona el botón "Añadir".</li> <li>4. El sistema despliega un formulario para añadir la marca personal.</li> <li>5. El usuario rellena los campos del formulario.</li> <li>6. El usuario presiona el botón "Confirmar".</li> <li>7. El sistema comprueba que los campos son correctos.</li> <li>8. El sistema muestra un mensaje de confirmación.</li> <li>9. El usuario presiona "Continuar".</li> <li>10. El sistema añade la nueva marca personal en el perfil.</li> </ol>
Flujo alternativo	3a. El usuario presiona el botón "Cancelar" 3b. El sistema cierra la alerta. 6a. El usuario presiona "Cancelar". 6b. El sistema cierra el desplegable. 7a. El sistema comunica el error al usuario. 7b. El usuario modifica los campos erróneos. 9a. El usuario presiona "Cancelar". 9b. El sistema cierra la alerta.
Postcondición	Se añade la marca personal al perfil.

Nombre del Caso de Uso	UC7 – Añadir un ejercicio NO-BÁSICO.
Descripción	El usuario crea un ejercicio.
Actor	Usuario.
Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta y el usuario debe de encontrarse en el apartado de creación de ejercicios "NO-BÁSICO".
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario rellena los campos del formulario.</li> <li>2. El usuario presiona el botón "Confirmar".</li> <li>3. El sistema comprueba que los campos son correctos.</li> <li>4. El sistema muestra un desplegable para seleccionar el día de la realización del ejercicio.</li> <li>5. El usuario selecciona el día el cual realizarlo.</li> <li>6. El usuario presiona el botón "Confirmar".</li> </ol>

	<p>7. El sistema muestra un mensaje de confirmación.</p> <p>8. El usuario presiona el botón "Continuar".</p> <p>9. El sistema añade el ejercicio a la base de datos.</p>
Flujo alternativo	<p>3a. El sistema comunica el error al usuario.</p> <p>3b. El usuario modifica los campos erróneos.</p> <p>6a. El usuario presiona el botón "Cancelar".</p> <p>6b. El sistema cierra el desplegable.</p> <p>8a. El usuario presiona el botón "Cancelar".</p> <p>8b. El sistema cierra la alerta.</p>
Postcondición	El ejercicio es añadido a la base de datos.

Nombre del Caso de Uso	UC8 – Añadir un ejercicio BASICO con planificación propia.
Descripción	El usuario añade una planificación de un ejercicio.
Actor	Usuario.
Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta y el usuario debe de encontrarse en el apartado de creación de ejercicios "BÁSICO", haber seleccionado el modo de ejercicio con planificación y de planificación propia.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario rellena los campos del formulario.</li> <li>2. El usuario presiona el botón "CREA TU PROPIO PLAN".</li> <li>3. El sistema redirige al usuario a una ventana con un formulario para crear tu planificación.</li> <li>4. El usuario rellena los campos del formulario.</li> <li>5. El usuario presiona el botón "Confirmar".</li> <li>6. El sistema comprueba que los campos son correctos.</li> <li>7. El sistema muestra un mensaje de confirmación.</li> <li>8. El usuario presiona el botón "Continuar".</li> <li>9. El sistema redirige al usuario a la ventana anterior y previsualiza la planificación.</li> <li>10. El usuario presiona el botón "Confirmar".</li> <li>11. El sistema comprueba que los campos son correctos.</li> <li>12. El sistema muestra un mensaje de confirmación.</li> <li>13. El usuario presiona el botón "Confirmar".</li> <li>14. El sistema añade la planificación a la base de datos.</li> </ol>
Flujo alternativo	<p>4a. El usuario presiona el botón "&lt;".</p> <p>4b. El sistema redirige al usuario a la ventana anterior.</p> <p>6a. El sistema comunica el error al usuario.</p> <p>6b. El usuario modifica los campos erróneos.</p> <p>8a. El usuario presiona el botón "Cancelar".</p> <p>8b. El sistema cierra la alerta.</p> <p>11a. El sistema comunica el error al usuario.</p> <p>11b. El usuario modifica los campos erróneos. (<i>vuelve al paso 10</i>)</p> <p>13a. El usuario presiona el botón "Cancelar".</p> <p>13b. El sistema cierra la alerta.</p>
Postcondición	La planificación es añadida a la base de datos

Nombre del Caso de Uso	UC9 – Eliminar ejercicio.
Descripción	El usuario elimina un ejercicio de la rutina.
Actor	Usuario.

Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta y el usuario debe de encontrarse en la ventana principal de la aplicación.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario presiona el botón con forma de lápiz.</li> <li>2. El sistema muestra un mensaje de confirmación.</li> <li>3. El usuario presiona el botón "Eliminar ejercicio".</li> <li>4. El sistema muestra un mensaje de confirmación.</li> <li>5. El usuario presiona el botón "Eliminar".</li> <li>6. El sistema elimina el ejercicio de la base de datos.</li> </ol>
Flujo alternativo	<ol style="list-style-type: none"> <li>3a. El usuario presiona el botón "Cancelar".</li> <li>3b. El sistema cierra la alerta.</li> <li>5a. El usuario presiona el botón "Cancelar".</li> <li>5b. El sistema cierra la alerta.</li> </ol>
Postcondición	El ejercicio se elimina de la base de datos.

Nombre del Caso de Uso	UC10 – Visualizar planificaciones.
Descripción	El usuario visualiza una planificación.
Actor	Usuario.
Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta, debe de haber como mínimo una planificación creada y el usuario debe de encontrarse el apartado de "Planificaciones".
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario presiona encima del botón azul de la planificación.</li> <li>2. El sistema muestra un desplegable en el cual incluye toda la información de la planificación.</li> </ol>
Flujo alternativo	No hay.
Postcondición	Se muestra la planificación.

Nombre del Caso de Uso	UC11 – Añadir nota.
Descripción	El usuario añade una nota a un ejercicio.
Actor	Usuario.
Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta, debe de haber como mínimo un ejercicio por visualizar y el usuario debe de encontrarse en la ventana principal de la aplicación.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario presiona el botón con forma de papel y bolígrafo.</li> <li>2. El sistema redirige al usuario hacia la ventana de "Notas".</li> <li>3. El usuario rellena los campos del formulario para añadir una nota.</li> <li>4. El usuario presiona el botón "Añadir nota".</li> <li>5. El sistema muestra un mensaje de confirmación.</li> <li>6. El usuario presiona el botón "Añadir".</li> <li>7. El sistema añade la nota a la base de datos.</li> </ol>
Flujo alternativo	<ol style="list-style-type: none"> <li>6a. El usuario presiona el botón "Cancelar".</li> <li>6b. El sistema cierra la alerta.</li> </ol>
Postcondición	Se añade la nota a la base de datos

Nombre del Caso de Uso	UC12 – Visualizar notas.
Descripción	El usuario visualiza las notas de un ejercicio.
Actor	Usuario.
Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta, debe de haber como mínimo un ejercicio por visualizar, debe de haber una nota creada como mínimo y el usuario debe de encontrarse en la ventana principal de la aplicación.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario presiona el botón con forma de papel y bolígrafo.</li> <li>2. El sistema redirige al usuario hacia la ventana de “Notas”.</li> <li>3. El usuario presiona encima de la nota con fecha: X.</li> <li>4. El sistema muestra un desplegable con la nota escrita.</li> </ol>
Flujo alternativo	<ol style="list-style-type: none"> <li>3a. El usuario presiona en el botón “Página principal”.</li> <li>3b. El sistema redirige al usuario a la ventana principal de la aplicación.</li> </ol>
Postcondición	La nota es mostrada al usuario.

Nombre del Caso de Uso	UC13 – Hablar con el Bot.
Descripción	El usuario quiere hablar con el Bot para hacerle preguntas.
Actor	Usuario.
Precondición	La aplicación esta iniciada, la sesión del usuario debe de estar abierta y el usuario debe de encontrarse en el apartado del Bot
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario introduce una pregunta al sistema.</li> <li>2. El sistema responde satisfactoriamente a su pregunta.</li> </ol>
Flujo alternativo	2a. El sistema no entiende la respuesta y responde con un mensaje de <i>fallback</i> .
Postcondición	El sistema responde a la pregunta del usuario.

### 3.3.2 Diagrama de casos de uso

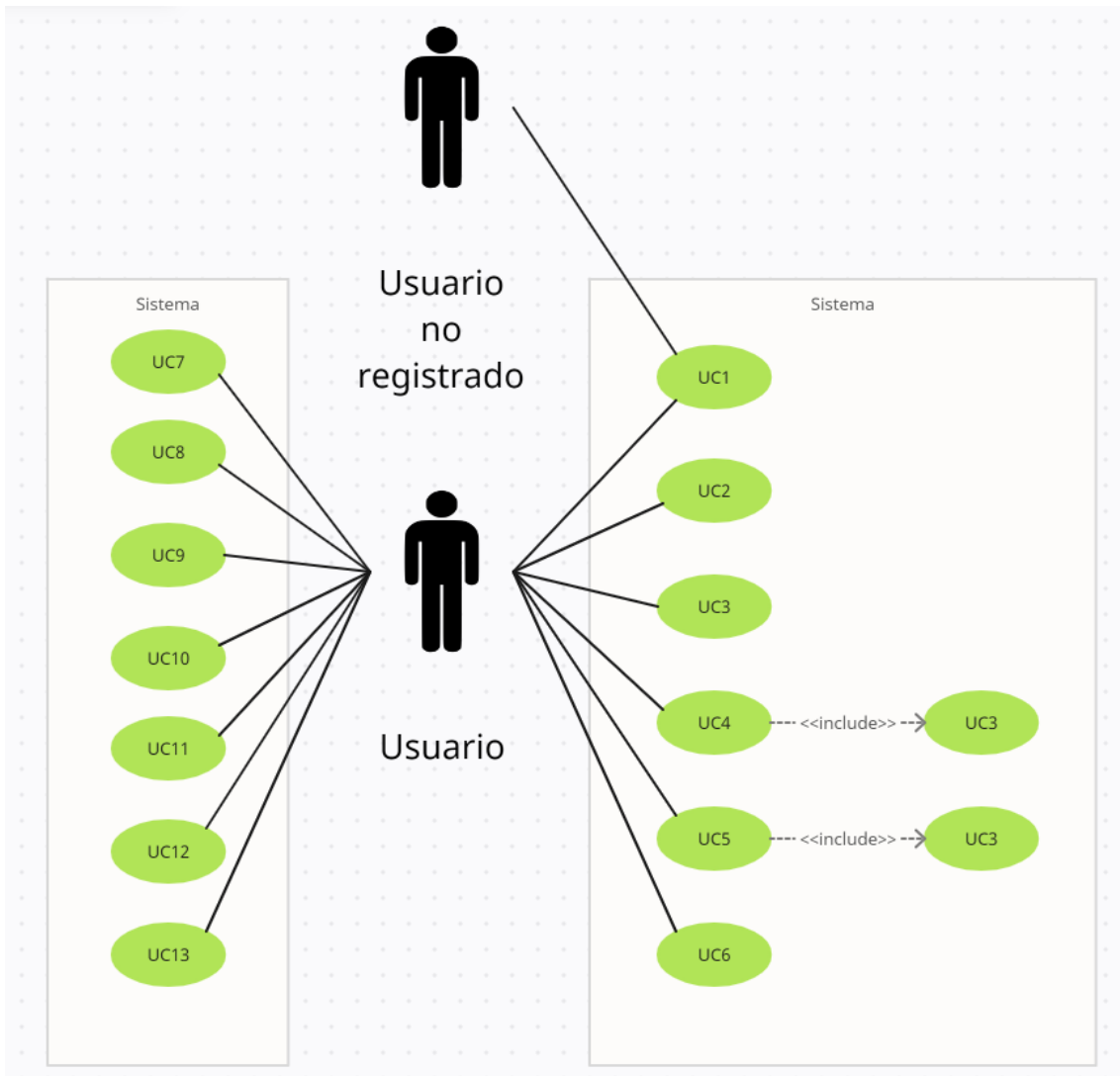


Figura 7: Diagrama de casos de uso

Leyenda del diagrama:

- ❖ UC1: Registro.
- ❖ UC2: Iniciar sesión.
- ❖ UC3: Visualizar perfil.
- ❖ UC4: Cerrar sesión.
- ❖ UC5: Eliminar cuenta de la aplicación.
- ❖ UC6: Añadir una marca personal.
- ❖ UC7: Añadir un ejercicio NO-BÁSICO.
- ❖ UC8: Añadir un ejercicio BASICO con planificación propia
- ❖ UC9: Eliminar ejercicio.
- ❖ UC10: Visualizar planificaciones.
- ❖ UC11: Añadir nota.
- ❖ UC12: Visualizar notas.

❖ UC13: Hablar con el Bot.

### 3.4 Requisitos no funcionales

Se describirán los aspectos de la aplicación que no son funcionales, pero son importantes para su correcto funcionamiento, como el rendimiento, la escalabilidad y la seguridad.

- **Seguridad:** El sistema estará protegido mediante el inicio de sesión por autenticación.
- **Disponibilidad:** El sistema va a estar disponible las 24h para cualquier usuario.
- **Actuación:** El sistema podrá manejar cualquier tipo de usuario sin degradación, ya que la aplicación no contendrá registros globales que cargar para cada uno.
- **Escalabilidad:** La idea es que la aplicación desarrollada sea una versión inicial, la cual se ira actualizando según las peticiones de los usuarios.
- **Usabilidad:** El sistema será intuitivo y fácil de utilizar.
- **Privacidad:** El sistema solo requerirá de una contraseña, la cual va a estar encriptada y va a ser imposible de descifrar por los desarrolladores y por cualquier usuario.
- **Fiabilidad:** El sistema no pedirá ningún tipo de dato alternativo que no sea el correo electrónico o la contraseña para iniciar sesión.

### 3.5 Restricciones

Se identificarán las restricciones bajo las cuales se desarrollará la aplicación, como plataformas soportadas, requisitos de hardware y software, y limitaciones legales y de cumplimiento.

La aplicación por desarrollar estará solamente dirigida para usuarios iPhone, cualquier iPhone será capaz de contener y visualizar la aplicación, ya que no va a demandar de grandes recursos, por no decir nulos.

## 4 Diseño

### 4.1 Prototipo de la aplicación

Inicialmente, se hizo un pequeño prototipo de las pantallas principales de la aplicación para poder tenerlas de referencia a la hora de codificarlas.

Desde un principio la aplicación iba a tener solo 3 pantallas, pero como más adelante se consideró una funcionalidad importante poder añadir usuarios para poder llevarse su progreso a cualquier iPhone, introduje una pantalla de perfil, la cual serviría para visualizar tus datos y modificarlos, y las pantallas referentes de iniciar y cerrar sesión. Estas no fueron prototipadas, ya que desde un principio la idea principal de la aplicación no incluía esta funcionalidad adicional.



Figura 8: Prototipo de diseño del *ChatBot*

La pantalla que se quería diseñar referente al *ChatBot* iba a ser muy sencilla, incluiría una interfaz de usuario parecida a la de WhatsApp [14].



Figura 9: Prototipo de diseño de la pantalla principal

Para la pantalla principal de la aplicación, se decidió que incluiría una lista de contenedores dónde en ellos se encontrarían los ejercicios a realizar en un día. Con un total de 7 contenedores, cumplirían la función de identificar un día de la semana individualmente.



Figura 10: Prototipo de diseño del planificador de ejercicios



Finalmente, para la última pantalla que estaba destinada a ser implementada, tenemos el planificador. El diseño del planificador iba a ser complejo, ya que este podría crear tanto planificaciones como ejercicios para rutinas. Al querer implementar las dos opciones en un mismo sistema, se idealizó que habría *switches*, los cuales me permitirían cambiar el modo de creación entre ejercicios no-básicos y básicos. Los ejercicios básicos podrían variar entre tener planificación o no, los cuales tengan planificación, podrían ser por planificación por defecto o tener una propia.

En la imagen de la figura 8, vemos el caso de creación de un ejercicio básico, sin planificación, el cual debería tener series, dónde en cada serie habría ejercicios.

El botón de confirmar indicaría el momento en el cual se crea el objeto y se añadiría a la base de datos de la aplicación.

## 4.2 Metodología de trabajo

Para este proyecto se ha seguido la metodología agile Kanban [15].

¿Qué es una metodología agile? Una metodología agile es aquella que consigue adaptar la forma en la que se va a realizar el trabajo con la naturaleza del proyecto. De esta manera proporcionaremos una mayor respuesta para desarrollar el proyecto [16].



Figura 11: Proceso en una metodología agile [17]

He escogido Kanban, ya que esta se expresa mediante tableros. Permite tener una buena organización de mis tareas y tener una visualización diaria de lo que tengo que hacer, donde podré evitar la sobrecarga y podré medir el tiempo estimado en el cual se deberían completar las tareas.

El tablero Kanban está estructurado en 3 columnas: *Backlog* o tareas a realizar, *In progress* o tareas en desarrollo, *To test* o a comprobar y *Done* o tareas completadas.

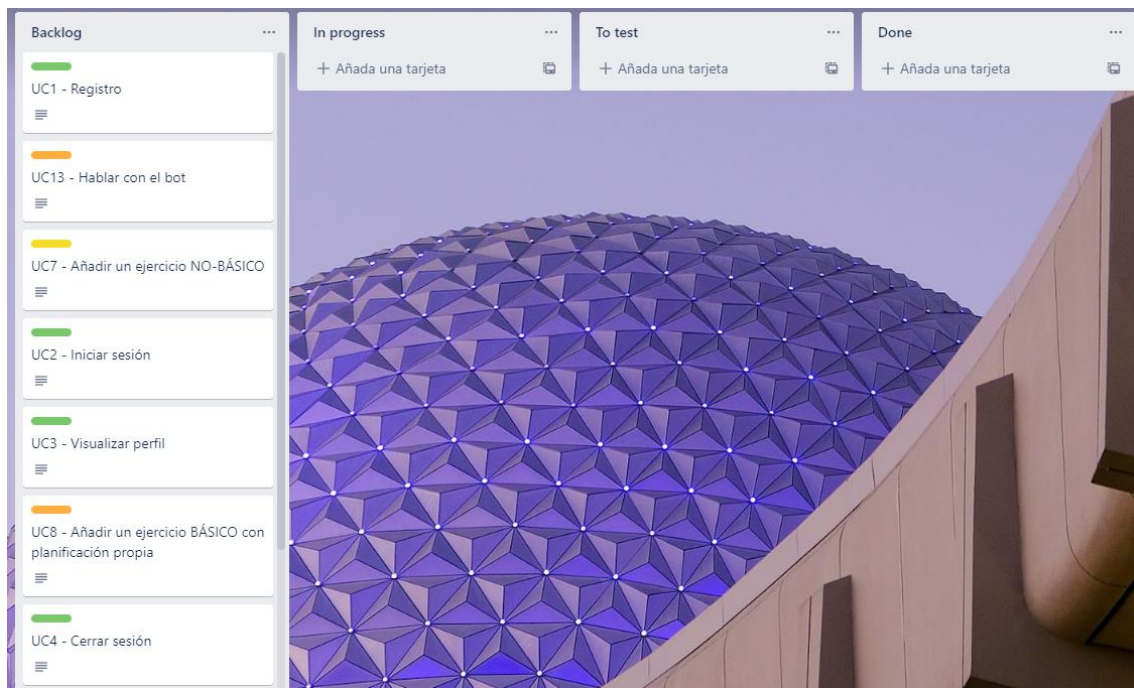


Figura 12: Tablero Kanban

En la figura 12 vemos un ejemplo del tablero con las tareas que han sido escogidas como las más relevantes.

Cada tarea del tablero ha sido evaluada y asignada con un nivel de dificultad, las tareas que he decidido poner han sido los casos de uso más relevantes desarrollados para el proyecto. Una vez se empieza a desarrollar un caso de uso, esta tarea se desplaza hacia la columna *In progress* hasta que se completa. Una vez programada la tarea, esta se prueba y finalmente se mueve a la columna de *Done*, donde permanecerán todas las tareas completadas.

## 5 Implementación y resultado

### 5.1 Tecnologías aplicadas

Estas son las tecnologías que han sido aplicadas al proyecto.

#### 5.1.1 Git y GitHub

Git [18] es un software de control de versiones que permite tener un registro de los cambios de archivos del proyecto, así como crear ramas de trabajo o subir los cambios a la nube, en este caso a Github [19].

Github es una plataforma de desarrollo colaborativo, en la cual se permite el almacenamiento del código desarrollado por el programador utilizando el sistema de control de versiones Git. Github ofrece distintas funcionalidades que facilitan el trabajo colaborativo en un proyecto, en mi caso no he usado ninguna excepto la de almacenaje de la aplicación, ya que he estado trabajando en solitario.

#### 5.1.2 Visual Studio Code y React Native

Visual Studio Code [20] es el editor de texto que he estado utilizando para programar la aplicación. Permite multitud de lenguajes, en mi caso he estado utilizando React Native.

React Native es un marco de trabajo basado en la librería de JavaScript React. [21]

#### 5.1.3 Expo / Expo Go

Expo [22] ha sido la principal herramienta la cual me ha permitido probar la aplicación y poder tener un visual de lo que estaba construyendo.

Es una plataforma la cual conecta mi teléfono móvil con el código para probarlo.

#### 5.1.4 Firebase

Firebase [23] es una plataforma utilizada para acompañar al desarrollo de aplicaciones ofreciendo distintas funcionalidades, de las cuales he extraído Firestore, la base de datos donde se almacenan toda la información del usuario, y Auth, que me permite autenticar los usuarios y mantener un registro de ellos.

#### 5.1.5 Google Cloud: Dialogflow

Dialogflow, es una de las muchas herramientas que ofrece Google. Es una plataforma de comprensión del lenguaje mediante IA. La he utilizado para hacer una integración de una interfaz de usuario en la aplicación, la cual me ha permitido construir el ChatBot.

#### 5.1.6 Postman

Postman [24] es una herramienta útil para realizar llamadas a una API. Se ha utilizado para comprobar la funcionalidad de Google Cloud: Dialogflow.

#### 5.1.7 Trello

Trello [25] es una aplicación tanto web como de dispositivo, la cual me ha permitido organizar mis tareas y tener un visual de las mismas mediante un tablero. Tablero Kanban.

### 5.2 Detalles de la implementación

#### 5.2.1 Estructura del código

Todos los archivos del proyecto tienen la misma estructura. React Native permite crear *functional components*, que son como clases de objetos, mediante una función. Estas funciones

pueden o no recibir datos como parámetros y su valor devuelto es un código JSX, la vista, para representar al usuario.

Por lo tanto, he decidido agrupar frontend y backend en cada fichero de la aplicación dónde tanto llamadas a la base de datos como a las APIs se hacen desde el mismo fichero.

- Sección 1: Imports, dependencias.

Estructura del *functional component*:

- Sección 2: Funciones, que pueden ser llamadas desde los componentes visuales.
- Sección 3: Componentes visuales.

## 5.2.2 Estructura de la base de datos

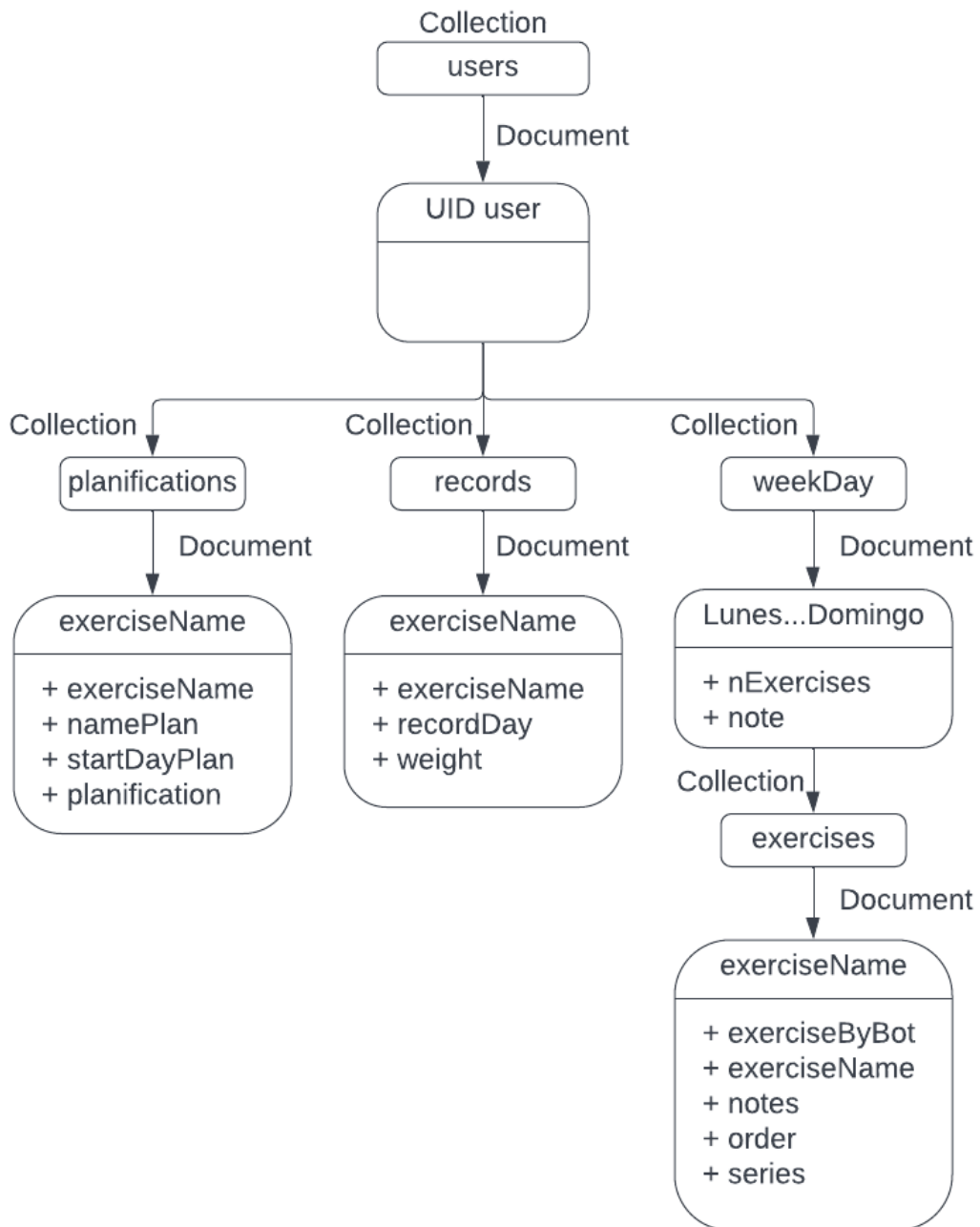


Figura 13: Diagrama de la base de datos

Firestore es una base de datos NoSQL orientada a documentos, no hay tablas sino colecciones de documentos, donde cada documento tiene una estructura similar a JSON, y los documentos pueden contener otras colecciones. La base de datos diseñada parte del punto `users`, lo que significa que cada usuario va a contener todos estos datos.

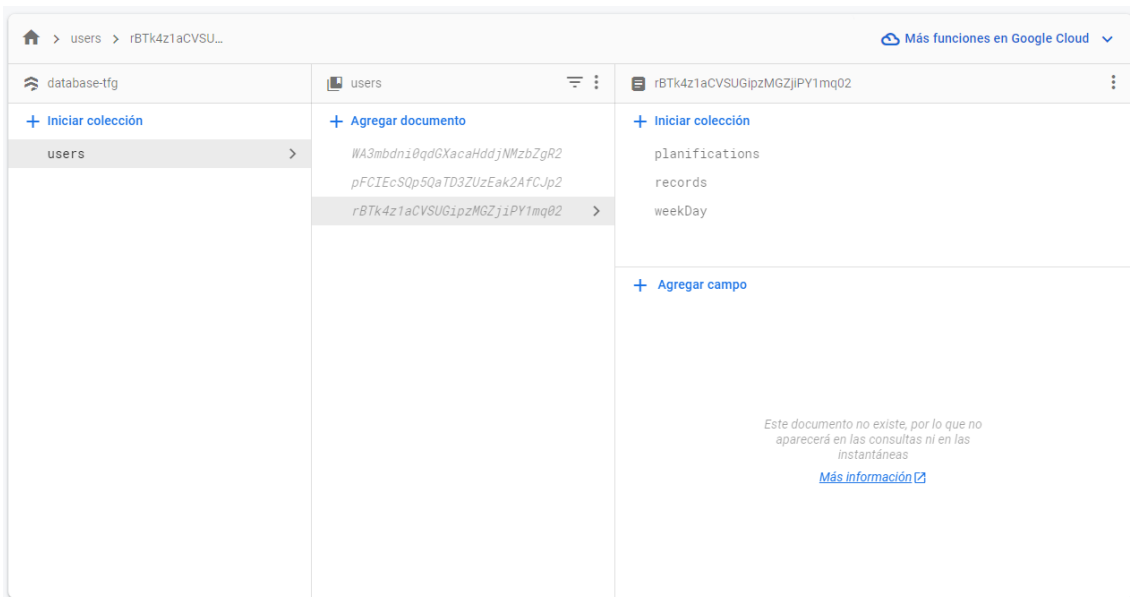


Figura 14: Colección de usuarios de la base de datos.

La principal colección en la cual se basa la base de datos es *users*, la cual cada documento de la colección tiene un identificador de usuario. La colección de *users* está compuesta por 3 sub-colecciones: *planifications*, utilizada para almacenar las planificaciones del usuario, *records*, utilizada para almacenar las marcas personales del usuario y *weekDay*, utilizada para almacenar los ejercicios de la rutina para cada día de la semana.

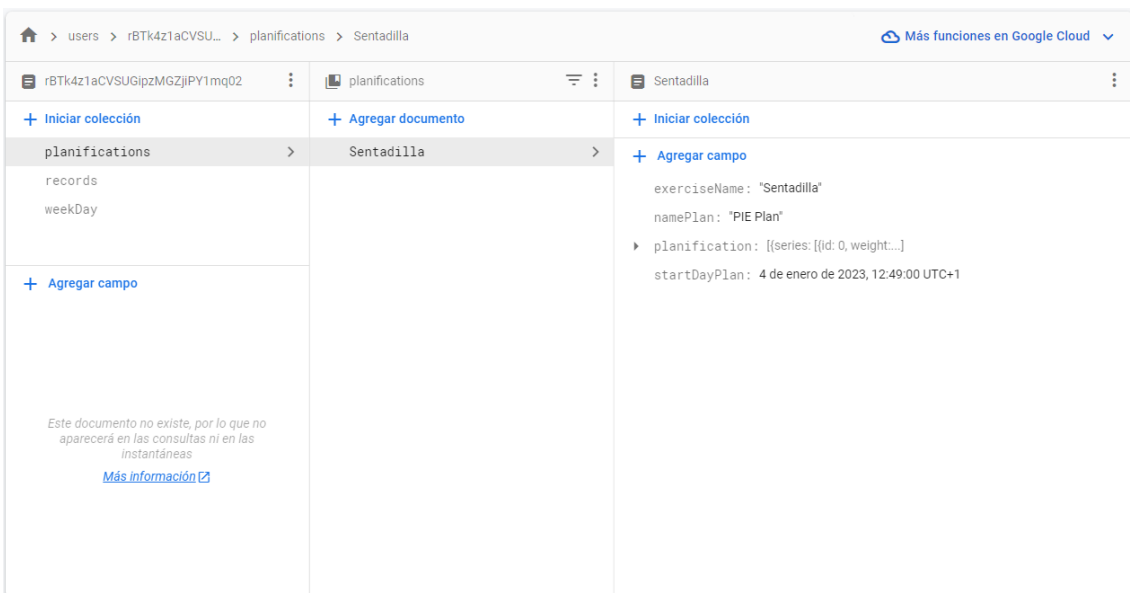


Figura 15: Colección de *planifications* de la base de datos.

La primera sub-colección que encontramos es la de *planifications*, la cual cada documento de la colección es una planificación creada de un ejercicio básico con ID -> nombre del ejercicio. Cada documento o planificación tiene los siguientes campos: *exerciseName* o nombre del ejercicio, *namePlan* o nombre de la planificación, *startDayPlan* o día en el que comienza la planificación y *planification* la cual contiene todos los datos de la planificación, es decir todos los días que la componen, cada serie que has de efectuar ese día, etc.

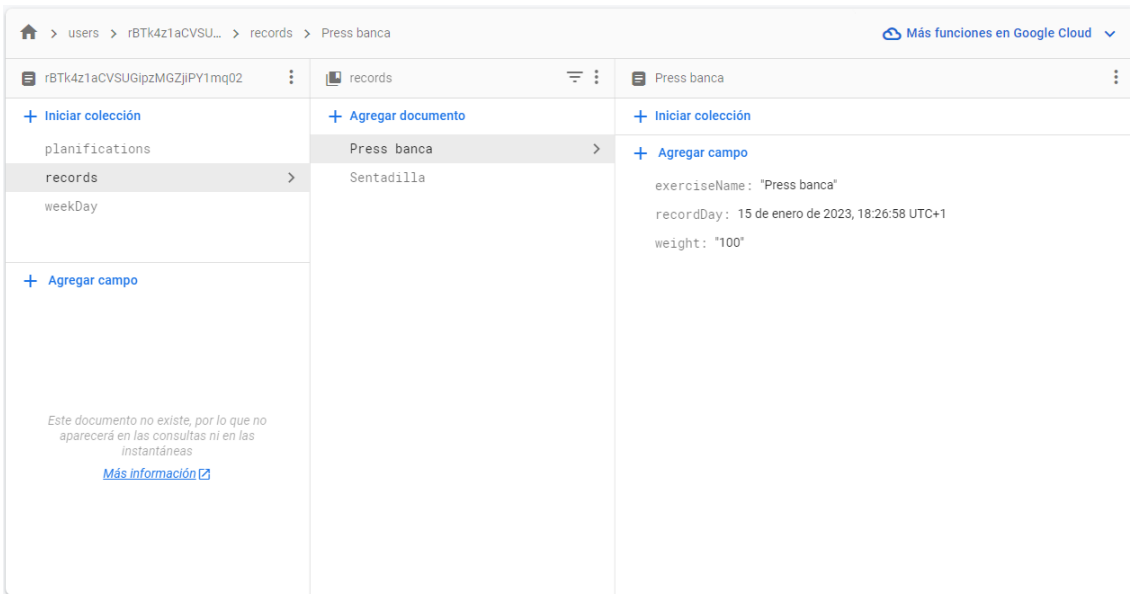


Figura 16: Colección de *records* de la base de datos.

La segunda sub-colección que encontramos es la de *records*, la cual cada documento de la colección es una marca personal con ID -> nombre del ejercicio. Cada documento o marca personal tiene los siguientes campos: *exerciseName* o nombre del ejercicio, *recordDay* o día en el que se batió la marca y *weight* o el peso de la marca.

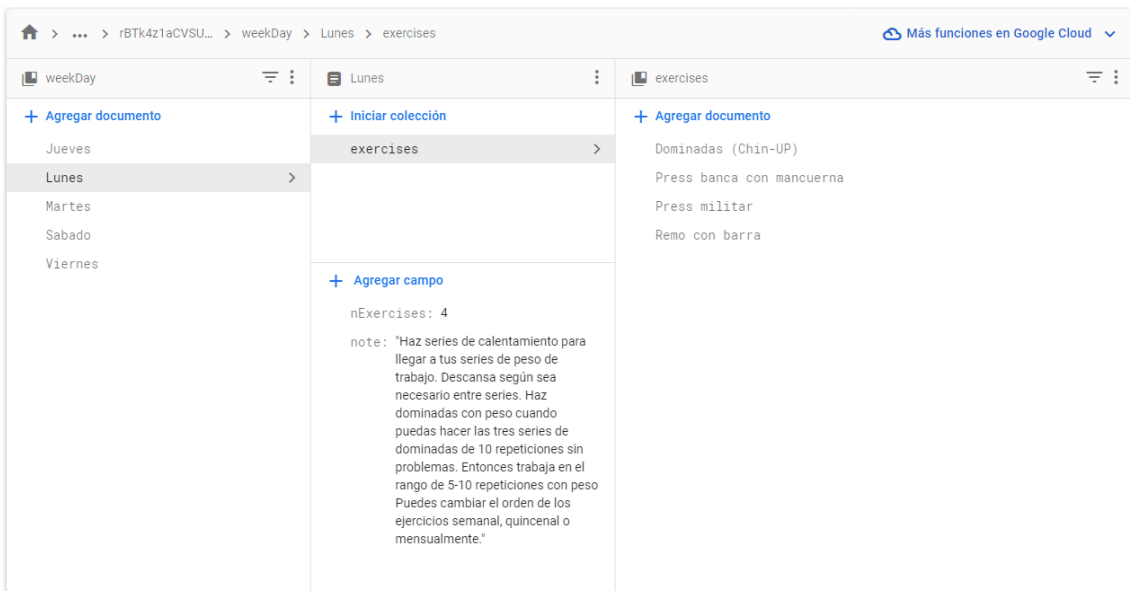


Figura 17: Colección de *weekDay* de la base de datos.

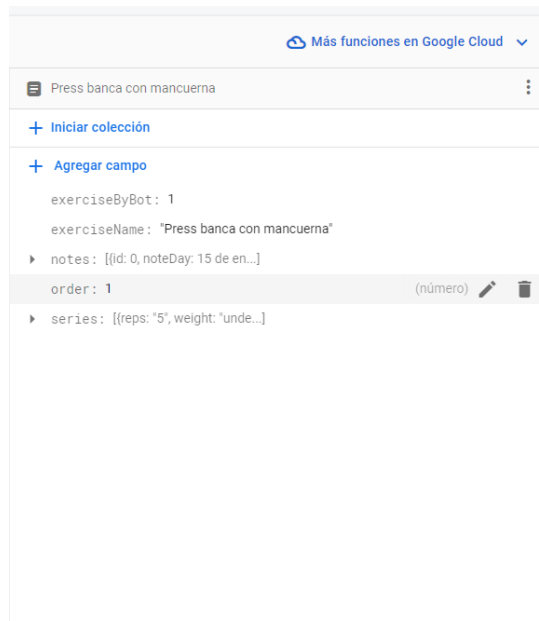


Figura 18: Datos de los documentos de la colección *exercises* de la colección *weekDay*

La última sub-colección que encontramos es la de *weekDay*, la cual cada documento de la colección es un día de la semana con ID -> de lunes hasta domingo. Cada documento tiene los siguientes campos: *nExercises* o número de ejercicios que se realizan ese día y *note* o nota.

Cada documento de la sub-colección *weekDay* contiene otra colección llamada *exercises*, donde cada documento de la colección es un ejercicio. Cada documento tiene los siguientes campos: *exerciseByBot* o ejercicio hecho por el Bot, *exerciseName* o nombre del ejercicio, *notes* o notas, *order* u orden del ejercicio y *series* o series del ejercicio.

### 5.2.3 Obtención de datos de la API

Google Cloud: Dialogflow es la herramienta la cual me ha ayudado a entrenar el Bot.

Cada pregunta que se le puede hacer al Bot significa un *intent*. Un *intent* es la resolución de uno de los casos introducidos por el usuario. Cada *intent* está compuesto por:

Training phrases ⓘ Search training phrases 🔍

⚠️ Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

🗨️  add user expression
🗨️ aló
🗨️ buenas noches
🗨️ buenas tardes
🗨️ buenos días
🗨️ hola
🗨️ hey
🗨️ saludos



Figura 19: Preguntas de entrada de un *Intent*.

- Preguntas de entrada o *training phrases*: Conjunto de preguntas o palabras clave que el Bot detectará del usuario mediante IA. En caso de que la IA detecte una de las preguntas de cualquier *intent*, esta seleccionará el flujo a seguir por ese *intent*.

NAME	ENTITY	VALUE
exercisesszone	@exercisesZone	\$exercisesszone
PROMPTS		
1	¿Que zona del cuerpo quieres entrenar?	
2	Enter a prompt variant	

Figura 20: Parámetros requeridos por el Bot.

- Parámetros requeridos: En caso de que la pregunta realizada requiera parámetros, el sistema indicará que le faltan datos a la pregunta para poder responderla mediante una pregunta.

Responses ⓘ

DEFAULT +

Text Response

1	¡Hola! Me llamo Astro, soy tu asistente virtual. ¿En que puedo ayudarte?
2	¡Hey! Mi nombre es Astro, en que puedo ayudarte?
3	¡Buenas! Soy Astro, tu asistente de preguntas de la aplicación. ¿En que puedo ayudarte?
4	Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation ⓘ

Figura 21: Respuestas del Bot.

- Respuestas: Frase aleatoria de las programadas que el Bot devolverá una vez se haya hecho una pregunta de entrada.

Una vez entrenado al Bot con varios *Intents*, es el momento de conectarlo a la aplicación.

Para conectarlo a la aplicación debemos de seguir los siguientes pasos:

1. Crear una cuenta de servicio de Google Cloud
2. Crear credenciales para tener una verificación de conexión a la Api:
  - a. Una clave para la API
  - b. Un ID de cliente Oauth2

3. Una vez tengamos las credenciales creadas, utilizaremos Postman para comprobar los *endpoints* de la API de Google Cloud: Dialogflow.
  - a. *Endpoint* para detectar *intents* en función de la *query* del usuario:  
<https://dialogflow.googleapis.com/v2beta1/projects/tfg-bot-hrov/agent/sessions/123456789:detectIntent>
4. Comprobado su funcionamiento, añadiremos a nuestro código una función para que, en función de la entrada de texto del usuario por pantalla, ejecute el *endpoint* con la *query* y nos devuelva el resultado del Bot.

## 5.3 Resultado

### 5.3.1 Pantallas

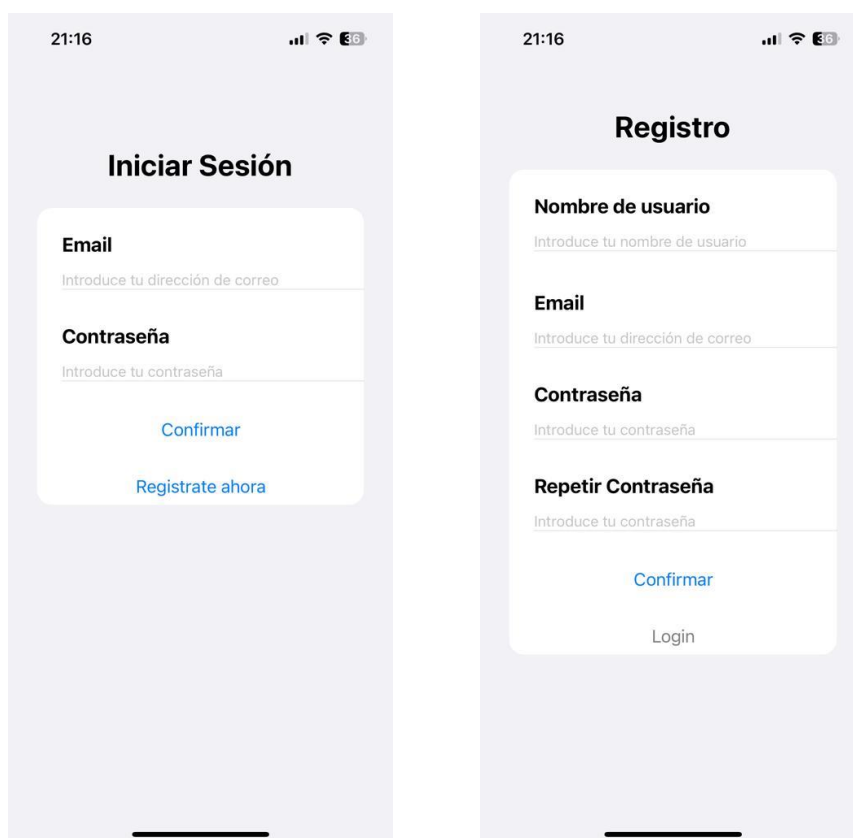


Figura 22: Pantallas de iniciar sesión y registro

La figura 24 representa las pantallas iniciales, las cuales se muestran una vez abres por primera vez la aplicación. Muestran los dos métodos de acceso a la aplicación. Primero el inicio de sesión, el cual pide el correo electrónico y contraseña de una cuenta previamente creada. Finalmente, el registro, el cual pide un nombre de usuario, el correo electrónico y una contraseña para darse de alta a los servicios de la aplicación.

Estas dos no se volverán a mostrar una vez hayas iniciado la aplicación, ya que esta detectara un inicio de sesión activo y entrara automáticamente. Para registrar o iniciar sesión con otra cuenta hará falta que cerremos la sesión.

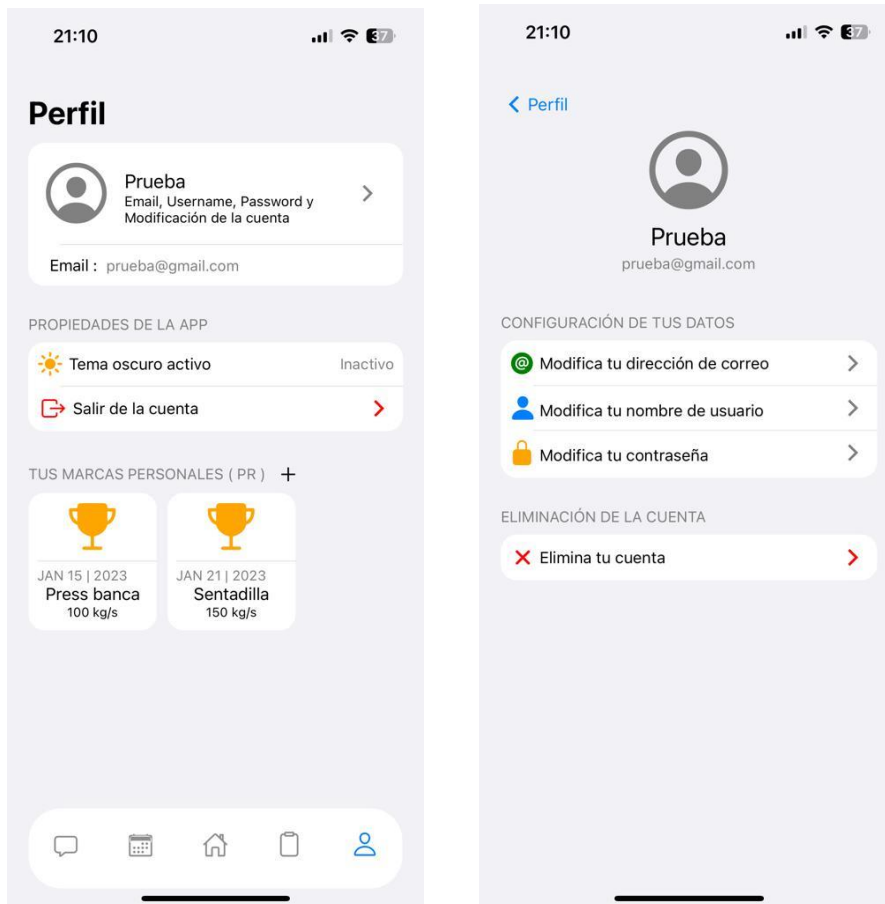


Figura 23: Pantallas del Perfil

La figura 25 representa las pantallas referentes al perfil del usuario. En la primera imagen observamos el Perfil una vez accedemos a él dónde podremos encontrar un resumen de nuestros datos en la parte superior. Seguidamente encontraremos un apartado para ver si tenemos o no el tema oscuro de la aplicación activado y un botón de cierre de sesión. En la parte más baja de la pantalla observamos un espacio dedicado a las marcas personales del usuario, el cual podrá añadir y eliminar.

En la segunda pantalla que observamos, podremos modificar nuestros datos del perfil así como nuestra dirección de correo electrónico hasta nuestra contraseña y nombre de usuario. También podremos eliminar la cuenta con todos los respectivos datos.

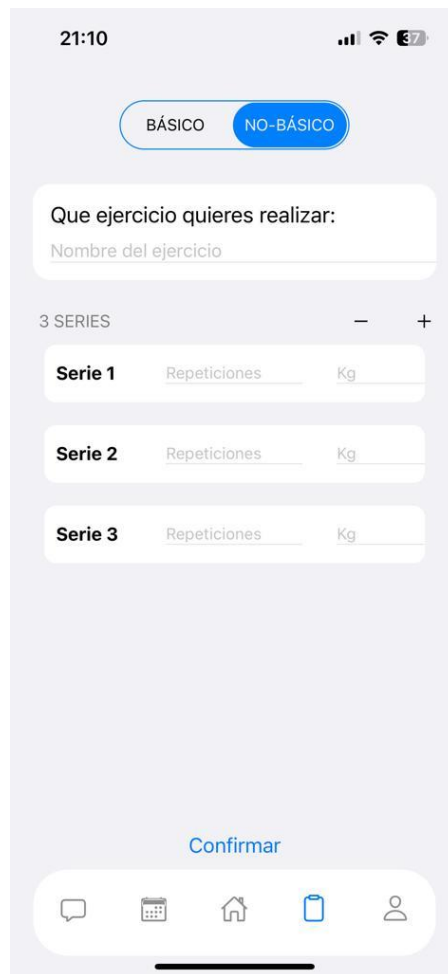


Figura 24: Pantalla del planificador de ejercicios

La figura 26 representa la pantalla de planificación de un ejercicio no-básico, en la cual encontramos los campos de: nombre del ejercicio y series a añadir, las cuales podremos introducir repeticiones y peso.

En la parte inferior de la pantalla vemos el botón de confirmar, donde una vez estén correctamente introducidos todos los datos, al apretarlo, desplegara un selector del día en que queremos realizar el ejercicio.

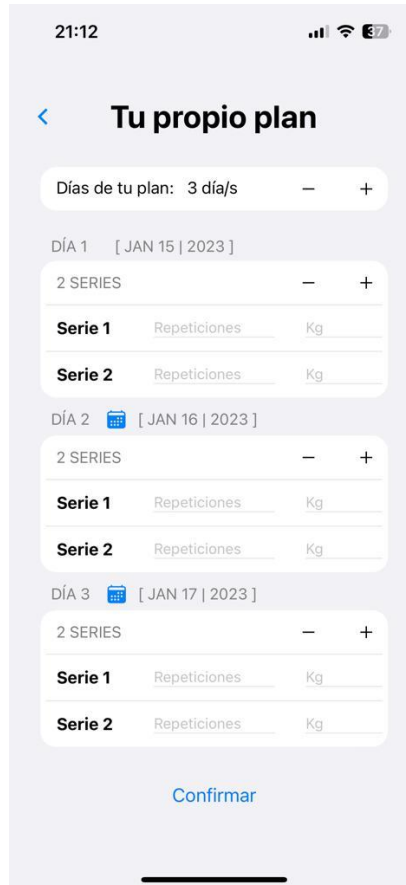
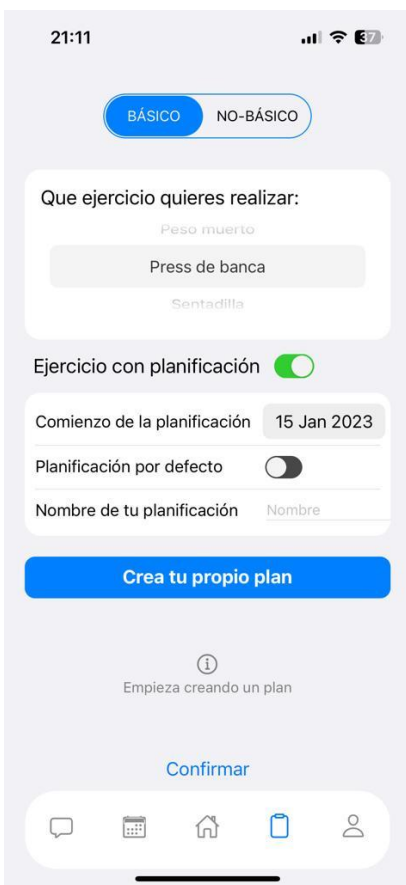


Figura 25: Pantallas del planificador de ejercicios

Esta ha sido sin duda una de las secciones más complicadas de la aplicación, ya que tiene varias opciones que podemos seleccionar.

- Sin planificación: Para este caso vemos la imagen de arriba a la izquierda de la figura 27. Pantalla compuesta por un seleccionador del ejercicio a realizar y las series para añadir, en las cuales podremos introducir la cantidad de repeticiones y el peso de cada serie.
- Con planificación por defecto: Para este caso vemos la imagen de arriba a la derecha de la figura 27. Pantalla compuesta por un seleccionador del ejercicio, campo para introducir el primer día de la planificación, el peso final a levantar y un seleccionador de la planificación.

En la parte inferior de la pantalla encontraremos una vista previa de nuestra planificación (se tiene que haber introducido el campo de peso final a levantar anteriormente para visualizar la estimación, para la imagen se han introducido 100kg de prueba) y el botón de confirmar, el cual una vez estén todos los campos correctamente introducidos podremos añadir la planificación.

- Con planificación propia: Para este caso vemos la imagen de abajo a la izquierda. Compuesta por los mismos campos que el caso anterior menos el del peso a levantar, el cual es substituido por el nombre de la planificación.

Encontramos además un botón para crear la planificación propia, el cual nos dirige a la pantalla de la imagen de abajo a la derecha, la cual está compuesta por todos los campos necesarios para crear una planificación: los días de la planificación, la cantidad de series de cada día y los campos respectivos de cada serie, las repeticiones y el peso.

Una vez estén introducidos correctamente los campos podremos confirmar la planificación con el botón que se encuentra en la parte inferior de la pantalla.



Figura 26: Pantalla principal de la aplicación

La figura 28 representa la pantalla principal de la aplicación. En ella encontramos un contenedor del día de la semana en cuestión. Cada contenedor estará relleno de ejercicios a realizar con sus respectivas series. Cada ejercicio podrá ser cambiado de orden, de nombre y eliminado. El botón amarillo de la parte superior indica información adicional sobre la rutina y el botón rojo indica que puedes eliminar toda la rutina de ese día.

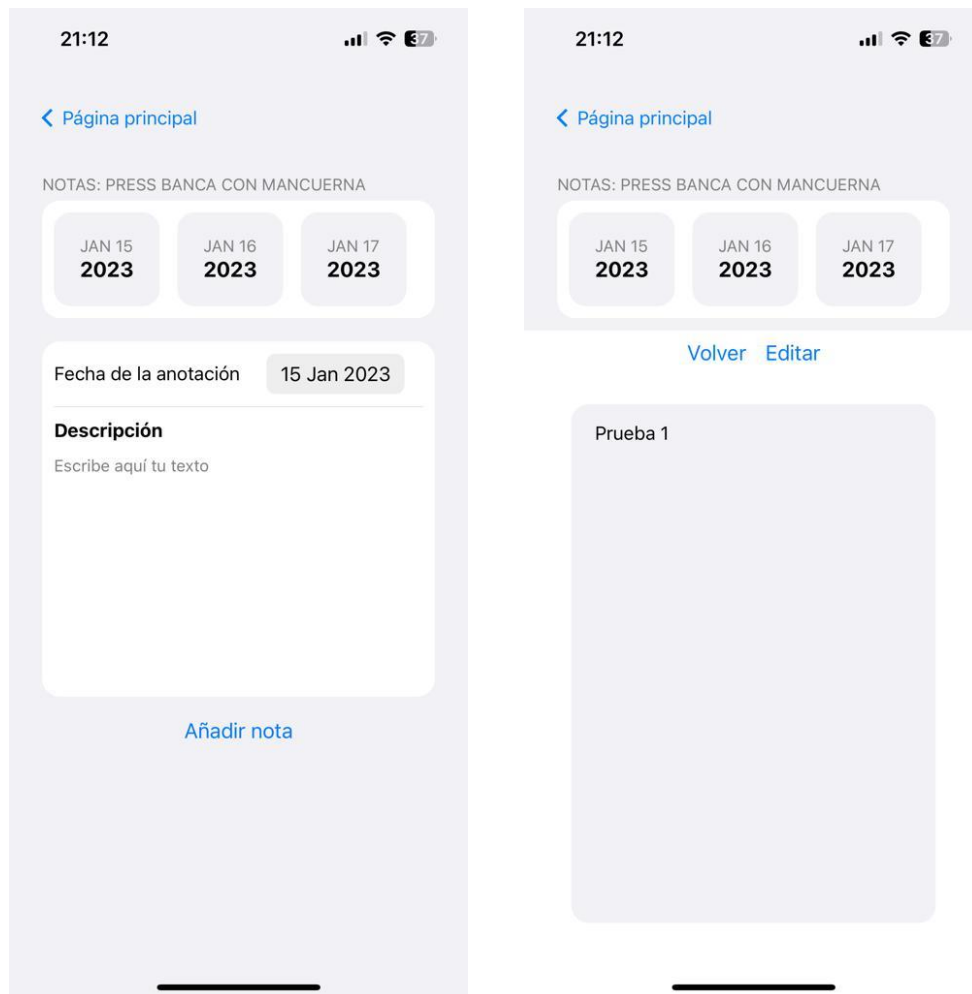


Figura 27: Pantallas de las notas

La figura 29 representa las pantallas en la cual se gestionan las notas para cada ejercicio. En la primera imagen encontramos el listado de notas totales en la parte superior y el formulario para añadir una en la parte inferior.

En la segunda imagen vemos el resultado una vez presionado el botón de una de las notas de la parte superior de la primera imagen.



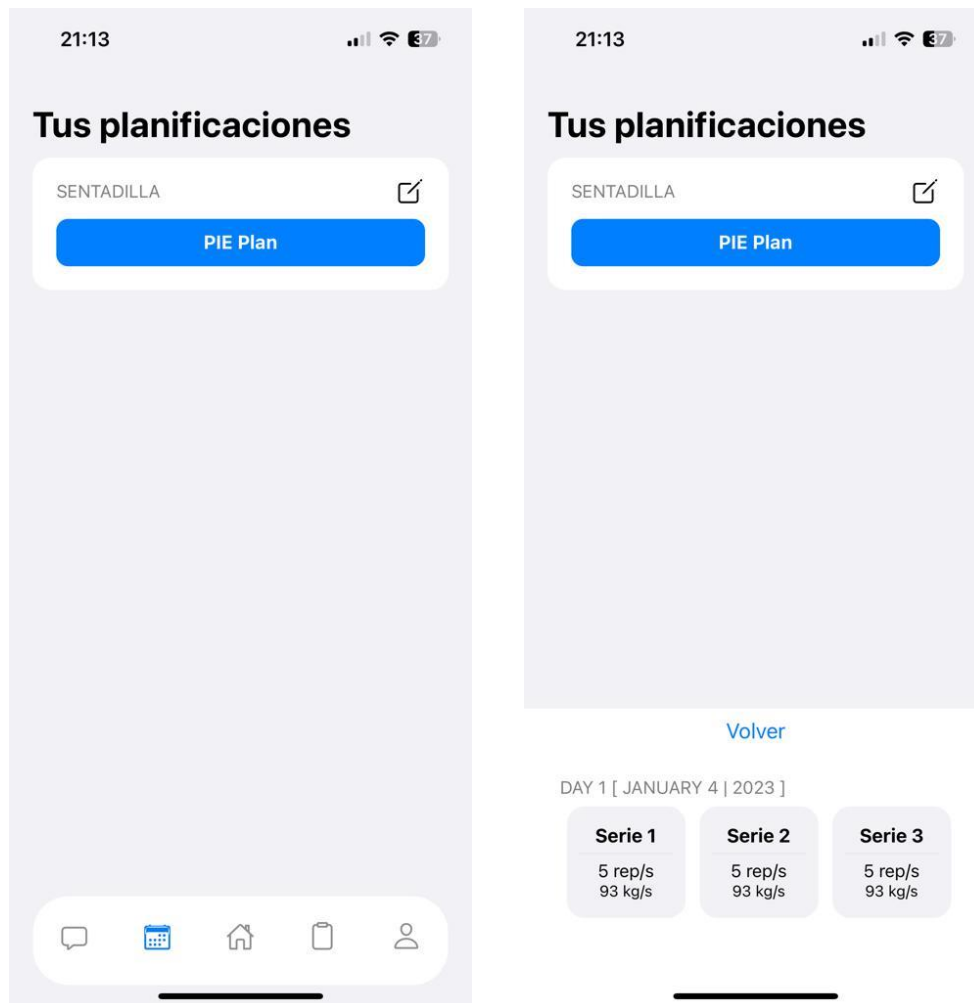


Figura 28: Pantallas de las planificaciones

La figura 30 (izquierda) representa la pantalla en la cual se muestran todas las planificaciones creadas. Cada planificación puede ser renombrada y eliminada pulsando el botón orientado a la derecha del nombre del ejercicio.

La figura 30 (derecha) representa el resultado de apretar el botón azul, el cual muestra un desplegable con la información de la planificación.

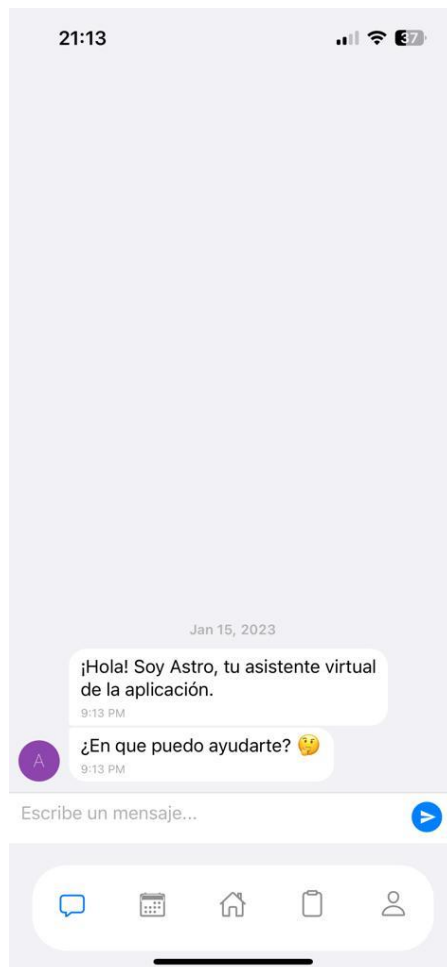


Figura 29: Pantalla del Bot

La figura 31 es una de las pantallas más importantes de la aplicación, la sección del Bot. En la cual encontramos un chat, parecido al estilo de WhatsApp. En esta pantalla podemos interactuar con el Bot.

A continuación, se muestra un ejemplo de interacción con el Bot para añadir una rutina:

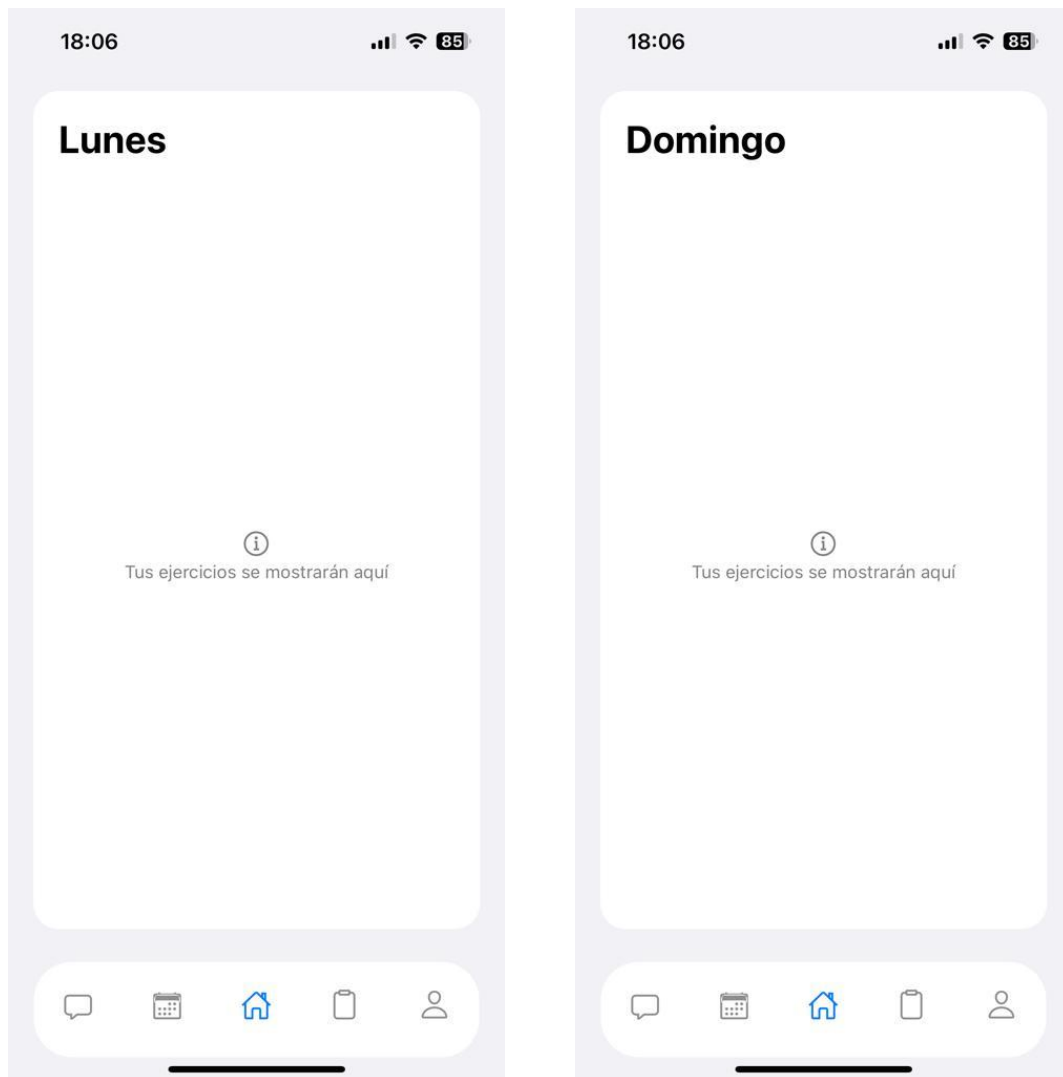


Figura 30: Pantalla principal de la aplicación sin ejercicios

La figura 30 representa el estado inicial en el que nos encontraríamos la aplicación una vez iniciada por primera vez. El siguiente paso por seguir es dirigirse a la pantalla del Bot y hacer nuestra petición.

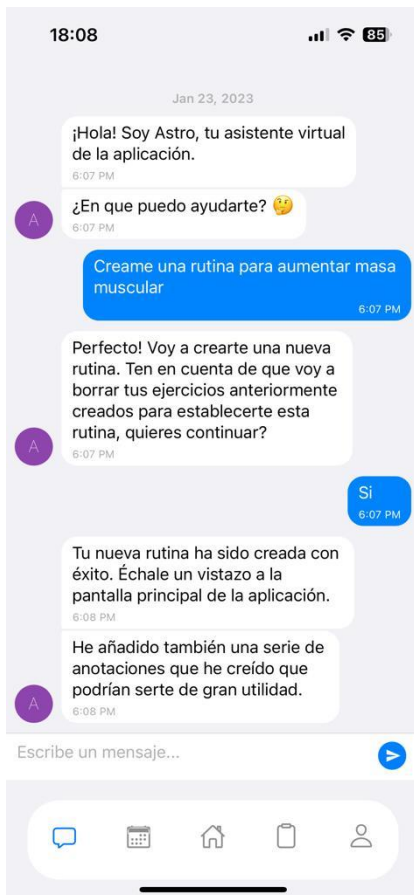


Figura 31: Pantallas de interacción con el Bot

- En la imagen de arriba a la izquierda de la figura 31 observamos la petición del usuario. El Bot a continuación responde conforme a la entrada del usuario.
- En la imagen de arriba a la derecha de la figura 31 vemos que en caso de que el usuario esté de acuerdo con el mensaje del Bot, la aplicación mostrará una alerta avisando de que el Bot va a añadir una rutina.
- En la imagen de abajo a la izquierda de la figura 31, el Bot confirma la creación de la rutina.
- En la imagen de abajo a la derecha de la figura 31 observamos que el Bot ha añadido correctamente los ejercicios a la pantalla principal de la aplicación.

### 5.3.2 Flujo de la aplicación



Figura 32: Barra de navegación

La barra de navegación de la aplicación está compuesta por 5 elementos:

1. Pantalla del Bot de conversación.
2. Pantalla de planificaciones.
3. Pantalla principal de la aplicación.
4. Planificador de ejercicios.
5. Perfil del usuario.

Cada clic a uno de estos botones de la barra de navegación redireccionará al usuario a la respectiva ventana.

### 5.3.2.1 Flujo general de la aplicación

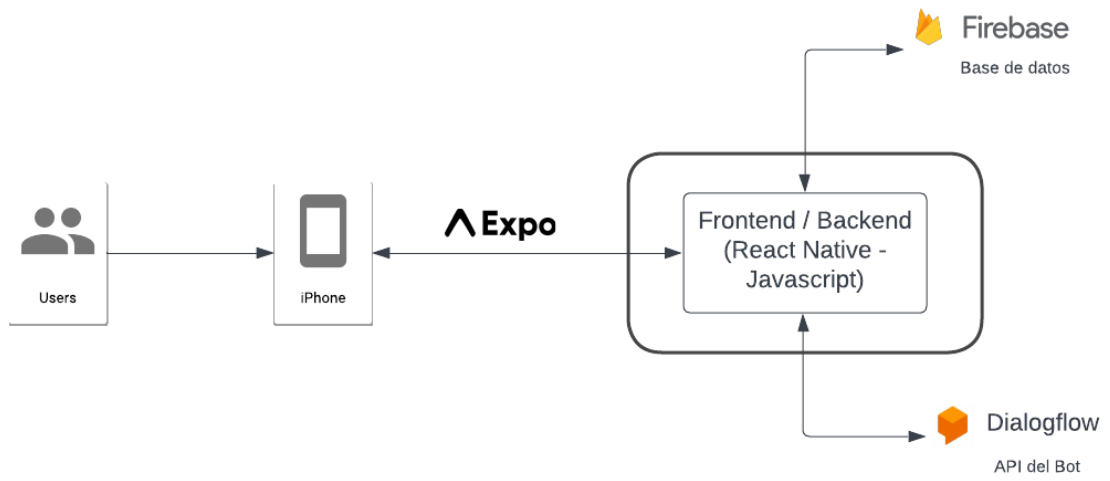


Figura 33: Diagrama de flujo general del sistema

## 6 Conclusiones

### 6.1 Conclusiones

El desarrollo de esta aplicación ha sido un proceso lleno de desafíos y aprendizajes valiosos dónde se han podido resolver todos los matices que abarcaban el problema a resolver mediante una buena metodología de trabajo.

Durante la realización de este proyecto, se ha trabajado con una metodología agile, metodologías de las cuales actualmente están siendo utilizadas por un gran número de empresas, además de ser un punto clave en cualquier tipo de proyecto en lo que se refiere a desarrollo de software. Es un método de trabajo muy eficiente cuando lo que se quiere es obtener resultados de forma segura y rápida.

A más, el Bot implementado, ha sido sin duda uno de los puntos claves finalmente de la aplicación, ya que además de resolver dudas a los usuarios, ha sido capaz de automatizar procesos de la aplicación, como bien podría ser la creación de una rutina de entrenamiento. La variedad de Inteligencias artificiales que hay actualmente implementadas en Bots conversacionales están muy avanzadas, nivel de las cuales me gustaría haber implementado en mi aplicación. Sin embargo, en esta primera versión de la aplicación, por limitaciones de tiempo, me he limitado a implementar un Bot de capacidades más limitadas. Personalmente, me hubiese gustado más haberle dedicado un tiempo razonable extra a la implementación de este.

En resumen, la conclusión es que, con una metodología clara y una estrategia bien planificada, se ha logrado desarrollar una aplicación móvil con un gran potencial de crecimiento.

### 6.1 Futuro trabajo

Ninguna aplicación es perfecta, una vez desarrollada siempre queda algún trabajo pendiente que hacer o alguna ampliación extra que se podría implementar añadiendo así más versiones de la aplicación.

Estas son unas de las implementaciones que podrían aportar más contenido y ayuda a la aplicación.

- Implementación de estadísticas: Con una sección con estadísticas en función de tus avances ayudarían a las personas a observar más visualmente sus progresos en función del tiempo.
- Sistema de recomendaciones: Un sistema de recomendaciones, facilitaría las decisiones del usuario en cuanto a las rutinas o que es lo mejor que puede realizar en cada etapa de tiempo.
- Sección alimentaria: Una sección alimentaria, completaría la aplicación totalmente, ya que siempre detrás de un buen entrenamiento, tiene que haber una buena alimentación.
- Crear una base de datos con ejercicios: En el momento de crear un ejercicio, estaría bien que en vez de escribir tú el ejercicio, tuvieses una base de datos con cada ejercicio ordenado por sección muscular del cuerpo, la cual ayudaría a entender mejor a usuarios novatos que ejercicio está realizando o creando.

- Añadir temporizadores y notificaciones. Mejoraría la usabilidad de la aplicación, notificaciones de cuando hayas de ir al gimnasio o temporizadores del tiempo de descanso entre ejercicios.
- Mejorar la dinámica y el diseño de la aplicación.



## 7 Referencias

OpenAI. ChatGPT: Optimizing Language Models for Dialogue.

[1] <https://openai.com/blog/chatgpt/>

Strong. Think less. Lift more.

[2] <https://www.strong.app/>

Hevy. Track workouts make progress.

[3] <https://www.hevyapp.com/>

Jefit. MANAGE & TRACK ALL YOUR WORKOUTS IN ONE PLACE.

[4] <https://www.jefit.com/>

React Native. Learn once, write anywhere.

[5] <https://reactnative.dev/>

Google Cloud: Dialogflow.

[6] <https://cloud.google.com/dialogflow>

Modelo PIP. Cómo PROGRAMAR FÁCIL con el MODELO PIP.

[7] [https://www.youtube.com/watch?v=T\\_xaM9GpoTM](https://www.youtube.com/watch?v=T_xaM9GpoTM)

Fitenium. LAS 4 PROGRAMACIONES DEL ENTRENAMIENTO.

[8] <https://fitenium.com/las-4-programaciones-del-entrenamiento/>

Apta VITAL SPORT. Ejercicios básicos de gimnasio. Cuáles son y su importancia.

[9] <https://aptavs.com/articulos/ejercicios-basicos-gimnasio>

Vitonica. Guía para principiantes (II): Press de banca.

[10] <https://www.vitonica.com/musculacion/guia-principiantes-ii-press-banca>

Vitonica. Guía para principiantes (VIII): Squats o sentadillas.

[11] <https://www.vitonica.com/musculacion/guia-para-principiantes-viii-squats-sentadillas>

Vitonica. Guía para principiantes (VII): Peso muerto.

[12] <https://www.vitonica.com/musculacion/guia-principiantes-vii-peso-muerto>

Vitonica. Guía para principiantes (XXXIV): Press frontal con barra o press militar.

[13] <https://www.vitonica.com/musculacion/guia-para-principiantes-xxxiv-press-frontal-con-barra-o-press-militar>

WhatsApp. Envía mensajes privados.

[14] <https://www.whatsapp.com/?lang=es>

apb. ¿En qué consiste la metodología Kanban y cómo utilizarla?

[15] <https://www.apd.es/metodologia-kanban/>

iebs.

[16] <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/#:~:text=Por%20definici%C3%B3n%20las%20metodolog%C3%ADas%20%C3%A1giles,las%20circunstancias%20espec%C3%ADficas%20del%20entorno.>

ain. Metodología Agile.

[17] <https://www.ain.es/actualidad/ain-pone-en-marcha-un-proyecto-que-ayudara-a-las-pymes-navarras-en-sus-procesos-de-transformacion-digital-y-organizativa-bajo-la-metodologia-agile/>

Git. --everything-is-local.

[18] <https://git-scm.com/>

Github.

[19] <https://github.com/>

Visual Studio Code. Code editing.Redefined.

[20] <https://code.visualstudio.com/>

Reactjs. A JavaScript library for building user interfaces.

[21] <https://reactjs.org/>

Expo.

[22] <https://expo.dev/>

Firebase.

[23] <https://firebase.google.com/?hl=es>

Postman.

[24] <https://www.postman.com/>

Trello. Trello unifica tus tareas, compañeros de equipo y herramientas.

[25] <https://trello.com/es>

## Anexos

Tema oscuro de la aplicación

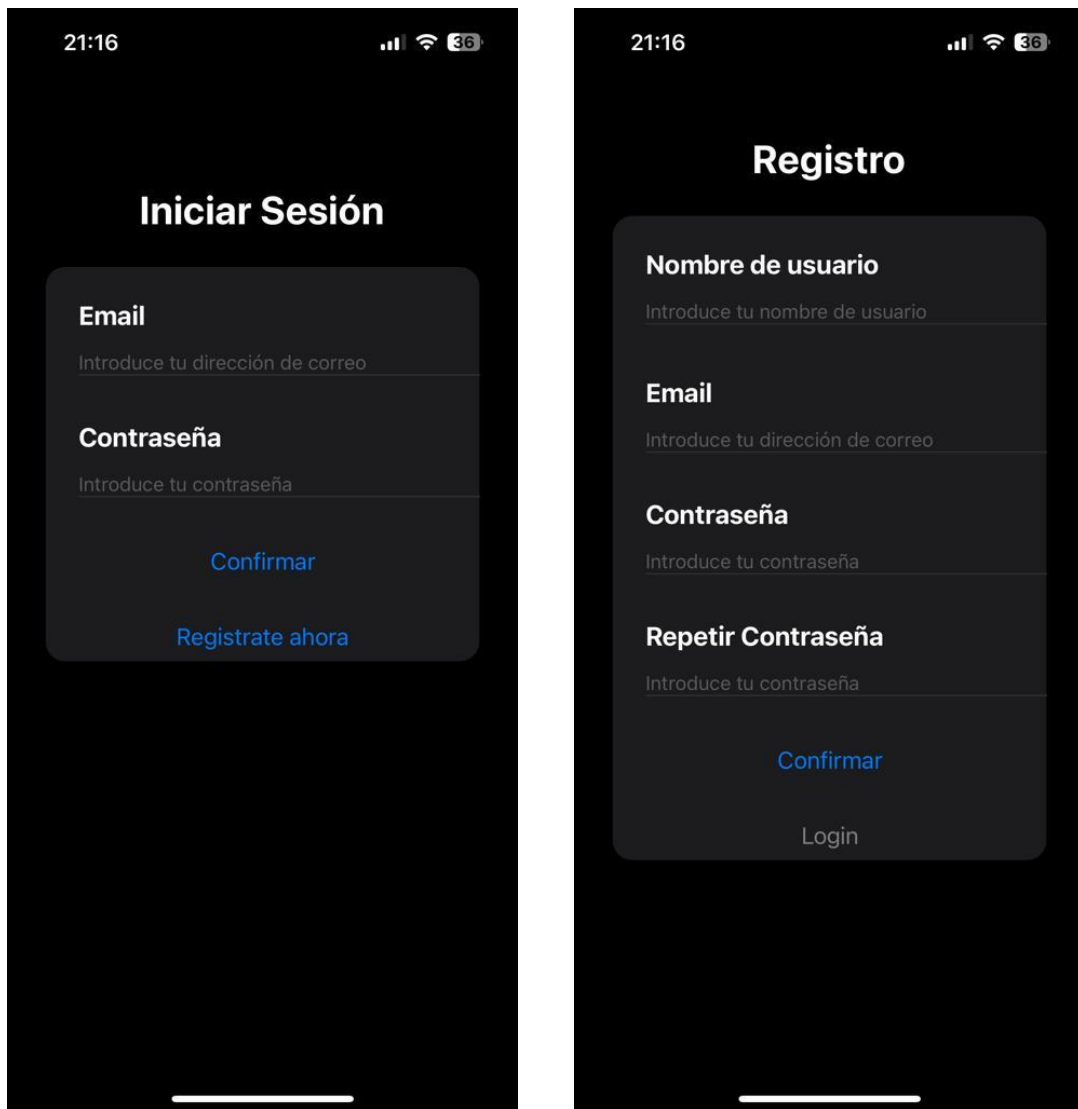


Figura 34: Pantallas de iniciar sesión y registro

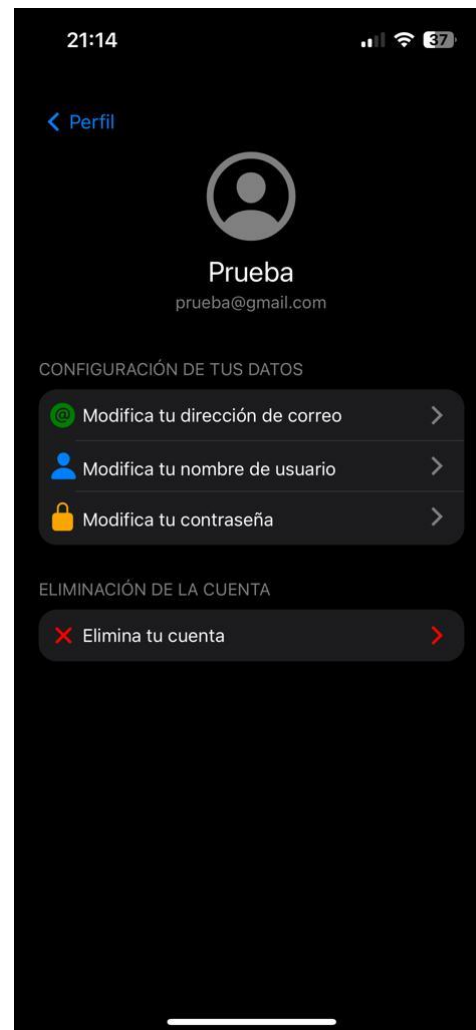
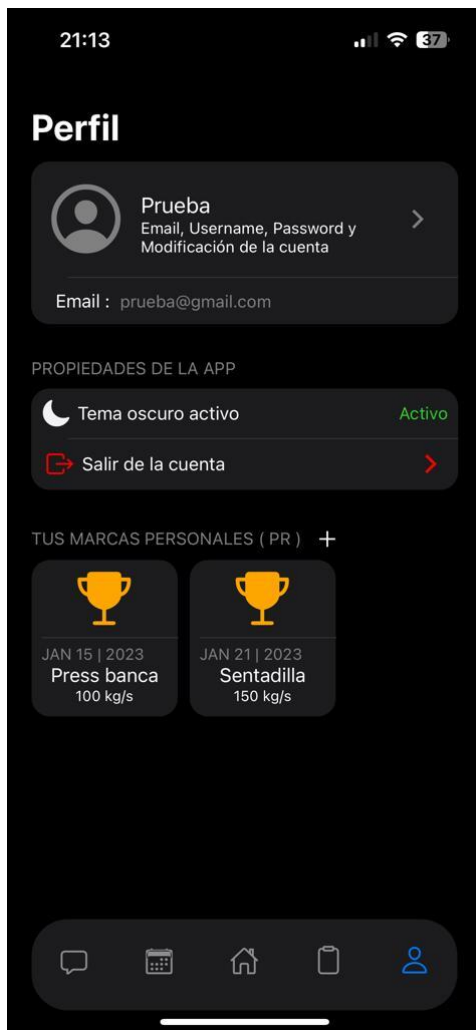


Figura 35: Pantallas del Perfil



Figura 36: Pantalla del planificador de ejercicios

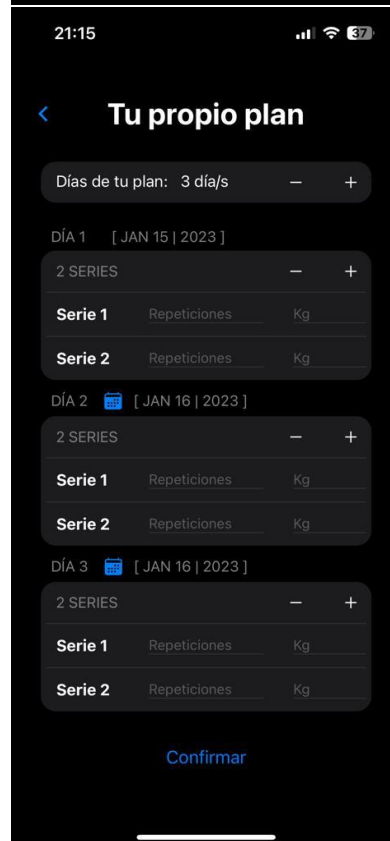
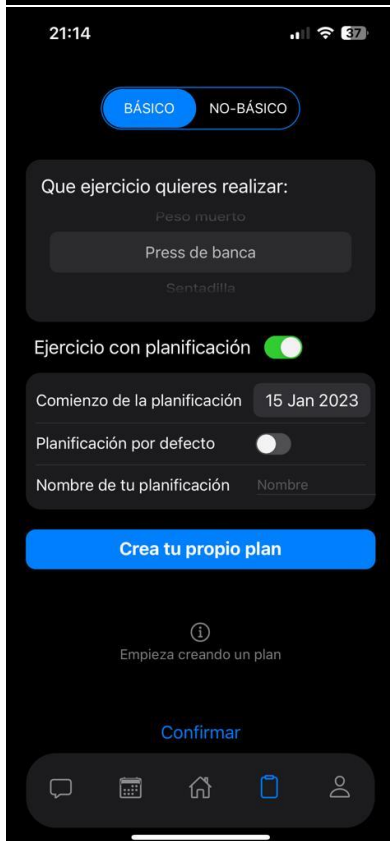
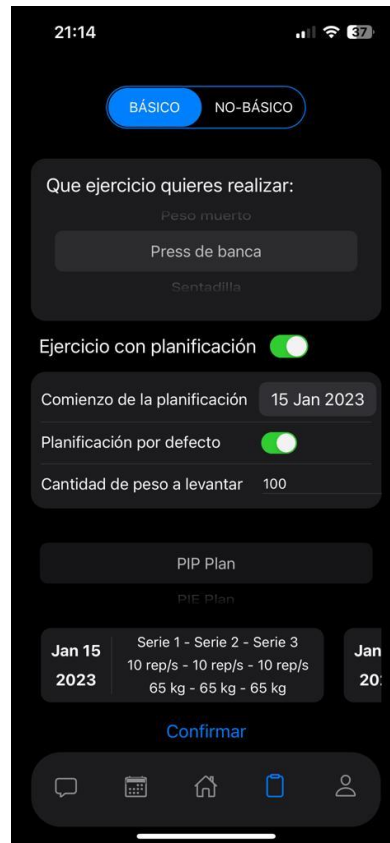


Figura 37: Pantallas del planificador de ejercicios

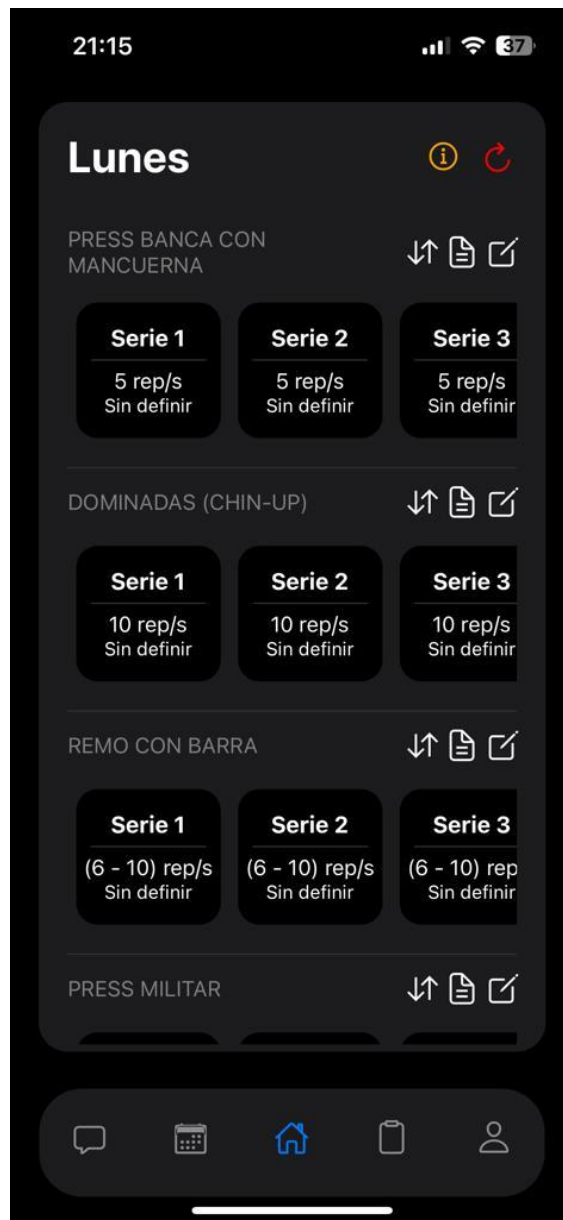


Figura 38: Pantalla principal de la aplicación





Figura 39: Pantallas de las notas

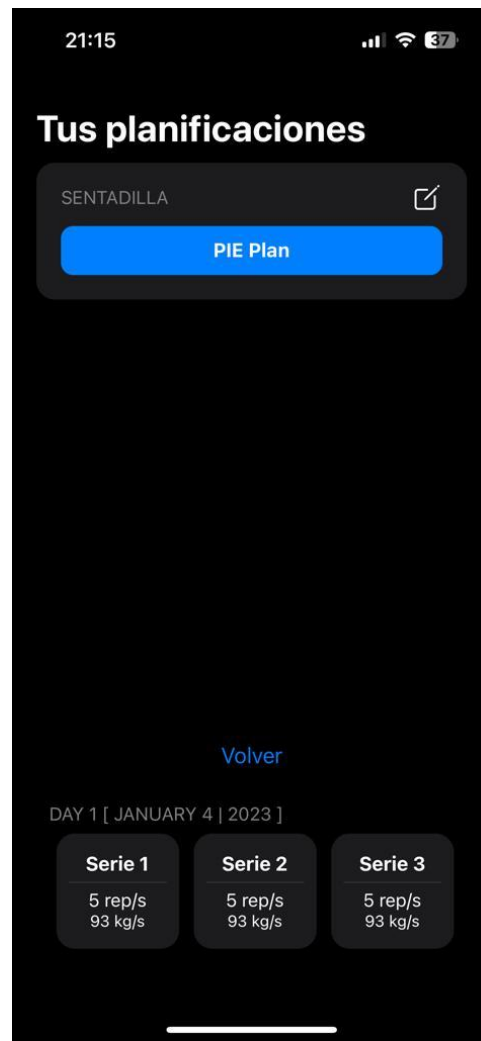
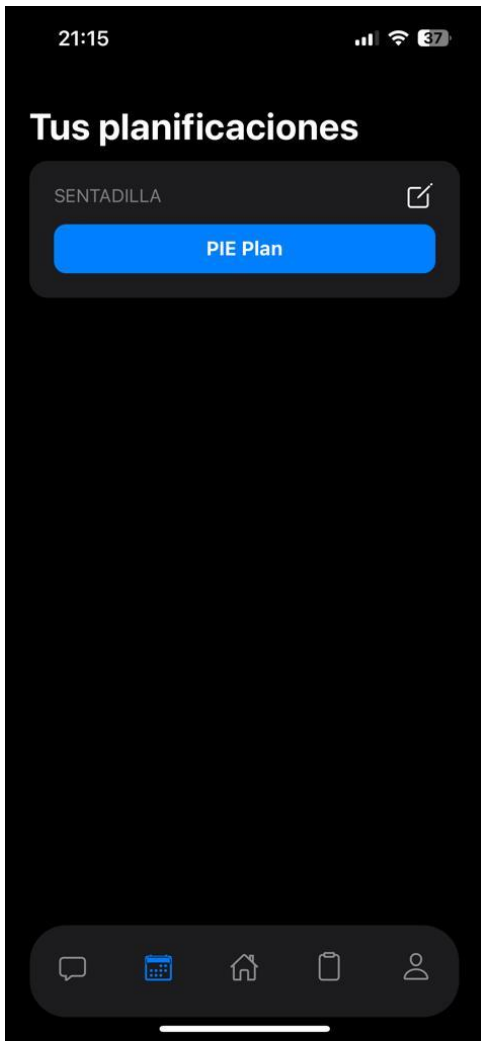


Figura 40: Pantallas de las planificaciones



Figura 41: Pantalla del Bot