



UNIVERSIDAD DE GRANADA

TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

ImagymBot

ChatBot de Telegram para fidelizar a los clientes de un gimnasio

Autor

Juan Manuel Castillo Nievas

Directora

Rosana Montes Soldado



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, 5 de julio de 2020

Dª. **Rosana Montes Soldado**, profesora titular del área de Lenguajes y Sistemas Informáticos del departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada, como tutora de este TFG

Informa

Que el presente trabajo, titulado **ImagymBot. ChatBot de Telegram para fidelizar a los clientes de un gimnasio**, ha sido realizado bajo mi supervisión por **Juan Manuel Castillo Nievas**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, firmo la presente en Granada a 30 de junio de 2020.

**Rosana Montes
Soldado**

Firmado digitalmente por
Rosana Montes Soldado
Fecha: 2020.06.30
12:41:58 +02'00'

Fdo. La directora

Índice general

1. Introducción	8
1.1. Motivación	8
1.2. Alcance	10
1.3. Objetivos generales	11
2. Estado del arte	13
2.1. Metodología de análisis	13
2.1.1. Fitify - Entrenamientos y Planes	15
2.1.2. Gym WP - Ejercicios y Rutinas para el gimnasio	18
2.1.3. BestFit: Rutina Gimnasio - Entrenamiento y Fitness	21
2.1.4. Fitvate - Gimnasio Rutina de ejercicio Entrenador	23
2.1.5. Total Fitness - Rutinas y Gimnasio	24
2.1.6. Home Workout - No equipment	28
2.1.7. Samsung Health	30
2.1.8. Muscle Booster - Fitness at home	33
2.2. Tabla comparativa de apps	37
3. Planificación del proyecto	41
3.1. Metodología de desarrollo: diseño centrado en el usuario	41
3.1.1. Design Thinking	43
3.2. Personas ficticias	44
3.3. Malla receptora de información	49
3.4. Definición de la propuesta	49
3.5. Recursos necesarios	51
4. Diseño de la Interfaz de Usuario	55
4.1. Tareas de usuario	55
4.2. Organización de la información	57
4.3. Bocetos wireframe	59
5. Diseño técnico	67
5.1. Requisitos funcionales	68
5.2. Requisitos no funcionales	69

5.3.	Diseño de la base de datos	70
5.3.1.	Mi objetivo de peso	70
5.3.2.	Mi objetivo de actividades cardio	70
5.3.3.	Retos	71
5.3.4.	Ejercicio del mes	71
5.3.5.	Rutinas y entrenamiento	71
5.3.6.	Mi ficha personal	72
6.	Implementación	76
6.1.	Cómo configurar el servicio de MySQL	80
6.2.	Implementando los Handlers del bot	81
6.2.1.	Respondiendo a las acciones del usuario	82
6.2.2.	Ejecutando sentencias MySQL	85
6.2.3.	Añadiendo Handlers dinámicos	86
6.2.4.	Añadiendo Fallbacks	87
6.3.	Comunicación entre ImagymBot y la plataforma web	88
6.3.1.	Archivo de entrada: Ejercicios del mes	89
6.3.2.	Archivo de entrada: Retos	90
6.3.3.	Archivo de entrada: Rutinas y entrenamiento	91
6.3.4.	Archivo de salida: Usuarios que han superado un reto	92
6.3.5.	Archivo de salida: Puntuación de los usuarios en el ejercicio del mes	92
6.4.	Problemas de programación más frecuentes	93
6.4.1.	Error al insertar/actualizar/borrar en MySQL	93
6.4.2.	Error al mostrar la ficha personal	94
7.	Pruebas	96
8.	Manual de uso y distribución	99
8.1.	Manual de uso	99
8.1.1.	Inicio	99
8.1.2.	Mi objetivo de peso	100
8.1.3.	Mi objetivo de actividades cardio	104
8.1.4.	Retos	108
8.1.5.	Ejercicio del mes	111
8.1.6.	Rutinas y entrenamiento	115
8.1.7.	Mi ficha personal	119
8.1.8.	Soporte	121
8.2.	Distribución del proyecto	123
9.	Conclusiones finales y trabajos futuros	124

Índice de figuras

1.1. Preferencias en la práctica deportiva. Fuente: Life Fitness [1]	9
1.2. Funcionalidades más usadas los gimnasios que cuentan con aplicación propia. Fuente: Life Fitness [1]	11
2.1. Pantalla de inicio y test inicial	16
2.2. <i>Timeline</i> principal	17
2.3. Programa <i>Empezar a perder peso</i>	17
2.4. Interfaz de entrenamiento	18
2.5. Interfaz de entrenamiento libre	19
2.6. Interfaz de inicio	20
2.7. Menú de inicio y seguimiento de peso diario	20
2.8. Planes de entrenamiento	21
2.9. Interfaz de inicio y regalo de prueba gratuita	22
2.10. Creación de un plan de entrenamiento	22
2.11. Proceso para realizar un entrenamiento	23
2.12. Pantalla inicial de la aplicación	24
2.13. Realización de un programa de entrenamiento	25
2.14. Desafío de sentadillas	25
2.15. Inicio de sesión de la aplicación	26
2.16. Menú y programas de entrenamiento	27
2.17. Tabla de ejercicios y cronómetro	27
2.18. Retos y control de peso/IMC/grasa	28
2.19. Inicio de Home Workout	29
2.20. Establecimiento de objetivo y seguimiento de peso/IMC	29
2.21. Calendario de entrenamientos y ejecución	30
2.22. Sección <i>Descubre</i>	31
2.23. Interfaz de inicio de Samsung Health	32
2.24. Seguimiento de calorías ingeridas	32
2.25. Interfaz de inicio de Samsung Health	33
2.26. Sección de retos	34
2.27. Entrenamientos de Samsung Health	34
2.28. Test inicial de Muscle Booster	36
2.29. Acceso al pago de Muscle Booster	36

3.1. Etapas del proceso de Design Thinking [2]	43
3.2. Usuario perteneciente al grupo 1	46
3.3. Usuario perteneciente al grupo 2	47
3.4. Usuario perteneciente al grupo 3	48
3.5. Tabla comparativa Whatsapp vs Telegram . Fuente: Iván Ramírez [3] .	52
3.6. Distintos tipos de suscripciones a Python Anywhere [4]	54
 4.1. Sitemap de ImagymBot	57
4.2. Sitemap de ImagymBot	58
4.3. Boceto de <i>Inicio</i>	59
4.4. Boceto de <i>Inicio > Mi objetivo de peso</i>	60
4.5. Boceto de <i>Inicio > Mi objetivo de actividad cardio</i>	61
4.6. Boceto de <i>Inicio > Retos</i>	62
4.7. Boceto de <i>Inicio > Ejercicio del mes</i>	63
4.8. Boceto de <i>Inicio > Rutinas y entrenamiento</i>	64
4.9. Boceto de <i>Inicio > Mi ficha personal</i>	65
4.10. Boceto de <i>Inicio > Soporte</i>	66
 5.1. Diagrama Entidad-Relación de la base de datos de ImagymBot	73
5.2. Paso a tablas	74
5.3. Base de datos de ImagymBot	75
 6.1. Primer paso para la creación de un bot	77
6.2. Nombre y usuario de nuestro bot	78
6.3. Token del bot creado	79
6.4. Interfaz de Database en PythonAnywhere	80
 7.1. Puntuaciones del cuestionario SUS	98
 8.1. Inicio de ImagymBot	100
8.2. Menú de <i>Inicio > Mi objetivo de peso</i>	101
8.3. Proceso de anotar datos	101
8.4. Proceso de establecer un objetivo de peso	102
8.5. Menú de <i>Inicio > Mi objetivo de peso</i> con un objetivo de peso establecido	103
8.6. Menú de <i>Inicio > Evolución</i>	103
8.7. Filtrar la evolución del peso por fecha	104
8.8. Menú de <i>Inicio > Mi objetivo de actividades cardio</i>	105
8.9. Proceso de registrar cardio	106
8.10. Proceso de establecer un objetivo personal de cardio	106
8.11. Sumando minutos al objetivo personal de cardio	107
8.12. <i>Inicio > Mi objetivo de actividades cardio > Ver mis registros en cardio</i> .	108
8.13. <i>Inicio > Retos</i>	109
8.14. Anotando el progreso del reto	110
8.15. Viendo próximos retos	110
8.16. Historial de retos del usuario	111

8.17. Eliminar inscripción del usuario de próximos retos	112
8.18. <i>Inicio > Ejercicio del mes</i>	113
8.19. Ranking del ejercicio del mes actual	113
8.20. Consultar historial de ejercicios del mes	114
8.21. Registrando cardio que afecta a la puntuación del ejercicio del mes	115
8.22. <i>Inicio > Rutinas y entrenamiento</i>	116
8.23. Viendo las rutinas de entrenamiento	117
8.24. Agregando rutina a favoritos y viendo tutorial de un ejercicio	117
8.25. Anotar rutina de entrenamiento	118
8.26. Consultar rutina de entrenamiento de otro día	119
8.27. <i>Inicio > Mi ficha personal</i>	120
8.28. Registrando la fecha de nacimiento	120
8.29. Modificando el género del usuario	121
8.30. Valoración del IMC del usuario	122
8.31. <i>Inicio > Soporte</i>	122

Índice de tablas

2.1. Tabla comparativa de aplicaciones	37
3.1. Malla receptora de información	50
4.1. Matriz de Tareas de Usuario	56

Capítulo 1

Introducción

1.1. Motivación

Nadie duda del gran crecimiento que ha tenido la actividad física y deportiva como parte de la vida de la sociedad en la que vivimos. Cada día más y más gente se anima a salir a andar por la calle, hacer senderismo, rutas en bicicleta, e incluso apuntarse al gimnasio. No solo la actividad física y deportiva ha impactado en nuestra sociedad, sino que también está cambiando la mentalidad de la gente en cuanto a la nutrición. Todos hemos oído hablar acerca del famoso reto del real food, lanzado por Carlos Ríos en el año 2017 [5]. En este reto, Carlos Ríos planteaba abandonar durante 30 días todos los alimentos ultraprocesados de nuestra dieta y experimentar el cambio que supone comer comida “real”. Esto quiere decir que durante 1 mes sólo vamos a poder alimentarnos de alimentos puramente reales, eliminando de nuestra dieta todos los productos ultraprocesados con gran cantidad de azúcares, aditivos y demás. El gran impacto de este reto en nuestra sociedad es evidente, pues así lo demuestran sus más de 2.000.000 de seguidores que ha conseguido Carlos Ríos a través de sus redes sociales desde que lanzó este gran proyecto [6].

Siguiendo un poco este contexto y volviendo a lo que se ha comentado inicialmente, nos encontramos en un momento de auge para las actividades físico-deportivas. De acuerdo al estudio de mercado realizado por Life Fitness en 2019 [1], España es actualmente el quinto país de Europa con más mercado fitness en cuanto a volumen de ingresos se refiere. Entre todos los encuestados, el deporte se valora en un 8 sobre 10. Curiosamente, de entre todas las personas encuestadas, las que más valoran el deporte son aquellas que acuden a un gimnasio y no aquellas que practican deporte de alguna otra forma. Aún podemos ver más datos que confirmen la importancia del deporte en la sociedad actual. La Figura 1.1 refleja las preferencias en cuanto a la práctica de deporte, y donde se aprecia que más del 65 % de los encuestados realizan su actividad física en un gimnasio.

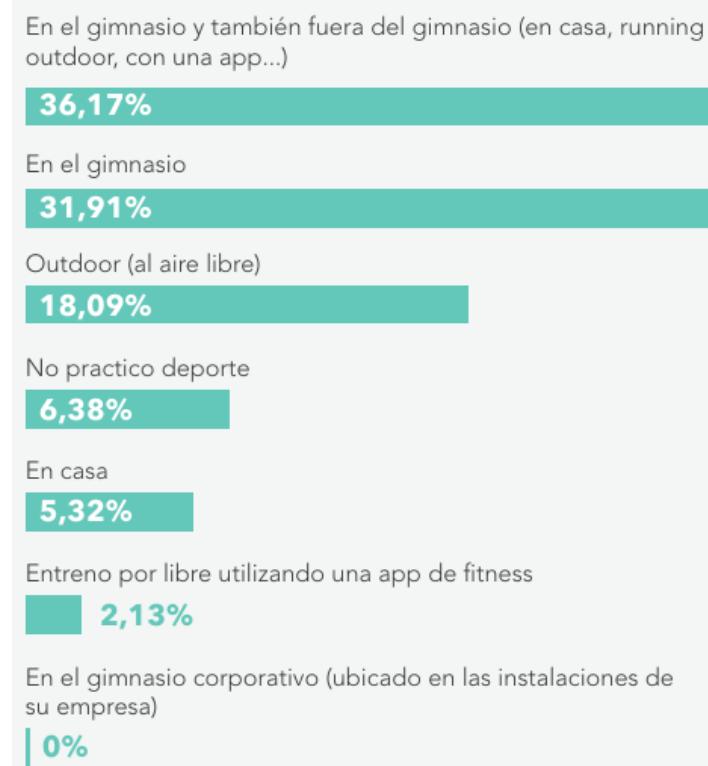


Figura 1.1: Preferencias en la práctica deportiva. Fuente: Life Fitness [1]

En el contexto de este proyecto nos planteamos la siguiente pregunta de investigación: ¿qué es lo que realmente motiva a la gente a apuntarse a un gimnasio? Pues siguiendo con el mismo estudio, las 3 razones principales por las cuales la gente decide apuntarse son las siguientes:

1. Encontrarse bien físicamente consigo mismo.
2. Preparación física para competiciones o trabajos.
3. El simple mero hecho de adelgazar.

Pudiera parecer que el motivo por el cual la gente acude al gimnasio es para ser feliz, pues en la sociedad actual se ha inculcado el pensamiento de que tener un cuerpo sano y fuerte es una buena razón para sentirse bien consigo mismo y con ello alcanzar un buen estado de salud mental y bienestar.

Dado que la demanda de usuarios en los gimnasios va cada vez más en aumento y por lo tanto la competitividad entre ellos para captar clientes es más fuerte, podemos hacernos otra pregunta: ¿qué puede ofrecer un gimnasio que le haga ser distinto a los demás y que sea la razón por la cual un cliente acepta ser socio? La respuesta es obvia dado en la época tecnológica en la que nos encontramos: el uso de la tecnología. Cada vez son más y más gimnasios los que tienen su propia APP que ofrecen un uso exclusivo a sus clientes. Y no es para menos, pues de acuerdo a la página web *statista* [7], en 2019 ya contábamos con más de 2.500 millones usuarios de **smartphones**; casi el doble de usuarios que en 2014.

Vivimos en una sociedad adicta al uso del **smartphone**. De hecho, la nomofobia es el miedo irracional a estar un intervalo de tiempo sin móvil. Hay algunos datos muy interesantes que cabe destacar [8]: un tercio de los españoles lleva el móvil consigo para el baño y un 75 % puede llegar a utilizarlo cuando mantiene relaciones sexuales para contestar notificaciones insistentes. Teniendo en cuenta estos datos, es obvio que uno de los mejores canales de comunicación entre un gimnasio y sus usuarios es, efectivamente, el smartphone. De acuerdo a Life Fitness [1], de todos los socios que acuden a un gimnasio que cuentan con una aplicación propia (por ejemplo, el gimnasio **BeOne** de Granada cuenta con su propia aplicación que está disponible tanto en **Google Play** como en **Apple Store**), el 70 % la tiene descargada. En la Figura 1.2 se muestran las funcionalidades que han tomado protagonismo en estas APPs y se compara su uso durante el año 2017 y 2018. Se puede ver que el gestionar rutinas y seguimientos con el entrenador se ha convertido en la mayor prioridad para un gimnasio.

1.2. Alcance

Este proyecto está pensado para llegar a todos los gimnasios y centros deportivos a nivel nacional. Esto puede aportar una gran innovación a sus centros que puede desembocar en el aumento de clientes. Al fin y al cabo vivimos en una sociedad en la que prima el boca a boca, es decir, si los clientes de un gimnasio están muy contentos por la incorporación de esta aplicación en su centro deportivo, muy probablemente lo recomendarán y animarán a más personas a hacerse socios o socias.

Este alcance se puede ver muy claramente en la situación vivida en 2020 debido al COVID-19. Debido al estado de alarma, todas las personas se han debido de quedar en casa y, además, todos los gimnasios y centros deportivos han sido cerrados. Esto se refleja en que todas las personas están buscando rutinas y ejercicios para hacer a través de internet. Si dichos gimnasios y centros deportivos contaran con una aplicación como **Imagym**, sus clientes podrían seguir haciendo rutinas y ejercicios a través del teléfono móvil u ordenador y podría haber una comunicación indirecta entre los monitores y

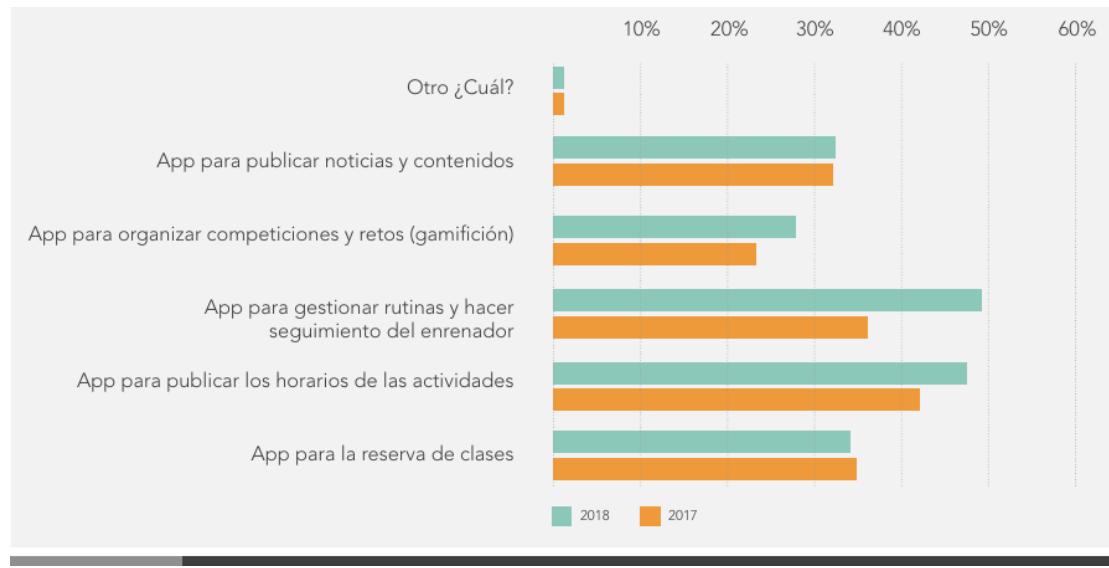


Figura 1.2: Funcionalidades más usadas los gimnasios que cuentan con aplicación propia. Fuente: Life Fitness [1]

clientes a pesar de estar todos en casa. Además, se incorporarían más socios al centro, ya que el pago de cuota (más reducida, obviamente) se puede hacer online. Esto desemboca en que los gimnasios y centros con **Imagym** pueden seguir ofreciendo un servicio a sus clientes y pueden seguir acogiendo a nuevos. En cambio, un centro que no cuente con esta aplicación no puede seguir dando su servicio y no obtiene beneficios.

Pero si ahora mismo todos los servicios deportivos se hacen online (Facebook, Instagram...), ¿qué tiene de diferente **Imagym**? Entre otras muchas cosas, se ofrecen retos y desafíos que permiten la competición entre todos los usuarios, búsqueda de rutinas de forma más sencilla (hay un chatbot), y competición con uno mismo en distintas actividades cardiovasculares. Todo esto y mucho más se verá más adelante en el Capítulo 3.

1.3. Objetivos generales

El principal objetivo de este proyecto es ofrecer una aplicación al mercado que verdaderamente sea **fiable** para todos los usuarios y que además consiga **unir** a los usuarios de la comunidad de gimnasios y centros deportivos. Actualmente estamos en un punto

en el que hay miles y miles de aplicaciones que ofrecen prácticamente lo mismo: ejercicios para hacer deporte en casa y guía de nutrición. El problema es que los usuarios no saben la fiabilidad de estos datos y todos estamos de acuerdo en que cada persona tiene una forma física y un metabolismo diferente, con lo cual no existe la guía definitiva de alimentación o de rutinas.

El mayor miedo que tienen las personas al apuntarse al gimnasio es ir a solas, sin compañía de nadie. Esto se debe a que sienten vergüenza de equivocarse al hacer ejercicios, o tienen miedo de no socializar con nadie. Con **Imagym** se pretende que entre los mismos usuarios de un gimnasio o centro deportivo haya una competitividad sana, permitiendo una comunicación indirecta entre ellos y facilitando la comunicación directa cuando se encuentren en el centro.

También es cierto que en un mismo gimnasio hay distintos monitores y cada uno de ellos puede tener más conocimiento o técnica en unas rutinas u otras. Con este proyecto se pretende dar visibilidad a cada uno de estos monitores, pudiendo ellos ofrecer sus propias rutinas a los socios del gimnasio. De esta manera, son los propios usuarios los que eligen las rutinas que quieren hacer, pudiendo alternar rutinas de un monitor u otro en distintos días de la semana. Así mismo, permite crear una competitividad sana entre monitores para ver quién consigue más seguidores en su rutina.

Capítulo 2

Estado del arte

En el Capítulo 1 hablábamos de que muchos de los gimnasios y centros deportivos actuales cuentan con una APP propia para ofrecer diversas funcionalidades a sus usuarios. Hoy en día contamos con un enorme número de aplicaciones tanto en **Google Play** como **Apple Store**, cada una ofreciendo una funcionalidad diferente: hacer ejercicio en casa, contar calorías quemadas, registro de calorías ingeridas a lo largo del día o incluso hacerse una dieta propia, entre otras funcionalidades. Es por ello que no solo debemos hablar de las APP "nativas" de los gimnasios, si no de aquellas que nos ofrecen en los **smartphones**, tanto **Android** como **iOS**.

Las aplicaciones más usadas por los usuarios son aquellas relacionadas con el control de peso y el control de la alimentación [9]. Para poder profundizar en este asunto, a continuación se presenta una investigación de mercado sobre las aplicaciones que podemos descargar en nuestro móvil que tienen que ver con todo lo hablado anteriormente.

2.1. Metodología de análisis

Esta investigación de mercado trata de responder a la pregunta más importante: ¿cómo de **fiables** son las aplicaciones que tanta gente usa para llevar un estilo de vida saludable? es interesante ver la enorme cantidad de aplicaciones disponibles y que cada una de ellas ofrece un servicio u objetivo diferente. De acuerdo a este artículo [10] de marzo de 2020, hay más de 250.000 aplicaciones en la categoría de **Health** en las tiendas de aplicaciones. En Febrero de 2020 utilicé **Google Play** para buscar aplicaciones sobre este tema. Al introducir la búsqueda **Gimnasio** me aparecieron una gran cantidad de

aplicaciones (**Google Play** ya no muestra el número de resultados de la búsqueda), y me quedé con las 5 primeras ya que el algoritmo prioriza las aplicaciones de calidad de acuerdo a tres factores: experiencia del usuario, estabilidad y calidad del listado [11]. Estas 5 aplicaciones son las siguientes:

- Fitify - Entrenamientos y planes
- Gym WP - Ejercicios y Rutinas para el gimnasio
- BestFit - Rutina Gimnasio, Entrenamiento y Fitness
- Fitvate - Gimnasio Rutina de ejercicio Entrenador
- Total Fitness - Rutinas y Gimnasio

Debido a la situación actual de estado de alarma, se ha incrementado el uso de aplicaciones de este estilo. Esto es una buena oportunidad para ver qué es lo que se está ofreciendo a los usuarios en una situación como esta. En esta investigación de la web *statista* [12] se presenta el top 10 de las aplicaciones más descargadas en marzo de 2020 (categoría salud y fitness). De este top 10 me he quedado con 3 aplicaciones en concreto porque las 7 restantes contienen contenido muy específico: solo es para mujeres, solo es para hombres, solo contiene ejercicios para abdominales, el entrenamiento es solo para 30 días... O incluso la aplicación *Coronavirus - SUS* que es la segunda más descargada, su contenido no tiene nada que ver con rutinas y entrenamientos, si no que ofrece consejos de salud y situación actual del COVID-19 Las 3 aplicaciones seleccionadas son las siguientes:

- Home Workout - No equipment
- Samsung Health
- Muscle Booster - Fitness at home

Es muy importante contar con el apoyo de profesionales de mundo del deporte en este tipo de mercado, pues ellos realmente cuentan con el conocimiento necesario para darle fiabilidad a todas las funcionalidades que incluyen las aplicaciones. Esta investigación de mercado recoge las funcionalidades ideales que en mi opinión una aplicación de este estilo debe tener y que no implican que el usuario deba tener unos conocimientos previos para poder utilizarla **adecuadamente**. Además, estas funcionalidades **SÍ** están presentes en **Imagym**. Los criterios a tener en cuenta son los siguientes:

1. Control de peso, IMC y/o grasa del usuario

2. Establecer objetivo de peso, masa muscular y/o grasa del usuario
3. Seguimiento de la evolución del peso, IMC y/o grasa del usuario
4. Registro de actividades de cardio (minutos, intensidad, distancia...)
5. Establecer objetivo personal de actividades cardio
6. Retos o desafíos de ejercicios propuestos por profesionales
7. Entrenamientos ofrecidos al usuario de distintos grupos musculares
8. La anotación del entrenamiento no requiere la intervención del usuario tras cada ejercicio
9. Rutinas establecidas por profesionales
10. Objetivos de actividades cardio propuestas por profesionales
11. Competición y clasificación entre los usuarios en los distintos retos o desafíos
12. No requiere cuenta en Google, Facebook, E-mail, o cualquier otra plataforma externa para acceder a la aplicación
13. No requiere hacer un test inicial para empezar a usar la aplicación
14. Todo el contenido es gratuito
15. Tiene contenido gratuito
16. No tiene contenido de pago
17. No contiene anuncios
18. Plataforma en la que se ofrece

2.1.1. Fitify - Entrenamientos y Planes

Fitify [13] cuenta con una puntuación media de 4.7 sobre 5 estrellas, más de 12 mil opiniones y más de 1 millón de descargas. Está disponible tanto en **Android** como en **iOS**. En principio parece que estamos ante una aplicación bastante aceptable y que, como se vio anteriormente la Sección 2.1, el algoritmo **Google Play** ha detectado de buena calidad. Para empezar, la aplicación nos pide entrar con una cuenta o empezar si no tenemos una. Yo no tengo cuenta, con lo cual voy a empezar desde cero tal y como se muestra en la Figura 2.1.

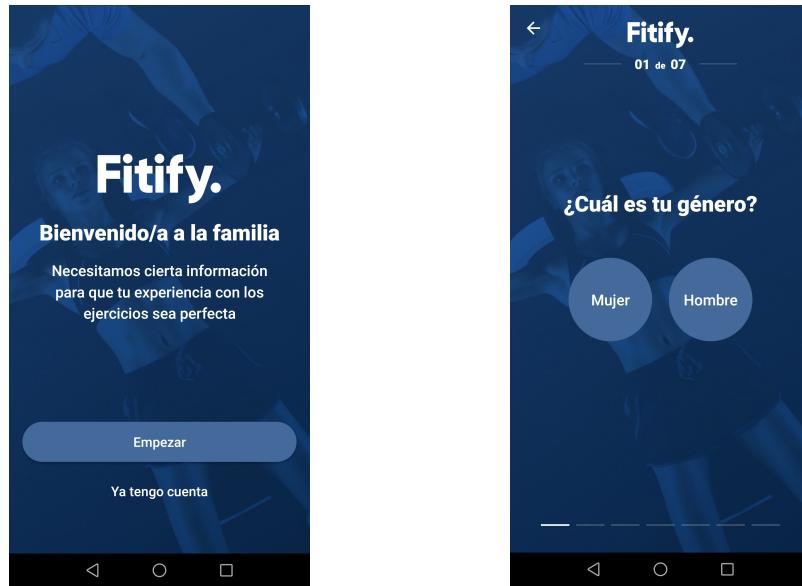
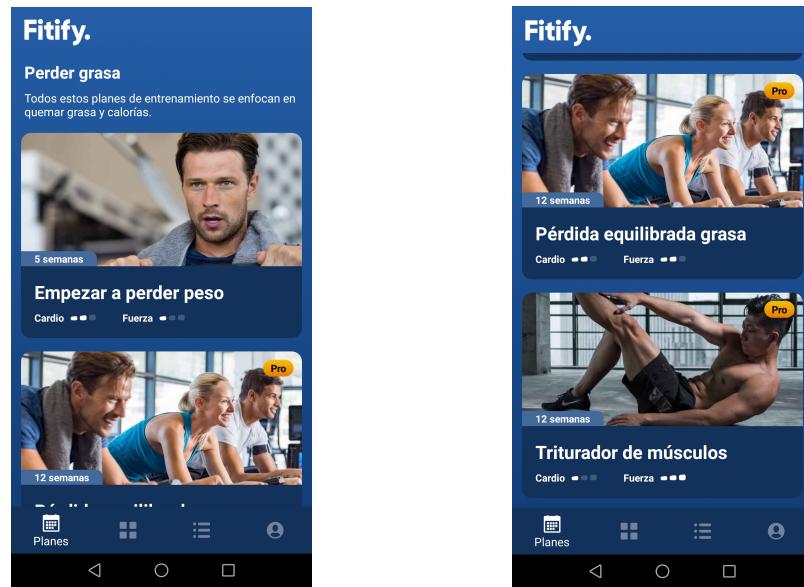
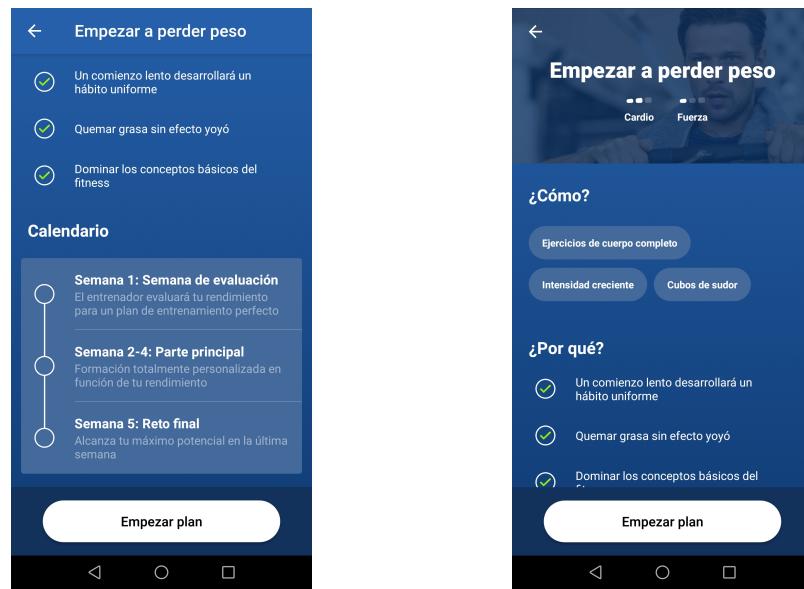


Figura 2.1: Pantalla de inicio y test inicial

Fitify nos hace un pequeño cuestionario de siete preguntas sobre información personal que es obligatorio para poder acceder a la aplicación: género, objetivo (perder grasa, ponerse en forma o desarrollar músculo), forma actual (bastante en forma, en forma, no muy en forma), años, altura, peso, y si deseamos obtener ofertas especiales y consejos fitness por correo electrónico (no se indica qué tipo de ofertas podemos recibir: meses de descuento, descuentos en alimentación, descuento en la suscripción...). A continuación, se nos ofrece un método de registro: Google, Facebook, Apple o E-mail. Es obligatorio tener cuenta en alguna plataforma para acceder al contenido. Una vez registrado, aparece un *Timeline* con diferentes tipos de entrenamientos. Esto se muestra en la Figura 2.2. Estos entrenamientos que se ven en la pantalla principal se dividen únicamente en tres subcategorías: perder grasa, ponerse en forma y desarrollar músculo.

Lo primero que me ha llamado la atención es que en cada subcategoría hay un primer programa que es totalmente gratis pero todos los demás programas pertenecen a la versión Pro de la aplicación, es decir, requiere una suscripción. Esta suscripción puede ser mensual (6.99 €/mes) o anual (59,99 €/año). En el caso de la suscripción anual se regalan 7 días de prueba gratuita. Para hacer una prueba, he seleccionado el programa *Empezar a perder peso*. Figura 2.3

Como se puede ver en la Figura 2.3, se muestra una pequeña descripción con la intensidad del programa y un calendario con un resumen de lo que se hará durante 5 semanas. Al darle a empezar, se muestra una barra de progreso y el día que toca. Al seleccionar el día se nos muestra el entrenamiento de hoy. Para una versión básica

Figura 2.2: *Timeline* principalFigura 2.3: Programa *Empezar a perder peso*

de la aplicación (gratuita), solo se puede seleccionar un entrenamiento. La versión Pro cuenta con más tipos de ejercicios o entrenamientos. En mi opinión, 5 semanas puede ser demasiado para los usuarios pues supone más de 1 mes de entrenamiento. Normalmente los gimnasios proponen rutinas durante 1 mes. Al empezar el entrenamiento, aparecen todos los ejercicios que componen el entrenamiento con el tiempo que dura cada uno. En la Figura 2.4 se muestra la interfaz de la cuenta atrás que dura un ejercicio.

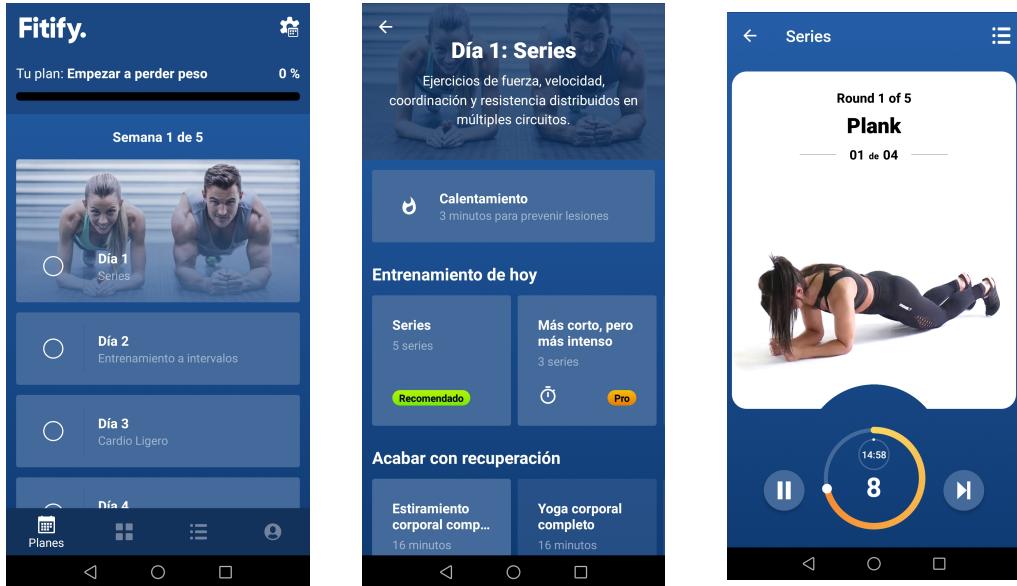


Figura 2.4: Interfaz de entrenamiento

Adicionalmente, Fitify cuenta con ejercicios independientes. Se pueden hacer ejercicios de diferentes categorías como abdominales, cuerpo superior, cuerpo inferior, etc. Simplemente se muestra un pequeño vídeo mostrando cómo se hace un determinado ejercicio, tal y como se muestra en la Figura 2.5. En resumen, esta aplicación cuenta con entrenamientos propuestos por la aplicación que pueden durar más de 1 mes y entrenamientos libres para el usuario.

2.1.2. Gym WP - Ejercicios y Rutinas para el gimnasio

Gym WP [14] cuenta con una media de 4.8/5 estrellas, 42 mil opiniones y más de 1 millón de descargas en **Android** (no está disponible en **iOS**). De acuerdo a su descripción, esta aplicación está destinada a personas que buscan una pérdida de peso, hipertrofia o monitoreo de su condición física.

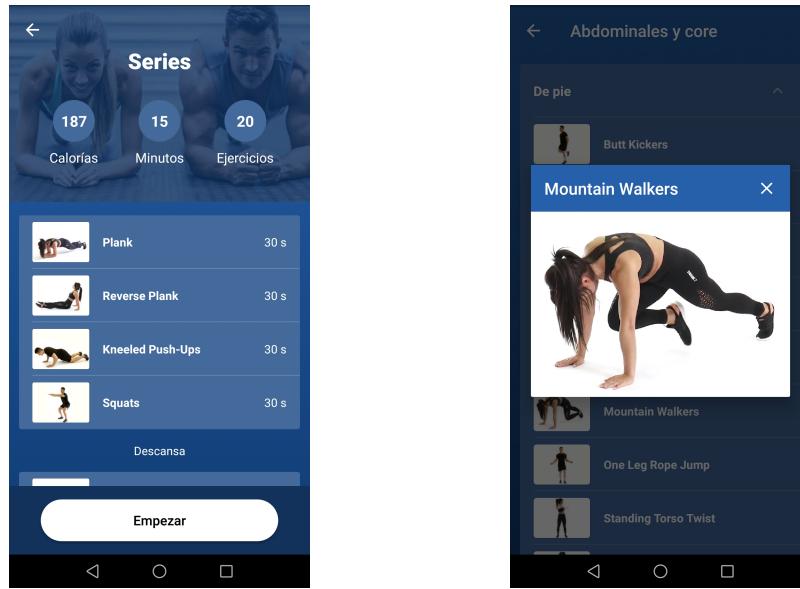


Figura 2.5: Interfaz de entrenamiento libre

Al iniciar la aplicación nos lleva directamente al menú de inicio, con lo cual no hay registro. Para ver cómo gestionan las cuentas de los usuarios, he entrado en el apartado *Mi cuenta* y aparezco como usuario anónimo, pero con opción de iniciar sesión. Este inicio de sesión sólo se puede hacer a través de una cuenta Google y sirve únicamente para una copia de seguridad de los datos, como se muestra en la Figura 2.6.

Siguiendo un poco la aplicación, en la Figura 2.7 se muestra el menú de inicio en el que se puede llevar un seguimiento del peso diario de los usuarios. Además, se indica el IMC, tasa de grasa y peso ideal. También se dispone de una evaluación física en la que se pueden introducir hasta 20 campos de medidas corporales. Desde mi punto de vista, esta evaluación física no es muy fiable, ya que para ello el usuario debe saber medir correctamente cada parte del cuerpo y además el usuario debe saber los nombres específicos de cada una de ellas.

También se observa que la aplicación es gratuita pero tiene publicidad que ocupa gran parte de la pantalla. Esta publicidad se puede quitar pagando una suscripción. Esta suscripción también se ve reflejada a la hora de hacer las rutinas de entrenamiento. En la Figura 2.8 se muestra los distintos entrenamientos que se ofrecen a los usuarios. Yo he seleccionado la rutina de piernas para hacer una pequeña prueba, y al entrar nos damos cuenta de que solamente se ofrecen dos entrenamientos gratuitos. Si el usuario quiere más entrenamientos, debe pagar la suscripción.

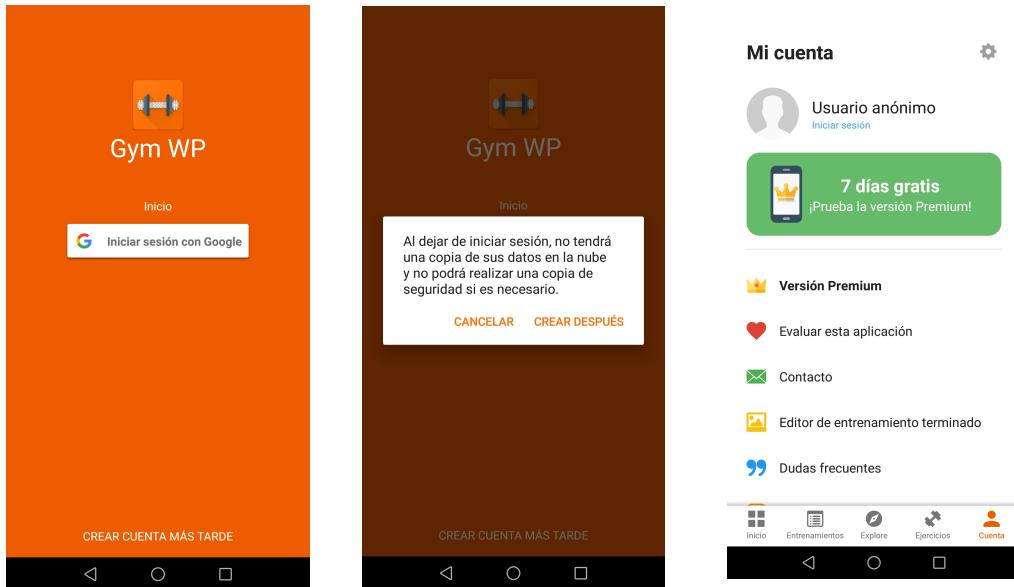


Figura 2.6: Interfaz de inicio

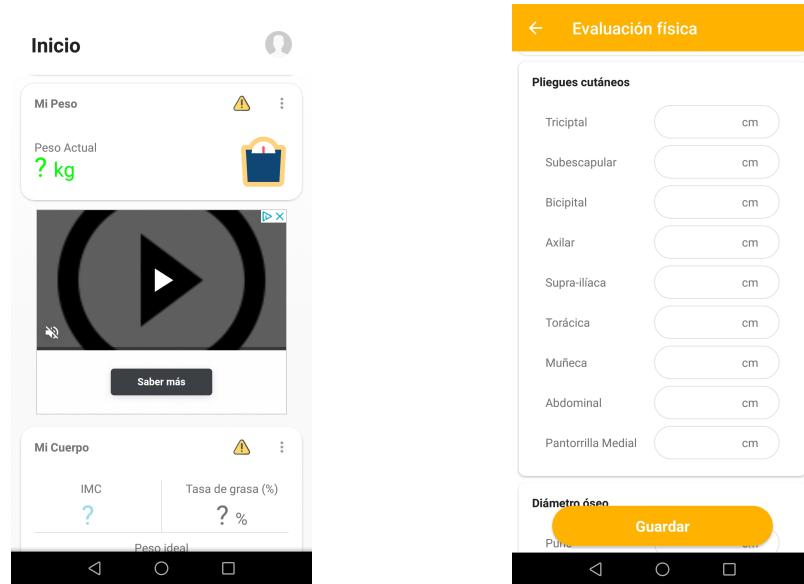


Figura 2.7: Menú de inicio y seguimiento de peso diario

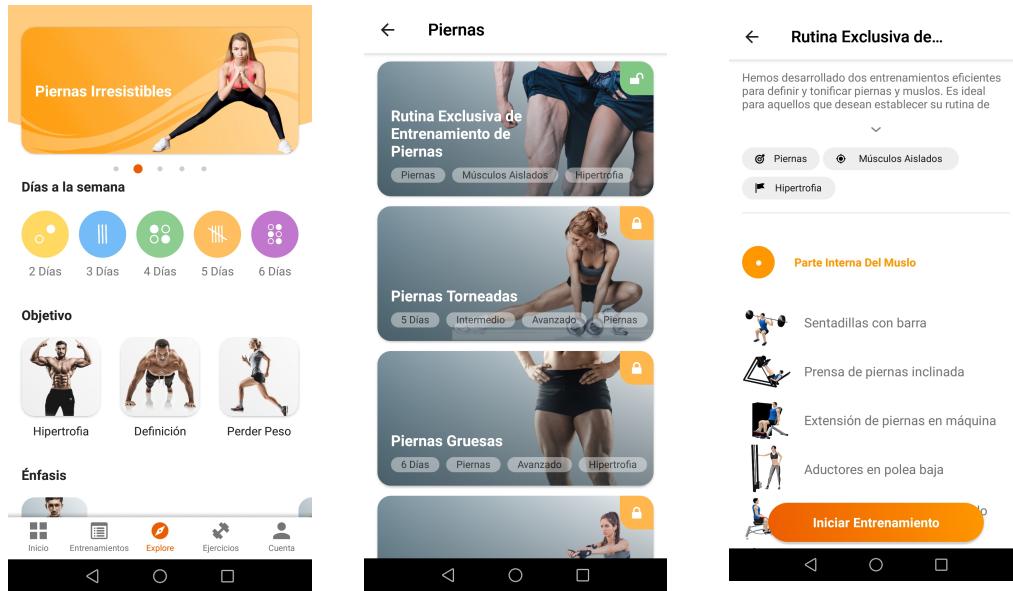


Figura 2.8: Planes de entrenamiento

2.1.3. BestFit: Rutina Gimnasio - Entrenamiento y Fitness

BestFit [15] está disponible tanto para **Android** como para **iOS**. Se define como un gimnasio entrenamiento con ejercicios, pesas y métodos adaptados para los usuarios, y en **Play Store** cuenta con una valoración de 4.3/5 estrellas.

Al entrar en la aplicación nos aparece una pantalla inicial donde se puede elegir entrar con registro o sin registro. En mi caso, he seleccionado sin registro. Se deben introducir unos datos iniciales básicos como el género, edad, altura y peso. Una vez hecho esto, nos lleva al menú de inicio donde nos dice que tenemos un regalo. Este regalo es 7 días de prueba gratuita para la versión premium de la aplicación, tal y como se muestra en la Figura 2.9.

Después de obtener la versión gratuita, únicamente nos deja crear un nuevo plan de entrenamiento. Debemos de elegir el objetivo de nuestro entrenamiento (ver Figura 2.10), la frecuencia de entrenamiento, la duración del entrenamiento y los músculos que queremos ejercitar. Una vez seleccionado todo, nos muestra un resumen de nuestra selección para confirmar que todo está bien.

Cuando iniciamos el entrenamiento que la aplicación ha generado para nosotros, se nos muestran los ejercicios que debemos hacer y en la parte superior un cronómetro

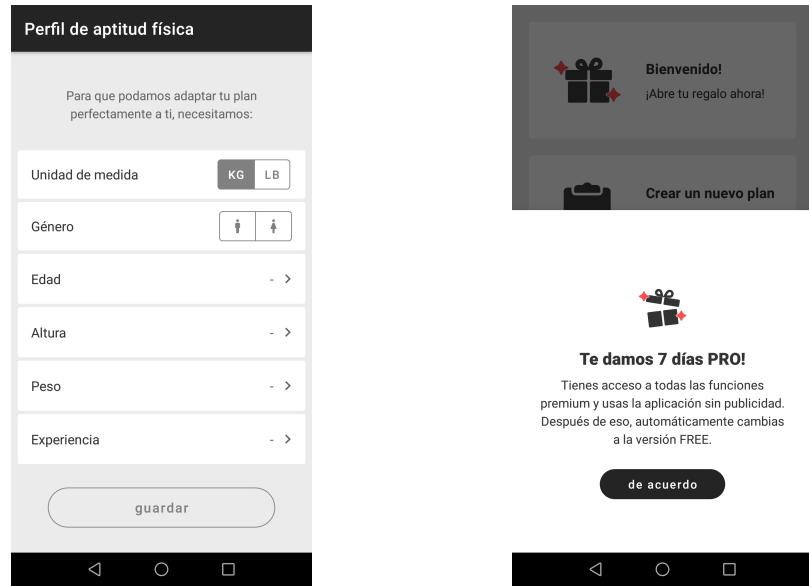


Figura 2.9: Interfaz de inicio y regalo de prueba gratuita

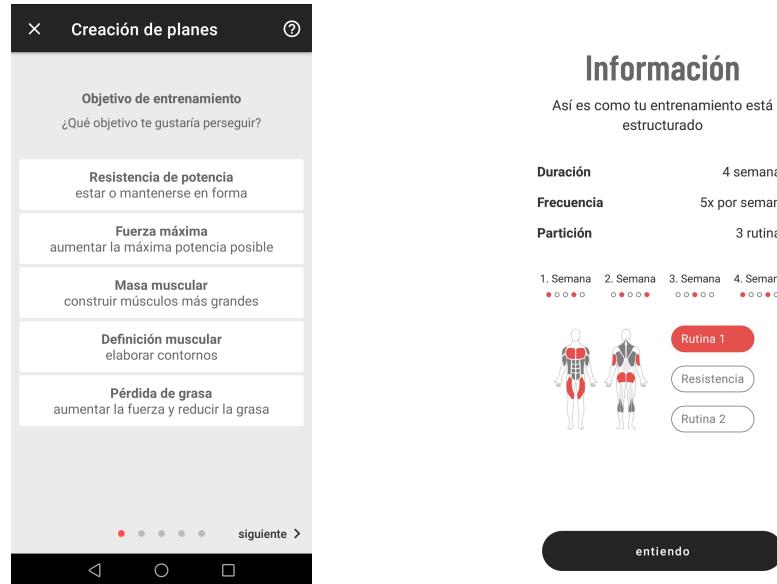


Figura 2.10: Creación de un plan de entrenamiento

con el tiempo que llevamos. Cuando seleccionamos un ejercicio en concreto, nos muestra un vídeo con la explicación del ejercicio, sugerencias de peso y campos para introducir el peso que vamos a coger (ver Figura 2.11). Este método es algo engoroso, ya que el usuario debe introducir el peso que va a coger cada vez que va a realizar un ejercicio. Esto hace que el usuario pase demasiado tiempo introduciendo datos en la aplicación y puede hacer que el usuario deje de utilizar la aplicación y busque otra que no lleve un proceso tan repetitivo.

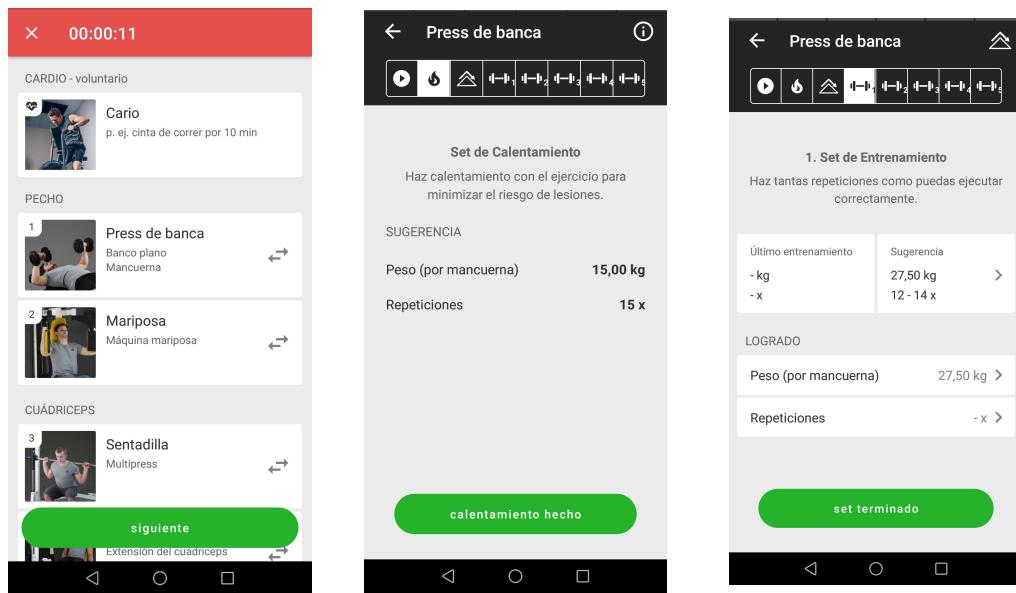


Figura 2.11: Proceso para realizar un entrenamiento

2.1.4. Fitvate - Gimnasio Rutina de ejercicio Entrenador

Fitvate [16] está disponible tanto para **Android** como para **iOS**. Se define como una aplicación de planificación de ejercicios de gimnasio para móvil y en **Play Store** cuenta con una puntuación de 4.7/5 estrellas, más de 11 mil opiniones y más de 1 millón de descargas.

Al iniciar la aplicación nos lleva directamente a los ejercicios. Los ejercicios se dividen en categorías según las distintas partes del cuerpo (ver Figura 2.12).

Dispone de una pestaña para los planes de entrenamiento. Se puede crear un plan propio de entrenamiento o elegir uno. El problema es que, al igual que en las aplicaciones anteriores, se muestra un programa de entrenamiento gratuito y todos los demás son

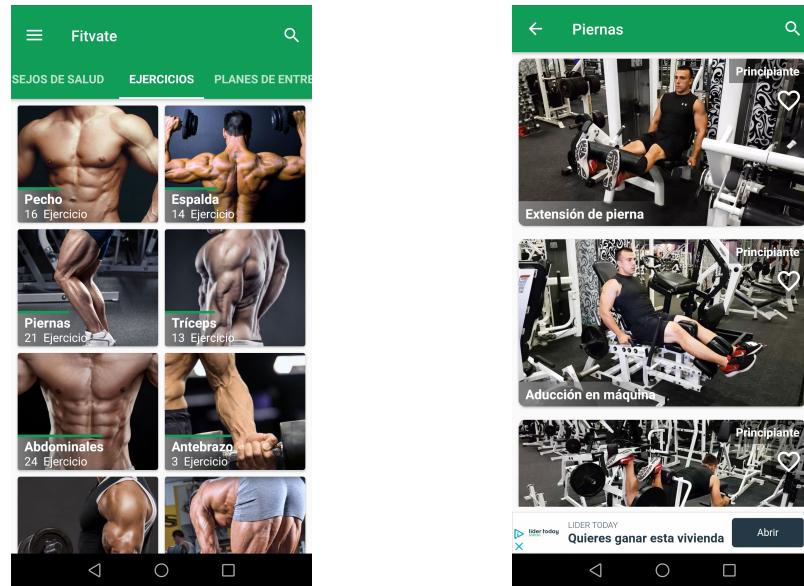


Figura 2.12: Pantalla inicial de la aplicación

de pago (hay que tener una suscripción premium), además de contener anuncios. Al seleccionar un programa de entrenamiento se nos muestra un resumen de los días de entrenamiento durante las semanas que dura. Al seleccionar un día, nos muestra los ejercicios asociados a ese día y una casilla para marcar que lo hemos hecho (ver Figura 2.13).

Algo nuevo que incorpora esta aplicación con respecto a las anteriores son los desafíos. El desafío que se puede hacer gratuitamente es uno de 300 sentadillas. En él, se establecen 15 semanas donde el usuario debe realizar cada día un número determinado de repeticiones, intercalando días de entrenamiento con días de descanso. Cada desafío es con uno mismo, es decir, no permite competir con los demás usuarios. Cada vez que se marca un día como "hecho" (ver Figura 2.14) se añade un porcentaje al desafío, así hasta completar el 100 %.

2.1.5. Total Fitness - Rutinas y Gimnasio

Total Fitness [17] se define como un **personal trainer** y está disponible tanto para **Android** como para **iOS**. De acuerdo a su información, dispone de las mejores rutinas para ponerse en forma y propone una guía de Fitness para los aficionados al gimnasio. En **Play Store** cuenta con una valoración de 4.5/5 estrellas, 57 mil opiniones y más de 1 millón de descargas.

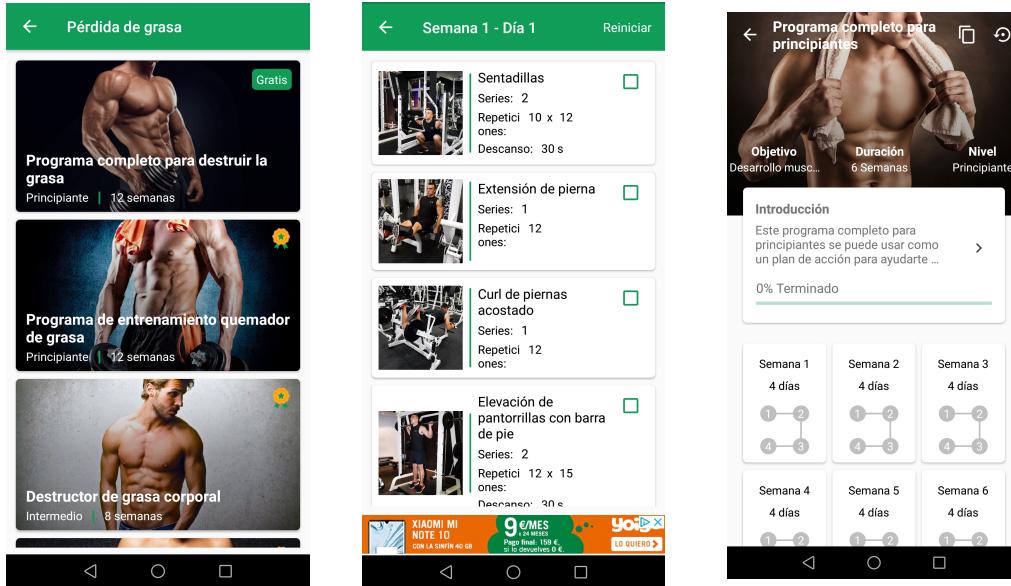


Figura 2.13: Realizaci ́on de un programa de entrenamiento

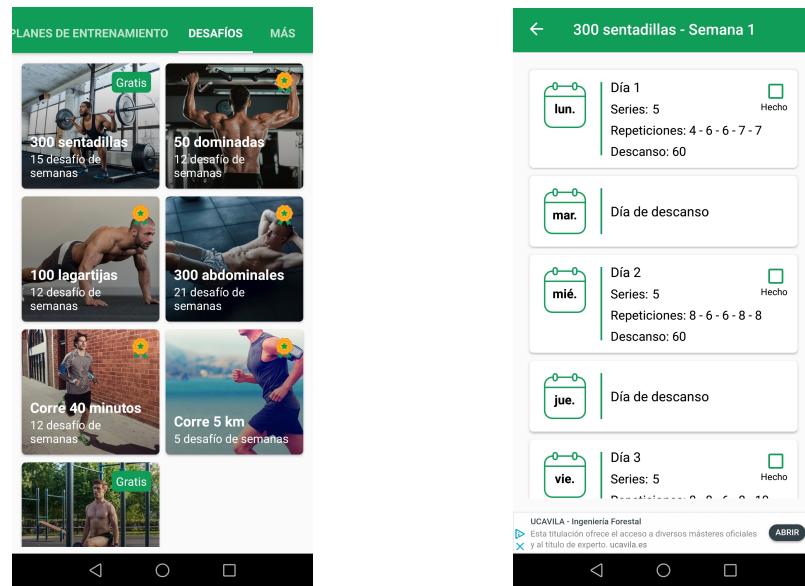


Figura 2.14: Desaf ́o de sentadillas

Inicialmente esta aplicación nos muestra una pantalla de inicio donde es obligatorio iniciar sesión mediante Google o Facebook. Una vez iniciada la sesión, se puede ver que podemos seleccionar la rutina del mes (en este caso Marzo) pero que solo está disponible para la versión premium de la aplicación (ver Figura 2.15).

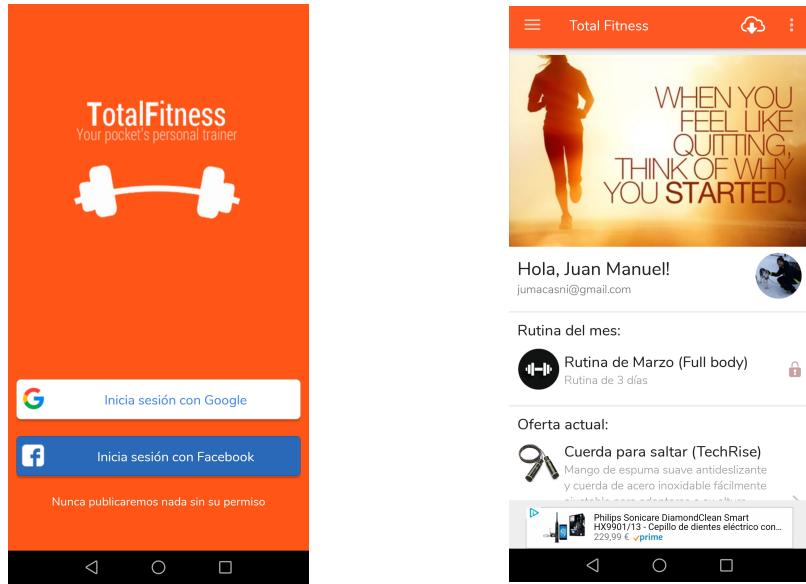


Figura 2.15: Inicio de sesión de la aplicación

En el menú podemos encontrar las secciones de *Rutinas*, *Retos*, *Nutrición* y *Utilidades*. En la pestaña de rutinas podemos encontrar diferentes programas de entrenamiento. De nuevo, al igual que en todas las aplicaciones anteriores, se muestran unos programas para la versión gratuita pero la mayoría de ellos se desbloquean al comprar la suscripción premium (ver Figura 2.16).

Al seleccionar un programa, se muestra un resumen de los días que componen el entrenamiento, así lo refleja la Figura 2.17, y al seleccionar un día nos muestra una tabla de ejercicios con un marcador que se puede seleccionar cuando hayamos hecho un ejercicio. Además, cuenta con un cronómetro en la parte superior para contar el tiempo que llevamos haciendo la rutina.

Total Fitness cuenta con una sección de retos muy similar a la que había en Fitvate (ver Sección 2.1.4). De hecho, el primer reto es de 300 sentadillas que es exactamente igual que el reto de prueba que había en Fitvate. En otra sección, la de *Utilidades*, podemos llevar un control de nuestro peso diario, así como de IMC y porcentaje de grasa (ver Figura 2.18).

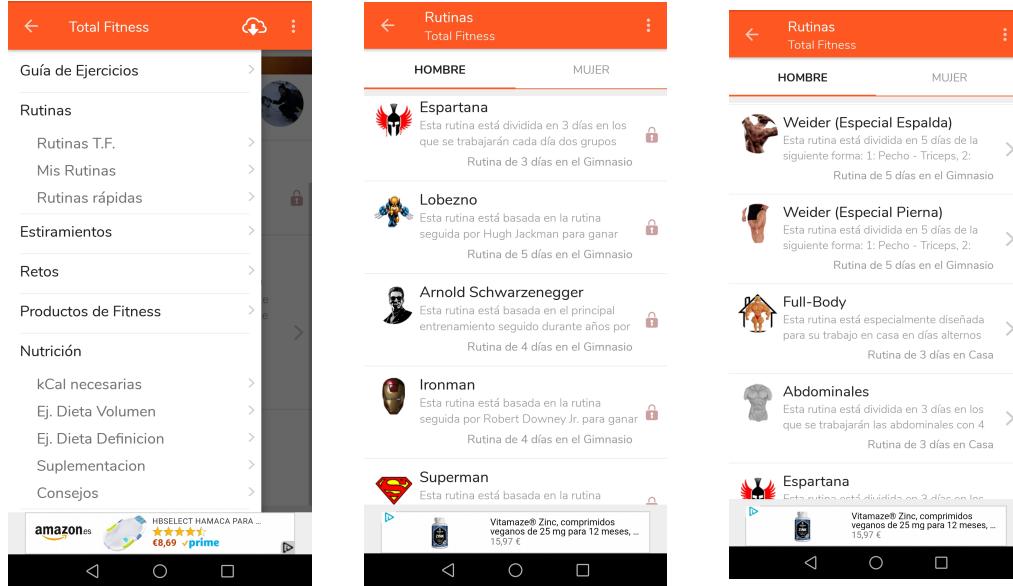


Figura 2.16: Menú y programas de entrenamiento

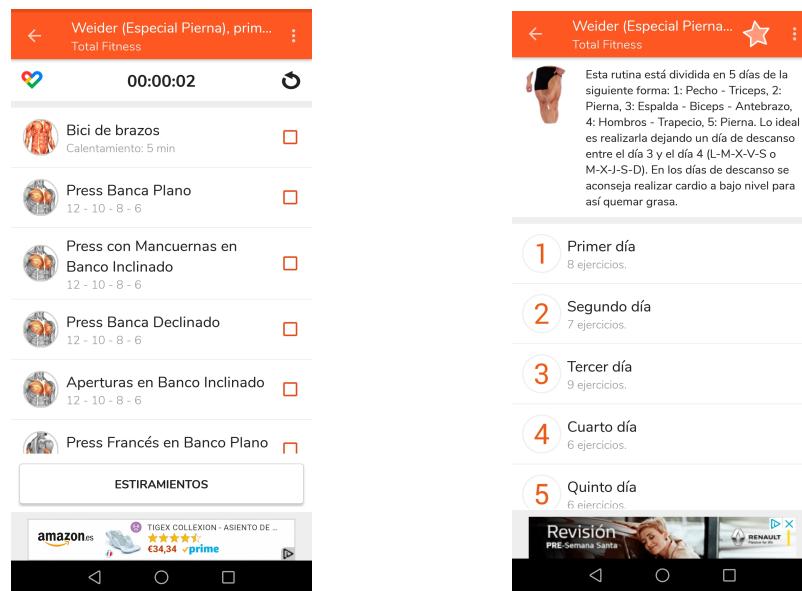


Figura 2.17: Tabla de ejercicios y cronómetro

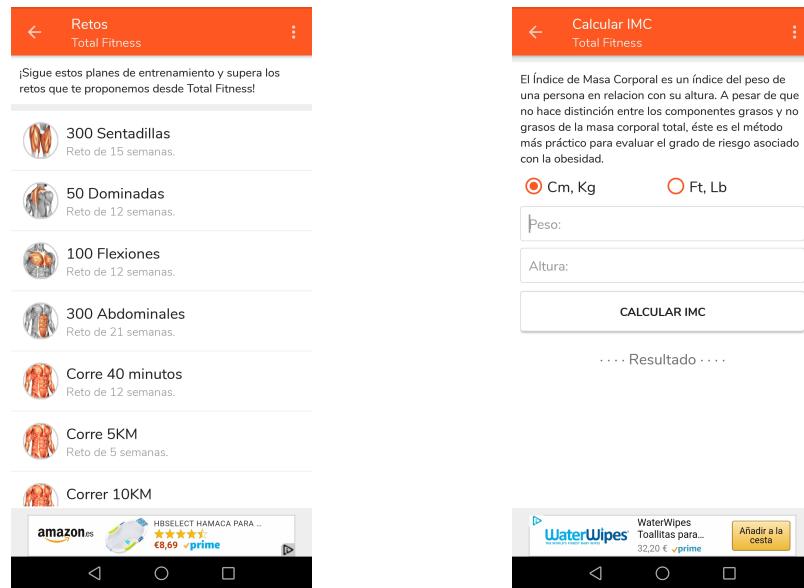


Figura 2.18: Retos y control de peso/IMC/grasa

2.1.6. Home Workout - No equipment

Home Workout [18] está disponible tanto en **Android** como en **iOS**. En Play Store tiene una puntuación de 4.8/5 estrellas y más de 50 millones de descargas. Forma parte del top 10 de aplicaciones descargadas en Marzo de 2020, siendo esta la número 1 [12], tal y como se dijo en la Sección 2.1.

Al entrar en la aplicación nos lleva directamente al *Timeline* principal y no nos pide ningún registro (ver Figura 2.19). Lo primero que me ha llamado la atención es que una llamada a la acción con el mensaje: *Establecer un objetivo*. También se ve que hay desafíos y entrenamientos por nivel, habiendo tres niveles: principiante, intermedio y avanzado.

Al pulsar el botón de *Establecer un objetivo*. Debemos seleccionar los días de entrenamiento por semana y el día de semana que queremos empezar. La configuración que he puesto se muestra en la Figura 2.20. Esto no ha supuesto mucho cambio, pues simplemente donde antes aparecía el botón ahora aparece un mini calendario con el día actual. Al seleccionar un día, nos lleva a un historial registrado de ese día en el que se nos muestra el número de ejercicios, calorías y minutos hechos, el peso e IMC de ese día y una gráfica con la evolución del peso.

En cuanto a los desafíos, he seleccionado el primer desafío que aparecía en la interfaz

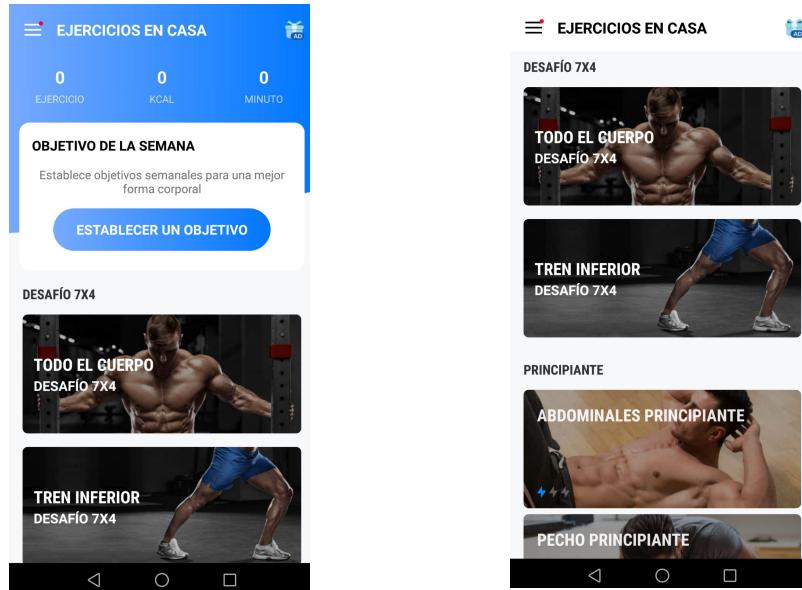


Figura 2.19: Inicio de Home Workout

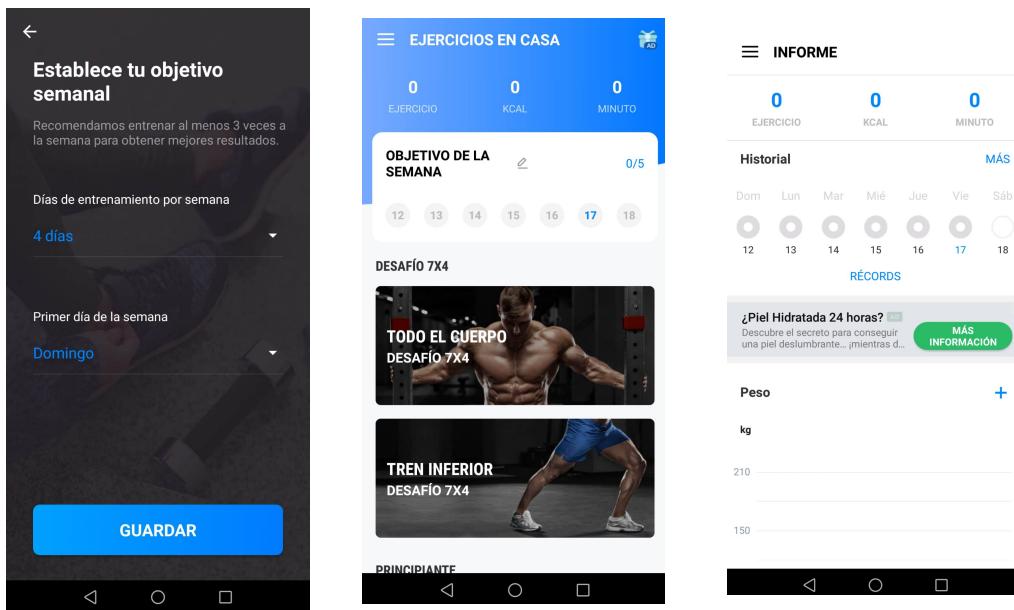


Figura 2.20: Establecimiento de objetivo y seguimiento de peso/IMC

principal, llamado *Todo el cuerpo* (Figura 2.21). Al entrar vemos una especie de calendario dividido en 4 semanas y el número de días restantes que nos faltan (acompañado de un porcentaje que irá subiendo conforme vayamos completando días). Al empezar el primer día nos indica los ejercicios que debemos hacer, el tiempo o repeticiones que debemos de aguantar o hacer. Al empezar el entrenamiento, se nos muestra un vídeo de muestra de cómo hacer el ejercicio y un contador hacia atrás. El problema que encuentro es que el contador empieza de manera automática, es decir, no deja margen para que el usuario suelte el móvil y empiece el ejercicio. Sin embargo, algo bueno que tiene es que utiliza la voz para indicar al usuario el ejercicio que debe hacer. Esta interfaz de desafíos es exactamente igual para los entrenamientos.

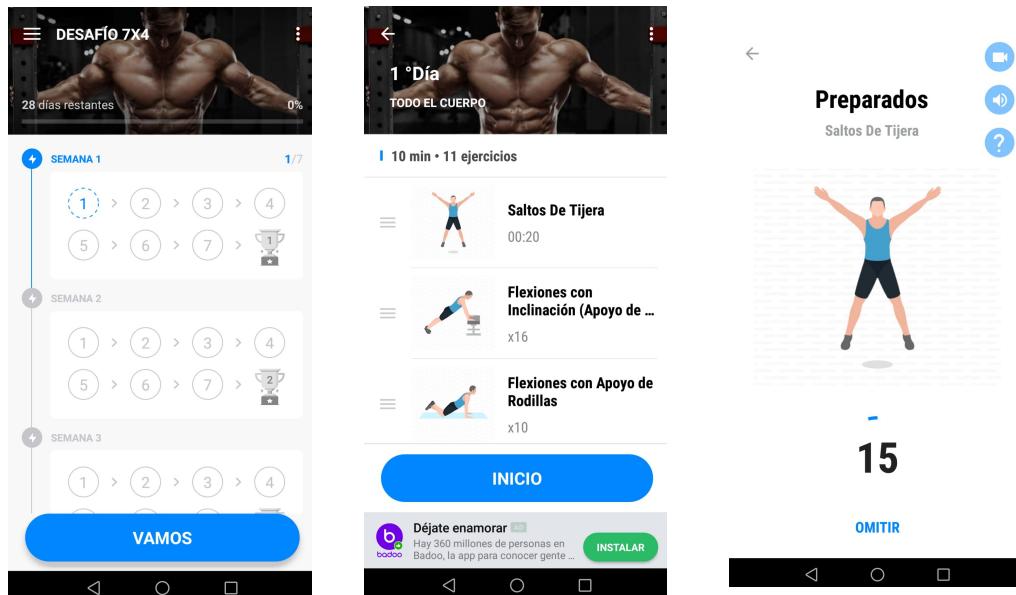


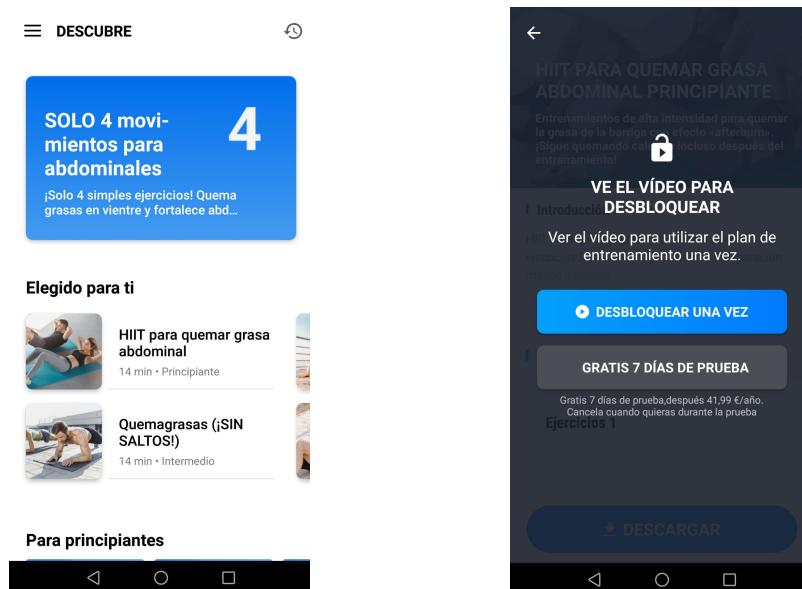
Figura 2.21: Calendario de entrenamientos y ejecución

En la sección de *Descubre* se pueden encontrar entrenamientos y desafíos más específicos (HIIT, Tabata, cardio sin saltos...), pero todos ellos requieren una suscripción, aunque podemos adquirir una prueba gratuita durante 7 días (Figura 2.22).

2.1.7. Samsung Health

Samsung Health [19] es la tercera aplicación más descargada en marzo de 2020 [12] con casi 2 millones de descargas (solo en marzo). Está disponible tanto para **Android** como para **iOS**, y en **Play Store** cuenta con una puntuación de 4.3/5 estrellas.

La pantalla inicial de esta aplicación nos ofrece datos de los pasos actuales que

Figura 2.22: Sección *Descubre*

llevamos (no hay registro inicial ni preguntas iniciales). También se puede seleccionar un entrenamiento, pero sólo de actividades cardio al aire libre (correr, andar, bicicleta...). Podemos llevar un registro de calorías ingeridas, un registro de horas de sueño (que se introduce manualmente), un seguimiento de peso y los vasos de agua que llevamos en el día (Figura 2.23).

Me ha parecido muy interesante la sección del contador de calorías de comida. Cuando el usuario quiere añadir una comida, se le muestran las distintas comidas (desayuno, almuerzo, merienda...) y un máximo de calorías que debe consumir (Figura 2.24). Este máximo lo ha calculado la aplicación y de hecho ese número es muy general ya que aún no he dado mis datos personales de edad, peso, etc. y seguramente ese no sea el número. Además, esas calorías pueden variar dependiendo de si mi objetivo es perder grasa o ganar músculo. Cuando se quiere registrar una comida se nos muestran una serie de alimentos (Figura 2.24) o podemos crear una comida nosotros. Desde mi experiencia, este seguimiento conlleva demasiado tiempo a los usuarios ya que una comida puede estar formada por muchos alimentos. Si el usuario debe registrar todos y cada uno de los alimentos con sus respectivas cantidades y calorías llegará un momento en el que se canse. Además, resulta tedioso el hecho de mirar el número exacto de gramos o de tazas de cada alimento. En conclusión, este seguimiento de calorías lo haría acudiendo a un nutricionista y no lo dejaría en manos de una aplicación.

En cuanto a los entrenamientos que puedo registrar, he seleccionado la actividad de andar. Se pueden establecer distintos objetivos, mostrados en la Figura 2.25. El problema

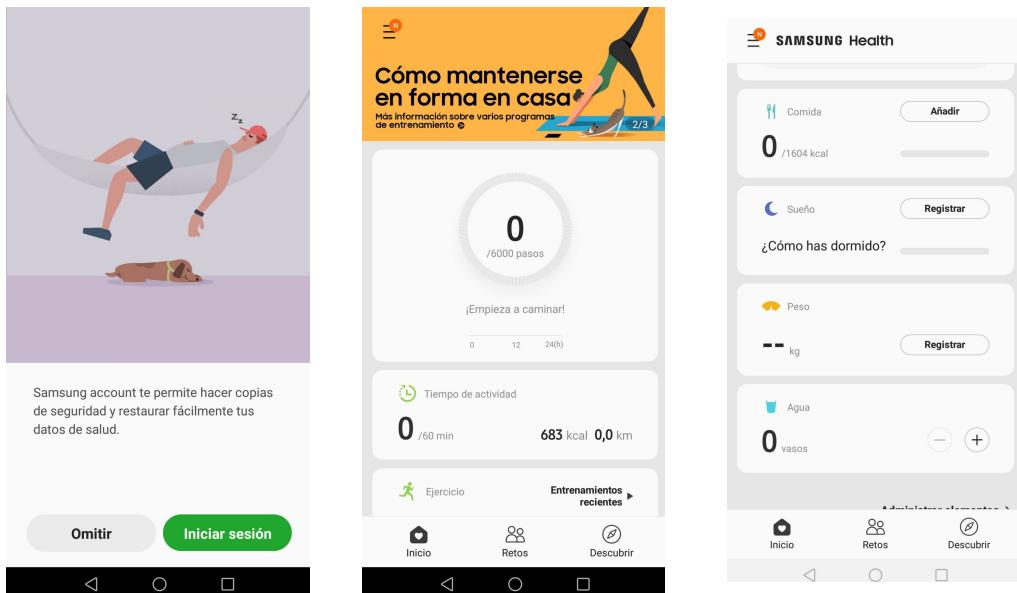


Figura 2.23: Interfaz de inicio de Samsung Health

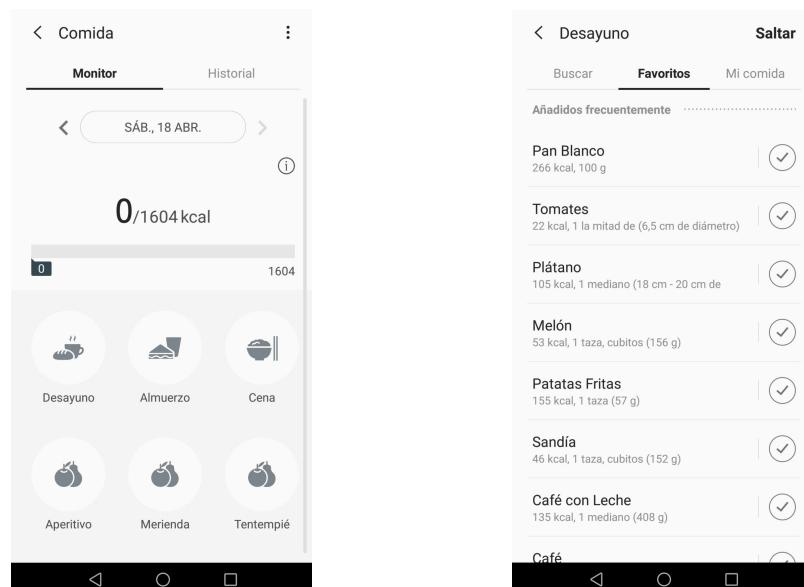


Figura 2.24: Seguimiento de calorías ingeridas

es que los kilómetros se cuentan siguiendo la actividad del GPS, y en una situación actual como la del COVID-19 no nos sirve de mucho. Pero por otro lado, esta funcionalidad de los objetivos me parece muy útil si estuviéramos en una situación normal. Además, podemos ver un historial de los entrenamientos registrados en los anteriores días (Figura 2.25).

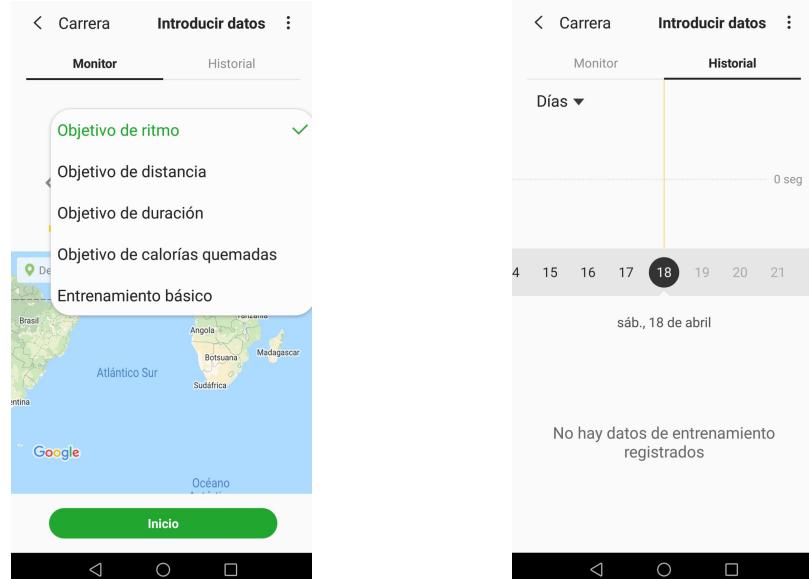


Figura 2.25: Interfaz de inicio de Samsung Health

Samsung Health cuenta con una pestaña llamada *Retos*. Se puede ver que actualmente hay un desafío global de abril (Figura 2.26) con el número de participantes. Además, si nos registramos con una cuenta y añadimos amigos, podemos competir con ellos y seguir las clasificaciones. Los retos van asociados al número de pasos de cada persona. Esta funcionalidad está muy bien y realmente es motivante para todos los usuarios, pero solamente hay una variedad de reto.

En la pestaña de *Descubrir* nos aparecen distintos entrenamientos. He elegido el Entrenamiento adaptativo de intervalos de alta intensidad (HIIT) y se nos muestra un calendario (Figura 2.27). Los días en verde son los días que dura este entrenamiento. Cuando seleccionamos el día actual, se nos muestran los distintos ejercicios a realizar con sus respectivas duraciones o repeticiones.

2.1.8. Muscle Booster - Fitness at home

Muscle Booster [20] es la octava aplicación más descargada de la categoría *salud y fitness* en marzo de 2020. Está disponible tanto en **Android** como en **Apple**, pero me

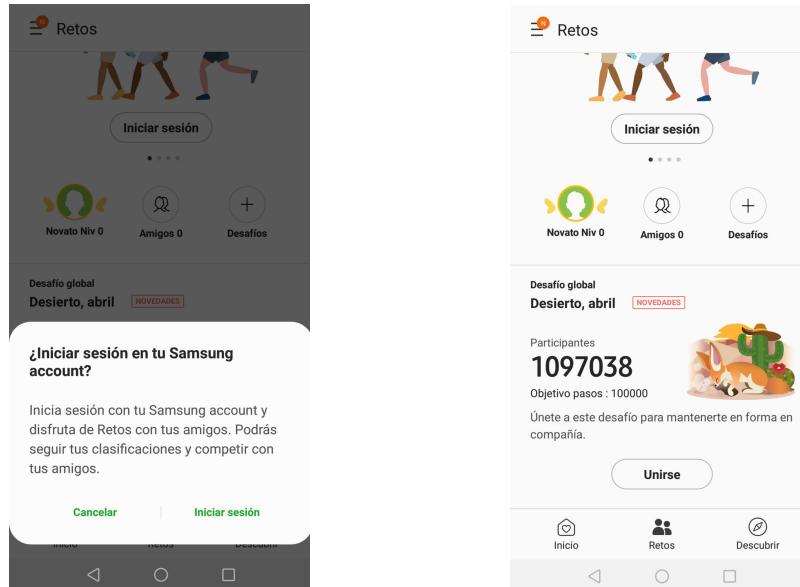


Figura 2.26: Sección de retos

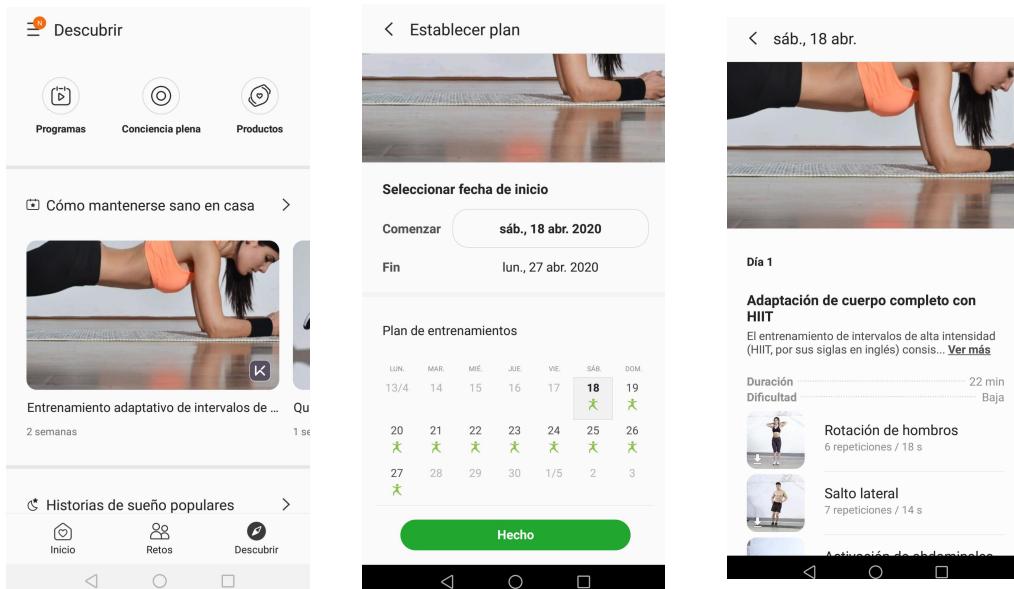


Figura 2.27: Entrenamientos de Samsung Health

he llevado una sorpresa cuando he visto que su puntuación en **Play Store** es de 2.3/5 estrellas.

Al iniciar la aplicación nos hace 14 preguntas (ver Figura 2.28):

1. Establecer objetivo
2. Elegir nuestro tipo de cuerpo
3. Cuerpo deseado
4. Áreas problemáticas del cuerpo
5. Condición física
6. Número de horas dedicadas a caminar diariamente
7. Frecuencia de hacer ejercicio
8. Número de flexiones que podemos hacer
9. Estatura
10. Peso actual
11. Objetivo de peso
12. Edad
13. Lugar de entrenamiento

Me parecen demasiadas preguntas obligatorias y algunas de ellas muy irrelevantes. Podrían hacer que los usuarios contestaran obligatoriamente las básicas (estatura, peso y edad) y las demás preguntas pudieran ser opcionales y contestar más adelante. De cualquier manera, al terminar las preguntas la aplicación nos dice que ya tiene un plan de entrenamiento hecho para nuestro objetivo de peso, sin preguntar al usuario qué ejercicios puede realizar y cuáles no (ver Figura 2.29). Al seguir adelante, la aplicación nos dice que para poder ver el entrenamiento hay que pagar obligatoriamente.

Creo que esta aplicación está muy mal hecha, pues no ofrece si quiera una mínima prueba gratuita. Los usuarios no saben qué entrenamiento es el que están pagando y si realmente les va a servir para algo. Muchos de ellos pueden llegar a pensar que puede ser algún tipo de estafa ya que no sabemos cuál es la fiabilidad de esta aplicación.

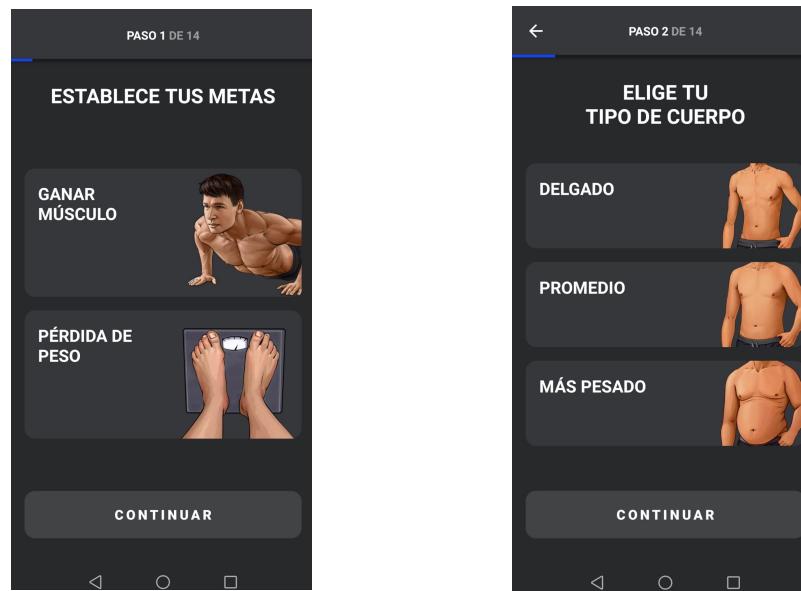


Figura 2.28: Test inicial de Muscle Booster

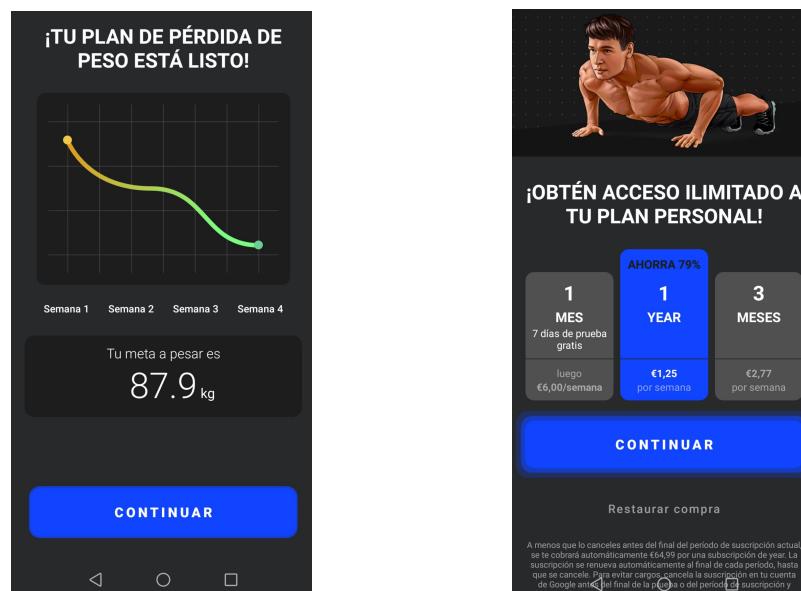


Figura 2.29: Acceso al pago de Muscle Booster

2.2. Tabla comparativa de apps

En la Tabla 2.1 se muestra una comparativa de los criterios que se vieron en la Sección 2.1. En **verde** se muestran las características que son positivas y que **SÍ** están disponibles en la aplicación. En **rojo** se muestran las características que son deseables pero que la aplicación **NO** las tiene. De acuerdo a esta tabla, a continuación se van a presentar las ventajas y desventajas de tener o no tener tales características.

	Fitify	Gym WP	BestFit	Fitvate	Total Fitness	Home Workout	Samsung Health	Muscle Booster
1 Control de peso, IMC y/o grasa del usuario	✗	✓	✗	✗	✓	✓	✓	NS/NC
2 Establecer objetivo de peso, masa muscular y/o grasa del usuario	✗	✓	✗	✗	✗	✗	✗	NS/NC
3 Seguimiento de la evolución del peso, IMC y/o grasa del usuario	✗	✓	✗	✗	✓	✓	✓	NS/NC
4 Registro de actividades cardio (minutos, intensidad, distancia...)	✓	✗	✓	✗	✗	✗	✓	NS/NC
5 Establecer objetivo personal de actividades cardio	✗	✗	✗	✗	✗	✗	✓	NS/NC
6 Ejercicios del mes de actividades cardio propuestos por profesionales	✗	✗	✗	✗	✗	✗	✗	NS/NC
Entrenamientos ofrecidos al usuario de distintos grupos musculares	✓	✓	✓	✓	✓	✓	✓	✓
8 Rutinas establecidas por profesionales	✓	✓	✓	✓	✓	✓	✓	NS/NC
La anotación del entrenamiento	✓	✓	✗	✓	✓	✓	✓	NS/NC
9 no requiere la intervención del usuario tras cada ejercicio	✓	✓	✗	✓	✓	✓	✓	NS/NC
10 Retos o desafíos de ejercicios propuestos por profesionales	✗	✗	✗	✓	✓	✓	✓	NS/NC
Competición y clasificación entre los usuarios en los distintos retos o desafíos	✗	✗	✗	✗	✗	✗	✓	NS/NC
12 No requiere cuenta en Google, Facebook, E-mail, o cualquier otra plataforma externa para acceder a la aplicación	✗	✓	✓	✓	✗	✓	✓	✓
13 No requiere hacer un test inicial para empezar a usar la aplicación	✗	✓	✗	✓	✓	✓	✓	✗
14 Todo el contenido es gratuito	✗	✗	✗	✗	✗	✗	✗	✗
15 Tiene contenido gratuito	✓	✓	✓	✓	✓	✓	✓	✗
16 No tiene contenido de pago	✗	✗	✗	✗	✗	✗	✗	✗
17 No contiene anuncios	✓	✗	✓	✗	✗	✗	✓	✓
18 Plataforma en la que se ofrece	Android iOS	Android	Android iOS	Android iOS	Android iOS	Android iOS	Android iOS	Android iOS

Tabla 2.1: Tabla comparativa de aplicaciones

Los tres primeros puntos de la tabla van de la mano. Desde mi punto de vista, un seguimiento del peso (incluyendo IMC y grasa) es básico para este tipo de aplicaciones. En algunas de ellas se ofrece un control de peso pero no se le propone al usuario que pueda establecer un objetivo. Cuando una persona decide embarcarse en el mundo del gimnasio, casi siempre busca un número objetivo en una báscula. Esta es una de las principales razones que mantienen a los usuarios motivados.

Los siguientes tres puntos (4, 5 y 6) tienen que ver con las actividades de cardio. Solo tres aplicaciones ofrecen un registro de actividades cardio y sólo una de ellas ofrece,

además, establecer un objetivo (caminar 10.000 pasos, por ejemplo). Las actividades cardio se pueden hacer fácilmente tanto en un gimnasio como al aire libre, con lo cual llevar un registro de los minutos, kilómetros, etc. es algo que puede llegar a motivar a los usuarios para ir superándose poco a poco. Un usuario puede establecerse un objetivo propio de caminar 10 kilómetros en una semana. Pero además, podemos hacer que los monitores de gimnasio propongan objetivos a los clientes, para que los usuarios sientan la fiabilidad de llegar a una meta propuesta por un experto.

Sin embargo, los siguientes tres puntos (7, 8 y 9) sí es algo que está presente en casi todas las aplicaciones, que son las rutinas de entrenamiento. Detecto situaciones en las que se puede dar una baja usabilidad o una experiencia de usuario no satisfactoria: en **BestFit** se proponen rutinas compuestas por varios ejercicios. Antes de hacer cada ejercicio, el usuario debe apuntar los kilos que va a coger para cada serie que va a hacer. Si un entrenamiento está compuesto por 10 ejercicios, el usuario debe parar 10 veces para coger el móvil e introducir esos datos. Está bien la idea de las otras aplicaciones, en la que simplemente se marca una casilla para indicar que se ha realizado el ejercicio, pero lo que no se debería hacer es pedir algo que hace que le implique mucho más tiempo al usuario.

Los siguientes dos puntos (10 y 11) tienen una funcionalidad que creo que es la que más motiva a los usuarios y es la que menos está presente: los retos. Un reto es una actividad propuesta a los usuarios de corta duración (no más de un mes). Por ejemplo, un reto de flexiones en el mes de abril en el que cada día deben hacer un número determinado de flexiones. Pero algo que sólo incorpora una aplicación y que puede realmente llevar al éxito entre los usuarios es la competición entre ellos. Se pueden hacer retos en los que se apunten los usuarios que quieran participar y cuando termine el reto se establezca una clasificación general para ver quién ha conseguido superarlo o no. Esta funcionalidad sólo está presente en **Samsung Health**, pero su problema es que solo hay un tipo de reto: dar un número de pasos durante el mes. Creo que si se amplía esta funcionalidad para un número mayor de retos (propuestos por los monitores de gimnasio o profesionales del sector) puede suponer una gran motivación para que los usuarios empiecen a utilizar la aplicación.

Pasando a los puntos 12 y 13, nos encontramos con dos funcionalidades que tienen que ver con el inicio de sesión. El inicio de sesión ideal debe ser rápido y directo: que nada más abrir la aplicación por primera vez entres a la pantalla inicial. Sin embargo, en algunas de las aplicaciones vistas nos encontramos con que hay un registro obligatorio a través de Google, Facebook, E-mail, etc. Aunque es raro, no todo el mundo dispone de una cuenta en estas plataformas, y aún cuando disponga de ellas, posiblemente nos llegue publicidad masiva al correo electrónico asociado a la cuenta. Además, en otras aplicaciones también se piden unos datos iniciales antes de entrar (edad, estatura, peso, objetivo...). Estos datos deberían ser opcionales, pues a lo mejor un usuario simplemente

quiere echar un vistazo a la aplicación en general, y si le piden estos datos le puede generar cierto rechazo. Desde mi punto de vista, estos datos los debe dar el usuario en caso de que quiera tener recomendaciones más precisas en cuanto a qué ejercicios hacer, entrenamiento, alimentación, etc. y no es algo que sea obligatorio, pues un usuario puede tener interés en mirar y hacer ejercicios sin tener que decir su peso, por ejemplo.

Los últimos puntos son los más importantes: ninguna aplicación ofrece un contenido **completamente** gratuito y algunas de ellas contienen anuncios. De hecho, **Muscle Booster** no ofrece ningún contenido sin pagar previamente. Un alto porcentaje de usuarios de estas aplicaciones ya estará pagando una suscripción mensual en un gimnasio o cualquier otro servicio de deporte público o privado. Hay cuatro tipos de usuarios:

- **Un usuario que ya está pagando un gimnasio u otro servicio deportivo y hace deporte:** nos encontramos con que la aplicación (con suscripción premium) ofrece unas rutinas que cualquier monitor de su gimnasio puede hacerle (aunque en algunos se debe pagar para ello). Además, un seguimiento básico semanal o mensual del peso se hace prácticamente en todos los gimnasios. En conclusión, estas aplicaciones no ofrecen a estos tipos de usuarios algo que no puedan hacer en su lugar deportivo.
- **Un usuario que no está apuntado a ningún gimnasio u otro servicio y tampoco hace deporte:** Puede ser que este usuario tenga conocimientos muy básicos acerca de la actividad física y deportiva, con lo cual estamos en la situación de un usuario que tiene que pagar una suscripción premium de una aplicación en la que se le ofrecen rutinas que probablemente no entienda o no disponga del material necesario para hacerlas. Son muchos los usuarios que quieren empezar a hacer ejercicio desde casa pero muy pocos de ellos consiguen aguantar un mes. La desmotivación surge cuando esta gente no ve resultados, y es porque no disponen del conocimiento experto para realizar lo que su cuerpo realmente necesita.
- **Un usuario que no está apuntado a ningún gimnasio y otro gimnasio pero sí hace deporte:** si hace deporte por su cuenta es porque sabe los ejercicios, rutinas y/o actividades cardiovasculares que debe hacer y de qué manera. También puede ser que haga deporte siguiendo canales de **YouTube** o **Instagram** y siga otras cuentas que le proporcionen conocimientos acerca de las actividades deportivas, con lo cual no necesitaría pagar una suscripción premium por algo que puede obtener de forma gratuita buscando en Internet.
- **Un usuario que está apuntado al gimnasio u otro servicio deportivo pero no hace deporte:** este puede ser el usuario menos predominante, y rara vez vamos a encontrar un usuario de este tipo que esté interesado en pagar una suscripción premium de una aplicación que prácticamente no le va a dar uso.

Sí que es cierto que casi todas ellas ofrecen mucho contenido gratuito: el seguimiento de peso, algunos ejercicios, rutinas... Pero estos ejercicios y/o rutinas gratuitos son muy escasos, y pueden hacer que los usuarios se cansen de hacer las mismas actividades todos los días. Además, con la situación actual del COVID-19 han sido muchos usuarios los que buscan rutinas y ejercicios a través de la plataforma de **YouTube**. Véase el caso de Patry Jordan con su proyecto **Gym Virtual** [21], en el que cuenta ya con más de 8 millones de suscripciones. Hay muchísimo contenido de este tipo totalmente gratuito, con lo cuál: **¿qué hay de diferente en estas aplicaciones que no se pueda conseguir de forma gratuita en Internet?** Prácticamente nada, y además no cuenta con una comunicación personalizada, ni existe un registro de la actividad, ni tampoco hay facilidad para acceder al material de forma estructurada.

Capítulo 3

Planificación del proyecto

3.1. Metodología de desarrollo: diseño centrado en el usuario

Para la planificación de este proyecto se va a utilizar un Diseño Centrado en el Usuario (DCU) [22], que es una filosofía de diseño en la que el proceso de diseño del producto final está enfocado a las necesidades, requerimientos y limitaciones de los usuarios finales. El usuario está involucrado en cada fase del desarrollo, garantizando que el producto final atiende a sus necesidades y características. Esto quiere decir que el usuario se vuelve co-autor del producto. Antes de describir esta metodología de desarrollo, es importante definir dos conceptos [23]:

- **Experiencia del usuario:** consiste en las emociones y sensaciones que siente el usuario cuando se relaciona o interactua con el producto. El objetivo de los diseñadores es que la experiencia del usuario sea lo más satisfactoria posible, obteniendo como resultado su fidelidad en el producto.
- **Experiencia en la web:** las interfaces gráficas de la web deben ofrecer funcionalidades que sean fáciles de entender para los usuarios. Hay dos componentes principales: facilidad de uso y entendimiento de funcionalidades. La **facilidad de uso** tiene que ver con los conocimientos, dominio de herramientas digitales, características sociográficas y psicográficas y ambientales o de contexto, que ayudan a que los diseñadores adapten las características funcionales para que el usuario entienda la interfaz. El **entendimiento de funcionalidades** es la manera en la que los diseñadores crean una interfaz que sea familiar para el usuario, reduciendo

la curva de aprendizaje normal cuando dicho usuario se enfrenta al producto.

En la metodología de desarrollo centrado en el usuario se destacan los siguientes principios [23]:

1. **Diseño para los usuarios y sus tareas:** El desarrollo de los sistemas informáticos deberá considerar siempre las características de sus usuarios, de sus tareas del mundo real, y de su entorno de trabajo.
2. **Consistencia:** ofrecer una interfaz que minimice la curva de aprendizaje del usuario cuando utilice el producto.
3. **Diálogo simple y natural:** debe haber un diálogo con una secuencia natural entre el usuario y el sistema, evitando la saturación de información sobre una simple tarea y empleando siempre la misma terminología.
4. **Reducción del esfuerzo mental del usuario:** el usuario debe esforzarse en realizar su tarea, no en aprender a usar la herramienta que realiza su tarea. Si emplean demasiado esfuerzo en ello, puede desembocar en la frustración y una mala experiencia de usuario.
5. **Proporcionar realimentación adecuada:** indicadores que ayuden al usuario a confirmar que su tarea ha sido completada (con éxito o no) o que aún sigue en proceso. Por ejemplo, cuando hacemos click en una carpeta de nuestro ordenador, se añade un círculo al lado del puntero para que sepamos que la carpeta está cargando.
6. **Proporcionar mecanismos de navegación adecuados:** proporcionar información que ayude al usuario a saber dónde se encuentra, asignando títulos a las páginas, barras de scroll, numeración de páginas, o lo que se suele utilizar en las páginas webs que se conoce como *migas de pan*, ofreciendo un historial de las áreas visitadas.
7. **Dejar que el usuario dirija la navegación:** el sistema debe ofrecer ayuda al usuario para que complete la tarea. Un buen ejemplo puede ser la ayuda de autocompletar los comandos que ofrece Overleaf [24].
8. **Presentar información clara:** ofrecer un contraste entre los distintos elementos de la pantalla, usando espacios, cajas y técnicas visuales.
9. **El sistema debe ser amigable:** el sistema debe explicarse a sí mismo lo suficiente como para minimizar las veces que el usuario se encuentra en situación de pedir ayuda.

10. **Reducir el número de errores:** el sistema debe dirigir al usuario para que tome la ruta correcta y evitar que cometa errores durante el camino.

En esta metodología de desarrollo se va a hacer uso del **Design Thinking** [2], cuyo objetivo es entender las necesidades de los usuarios y proponer una solución a ellas. De acuerdo a Tim Brown, actual CEO de IDEO [25], “Es una disciplina que usa la sensibilidad y métodos de los diseñadores para hacer coincidir las necesidades de las personas con lo que es tecnológicamente factible y con lo que una estrategia viable de negocios puede convertir en valor para el cliente, así como en una gran oportunidad para el mercado”.

3.1.1. Design Thinking

En la Figura 3.1 se muestran las cinco etapas que componen este proceso. Estas fases no tienen por qué ser lineales ni consecutivas, se pueden ir hacia adelante y hacia atrás e incluso saltar a una etapa no consecutivas.

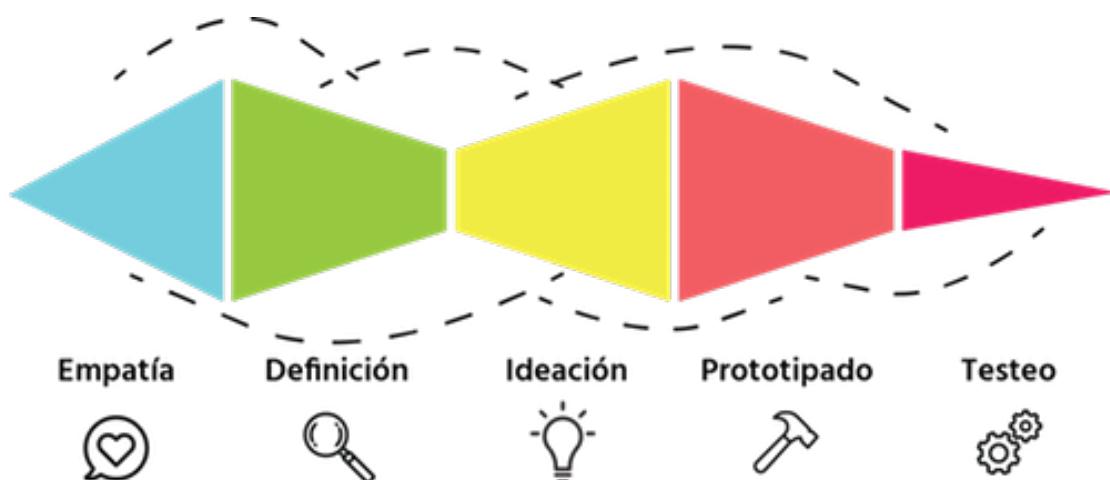


Figura 3.1: Etapas del proceso de **Design Thinking** [2]

El proceso se muestra a continuación [26]:

1. **Empatizar:** los diseñadores deben ser conscientes en todo momento de que van a resolver problemas de personas ajenas, con lo cual convierte a la empatía como algo esencial del proceso. Se debe hacer un esfuerzo para comprender lo que hacen y los

usuarios y el por qué lo han hecho, además de sus necesidades físicas y emocionales, cómo conciben el mundo y qué es significativo para ellos.

2. **Definir:** después de que los diseñadores se han puesto en la piel de los usuarios y se ha reunido suficiente información, lo siguiente es crear coherencia entre dicha información y definir el problema.
3. **Idear:** en esta etapa empieza el proceso de diseño y lo que se conoce como **brainstorming**. En esta lluvia de ideas se generan tantas ideas como sea posible para poder obtener múltiples soluciones en vez de una única. De acuerdo a Plattner, “Todas las ideas son válidas y se combina todo desde el pensamiento inconsciente y consciente, pensamientos racionales y la imaginación.”
4. **Prototipar:** en esta etapa se construyen prototipos de cualquier tipo (un post-it, un cartón, una actividad, etc). La clave es obtener un prototipo que sea palpable y que ayude a visualizar de manera fácil una posible solución. No hay que dedicar demasiado tiempo a un prototipo, pues la idea es hacer algo de manera rápida y barata de forma que podamos obtener un feedback de los usuarios.
5. **Testear:** esta etapa consiste en obtener el feedback y opiniones de los usuarios acerca de los prototipos que hemos creado. Además, esto puede desembocar en que los diseñadores creen más empatía con los usuarios debido a nuevos problemas que les hayan podido surgir. El objetivo de esta etapa es refinar las soluciones y mejorarlas.

En cada etapa existen diversas técnicas y herramientas que nos ayudan a conseguir el objetivo de la fase. En la página web de **Design Thinking** [2] podemos encontrar estas técnicas. Las técnicas que se han seguido para el desarrollo de este proyecto se van a presentar en las siguientes secciones.

3.2. Personas ficticias

Este método se utiliza para representar a distintos usuarios con distintas características. No necesariamente deben ser usuarios potenciales del producto, pero sí centrarse en las necesidades de los usuarios más importantes [27]. Tal y como se vio en la Sección 2.2, en este proyecto hay cuatro tipos de usuarios, de los cuales tres de ellos son los realmente importantes:

- Grupo 1: usuarios que van al gimnasio y hacen deporte
- Grupo 2: usuarios que no van al gimnasio y hacen deporte

- Grupo 3: usuarios que no van al gimnasio y no hacen deporte

En la Figura 3.2 se muestra un usuario que pertenece al grupo 1. Se trata de un usuario que lleva yendo al gimnasio varios años y al que suele ir frecuentemente.

En la Figura 3.3 se muestra un usuario perteneciente al grupo 2. Este hombre no va al gimnasio pero sale a correr varias veces a la semana. Además, cuando tiene tiempo libre aprovecha para realizar más actividades al aire libre con sus hijos.

En la Figura 3.4 nos encontramos con una usuaria perteneciente al grupo 3. Esta joven nunca ha ido a un gimnasio y tampoco hace deporte por su cuenta, pero ahora que ya ha acabado el instituto, se plantea el empezar a realizar alguna actividad deportiva.

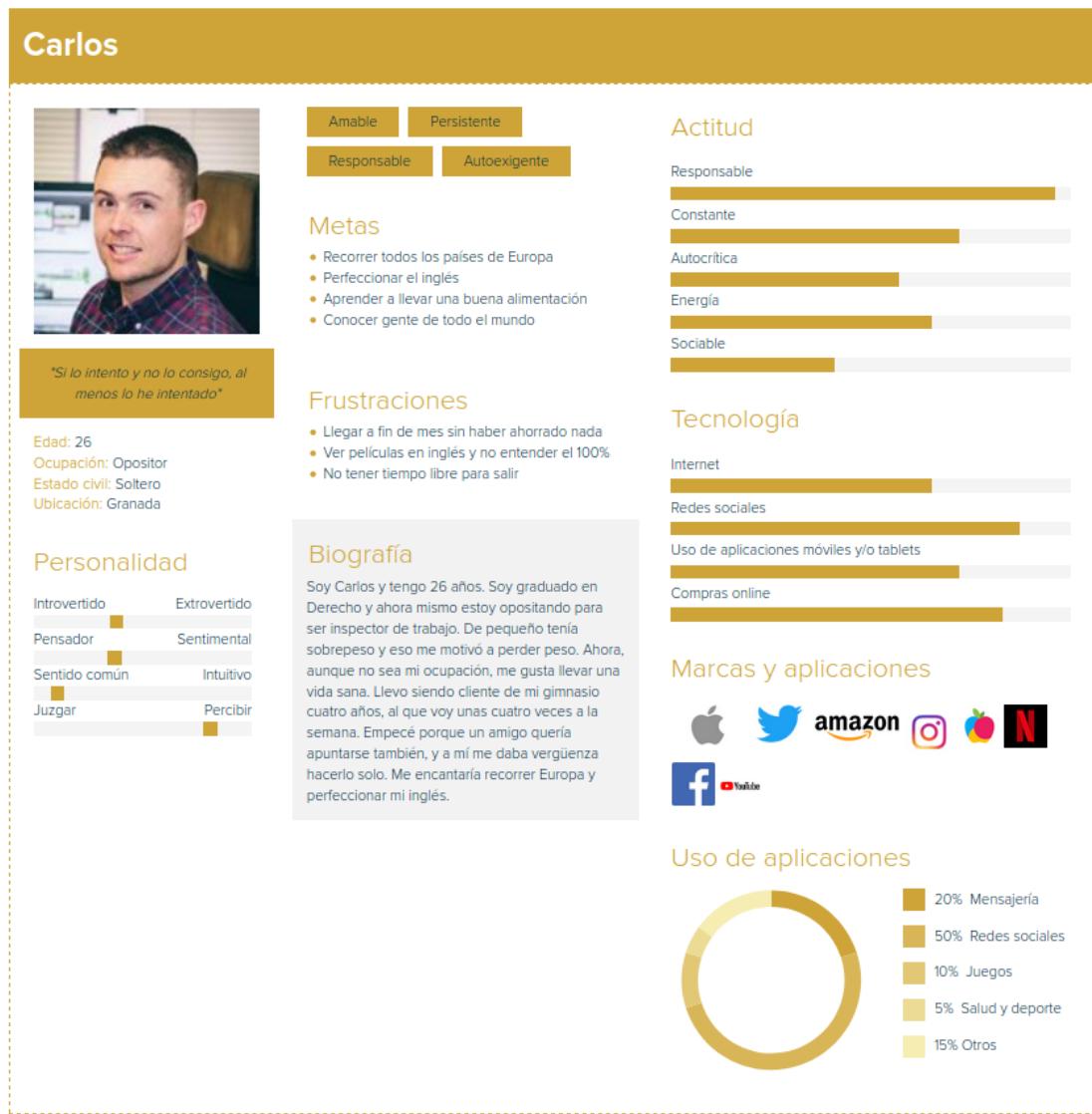


Figura 3.2: Usuario perteneciente al grupo 1



Figura 3.3: Usuario perteneciente al grupo 2

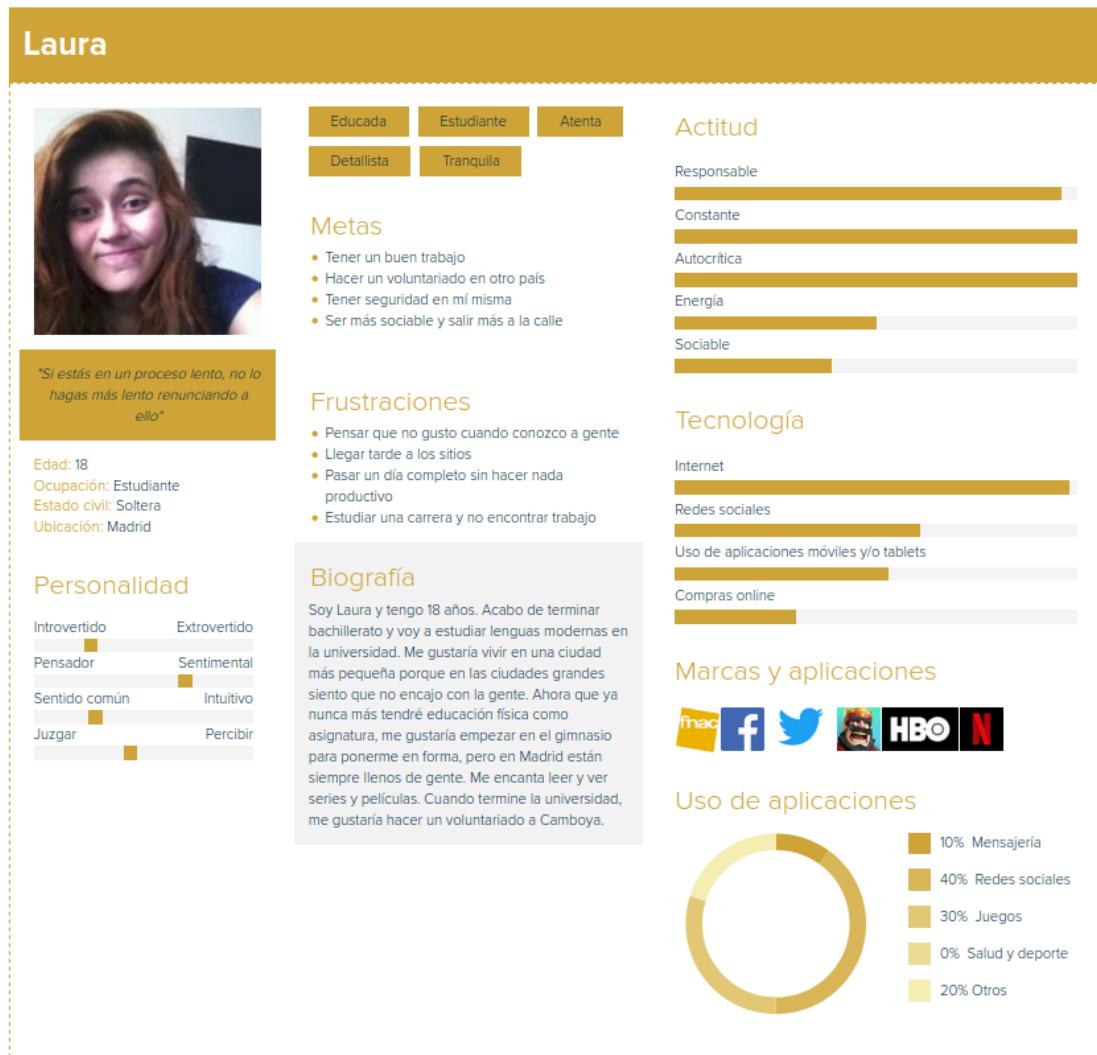


Figura 3.4: Usuario perteneciente al grupo 3

3.3. Malla receptora de información

Una vez que se han creado unas personas ficticias, con ellas en mente pasamos a hacer la malla receptora. Esta técnica se utiliza para recopilar información y recoger tantas ideas como sea posible. Esta información se divide en cuatro cuadrantes. Los pasos para hacer la malla receptora son los siguientes [28]:

1. Dividir un folio, pizarra u otra cosa en cuatro cuadrantes
2. Se dibuja un signo más (+) en el cuadrante superior izquierdo, un signo igual (=) en el cuadrante superior derecho, una interrogación (?) en el cuadrante inferior izquierdo (?) y una admiración (¡) en el cuadrante inferior derecho
3. En los cuadrantes se anota lo siguiente:
 - +: cosas importantes, notables o positivas
 - =: críticas constructivas
 - ?: preguntas que revelan la experiencia
 - ¡: ideas que surgen durante la experiencia

La malla receptora de este proyecto se muestra en la Tabla 3.1. En este caso, en cada cuadrante se ha anotado lo siguiente:

- +: aspectos positivos de las aplicaciones evaluadas
- =: aspectos negativos de las aplicaciones evaluadas
- ?: preguntas que le pueden surgir a cualquier persona
- ¡: ideas que quiero implementar y las que no

3.4. Definición de la propuesta

Imagym es una propuesta nueva que cambia el método de hacer deporte en los distintos gimnasios y centros deportivos. Se trata de una plataforma web en la que tanto los clientes como los trabajadores pueden acceder y pueden darle un uso diferente dependiendo de su rol. Independientemente de ello, el acceso a la web es totalmente personalizado, es decir, al entrar en la web cada usuario tendrá una interfaz diferente dependiendo del contenido que le ofrece su gimnasio. Además, lo que hace que sea diferente es que cuenta con un **bot de Telegram**, **ImagymBot**, que permite una interacción directa con los

<ul style="list-style-type: none"> ■ Control de peso ■ Seguimiento del peso, IMC y/o grasa ■ La anotación de ejercicios se realiza mediante un checkmark ■ Registro de actividades cardio ■ Rutinas de entrenamiento ■ Vídeos que muestran la realización de ejercicios ■ Posibilidad de hacer ejercicios individuales sin hacer un entrenamiento específico ■ Retos y desafíos ■ Acceso inmediato al contenido de la aplicación sin registro previo ■ Se indica al usuario mediante voz el ejercicio y repeticiones que debe hacer 	<ul style="list-style-type: none"> ■ Al realizar una rutina, antes de realizar cada ejercicio hay que anotar los kilos que se van a coger ■ Solamente se ofrecen una o dos rutinas de entrenamiento de forma gratuita ■ Poca variedad de retos y desafíos gratuitos ■ Competición entre usuarios en una sola variedad de reto ■ Utilizar la aplicación requiere un registro a través de una plataforma externa ■ Para empezar a utilizar la aplicación, el usuario debe realizar un test con preguntas personales ■ Contador de calorías de comida impreciso ■ El contador de las repeticiones de los ejercicios empieza de forma automática
+	=
<ul style="list-style-type: none"> ■ ¿Consumo muchos datos móviles? ■ ¿Se puede usar sin conexión a Internet? ■ ¿Ocupa poco espacio en la memoria? ■ ¿Cuál es la frecuencia con la que debo entrenar? ■ ¿Recibo notificaciones cuando me toque hacer un entrenamiento? ■ ¿Al cambiar de móvil se perderá todo mi progreso? 	<ul style="list-style-type: none"> ■ Seguimiento de la evolución del peso, IMC y/o grasa ■ Establecer objetivo de peso, IMC y/o grasa ■ Registro de actividades cardio (minutos, intensidad, distancia, etc.) ■ Establecer objetivo personal de actividad cardio ■ Ejercicios del mes de actividades cardio propuestos por profesionales ■ Seguimiento de la evolución en las actividades cardio ■ Entrenamientos de agrupados por objetivos/músculos ■ Rutinas establecidas por profesionales ■ Registro de rutinas y/o entrenamientos mediante checkmark ■ Cronómetro y contador de repeticiones de los ejercicios ■ Vídeo-tutoriales de cómo hacer los ejercicios ■ Retos y desafíos ■ Competición entre los usuarios en los retos y desafíos ■ Registro de funcionalidades mediante voz ■ Registro de información personal sólo si el usuario así lo deseé, obteniendo más precisión en las funcionalidades ■ Todo el contenido completamente gratuito ■ Conversación natural y expresión de emociones con el bot

Tabla 3.1: Malla receptora de información

clientes, de forma que pueden ver todo el contenido de su gimnasio a través del bot y sin necesidad de instalar aplicaciones adicionales.

Hay una comunicación bidireccional entre ImagymBot y la plataforma web. Se podría decir que ImagymBot está más enfocado a las tareas que pueden realizar los usuarios, mientras que la plataforma web está orientada a que los trabajadores del gimnasio introduzcan el contenido y este sea visible en el bot.

Como todo este proyecto es bastante grande, se ha dividido en dos trabajos. Este TFG abarca toda la planificación y desarrollo de **ImagymBot**, aunque se explicarán nociones básicas de la plataforma web cuando las tareas del bot impliquen una comunicación con ella. En esta propuesta se implementarán las siguientes funcionalidades:

1. Control de peso, músculo y grasa
2. Seguimiento de la evolución del peso, IMC, músculo y/o grasa
3. Establecer objetivo de peso, músculo y/o grasa
4. Registro de actividades cardio (minutos, intensidad, distancia, etc)
5. Establecer objetivo personal de actividad cardio
6. Seguimiento de la evolución en las actividades cardio
7. Ejercicio del mes propuesto por el gimnasio
8. Rutinas establecidas por profesionales
9. Registro de rutinas mediante **checkmark**
10. Retos y desafíos
11. Competición entre los usuarios en los retos y desafíos
12. Registro de información personal sólo si el usuario así lo deseé, obteniendo más precisión en las funcionalidades
13. Contenido completamente gratuito

3.5. Recursos necesarios

ImagymBot es un chatbot, con lo cual lo primero que se necesita es una aplicación de mensajería que permita a los usuarios interactuar con el bot. Como opciones se ha

barajado el uso de **Whatsapp** [29] y **Telegram** [30], ya que son las dos aplicaciones de mensajería más populares del mercado. Iván Ramírez explica en su artículo [3] las diferencias entre ambas aplicaciones, distinguiendo entre varias categorías. En la Figura 3.5 se muestra una tabla comparativa que indica el ganador de cada categoría, siendo **Telegram** el vencedor con más puntos acumulados.

	WHATSAPP	TELEGRAM
USUARIOS	1	0
PRIVACIDAD Y SEGURIDAD	0	1
CHAT DE TEXTO	1	1
COMPARTIR	0	1
LLAMADAS	1	0
MULTIPLATAFORMA	0	1
PERSONALIZACIÓN	0	1
TOTAL	3	5

Figura 3.5: Tabla comparativa **Whatsapp** vs **Telegram**. Fuente: Iván Ramírez [3]

En este proyecto se ha elegido el uso de **Telegram**, y a continuación voy a presentar las diferencias más importantes que me han hecho descartar **Whatsapp**:

1. **Número de teléfono:** **Telegram** permite la interacción entre usuarios con un simple nombre de usuario, valga la redundancia. Esto implica que se puede acceder a grupos sin que los restantes miembros sepan nuestro número de teléfono, y también que se puede empezar una conversación con otro usuario sin saber su número. En **Whatsapp** esto no es así, pues siempre se va a mostrar nuestro número de teléfono a los demás participantes del grupo, y para empezar una conversación con otra persona requiere tener su número.
2. **Versión web:** **Telegram** permite su uso a través de la web independientemente de si el teléfono está conectado a Internet. En **Whatsapp**, para tener acceso a la web

es necesario que nuestro teléfono esté encendido y además conectado a Internet.

3. **Copia de seguridad:** **Telegram** no necesita copia de seguridad, pues todo está almacenado en su propia nube. **Whatsapp** hace uso de **Google Drive** para realizar la copia de seguridad, implicando que hay que tener una cuenta en **Gmail** para poder tener esta funcionalidad.
4. **Descarga de archivos:** **Telegram** permite visualizar los archivos, imágenes y vídeos sin necesidad de descargarlos previamente en nuestro dispositivo. En **Whatsapp**, para ver los archivos y demás es necesario guardarlos en nuestro dispositivo, consumiendo de esta manera la memoria de almacenamiento local.

También se necesita un lenguaje de programación. He elegido **Python** como lenguaje porque nunca he tenido la oportunidad de aprenderlo y siempre había oído tiene una sintaxis muy limpia, ya que es necesario tener todo el código bien tabulado para poder compilarlo. Además, **Python** es el tercer lenguaje de programación más popular del mundo, solo siendo superado por **Java** y **C** [31], con lo cual es otra razón de peso para querer aprenderlo.

Python cuenta con una API cuya instalación y documentación se encuentra en un repositorio de **Github** [32]. Su instalación se hace muy sencilla mediante el uso de comandos en una terminal. También cuenta con ejemplos explicados que podemos descargar nosotros mismos y probarlos.

Este proyecto necesita también una base de datos para almacenar toda la información del bot. **Python** cuenta con un paquete llamado **pymysql** [33] que permite la ejecución de **MySQL** de forma muy sencilla.

El bot debe estar activo 24 horas al día, todos los días del año, con lo cual se precisa de un servidor. En el **Github** de la API de **Telegram** proponen diversos lugares donde se puede alojar nuestro bot. He seleccionado **Python Anywhere** [4] porque su interfaz me pareció la más intuitiva y sencilla de usar. También es importante destacar que permite el alojamiento de la base de datos, con lo cual me permite ahorrar el tener que buscar otro servidor externo para ella. Además, la versión gratuita de **Python Anywhere** ofrece las suficientes funcionalidades para hospedar mi bot, funcionalidades que en otros lugares se podían acceder sólo mediante pago.

Las funcionalidades que ofrece **Python Anywhere** para los distintos tipos de suscripciones se muestran en la Figura 3.6. Lo más destacable es que con la versión gratuita se cuenta con hasta dos consolas abiertas, que son suficientes para tener una consola ejecutando el **script** de **Python** y otra consola para hacer las operaciones **MySQL**.

	Custom: €5 to €500/month	Startup: €99/month	Web Developer: €12/month	Hacker: €5/month	Beginner: €0/month
Monthly billing, cancel at any time Or get in touch to hear about discounts if you pay for a year up front, or want to pay annually in bitcoins	✓ 30-day no-quibble money-back guarantee				
Python web applications At <code>your-username.eu.pythonanywhere.com</code>	You choose ✓	3 ✓	2 ✓	1 ✓	1 ✓
At your own domain	✓	✓	✓	✓	✗
Web workers per web app	You choose 25		3	2	1
Python, Bash, MySQL consoles	Unlimited	Unlimited	Unlimited	Unlimited	Up to 2
SSH access to your account via <code>ssh.eu.pythonanywhere.com</code>	✓	✓	✓	✓	✗
Access to external internet sites from your code e.g. <code>urllib</code> or <code>wget</code>	✓	✓	✓	✓	Specific sites via HTTP(S) only
Gold star next to your name in the forums :-)	✓	✓	✓	✓	✗
Share your consoles with other people	✓	✓	✓	✓	✓
Scheduled tasks Like cron, but simpler	20, hourly or daily	1 daily task			
Always-on tasks Tasks that are always running	You choose 1		1	1	✗
CPU allowance more info	You choose 10,000 seconds		4,000 seconds	2,000 seconds	100 seconds
Bandwidth more info	Depends on price	High	Medium	Low	Low
Private file storage	You choose 50GB		5GB	1GB	512MB

Figura 3.6: Distintos tipos de suscripciones a **Python Anywhere** [4]

Capítulo 4

Diseño de la Interfaz de Usuario

4.1. Tareas de usuario

En esta sección se va a presentar la Matriz de Tareas de Usuario. Esto consiste en construir una tabla donde las tareas de los usuarios están en las filas y en las columnas están los tres grupos de usuario, tal y como se vio en la Sección 3.2. La intersección de la tarea con el grupo de usuario es un número del 0-4 que representa la frecuencia con la que ese usuario va a acceder a dicha tarea. La frecuencia es la siguiente:

- **0:** solo la primera vez
- **1:** rara vez
- **2:** normal
- **3:** mucho
- **4:** muchísimo

La Matriz de Tareas de Usuario se presenta en la Tabla 4.1. Las tareas se han ordenado por orden de importancia, es decir, las primeras tareas serán aquellas que más frecuencia de uso tengan por parte de todos los usuarios y las últimas tareas serán aquellas que menos frecuencia de uso tengan. En **negrita** se han marcado las tareas que van a ser más frecuentes.

	Van al gimnasio y hacen deporte	No van al gimnasio y hacen deporte	No van al gimnasio y no hacen deporte
Establecer objetivo de peso	4	4	4
Establecer objetivo personal de actividad cardio	4	4	4
Apuntarse a ejercicio del mes	4	4	4
Consultar rutinas de entrenamiento	4	4	4
Consultar ranking del ejercicio del mes	4	4	4
Consultar historial de ejercicios del mes	4	4	4
Ver retos	4	4	4
Apuntarse a un reto	4	4	4
Ver clasificación de reto	4	4	4
Consultar historial de retos	4	4	4
Anotar peso	3	4	4
Ver evolución de peso	3	3	4
Anotar actividad cardio	3	3	3
Consultar historial de rutinas de entrenamiento	4	3	2
Consultar historial de actividades cardio	2	3	3
Seguir rutina de entrenamiento	3	3	2
Anotar rutina de entrenamiento (checkmark)	3	3	2
Cambiar rutina de entrenamiento	3	2	1
Ver tutorial de ejercicio	0	2	4
Ver progreso del reto actual	2	2	2
Eliminar objetivo de peso	1	1	2
Eliminar objetivo personal de actividad cardio	1	1	2
Terminar mi reto actual	1	1	2
Desapuntarse de un reto	1	1	1
Desapuntarse de un ejercicio del mes	1	1	1
Registrar información personal adicional	1	1	1

Tabla 4.1: Matriz de Tareas de Usuario

4.2. Organización de la información

En esta Sección se va a crear el **sitemap** (organización de la información) de **ImagymBot**. El **sitemap** es muy típico de las páginas webs. Consiste en un listado de todas las páginas que componen el sitio web [34]. Si aplicamos esto a **ImagymBot** que es un chatbot, el **sitemap** está compuesto por todos los botones que permiten que los usuarios naveguen a través del bot a las distintas funcionalidades. En las Figuras ?? y ?? se muestra la organización de la información de **ImagymBot**.

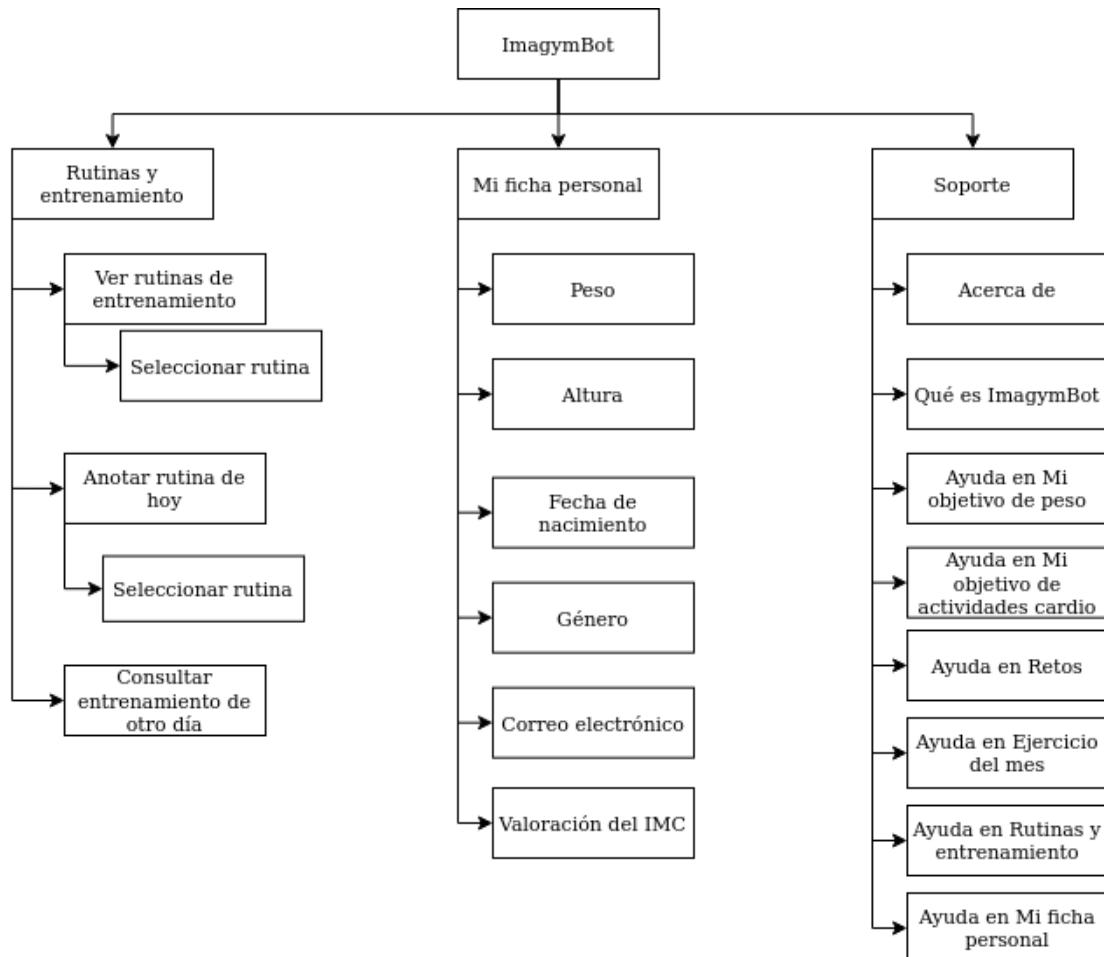


Figura 4.1: Sitemap de **ImagymBot**

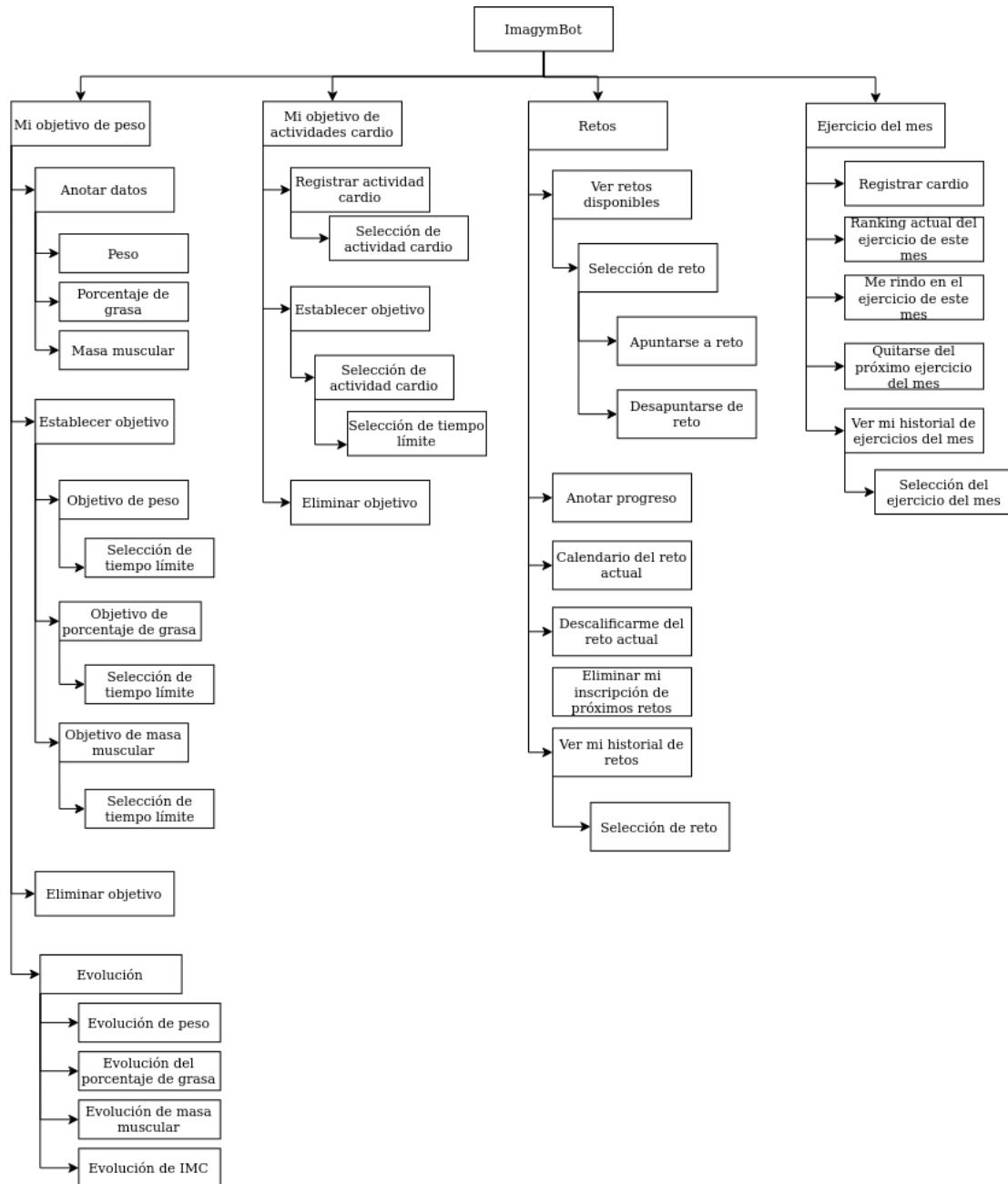


Figura 4.2: Sitemap de ImagymBot

4.3. Bocetos wireframe

En las Figuras de la 4.3 a la 4.10 se muestran unos bocetos que muestran una idea de la navegación de **ImagymBot**. Dentro de cada boceto se muestran dos capturas para explicar la situación del menú y detalle.



Figura 4.3: Boceto de *Inicio*



Figura 4.4: Boceto de *Inicio > Mi objetivo de peso*



Figura 4.5: Boceto de *Inicio > Mi objetivo de actividad cardio*

Figura 4.6: Boceto de *Inicio > Retos*



Figura 4.7: Boceto de *Inicio > Ejercicio del mes*



Figura 4.8: Boceto de *Inicio > Rutinas y entrenamiento*

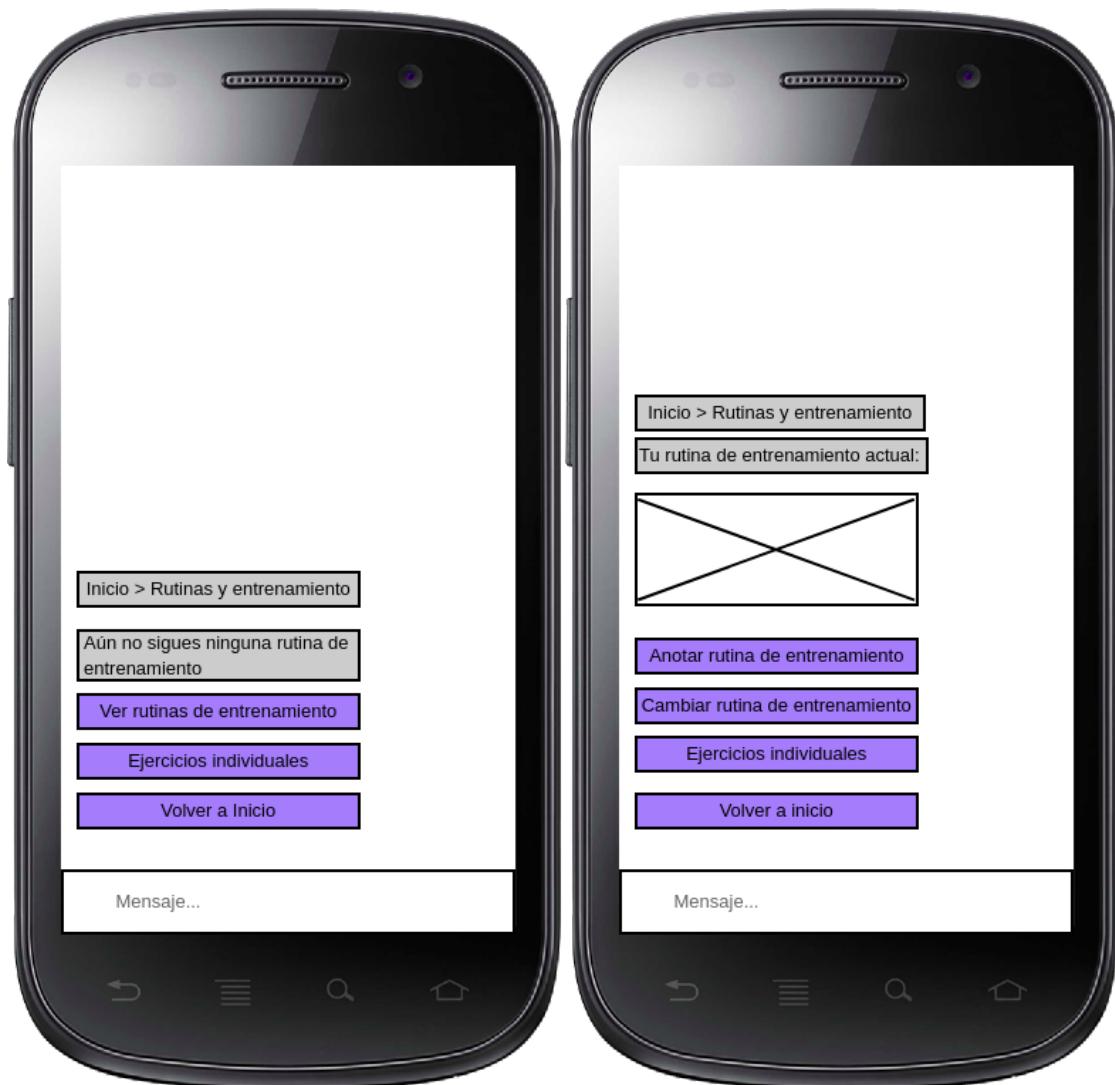


Figura 4.9: Boceto de *Inicio > Mi ficha personal*



Figura 4.10: Boceto de *Inicio > Soporte*

Capítulo 5

Diseño técnico

Como ya se adelantó en la Sección 3.5, este bot se va a hacer para **Telegram** usando **Python**. **Python** cuenta con una API para hacer bots en **Telegram**, cuya documentación está disponible en **Github** [32].

Como entorno de trabajo se va a usar el editor de textos **Sublime Text**, que es un editor gratuito y que posee una gran comunidad. Tiene muchos atajos de teclado que hacen la programación mucho más sencilla [35].

Para el registro de algunos datos del bot se va a usar una base de datos en común con la web. Los monitores y dueños de los gimnasios y centros deportivos usarán la web para proponer los retos, ejercicios del mes y rutinas de entrenamiento a los usuarios. Estos datos se guardarán en la base de datos y el bot los usará para proporcionar dicha información a los usuarios.

El registro de los usuarios se hace de una forma muy sencilla. **ImagymBot** pedirá una clave a los usuarios cuando estos inicien el bot. Esta clave la proporcionan los centros deportivos a sus usuarios y es única para cada centro deportivo. Esta idea es similar a la de la plataforma de cuestionarios **Kahoot** [36], donde el creador del cuestionario proporciona un código y todos los usuarios insertan dicho código para poder realizar el cuestionario. Una vez el usuario de **ImagymBot** inserta la clave, se le registrará automáticamente y se le proporcionará una clave provisional para acceder a la web. Su usuario es su nombre de usuario de **Telegram** y podrá cambiar su contraseña desde la web, además de modificar o añadir sus datos personales.

Para la gestión de la base de datos se va a usar **MySQL**. **Python** dispone de una

API llamada **pymysql** [33] que sirve para interactuar con bases de datos **MySQL** de una forma muy sencilla. Su web proporciona el método de instalación y una documentación muy fácil de entender.

ImagymBot creará gráficas e imágenes para mostrar a los usuarios distinta información así como su evolución de peso o el calendario de un reto específico. Para ello, **Python** cuenta con una librería llamada **Matplotlib** que permite crear gráficas y tablas [37].

Desde la Sección 5.1 a la Sección 5.3 se describen los requisitos funcionales, no funcionales, y diseño de la base de datos, respectivamente:

- Los **requisitos funcionales** son aquellos que describen las funcionalidades que el sistema proporcionará a las distintas entradas que se le presenten, ya sea de un usuario o de una interacción con otro sistema.
- Los **requisitos no funcionales** son aquellos que describen las propiedades del sistema, tales como el rendimiento, la seguridad y la disponibilidad.
- En el **diseño de la base de datos** se describe la representación de la información necesaria para realizar el modelo entidad-relación de la base de datos, su paso a tablas, y por último el diagrama completo la base de datos con todos datos y sus tipos correspondientes.

5.1. Requisitos funcionales

- El sistema permitirá el registro de usuarios a partir de una clave común del gimnasio, generando una clave privada para cada usuario
- El sistema permitirá anotar o modificar los datos personales del usuario
- El sistema permitirá registrar el peso, grasa y/o músculo diario de los usuarios
- El sistema permitirá establecer un objetivo de peso, grasa y/o músculo de los usuarios, con una fecha de inicio y una fecha fin
- El sistema permitirá registrar las actividades cardio que hagan los usuarios a lo largo del día
- El sistema permitirá a los usuarios establecer un objetivo personal de una determinada actividad de cardio
- El sistema permitirá consultar las actividades cardio registradas durante un día cualquiera

- El sistema deberá mostrar todos los retos disponibles del gimnasio a los usuarios
- El sistema deberá permitir que los usuarios se apunten/desapunten a los retos del gimnasio
- El sistema deberá informar a los usuarios cuando empiece un reto
- El sistema deberá recordar cada día del reto a los usuarios que deben anotar su progreso
- El sistema permitirá consultar el historial de retos de los usuarios
- El sistema deberá mostrar el próximo ejercicio del mes a los usuarios
- El sistema permitirá que los usuarios se apunten/desapunten del próximo ejercicio del mes
- El sistema deberá informar a los usuarios cuando empiece el ejercicio del mes
- El sistema permitirá consultar el historial de ejercicios del mes de los usuarios
- El sistema deberá establecer una clasificación del ejercicio del mes de acuerdo a la puntuación de los usuarios
- El sistema deberá ofrecer todas las rutinas disponibles del gimnasio a los usuarios
- El sistema permitirá anotar los ejercicios de las rutinas que hacen los usuarios
- El sistema permitirá a los usuarios seguir sus rutinas favoritas
- El sistema permitirá consultar los ejercicios anotados durante cualquier día

5.2. Requisitos no funcionales

- El sistema estará disponible para todas las plataformas que soporten **Telegram**
- El sistema debe estar disponible 24 horas al día durante todo el año
- El sistema no debe tardar más de 3 segundos en dar una respuesta a los usuarios
- El sistema deberá cumplir con la Ley de Protección de Datos
- El sistema controlará el acceso de usuarios y sólo podrán acceder usuarios que estén inscritos en un gimnasio
- El sistema deberá ser tolerante a los fallos y los errores deben ser reportados por el propio sistema

- El sistema ofrecerá un manual de ayuda de cada una de las funcionalidades del sistema
- El sistema debe cumplir con la política de privacidad de **Telegram**

5.3. Diseño de la base de datos

En la Figura 5.1 se muestra el modelo entidad-relación de la base de datos de **ImagymBot**. Una vez se tiene este modelo, el siguiente paso es el paso a tablas, que se muestra en la Figura 5.2. Por último, en la Figura 5.3 se muestran todas las tablas resultantes con los datos y sus tipos correspondientes. La base de datos representa la siguiente información (agrupada por funcionalidades de acuerdo al **sitemap** de la Sección 4.2):

5.3.1. Mi objetivo de peso

- Un usuario se apunta a un gimnasio y un gimnasio tiene muchos usuarios.
- Los usuarios pueden anotar su peso, grasa o músculo una vez al día, pudiendo modificar estos datos a lo largo del día.
- Los usuarios pueden proponerse un objetivo de peso, grasa o músculo, pero no pueden tener dos objetivos a la vez.
- Un objetivo de peso tiene una fecha de inicio y una fecha fin.

5.3.2. Mi objetivo de actividades cardio

- Cada gimnasio está equipado con máquinas de actividades cardio.
- Un usuario del gimnasio puede registrar el tiempo, distancia, nivel y calorías que ha hecho de una determinada actividad cardio del gimnasio en un día. En el mismo día puede anotar tantas actividades cardio como quiera.
- Los usuarios pueden proponerse objetivos personales de cardio. Un objetivo personal de cardio consiste en hacer un mínimo de minutos, kilómetros o calorías de una actividad cardio en un mes.

5.3.3. Retos

- Los monitores de un gimnasio pueden proponer un reto de un ejercicio.
- El reto consiste en hacer un número determinado de repeticiones del ejercicio cada día durante el tiempo que dure el reto.
- Cada reto tiene un nivel de dificultad, siendo 1 el nivel fácil y 4 el nivel experto.
- Los usuarios del gimnasio pueden apuntarse al reto, pero no pueden apuntarse a un reto si sus fechas coinciden con otro al que ya está apuntado.
- Los usuarios no pueden apuntarse a un reto que ya ha empezado.
- Los usuarios pueden desapuntarse de un reto que aún no ha empezado o descalificarse del reto que están haciendo.
- Los usuarios deben anotar su progreso cada día del reto.

5.3.4. Ejercicio del mes

- Un gimnasio propone un ejercicio del mes que consiste en hacer un mínimo de minutos, distancia o calorías en una determinada actividad cardio durante un mes.
- Sólo puede haber un ejercicio del mes en cada mes.
- Los usuarios del gimnasio deciden si quieren apuntarse o no al ejercicio del mes.
- Los usuarios no pueden apuntarse a un ejercicio del mes que ya ha empezado.
- Los usuarios recibirán puntos cada vez que registren la actividad cardio del ejercicio del mes.
- Un usuario podrá descalificarse del ejercicio del mes siempre que quiera.
- Cuando termine el mes se hará un ranking con todas las puntuaciones de todos los usuarios que han terminado el ejercicio del mes.

5.3.5. Rutinas y entrenamiento

- Los monitores de un gimnasio ofrecen rutinas de entrenamiento a los usuarios.
- Un monitor del gimnasio puede ofrecer tantas rutinas como quiera, pero solo se mostrará a los usuarios la rutina más reciente.

- Una rutina está formada por varios ejercicios y cada ejercicio tendrá un número de repeticiones.
- Un ejercicio ejercita varios músculos.
- Los usuarios pueden seguir rutinas (añadirlas a sus favoritos)

5.3.6. Mi ficha personal

- Los usuarios acceden a **ImagymBot** mediante una clave del gimnasio.
- Un gimnasio puede estar instalado en diferentes zonas geográficas.
- Cuando un usuario accede a **ImagymBot**, se genera una clave que le servirá para acceder a la web.
- Un usuario podrá anotar o modificar los siguientes datos:
 - Altura
 - Fecha de nacimiento
 - Género
 - Correo electrónico

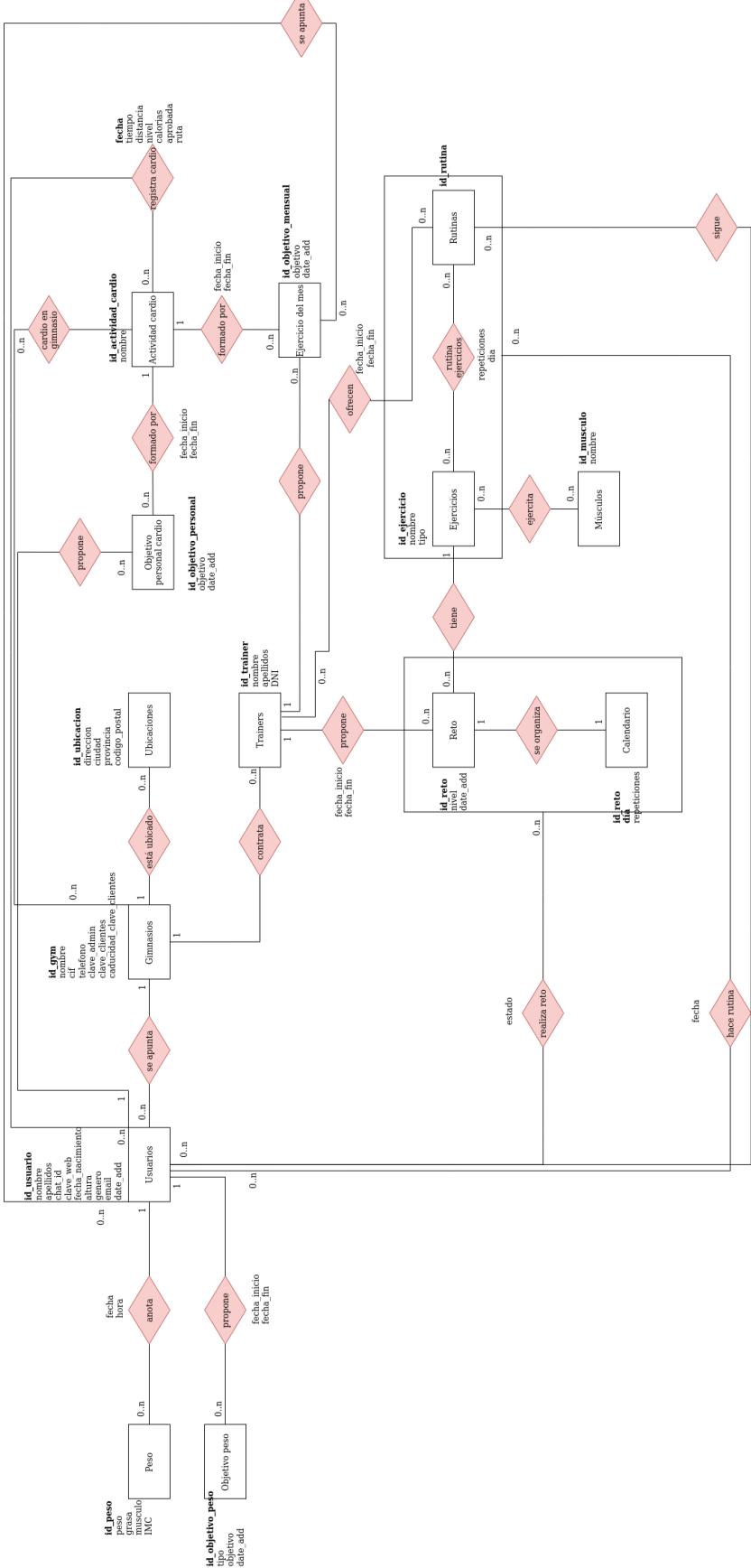


Figura 5.1: Diagrama Entidad-Relación de la base de datos de ImagymBot

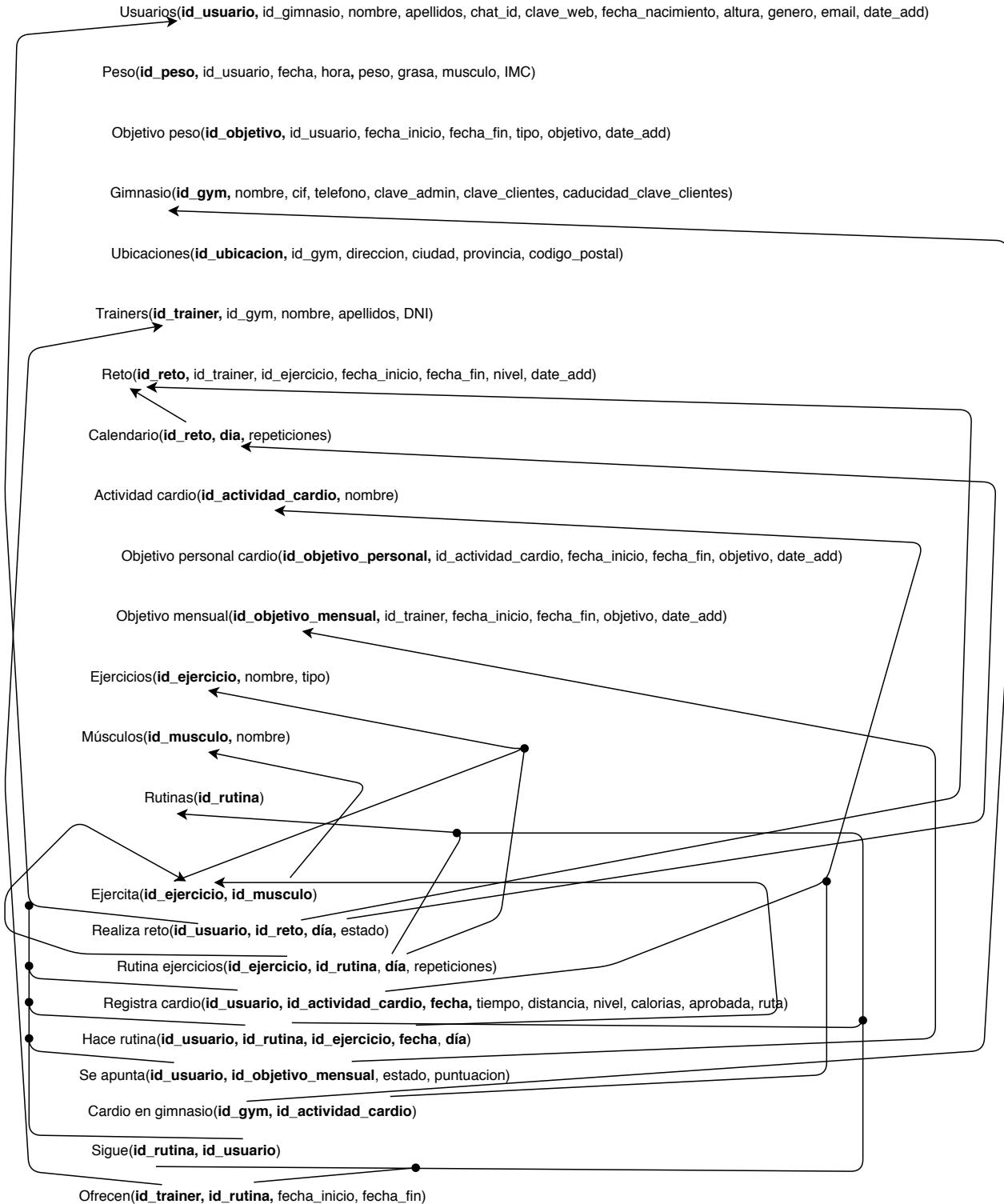


Figura 5.2: Paso a tablas

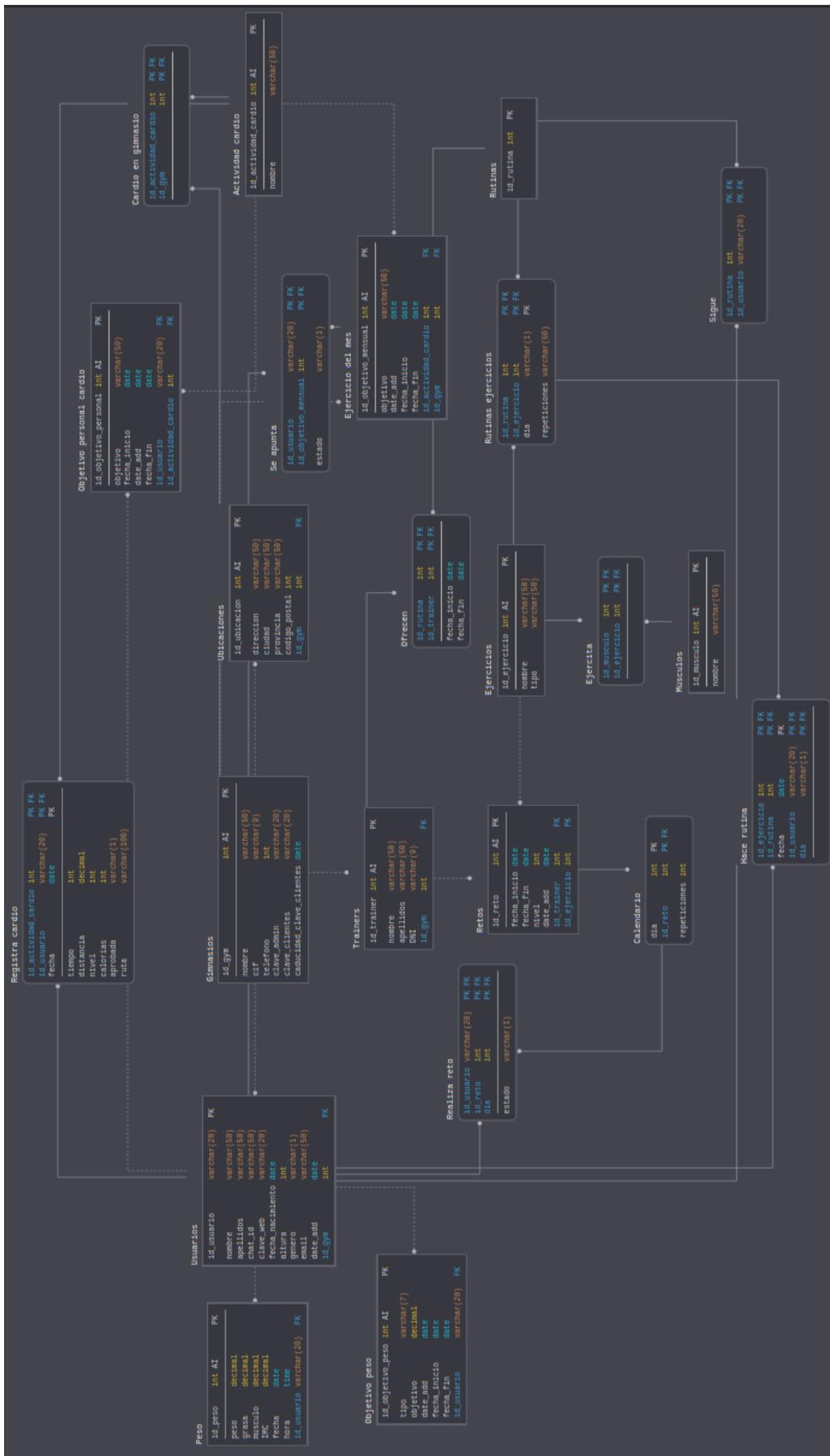


Figura 5.3: Base de datos de ImagymBot

Capítulo 6

Implementación

El primer paso de la implementación abarca lo que es la selección de un lenguaje para programar el bot y la creación del propio bot en **Telegram**. Tal y como mencioné en la Sección 3.5, un bot en esta plataforma se puede crear en diferentes lenguajes de programación, y en este proyecto se ha hecho uso de **Python**.

La creación del propio bot se hace desde la misma aplicación. Hay que buscar al usuario **@BotFather** y nos enviará un mensaje con varios comandos para utilizar, tal y como muestro en la Figura 6.1. A continuación, se pulsa (o se escribe) el comando **/newbot** y **BotFather** nos pedirá el nombre de nuestro bot y su nombre de usuario, teniendo este último que acabar obligatoriamente en ‘bot’(ver Figura 6.2). Una vez hecho esto, **BotFather** nos mandará el **token** para acceder a la **API** (ver Figura 6.3).

Una vez tenemos el **token**, el siguiente paso es conectarse con el bot mediante la **API**. En el repositorio de **Github** de **Telegram** cuentan con varios ejemplos muy simples de bots conversacionales [32]. En concreto, para tomar una simple plantilla y empezar **ImagymBot**, utilicé el primer ejemplo que ofrecen en su web llamado **conversationbot.py** [38]. El **token** debemos ponerlo en la función **main** del bot.

Un bot conversacional debe tener en cuenta los contextos en los que se mueven los usuarios. Por ejemplo, si un usuario le pregunta al bot: ‘¿tienes hermanos?’, y posteriormente le pregunta: ‘¿cuántos tienes?’, el bot sabe que se refiere a cuántos hermanos tiene, pues sabe que está en el contexto de los hermanos. Estos contextos se definen al principio de nuestro bot. En el caso de **ImagymBot** se disponen de varios contextos. Se le aparecerán distintos menús al usuario dependiendo de la sección donde se encuentre. Si el usuario está actualmente en la sección de *Mi objetivo de peso* y, por ejemplo, pulsa un botón que lleva a la sección *Mi ficha personal* y que estaba en mensajes anteriores, el

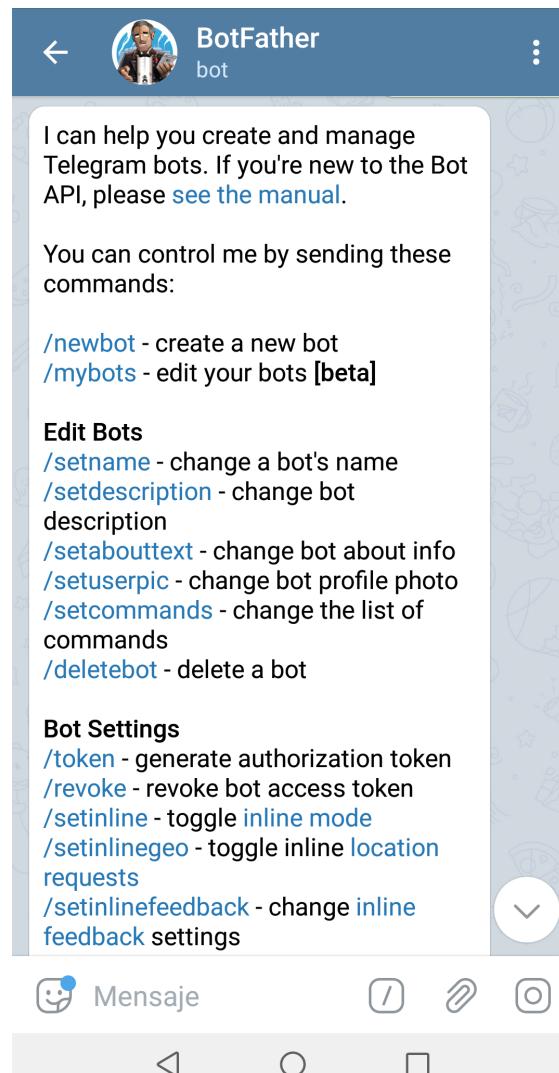


Figura 6.1: Primer paso para la creación de un bot

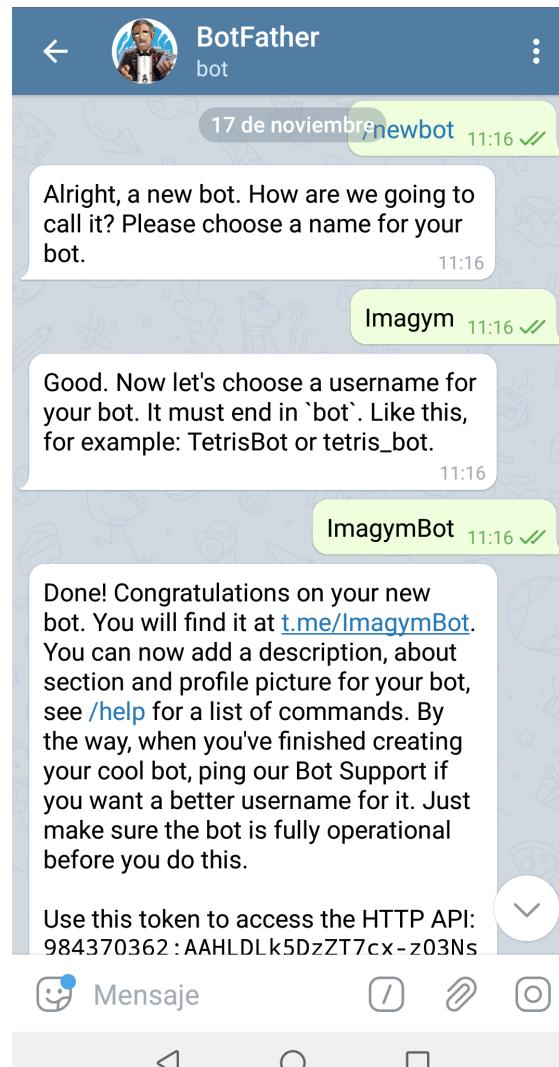


Figura 6.2: Nombre y usuario de nuestro bot

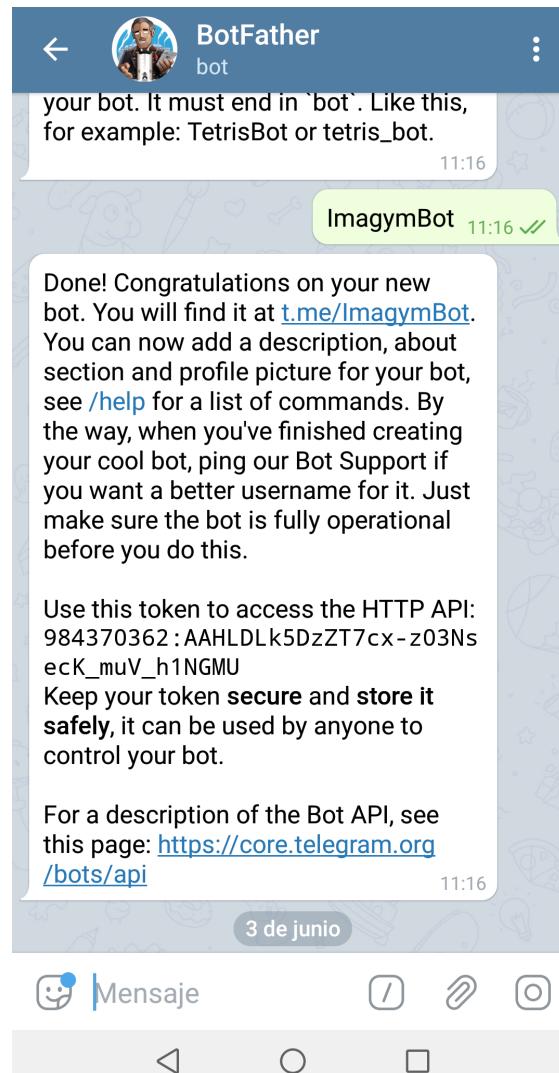


Figura 6.3: **Token** del bot creado

bot sabrá que ha pulsado un botón que no corresponde con el contexto actual e informará de ello al usuario.

6.1. Cómo configurar el servicio de MySQL

Como ya se comentó en la Sección 3.5, se ha utilizado **PythonAnywhere** [4] para hospedar el bot y la base de datos. Desde su web, simplemente se debe ir la sección **Database** y poner un nombre para crear la base de datos, tal y como muestra en la Figura 6.4. En esta misma interfaz se nos proporcionan los valores que debemos utilizar para conectar con la base de datos desde nuestra aplicación y además se puede cambiar la contraseña de dicha base de datos.

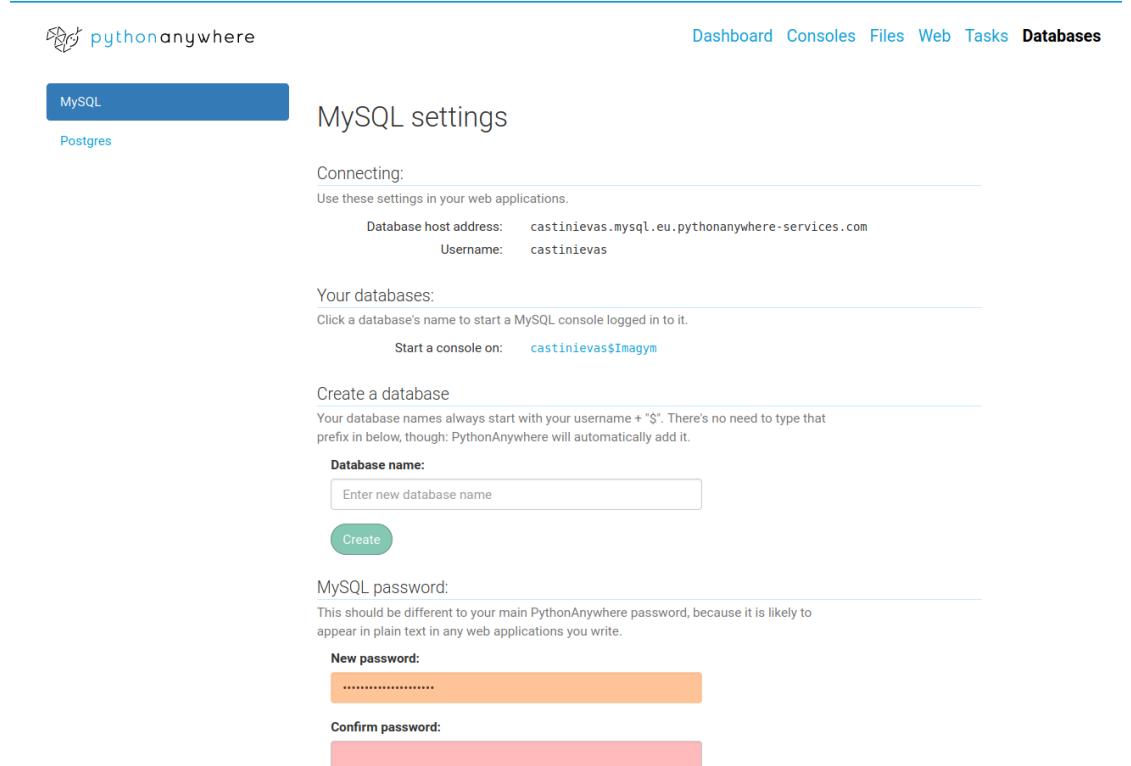


Figura 6.4: Interfaz de **Database** en **PythonAnywhere**

Para conectar **ImagymBot** con la base de datos, ya se comentó en la Sección 5 que se iba a utilizar la librería **pymysql** [33]. Se debe utilizar la sentencia que se muestra

en la línea de código en la Figura 6.1 para realizar la conexión. Como ya mencioné anteriormente, los valores que debemos pasar se muestran en **PythonAnywhere**.

```
1 db = pymysql.connect("Servidor", "Usuario", "Password", "BaseDatos")
```

Listing 6.1: Conexión con la base de datos desde **Python** con **pymysql**

6.2. Implementando los Handlers del bot

El bot va a manejar diferentes manejadores o **handlers** dependiendo de lo que haga el usuario. En **ImagymBot** se van a utilizar tres manejadores: **CallbackQueryHandler**, **MessageHandler** y **CommandHandler**:

- **CallbackQueryHandler(*nombre_funcion, pattern*)**: este manejador ejecuta la función **nombre_funcion** cuando el usuario pulsa un botón cuyo identificador coincide con el **pattern**.
- **MessageHandler(*filtros, nombre_funcion*)**: este manejador activa la función **nombre_funcion** cuando el usuario envía algo (texto, fotos, audios, vídeos, etc). Los filtros permiten indicar qué tipos de mensajes va a aceptar este manejador. Por ejemplo, en **ImagymBot** se ha utilizado mayoritariamente como filtro **Filters.text & (Filters.command)**, que significa aceptar todos los mensajes de texto que no sean comandos.
- **CommandHandler(*nombre_comando, nombre_funcion*)**: este manejador activa la función **nombre_funcion** cuando el usuario introduce mediante texto el comando **nombre_comando**

Cada función **nombre_funcion** se definirá con dos parámetros: **update** y **context**, que contienen toda la información del mensaje (nombre del usuario que ha mandado el mensaje, su alias, apellidos, identificador del mensaje, etc). El código de la definición de una función se muestra en la Figura 6.2.

```
1 def nombre_funcion(update, context):
```

Listing 6.2: Definición de una función asociada a un **handler**

Cuando un usuario accede a cualquier bot de **Telegram**, se le aparecerá un botón de **Iniciar**. Este botón lo que hace es simplemente activar el comando **/start**, con lo cual he implementado el manejador [**CommandHandler('start', start)**] para ejecutar la función **start** cuando el usuario lo inicie.

6.2.1. Respondiendo a las acciones del usuario

Para responder al usuario se utilizan dos funciones diferentes de la **API** dependiendo de si el usuario ha escrito algo o ha pulsado un botón.

Respondiendo con texto y botones

Si el usuario ha escrito algo, ya sea un comando o simplemente texto, se utiliza la función **reply_text**. Este código se implementa en las funciones que llaman los **CommandHandler** y **MessageHandler**. Como ejemplo, en la Figura 6.3 se muestra una parte del código que se utiliza cuando el usuario utiliza el comando **/start**:

- **Líneas 1-3:** el bot envía el mensaje que se encuentra en el parámetro **text** al usuario.
- **Líneas 5-8:** se crearán una lista de botones que se enviarán al usuario. En este caso, se envía un único botón cuyo texto será **Empezar Imagym** y cuyo identificador será **start_menu**. Esto significa que habrá un **CallbackQueryHandler** cuyo **pattern** será **start_menu**.
- **Líneas 10-12:** se utiliza la función **reply_text** con un parámetro adicional, **reply_markup**, cuyo valor es la lista de botones que se ha creado justo antes.

```

1 update.message.reply_text(
2     text="Bienvenido/a de nuevo, "+name_user
3 )
4
5 keyboard = [
6     [InlineKeyboardButton("Empezar Imagym", callback_data='start_menu')]
7 ]
8 reply_markup = InlineKeyboardMarkup(keyboard)
9 update.message.reply_text(
10    text="Pulsa el botón para empezar",
11    reply_markup=reply_markup
12 )

```

Listing 6.3: Uso de la función **reply_text** para responder al usuario

Cuando el usuario pulsa un botón, se utiliza la función **send_message**. Este código se implementa en las funciones llamadas por los **CallbackQueryHandler**. Como ejemplo, en la Figura 6.4 se muestra el código asociado a cuando el usuario pulsa el botón que se ha creado en el código anterior:

- **Líneas 1 y 2:** se coge la información necesaria de los parámetros **update** y **con-**

text (ver Figura 6.2).

- **Líneas 3-7:** se utiliza la función **send_message**. Los parámetros son iguales que en la función **reply_text**, con la adición del parámetro **chat_id** que es el identificador del chat del usuario que ha pulsado el botón. El parámetro **parse_mode** se utiliza para que el bot pueda **parsear** el mensaje y de esta manera poder enviar texto en negrita o cursiva.
- **Líneas 8-16:** se crea la lista de botones que se va a enviar al usuario.
- **Líneas 17-23:** se envía un mensaje al usuario con los botones que se han creado.

```

1 query = update.callback_query
2 bot = context.bot
3 bot.send_message(
4     chat_id = query.message.chat_id,
5     text="<b>argando inicio...</b>",
6     parse_mode='HTML'
7 )
8 keyboard = [
9     [InlineKeyboardButton("Mi objetivo de peso", callback_data='inicio_peso'),
10      ],
11     [InlineKeyboardButton("Mi objetivo de actividades cardio", callback_data='
12         inicio_cardio')], [
13         [InlineKeyboardButton("Retos", callback_data='inicio_retos')], [
14             [InlineKeyboardButton("Ejercicio del mes", callback_data='inicio_ejercicio
15                 ')], [
16                 [InlineKeyboardButton("Rutinas y entrenamiento", callback_data='
17                     inicio_rutinas')], [
18                     [InlineKeyboardButton("Mi ficha personal", callback_data='inicio_ficha')], [
19                         [InlineKeyboardButton("Soporte", callback_data='inicio_soporte')]]]]]]]
20 reply_markup = InlineKeyboardMarkup(keyboard)
21 bot.send_message(
22     chat_id = query.message.chat_id,
23     text="<b>Inicio</b>",
24     parse_mode='HTML',
25     reply_markup = reply_markup
26 )

```

Listing 6.4: Uso de la función **send_message** para responder al usuario

Otros tipos de respuesta

Un bot de **Telegram** también puede responder con imágenes, audios, documentos, entre otros. Para ello se utilizan otras funciones muy similares a las que se han explicado en la Sección 6.2.1. Aquí se muestran algunas funciones obtenidas de su documentación [32] que se utilizan para enviar otros tipos de respuesta o mensajes:

- **reply_animation(*args, **kwargs):** para responder con una animación.
- **reply_audio(*args, **kwargs):** para responder con un audio.

- **reply_contact(*args, **kwargs)**: para responder con un contacto.
- **reply_document(*args, **kwargs)**: para responder con un documento.
- **reply_photo(*args, **kwargs)**: para responder con una foto.
- **reply_video(*args, **kwargs)**: para responder con un vídeo.

Todas estas funciones tienen también asociada una función como era **send_message** a **reply_text**. Recordemos que realizan la misma función, pero en la primera se debe especificar el identificador del chat del usuario al que queremos enviar el mensaje.

ImagymBot envía fotos a los usuarios para mostrarles el calendario de un reto, entre otros. El código se muestra en la Figura 6.5:

- **Línea 1:** se escribe la ruta de la foto que se quiere enviar.
- **Línea 2:** se utiliza la función **open()** para abrir imagen de la ruta que hemos especificado. El parámetro **rb** indica que se abre con representación binaria.
- **Líneas 3-6:** se usa la función **send_photo** para enviar la foto al chat del usuario.

```

1 table_reto_path = "RUTA"
2 pic = open(table_reto_path, 'rb')
3 bot.send_photo(
4     chat_id = query.message.chat_id,
5     photo = pic
6 )

```

Listing 6.5: Uso de la función **send_photo** para enviar una foto al usuario

Recibiendo fotos del usuario

Este bot pide una foto al usuario cuando registra una actividad cardio que le añade puntos a su marcador en el ejercicio del mes, pudiendo una persona externa verificar dicho registro. Este mensaje se maneja a través de un **MessageHandler** cuyo filtro asociado es **Filters.photo**. En la Figura 6.6 se muestra cómo se descarga la imagen del usuario en nuestro ordenador o servidor:

- **Línea 1:** se obtiene una lista con diferentes tamaños de la imagen que ha pasado el usuario.
- **Línea 2:** se elige el tamaño que está en la primera posición, que es la imagen con más calidad.
- **Líneas 4-6:** se abre la conexión con la base de datos

- **Líneas 9-12:** se selecciona la fecha y el identificador de la actividad cardio más reciente, que es la actividad de la que el usuario ha enviado la foto. Esto es simplemente para el nombre de la foto.
- **Líneas 14-15:** se construye el nombre de la imagen en un **string**. El nombre de la imagen será ***id-actividad-cardio_fecha_usuario***
- **Línea 17:** se usa la función **download()** especificando una ruta, que será donde se descargará la imagen.

```

1 array_photos = update.message.photo
2 file = context.bot.getFile(array_photos[-1].file_id)
3
4 db = pymysql.connect("localhost", "root", "password", "ImagymServer")
5 db.begin()
6 cur = db.cursor()
7 username = update.message.from_user.username
8
9 cur.execute("SELECT id_actividad_cardio,fecha FROM Registra_cardio WHERE
    id_usuario='"+username+"' AND fecha=(SELECT MAX(c2.fecha) FROM Registra_cardio
    c2 WHERE id_usuario='"+username+"');");
10 resultado = cur.fetchall()
11 id_actividad_cardio=resultado[0][0]
12 fecha=resultado[0][1]
13
14 name_image = str(id_actividad_cardio)+"_"+fecha.strftime(' %d-%m-%Y-%H:%M:%S ')+"_"+
    username
15 ruta = "RUTA DE LA IMAGEN"+name_image
16
17 file.download(ruta)

```

Listing 6.6: Obteniendo una imagen que envía el usuario

6.2.2. Ejecutando sentencias MySQL

La información que proporciona el usuario se debe guardar en la base de datos. Para ello, con **pymysql** se utilizan una serie de funciones para ejecutar las sentencias en nuestra base de datos, mostradas en la Figura 6.7:

- **Línea 1:** se realiza la conexión con la base de datos.
- **Línea 2:** se comienza la transacción.
- **Línea 3:** se crea un cursor para ejecutar una sentencia.
- **Líneas 4-5:** si es una consulta, se obtiene el resultado con **fetchall()**. Si es una inserción, borrado o alteración de la base de datos, se utiliza **commit** para guardar los cambios.
- **Líneas 6-7:** se cierra el cursor y por último se cierra la conexión con la base de datos.

```

1 db = pymysql.connect("Servidor", "Usuario", "Password", "BaseDatos")
2 db.begin()
3 cur = db.cursor()
4 cur.execute("SENTENCIA")
5 resultado = cur.fetchall()
6 db.commit()
7 cur.close()
8 db.close()

```

Listing 6.7: Ejecutando una sentencia SQL en Python

6.2.3. Añadiendo Handlers dinámicos

En la función **main** de nuestro código se especifican todos los **handlers** con todas las funciones que vamos a usar. Pero debemos tener en cuenta que hay ocasiones en las que debemos de agregar manejadores de forma dinámica, ya que el contenido del bot va a ser dinámico y es el contenido que ofrecen los gimnasios. Por ejemplo, un gimnasio propone un nuevo reto. Entonces el usuario va a la sección de retos. Debemos crear un **handler** para consultar y poder apuntarse a ese nuevo reto (ver Figura 6.8):

- **Línea 1:** se recorren todos los identificadores de retos que tiene el gimnasio actualmente.
- **Línea 2:** se crea un botón con el nombre del reto y cuyo **callback_data** es el texto **inicio_ver_retos_id_reto**.
- **Líneas 3-4:** se añade el botón creado a la lista de botones que se va a enviar al usuario.
- **Línea 5:** se crea un **CallbackQueryHandler** que llama a la función **ver_reto** cuando se pulsa el botón que tiene asociado el **callback_data** que hemos creado antes.
- **Línea 6:** se crea un **CallbackQueryHandler** que llama a la función **ver_reto.apuntarse** cuando se pulsa el botón que tiene asociado el **callback_data**, que se creará en la función **ver_reto**.
- **Líneas 8-9:** se añade el **CallbackQueryHandler** creado en la línea 5 a la conversación cuando el bot está en el contexto **INICIO_RETOS_VER**.
- **Líneas 11-12:** se añade el **CallbackQueryHandler** creado en la línea 6 a la conversación cuando el bot está en el contexto **INICIO_RETOS_VER_RETO**.

```

1 for id_reto in resultado:
2     button = InlineKeyboardButton(name_button, callback_data="inicio_retos_ver_"+
3         str(id_reto[0]))
        keyboard = []

```

```

4     keyboard.append(button)
5     callback_query_retos_ver = CallbackQueryHandler(ver_reto, pattern="
6         inicio_retos_ver_"+str(id_reto[0]))
7     callback_query_retos_ver_apuntarse = CallbackQueryHandler(ver_reto_apuntarse,
8         pattern="inicio_retos_ver_apuntarse_"+str(id_reto[0]))
9
10    if not callback_query_retos_ver in conv_handler.states[INICIO_RETOS_VER]:
11        conv_handler.states[INICIO_RETOS_VER].append(callback_query_retos_ver)
12
13    if not callback_query_retos_ver_apuntarse in conv_handler.states[
14        INICIO_RETOS_VER_RETO]:
15        conv_handler.states[INICIO_RETOS_VER_RETO].append(
16            callback_query_retos_ver_apuntarse)

```

Listing 6.8: Creación de **handlers** en dinámico

Se ha creado una única función que sirve para manejar muchos botones con un **callback_data** diferente. Esto es posible debido a que se puede obtener el **callback_data** del botón que ha pulsado el usuario. Por ejemplo, la función **ver_reto** se ejecutará cuando el botón **callback_data** de un botón tenga la estructura **inicio_ver_retos_id_reto**. El identificador del reto se podrá obtener usando el código que se muestra en la Figura 6.9:

- **Línea 1:** se obtiene la consulta del botón pulsado generada por el usuario.
- **Línea 2:** se obtiene el texto asociado al **callback_data** del botón pulsado.
- **Líneas 3-4:** en este caso, el **callback_data** es **inicio_ver_retos_id_reto**. Se utiliza la función **split()** para separar el texto en función de los guiones bajos, y nos quedamos con el último elemento que es el identificador del reto, que es el que nos servirá para coger toda la información de ese reto específico en la base de datos.

```

1 query = update.callback_query
2 id_reto_callback = query.data
3 id_reto = id_reto_callback.split('_',4)
4 id_reto = id_reto[3]

```

Listing 6.9: Obtener el **callback_data** del botón que ha pulsado el usuario

6.2.4. Añadiendo Fallbacks

Como ya se adelantó al principio del Capítulo 6, el bot responderá dependiendo del contexto en el que se encuentre. Por ejemplo, el usuario pulsa el botón de *Empezar Imagym* (ver Figura 6.4), el contexto pasará a ser **INICIO**. Este contexto estará formado por todos los **handlers** posibles en ese estado (en este caso, los botones del menú inicial), mostrados en la Figura 6.10. Supongamos que el usuario navega por los menús, enviando el bot más mensajes con diferentes botones. El usuario puede ir hacia arriba de la conversación y pulsar un botón del menú inicial, pero ya no estaría en ese contexto.

```

1 INICIO: [
2     MessageHandler(Filters.text & (~Filters.command), any_message),
3     CallbackQueryHandler(show_inicio_peso, pattern='inicio_peso'),
4     CallbackQueryHandler(show_inicio_cardio, pattern='inicio_cardio'),
5     CallbackQueryHandler(show_inicio_retos, pattern='inicio_retos'),
6     CallbackQueryHandler(show_inicio_ejercicio, pattern='inicio_ejercicio'),
7     CallbackQueryHandler(show_inicio_rutinas, pattern='inicio_rutinas'),
8     CallbackQueryHandler(show_inicio_ficha, pattern='inicio_ficha'),
9     CallbackQueryHandler(show_inicio, pattern='show_inicio'),
10    CallbackQueryHandler(show_inicio_soporte, pattern='inicio_soporte'),
11 ]

```

Listing 6.10: **Handlers** asociados al estado **INICIO**

Para este tipo de situaciones existen los **fallbacks**. Consiste en una lista de todos los **handlers** que existen pero que el usuario ejecuta cuando no está en el contexto adecuado. Entonces estos **handlers** tendrán el mismo **pattern** pero diferente función. Por ejemplo, para el caso anterior, los **handlers** en la lista de **fallbacks** serán los mostrados en la Figura 6.11. Todos ellos tienen el mismo **pattern** que en los **handlers** de la Figura 6.10 pero llamarán a la función **usuario_pulsa_boton_anterior**. Esta función enviará un mensaje al usuario diciendo que no se le permite pulsar el botón en estos momentos.

```

1 fallbacks=[ 
2     CallbackQueryHandler(usuario_pulsa_boton_anterior, pattern='start_menu'),
3     CallbackQueryHandler(usuario_pulsa_boton_anterior, pattern='inicio_peso'),
4     CallbackQueryHandler(usuario_pulsa_boton_anterior, pattern='inicio_cardio',
5                           '),
6     CallbackQueryHandler(usuario_pulsa_boton_anterior, pattern='inicio_retos')
7     ,
8     CallbackQueryHandler(usuario_pulsa_boton_anterior, pattern='
9         inicio_ejercicio'),
10    CallbackQueryHandler(usuario_pulsa_boton_anterior, pattern='inicio_rutinas
11   '),
12    CallbackQueryHandler(usuario_pulsa_boton_anterior, pattern='inicio_ficha')
13    ,
14    CallbackQueryHandler(usuario_pulsa_boton_anterior, pattern='show_inicio')
15 ]

```

Listing 6.11: **Fallbacks**

6.3. Comunicación entre ImagymBot y la plataforma web

Existe una comunicación bidireccional entre **ImagymBot** y la plataforma web. Los monitores de un gimnasio disponen de su propio usuario y contraseña para acceder a la web. Desde ella podrán proponer nuevos retos, nuevos ejercicios del mes y nuevas rutinas de entrenamiento. Esto lo podrán hacer desde sencillos formularios. Se dispone de una base de datos compartida, si bien se pueden habilitar temporalmente otras vías

de importación/exportación, así como hojas de cálculo empleando formato **csv**. Para introducir estos datos en las tablas correspondientes de la base de datos se empleará una estructura de **archivos de entrada csv** que se explican desde la Sección 6.3.1 a la Sección 6.3.3.

La plataforma web también necesita información del bot, en concreto necesita saber qué usuarios han superado los retos para colocarles la insignia en su perfil. También necesita saber la puntuación conseguida en los ejercicios del mes para mostrar un ranking de todos los usuarios. La estructura de los **archivos de salida** del bot al gimnasio se explican desde la Sección 6.3.4 a la Sección 6.3.5.

6.3.1. Archivo de entrada: Ejercicios del mes

Un monitor propone un nuevo ejercicio del mes para su gimnasio. En la interfaz web se le pedirá llenar un formulario facilitando los siguientes datos:

- El mes en el que se realizará el ejercicio
- Actividad cardio
- Objetivo

Entonces el archivo **csv** tendrá la siguiente estructura:

objetivo,fecha_inicio,fecha_fin,fecha_hoy,id_gimnasio,id_actividad_cardio

Por ejemplo, supongamos que hoy es día **16 de junio de 2020**. El gimnasio cuyo identificador es *kronos* propone un ejercicio del mes para el próximo mes de **septiembre**, cuyo objetivo es realizar un mínimo de **300 minutos** en la máquina de elíptica, cuyo identificador es **eliptica**. La línea del archivo csv quedaría de la siguiente manera:

"300 tiempo",01-09-2020,30-09-2020,16-06-2020,kronos,eliptica

Si fuera un objetivo de kilómetros, el objetivo sería *"300 distancia"*. Si fuera un objetivo de calorías, el objetivo sería *"300 calorias"*.

6.3.2. Archivo de entrada: Retos

Esta sección tendrá dos archivos **csv**: uno para la información general de reto y otro para el calendario asociado a ese reto.

Un monitor o monitora propone un nuevo reto de un ejercicio. En la web primero deberá llenar un formulario en el que proporciona los siguientes primeros datos:

- Fecha de inicio
- Fecha final
- Ejercicio
- Nivel de intensidad

El archivo **csv** de los retos tendrá la siguientes estructura:

nivel,fecha_inicio,fecha_fin,fecha_hoy,id_entrenador,id_ejercicio

Supongamos que hoy es día **18 de junio de 2020** y que el monitor cuyo id en la base de datos es **juan_castillo** propone un nuevo reto de flexiones, cuyo identificador de ejercicio es **flexiones**. Empieza el **1 de julio de 2020** y termina el **07 de julio de 2020**. El nivel de intensidad es para principiantes, es decir, nivel **1**. La línea en el archivo **csv** quedaría de la siguiente manera:

1,01-07-2020,07-07-2020,18-06-2020,juan_castillo,flexiones

De esta manera se genera un identificador del reto. A continuación se le abrirá una nueva interfaz web de un calendario con la duración del reto. En este caso, el calendario tendrá 15 días que irán desde el 01-07-2020 hasta el 15-07-2020. En cada día deberá anotar un número, que serán las repeticiones de flexiones que los usuarios deberán realizar cada día. La estructura del archivo **csv** de este calendario será:

id_reto,dia,repeticiones

Donde día será un número del 1 al 7, siendo 1 el día 01-07-2020 y 7 el día 07-07-2020. Si el monitor no especifica unas repeticiones para un día, las repeticiones se pondrán en **NULL**. Suponiendo que el identificador del reto que se ha generado es **flexiones_nivel1**, un ejemplo de las líneas del archivo **csv** serían:

*flexiones_nivel1,1,10
flexiones_nivel1,2,10*

```
flexiones_nivel1,3,15  
flexiones_nivel1,4,NULL  
flexiones_nivel1,5,15  
flexiones_nivel1,6,20  
flexiones_nivel1,7,20
```

Se puede ver que el día 4, que correspondería al día 04-07-2020, no hay repeticiones, con lo cual ese día habría un descanso para los usuarios del reto.

6.3.3. Archivo de entrada: Rutinas y entrenamiento

Esta sección también consta de dos archivos **csv**: uno para la rutina y el entrenador y otro para los ejercicios asociados a esa rutina.

Un monitor o monitora propone una nueva rutina. En la web debe proporcionar únicamente la fecha de inicio y fecha fin de la rutina. Si la fecha final se deja en blanco, se insertará como **NULL**. El archivo **csv** tendrá la siguiente estructura:

```
fecha_inicio,fecha_fin,fecha_hoy,id_entrenador
```

Si el monitor cuyo id es **juan_castillo** propone una rutina cuya fecha de inicio es el día **1 de julio de 2020** y hoy es dia **16 de junio de 2020**, la línea en el primer archivo **csv** será:

```
01-07-2020,NULL,16-06-2020,juan_castillo
```

Ahora se generará un identificador para esta rutina. Al monitor se le mostrará una nueva interfaz web con todos los ejercicios disponibles en su gimnasio y que puede añadir. Comenzará a añadir ejercicios por día de la semana: primero el lunes, luego el martes, y así sucesivamente. Cada vez que añada un ejercicio, deberá proporcionar las repeticiones de dicho ejercicio en formato de texto. La estructura de este archivo **csv** será:

```
id_rutina,id_ejercicio,dia,repeticiones
```

El identificador de la rutina que se genera es **rutina_juan_castillo_5**. Para un ejemplo sencillo, supongamos que este entrenador añade los siguientes ejercicios:

- Lunes:
 - Press con mancuernas: 4x20

- Sentadillas: 3x15
- Miércoles
 - Lunges: 5x20
 - Dominadas: 5x20

Las líneas que se añaden al segundo archivo **csv** serían las siguientes:

```
rutina_juan_castillo_5,press_con_mancuernas,lunes,4x20
rutina_juan_castillo_5,sentadillas,lunes,3x15
rutina_juan_castillo_5,lunges,miercoles,5x20
rutina_juan_castillo_5,dominadas,miercoles,5x20
```

6.3.4. Archivo de salida: Usuarios que han superado un reto

La base de datos contiene una tabla con la información de cada usuario en cada reto. El archivo **csv** que se genera tiene la siguiente estructura:

id_reto,id_usuario,estado

- **id_reto:** el identificador del reto
- **id_usuario:** el identificador del usuario, que es su alias de **Telegram**
- **estado:** su estado en el reto, que puede ser:
 - **A:** el usuario está **apuntado** el reto pero aún no ha empezado
 - **C:** el usuario ha **conseguido** el reto
 - **D:** el usuario ha sido **descalificado** del reto
 - **R:** el reto ha empezado y aún no ha acabado, y el usuario sigue **calificado**

6.3.5. Archivo de salida: Puntuación de los usuarios en el ejercicio del mes

La base de datos contiene una tabla con la información de todos los usuarios apuntados a un ejercicio del mes y la puntuación que llevan acumulada. El archivo **csv** que se exportará tendrá la siguiente estructura:

id_ejercicio_del_mes,id_usuario,puntuacion,estado

- **id_ejercicio_del_mes:** el identificador del ejercicio del mes
- **id_usuario:** el identificador del usuario, que es su alias de **Telegram**
- **estado:** su estado en el ejercicio del mes, que puede ser:
 - **A:** el usuario está **apuntado** al ejercicio del mes pero aún no ha empezado
 - **C:** el ejercicio del mes ha **terminado**, la puntuación es la puntuación final
 - **D:** el usuario ha decidido **abandonar** el ejercicio del mes
 - **R:** el ejercicio del mes ha empezado y aún no ha acabado, y el usuario sigue **participando**

6.4. Problemas de programación más frecuentes

Durante el desarrollo de **ImagymBot** he contado con la ayuda de conocidos y familiares que me han ayudado a encontrar errores de programación y sobre todo a mejorar la interacción del usuario con el bot. En esta sección explico algunos de los errores de programación más frecuentes.

6.4.1. Error al insertar/actualizar/borrar en MySQL

Hay un error que ha sido bastante frecuente a la hora de actualizar o borrar una tabla en la base de datos. Por ejemplo, cuando un usuario establece un objetivo de una actividad cardio, al final debe confirmar el objetivo o no. Los pasos para establecer un objetivo de cardio son los siguientes:

1. El usuario selecciona la actividad cardio
2. El usuario indica el objetivo
3. El usuario confirma el objetivo

El objetivo se inserta en la base de datos en el paso 1, y a partir de ahí se modificarán las columnas correspondientes en los siguientes pasos. Si el usuario **NO** confirma el objetivo, se debe borrar de la base de datos. Para ello tenía la sentencia de la Figura 6.12:

```
1 DELETE FROM Objetivo_personal_cardio WHERE id_usuario='"+username+"', AND  
fecha_inicio=(SELECT MAX(c2.fecha_inicio) FROM Objetivo_personal_cardio c2  
WHERE id_usuario='"+username+"' AND estado='P'))
```

Listing 6.12: Sentencia que borra el objetivo personal de cardio más reciente de un usuario

El error que daba es el siguiente:

You can't specify target table 'Objetivo_personal_cardio' for update in FROM clause

Buscando información parece ser que **MySQL** no deja ejecutar sentencias de insertar/actualizar/borrar donde se referencia a una misma tabla en una subsentencia. La solución fue hacer la misma sentencia pero usando **SELECT** en lugar de **DELETE**. Con el **SELECT** se coge el identificador de la fila y después ejecuto una sentencia **DELETE** para borrar la fila con ese identificador. Las dos sentencias se muestran en la Figura 6.13.

```

1 cur.execute("SELECT id_objetivo_personal FROM Objetivo_personal_cardio WHERE
    id_usuario='"+username+"', AND fecha_inicio=(SELECT MAX(c2.fecha_inicio) FROM
    Objetivo_personal_cardio c2 WHERE id_usuario='"+username+"', AND estado='P')")
2 resultado = cur.fetchall()
3
4 if resultado:
5     id_objetivo_personal = resultado[0][0]
6     cur.execute("DELETE FROM Objetivo_personal_cardio WHERE
    id_objetivo_personal="+str(id_objetivo_personal)+";")
7     db.commit()

```

Listing 6.13: Setencias **SELECT** y **DELETE** para borrar el objetivo personal de cardio del usuario más reciente

6.4.2. Error al mostrar la ficha personal

Este error fue bastante frecuente al inicio porque el error causante era una sentencia mal hecha que se usaba bastante en diferentes funcionalidades y dio lugar a muchas situaciones de error. La más comentada por los usuarios fue que no cargaba la ficha personal. Esto es porque en la ficha personal muestra el peso más reciente del usuario. Entonces, la sentencia que usaba para seleccionar el peso más reciente es la que se muestra en la Figura 6.14.

```

1 SELECT peso,fecha FROM Peso WHERE id_usuario='"+username_user+"', AND fecha=(SELECT
    MAX(p2.fecha) FROM Peso p2 WHERE id_usuario='"+username_user+"');

```

Listing 6.14: Sentencia para conseguir el peso más reciente (versión errónea)

El problema de esta sentencia es que coge el peso de la fila con la fecha más reciente, pero no necesariamente esa fila va a tener un peso. Puede ser que el usuario ese día solo registrara su porcentaje de grasa, con lo cual la fila con la fecha más reciente tendría el peso y músculo a **NULL**. Entonces al mostrar el peso al usuario daba un error de

que no se podía mostrar algo que estaba **NULL**. La solución fue muy sencilla y fue simplemente añadir una condición más: que el peso no sea nulo (ver Figura 6.15).

```
1 SELECT peso,fecha FROM Peso WHERE id_usuario='"+username_user+"', AND fecha=(SELECT  
    MAX(p2.fecha) FROM Peso p2 WHERE id_usuario='"+username_user+"', AND peso IS  
    NOT NULL;")
```

Listing 6.15: Sentencia para conseguir el peso más reciente (versión buena)

Capítulo 7

Pruebas

Para evaluar la usabilidad de **ImagymBot** se ha utilizado un cuestionario SUS (**System Usability Scale**). Este método se ha utilizado durante casi 30 años, siendo inventado por John Brooke en 1986 [39]. Es una de las maneras más eficientes y precisas de evaluar nuestro sistema. El cuestionario SUS cuenta con dos características muy importantes:

- **Barato:** no requiere demasiados recursos, es un cuestionario que se puede hacer tanto a papel como **online**.
- **Rápido:** hay plantillas disponibles con las preguntas de este cuestionario, con lo cual simplemente necesitamos copiar y pegar las preguntas.

El cuestionario cuenta con 10 preguntas, y cada una de ellas se deberá responder con un número en una escala del 1 al 5, significando 1 que el usuario está completamente en desacuerdo y 5 que el usuario está completamente de acuerdo.

Una vez se tienen todas las respuestas de todos los participantes, se calcula la puntuación de cada usuario:

- Por cada pregunta impar, se resta 1 de la puntuación
- Por cada pregunta par, se resta 5 al valor de la respuesta
- Se suman todos los valores resultantes y se multiplica todo por 2.5

El resultado que se obtiene está sobre 100. No es un porcentaje, pero es una forma fácil de ver el resultado. De acuerdo al informe de Nathan Thomas [39], el significado de la

puntuación es el siguiente:

- **80.3 o más:** a los usuarios les encanta la aplicación y la recomendarían a sus amigos
- **68 aproximadamente:** está bien la aplicación, pero se pueden mejorar algunos aspectos
- **51 o menos:** se debe considerar el mejorar urgentemente la aplicación y priorizar la usabilidad

El cuestionario está formado por las siguientes 10 preguntas:

1. Creo que me gustará visitar con frecuencia esta aplicación
2. Encontré la aplicación innecesariamente compleja
3. Me pareció fácil de usar esta aplicación
4. Creo que necesitaría del apoyo de un experto para recorrer la aplicación
5. Encontré las funciones de la aplicación bastante bien integradas
6. Pensé que había demasiada inconsistencia en la aplicación
7. Imagino que la mayoría de las personas aprenderían a utilizar fácilmente la aplicación
8. Encontré la aplicación muy grande al recorrerla
9. Me sentí muy confiado/a en el manejo de la aplicación
10. Necesito aprender muchas cosas antes de manejarla en la aplicación

Este cuestionario se ha pasado a usuarios finales de la aplicación y se les ha pedido que lo rellenen. Las respuestas de cada usuario se muestran en la Figura 7.1 junto con el cálculo de la puntuación final obtenida. Las filas son las respuestas de un usuario y las columnas son las respuestas de una misma pregunta.

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	PUNTUACIÓN
3	1	5	1	4	2	4	3	4	1	80
4	2	4	2	5	2	5	2	5	2	82,5
4	2	5	1	4	3	5	4	3	1	75
5	1	5	1	5	1	5	1	4	1	97,5
5	2	5	1	5	1	5	3	5	2	90
4	2	4	1	4	1	4	4	4	2	75
5	1	5	1	5	1	5	1	5	1	100
4	1	4	1	5	1	5	2	4	1	90
5	2	5	1	5	1	5	1	5	1	97,5
5	1	5	1	4	2	5	1	5	1	95
5	1	5	1	4	2	5	1	5	1	95
5	1	5	2	5	2	5	1	4	4	85
5	1	5	2	5	2	5	1	4	4	85

Figura 7.1: Puntuaciones del cuestionario SUS

En general hay puntuaciones bastante altas, por encima de 80. Sí que es cierto que para algunos usuarios la puntuación es un poco más baja, pero nunca menor que 70. Hay algunas cosas que se podrían mejorar. En concreto, los usuarios encuentran la aplicación un poco grande de recorrer, pero se obtiene una puntuación muy positiva.

Capítulo 8

Manual de uso y distribución

8.1. Manual de uso

En esta Sección se va a presentar un manual de usuario para explicar el funcionamiento de **ImagymBot**. A parte de esta documentación, se ha hecho un pequeño vídeo tutorial explicando todas las funcionalidades de manera muy resumida. Dicho vídeo se encuentra en el siguiente enlace:

<https://youtu.be/0FhwvvKksKw>

8.1.1. Inicio

Para iniciar el bot se debe buscar en **Telegram** el usuario **@ImagymBot**. Al iniciarlo, se pedirá una contraseña del gimnasio del usuario. Esta contraseña la debe repartir el propio gimnasio. Para probar el bot, se puede introducir la contraseña *clientes*. Una vez introducida, se le dará la bienvenida y se le concederá la contraseña para acceder a la web. El usuario se encontrará en el menú de inicio del bot, tal y como se muestra en la Figura 8.1.



Figura 8.1: Inicio de **ImagymBot**

8.1.2. Mi objetivo de peso

En *Mi objetivo de peso* los usuarios pueden hacer tres cosas:

1. Anotar el peso, porcentaje de grasa y/o porcentaje de músculo diario
2. Establecer un objetivo de peso, porcentaje de grasa y/o porcentaje de músculo
3. Ver la evolución del peso, porcentaje de grasa, porcentaje de músculo e IMC

El submenú que se muestra al usuario se encuentra en la Figura 8.2. El bot primero enviará dos mensajes: en el primero mostrará los datos de la última vez que registró algún dato, y en el segundo mensaje se mostrará el objetivo del usuario.

Anotar datos

El proceso de anotar datos se muestra en la Figura 8.3:

1. Se le muestra un menú al usuario para elegir lo que quiere anotar
2. El bot pide al usuario que le diga su peso (en este ejemplo se ha elegido anotar el peso). Como el usuario no tiene registrada su altura, también se le pide esta información por si el usuario quiere que se calcule su IMC, pero no es obligatorio.
3. Una vez anotado, se volverá al menú de anotar datos. Las opciones con un ícono de un lápiz son aquellas que aún no se han registrado en el día de hoy.

Figura 8.2: Menú de *Inicio > Mi objetivo de peso*

Figura 8.3: Proceso de anotar datos

Establecer objetivo

Se puede establecer un objetivo de peso, grasa o músculo. El proceso de establecer un objetivo se muestra en la Figura 8.4:

1. El usuario elige qué objetivo quiere. En este caso solo nos deja establecer un objetivo de peso porque el usuario no ha registrado ningún dato de grasa o músculo.
2. El usuario escribe su objetivo de peso y elige un plazo.
3. Se le mostrará al usuario un resumen de lo que ha elegido y confirmará su objetivo.

Una vez establecido el objetivo, el bot volverá al menú *Inicio > Mi objetivo de peso* y en el segundo mensaje se mostrará toda la información del objetivo (ver Figura 8.5).



Figura 8.4: Proceso de establecer un objetivo de peso

Evolución

Se puede ver la evolución del peso, grasa, músculo o IMC siempre que haya datos disponibles. En este caso se ha seleccionado el peso, y lo primero que mostrará el bot será una gráfica con la evolución de todo el peso registrado por el usuario (ver Figura 8.6).

El usuario podrá filtrar esta evolución mediante un rango de fechas usando el coman-

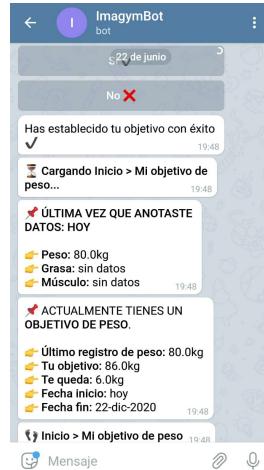


Figura 8.5: Menú de *Inicio > Mi objetivo de peso* con un objetivo de peso establecido



Figura 8.6: Menú de *Inicio > Evolución*

do **/rango**. Si el usuario introduce dos fechas, se mostrará la evolución desde la primera fecha hasta la segunda fecha. Si el usuario introduce una fecha, se mostrará la evolución desde esa fecha hasta el día de hoy (ver Figura 8.7).

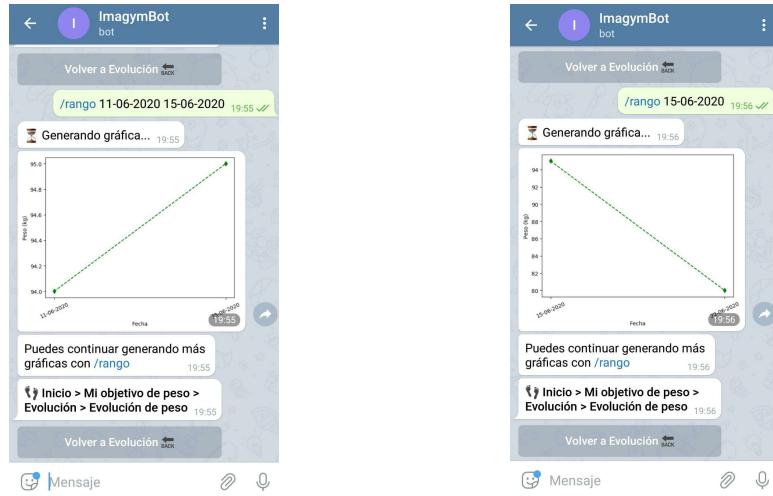


Figura 8.7: Filtrar la evolución del peso por fecha

8.1.3. Mi objetivo de actividades cardio

En *Mi objetivo de actividades cardio* los usuarios pueden hacer las siguientes tareas:

1. Registrar una actividad cardio del gimnasio
2. Ver todo el cardio que se hizo un día específico
3. Establecer un objetivo de una actividad cardio

El submenú que se muestra al usuario se encuentra en la Figura 8.8. El bot envía dos mensajes: en el primero muestra las actividades cardio que ha registrado el usuario hoy; en el segundo mostrará el informe de su objetivo personal de cardio. En este caso, el usuario aún no ha registrado ninguna actividad cardio hoy ni tampoco tiene un objetivo personal de cardio.



Figura 8.8: Menú de *Inicio > Mi objetivo de actividades cardio*

Registrar cardio

El proceso de registrar cardio se muestra en la Figura 8.9:

1. El usuario selecciona la actividad cardio que quiere registrar
2. Se introducen los minutos, kilómetros, intensidad y calorías mediante el comando **/cardio**
3. El bot envía un resumen de lo que se va a registrar y el usuario confirma el registro

Establecer objetivo

Un objetivo personal de cardio tiene una duración de 1 mes a partir del día que el usuario lo establece. Consiste en hacer un mínimo de minutos, kilómetros o calorías de cualquier actividad cardio. El proceso de establecer un objetivo personal de cardio se muestra en la Figura 8.10:

1. El usuario selecciona la actividad cardio del objetivo
2. El usuario indica los minutos, kilómetros o calorías mediante los comandos **/minutos**, **/distancia** o **/calorías**, respectivamente
3. El bot envía un resumen del objetivo que se va a establecer y el usuario confirma el objetivo



Figura 8.9: Proceso de registrar cardio



Figura 8.10: Proceso de establecer un objetivo personal de cardio

Una vez que se ha establecido el objetivo personal de cardio, cada vez que el usuario registre cardio que influya en dicho objetivo se sumará a su contador. En este ejemplo, se ha establecido un objetivo de un mínimo de 200 minutos en la cinta de correr. En la Figura 8.11 se muestra lo que pasa cuando se registran 40 minutos en la cinta de correr.



Figura 8.11: Sumando minutos al objetivo personal de cardio

Ver mis registros en cardio

El bot enviará un primer mensaje al usuario con todo el cardio que ha registrado en el día actual (ver Figura 8.12). El usuario puede consultar las actividades cardio que hizo en un determinado día usando el comando **/rango**.

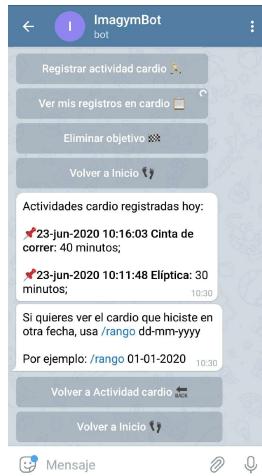


Figura 8.12: *Inicio > Mi objetivo de actividades cardio > Ver mis registros en cardio*

8.1.4. Retos

En *Retos* los usuarios pueden hacer las siguientes tareas:

1. Ver los próximos retos del gimnasio y apuntarse o desapuntarse de ellos
2. Si el usuario está realizando un reto:
 - a) Puede anotar su progreso
 - b) Puede descalificarse del reto
 - c) Puede ver el calendario del reto con su progreso actual
3. Puede consultar su historial de retos

Un reto consiste en hacer un determinado ejercicio a lo largo de un período de tiempo. Se harán un número de repeticiones cada día. Los usuarios deben anotar cada día del reto que han hecho las repeticiones. Si algún día no se anota el progreso, se les descalificará del reto. El objetivo es completar todos los días del reto.

El submenú que se muestra al usuario se encuentra en la Figura 8.13. Si el usuario está apuntado a un reto que se está realizando actualmente, el bot enviará un primer mensaje con la información de dicho reto. Si el usuario está apuntado a retos que se realizarán próximamente, se enviará un segundo mensaje con la información del que sea el reto más próximo.



Figura 8.13: *Inicio > Retos*

Anotar progreso y Calendario de mi reto actual

En la Figura anterior el bot envía un mensaje de que actualmente el usuario está apuntado a un reto de **lunge**s y que debe anotar su progreso. De no anotar su progreso en todo el día, al día siguiente se le descalificará del reto.

En la Figura 8.14 se muestra el proceso de anotar el progreso del reto. También se muestra una captura de lo que el bot envía al usuario cuando este pulsa el botón de **Calendario de mi reto actual**, en el que se muestra un calendario del reto con las repeticiones que se debe hacer cada día y en azul aquellos días que ya se han superado.

Ver próximos retos

Los usuarios pueden ver los próximos retos que los monitores del gimnasio tienen preparados para ellos. En la Figura 8.15 se muestra la interfaz que se muestra a los usuarios cuando ven un reto. Primero, se les muestra una lista de botones con todos los próximos retos, señalando con un ícono de **check** aquellos retos a los que ya se ha apuntado el usuario (en este caso está apuntado a los dos).

Cuando un usuario pulsa un reto, se le mostrará el calendario asociado a ese reto y el bot le enviará la información necesaria. El usuario puede apuntarse o desapuntarse



Figura 8.14: Anotando el progreso del reto

simplemente pulsando el botón correspondiente. El bot también informa al usuario del número de personas del gimnasio que ya están apuntadas al reto.



Figura 8.15: Viendo próximos retos

Historial de retos

El usuario puede consultar el historial de todos los retos en los que ha participado. Se le mostrará una lista de botones con el nombre del reto y un ícono que indicará si superó

el reto o no, tal y como se muestra en la Figura 8.16. Al pulsar un reto, se mostrará el calendario con los días que superó el usuario y la cantidad de usuarios que llegaron al final de reto.

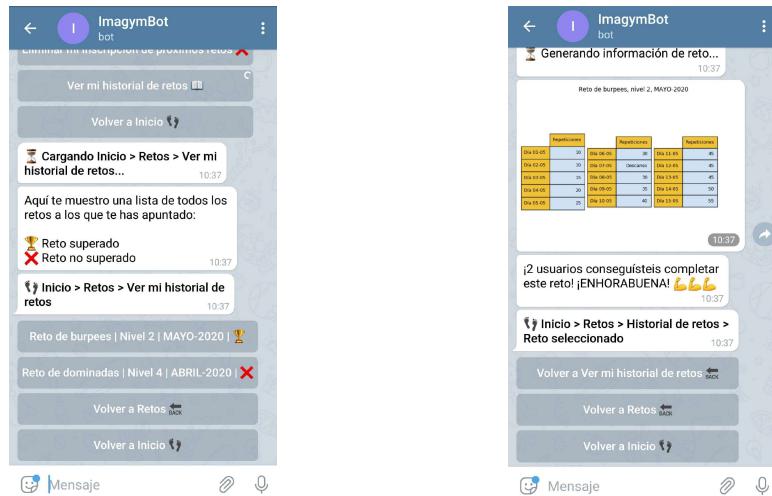


Figura 8.16: Historial de retos del usuario

Eliminar inscripción de próximos retos

El usuario puede eliminar su inscripción de los próximos retos a los que está apuntado sin necesidad de entrar en la información del reto, como se vio en la Sección 8.1.4. El usuario elegirá el reto del que quiere desapuntarse y simplemente deberá confirmar la operación (ver Figura 8.17).

8.1.5. Ejercicio del mes

En *Ejercicio del mes* los usuarios pueden hacer las siguientes tareas:

1. Apuntarse al ejercicio del mes
 2. Registrar cardio
 3. Consultar el ranking del ejercicio del mes actual
 4. Consultar su historial de ejercicios del mes

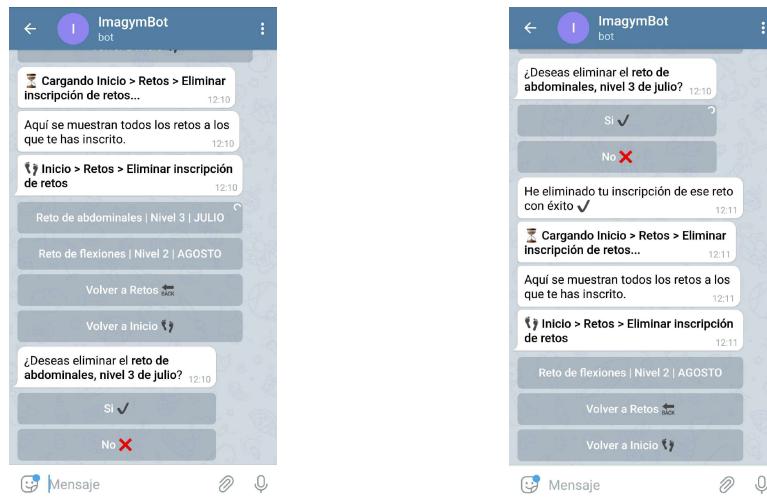


Figura 8.17: Eliminar inscripción del usuario de próximos retos

Un ejercicio del mes consiste en que el gimnasio propone hacer un mínimo de minutos, kilómetros o calorías durante un mes de una determinada actividad cardio. Cada vez que los usuarios registren cardio que influya en el ejercicio del mes, se les sumará puntos a su marcador. Una vez que lleguen al mínimo propuesto por el gimnasio, se sumarán puntos adicionales en función de su IMC.

El submenú que se muestra al usuario se encuentra en la Figura 8.18. El bot enviará un primer mensaje con la información del ejercicio de este mes. En este caso, el ejercicio de este mes consiste en hacer un mínimo de 700 minutos en las máquinas escaladoras. En el segundo mensaje aparecerá la información del ejercicio que dará lugar el próximo mes junto con el número de personas que ya hay apuntadas.

Ranking actual del ejercicio de este mes

Los usuarios pueden consultar en cualquier momento el ranking del ejercicio del mes actual. Se mostrará un top 10 de los usuarios con más puntuación (ver Figura 8.19). En negrita se mostrará nuestro usuario. Si no aparecemos en el top 10, se mostrará una línea adicional con nuestra puntuación.

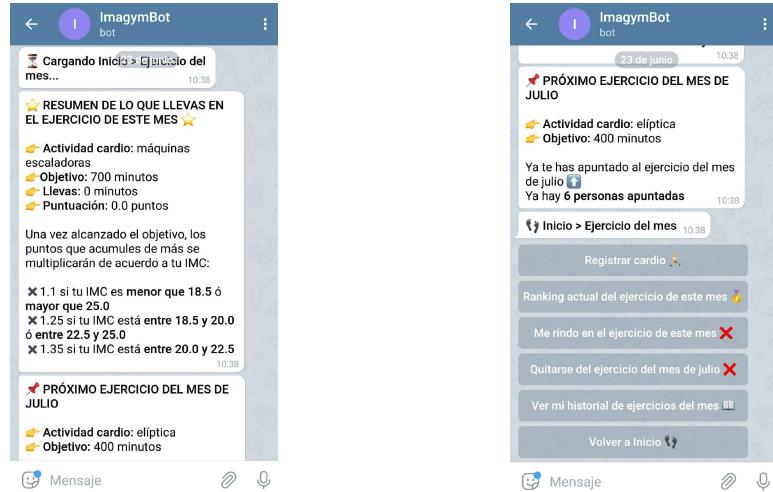
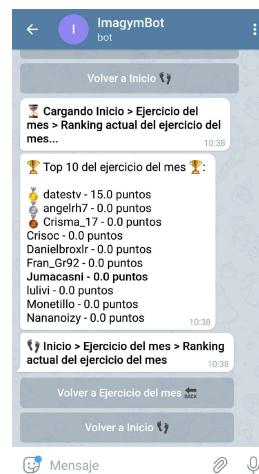
Figura 8.18: *Inicio > Ejercicio del mes*

Figura 8.19: Ranking del ejercicio del mes actual

Consultar historial de ejercicios del mes

Se le mostrará al usuario una lista de todos los ejercicios del mes en los que ha participado (ver Figura 8.20). Al pulsar un ejercicio del mes, se le mostrará un top 10 de los usuarios que obtuvieron más puntuación, de manera similar a como se hizo en la Sección anterior.

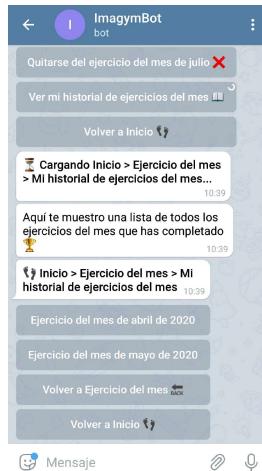


Figura 8.20: Consultar historial de ejercicios del mes

Registrar cardio que suma puntos al marcador del ejercicio del mes

Los usuarios pueden registrar el cardio desde *Mi objetivo de actividades cardio* o bien desde este menú. De cualquier forma, si el usuario registra cardio que va a sumar puntos a su puntuación en el ejercicio del mes, se le pedirá una prueba que verifique dicha puntuación.

En la Figura 8.21 se muestra un ejemplo de lo que pasa cuando se registran 50 minutos en las máquinas escaladoras (recordemos que el objetivo de este mes es hacer 700 minutos en las máquinas escaladoras). El bot informará al usuario de que va a sumar 50 minutos al ejercicio del mes. Cuando el usuario confirme esto, se le pedirá una foto de la máquina escaladora en la que se vea que se han hecho 50 minutos. Cuando un moderador externo apruebe la foto, se sumarán los puntos a su marcador.

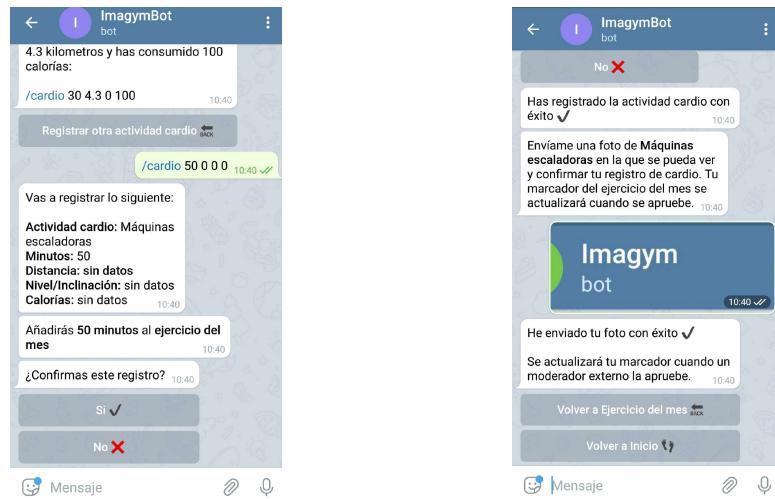


Figura 8.21: Registrando cardio que afecta a la puntuación del ejercicio del mes

8.1.6. Rutinas y entrenamiento

En *Rutinas y entrenamiento* los usuarios pueden hacer las siguientes tareas:

1. Ver las rutinas disponibles de los monitores del gimnasio
2. Anotar rutina de hoy
3. Consultar el entrenamiento que se hizo otro día

Una rutina de un monitor consiste en una serie de ejercicios para hacer cada día de la semana. El usuario puede anotar los ejercicios de las rutinas que quiera, es decir, puede anotar que ha hecho un ejercicio de la rutina de un monitor y que ha hecho dos ejercicios de la rutina de otro monitor. El usuario solo puede anotar la rutina de aquellas rutinas que tengan ejercicios disponibles para el día actual, es decir, si hoy es miércoles, el usuario podrá anotar aquellas rutinas que tengan ejercicios para hacer los miércoles.

El submenú que se muestra al usuario se encuentra en la Figura 8.22. Se le mostrará al usuario los ejercicios que lleva anotados hoy. En este caso, el usuario aún no ha anotado ningún ejercicio.



Figura 8.22: *Inicio > Rutinas y entrenamiento*

Ver rutinas de entrenamiento

En la Figura 8.23 se muestra el submenú que sale para ver todas las rutinas de entrenamiento. Se mostrarán botones con el nombre del monitor de la rutina. Al pulsar una rutina, el bot enviará un mensaje con los ejercicios que hay que hacer cada día de la semana y sus respectivas repeticiones. Además, el usuario pulsar (o escribir) los comandos que se muestran en azul para ver una imagen tutorial del ejercicio.

El usuario puede agregar las rutinas a favoritos simplemente pulsando el botón correspondiente. Esto influirá en la forma de anotar la rutina de hoy que se explicará en la siguiente Sección. En la Figura 8.24 se muestra una captura de lo que se muestra cuando el usuario pulsa el botón de agregar a favoritos y otra captura que muestra una imagen tutorial de un ejercicio.

Anotar rutina de hoy

En este submenú se mostrarán todas las rutinas que tienen ejercicios para hacer el día de hoy. En concreto, en la Figura 8.25 se muestran las rutinas que tienen ejercicios para hacer el día actual, que es el martes. Las rutinas que están agregadas a favoritos aparecerán en primer lugar.

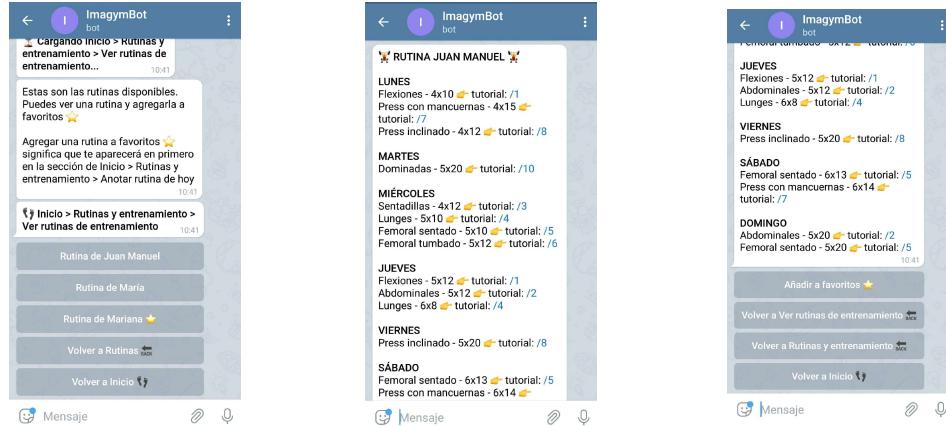


Figura 8.23: Viendo las rutinas de entrenamiento

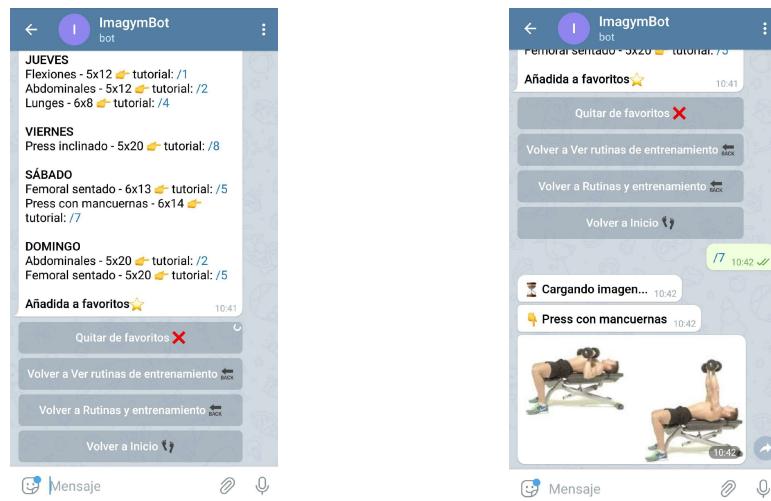


Figura 8.24: Agregando rutina a favoritos y viendo tutorial de un ejercicio

Al pulsar una rutina de un monitor, se mostrarán botones asociados a los ejercicios que tiene su rutina. En este ejemplo se muestran dos botones: uno para **sentadillas** y otro para **lunges**. El usuario simplemente debe pulsar el botón para registrar el ejercicio y se activará un icono de **check** en el botón que indica que ha sido registrado. Si por cualquier razón el usuario quiere eliminar el registro de un ejercicio, simplemente deberá pulsar el mismo botón y se quitará el icono.



Figura 8.25: Anotar rutina de entrenamiento

Consultar entrenamiento de otro día

Los usuarios podrán consultar los ejercicios que anotaron un día en específico mediante el comando **/consultar** (ver Figura 8.26). El bot primero enviará un mensaje con el último entrenamiento registrado, en el que se muestran los dos ejercicios que hemos anotado de acuerdo a la Figura anterior.



Figura 8.26: Consultar rutina de entrenamiento de otro día

8.1.7. Mi ficha personal

En *Mi ficha personal* los usuarios podrán registrar/modificar sus datos personales, así como ver una valoración de su IMC siempre que haya anotado su peso y su altura.

El submenú que se muestra al usuario se encuentra en la Figura 8.27. Para registrar o modificar cualquier cosa, los usuarios simplemente deben pulsar el botón correspondiente. En este ejemplo el usuario ha registrado todos sus datos personales a excepción de su fecha de nacimiento y su correo electrónico.

Registrando y modificando la ficha personal

En la Figura 8.28 se muestra el registro de la fecha de nacimiento del usuario. El usuario pulsa el botón de la fecha de nacimiento, escribe la fecha y el bot actualizará la ficha personal.

En la Figura 8.29 se muestra la modificación del género del usuario. El usuario debe pulsar el botón del género que le corresponda y el bot actualizará la ficha personal.

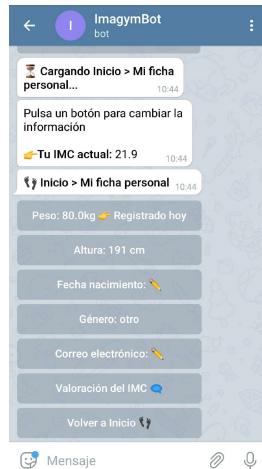
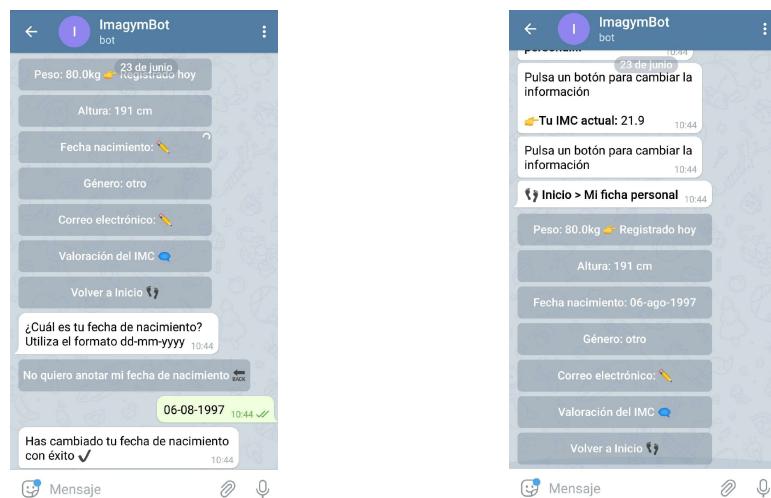
Figura 8.27: *Inicio > Mi ficha personal*

Figura 8.28: Registrando la fecha de nacimiento

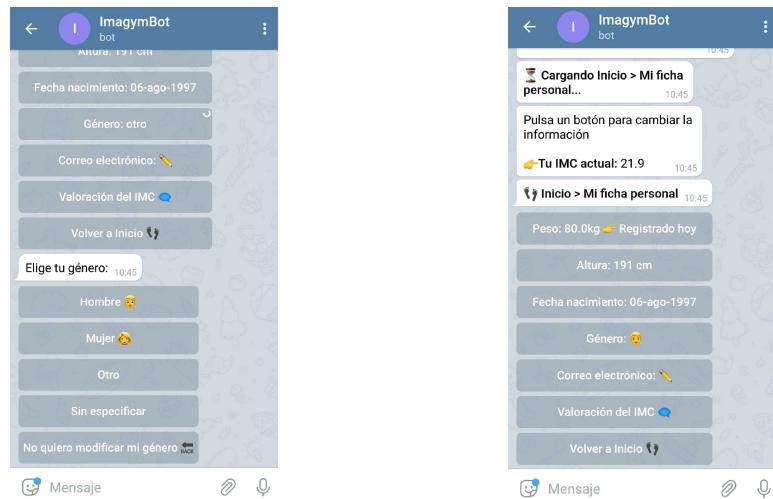


Figura 8.29: Modificando el género del usuario

Valoración del IMC

El bot puede valorar el IMC del usuario. Al pulsar el botón, el bot enviará una foto con los rangos en los que se mueve el IMC y una pequeña valoración acerca del IMC actual del usuario, tal y como se muestra en la Figura 8.30.

8.1.8. Soporte

En *Soporte* se ofrecerá información acerca del bot y la política de protección de datos. También ofrecerá ayuda acerca de cada funcionalidad del bot, al igual que se ha hecho durante toda esta Sección (ver Figura 8.31). En concreto, en la captura de la derecha se ofrece información acerca de **Mi objetivo de peso**.

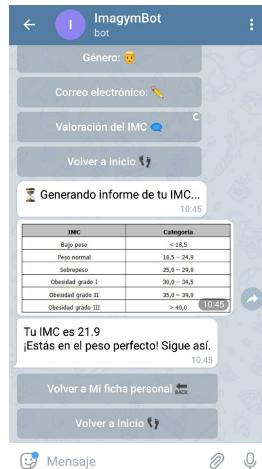
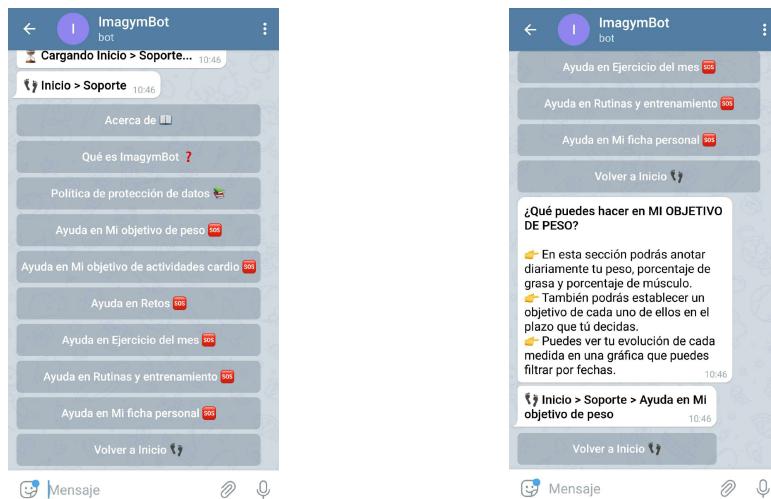


Figura 8.30: Valoración del IMC del usuario

Figura 8.31: *Inicio > Soporte*

8.2. Distribución del proyecto

ImagymBot está pensado para ser distribuido por todos los gimnasios y centros deportivos a nivel nacional. A cada uno de ellos se le ofrece una nueva forma de comunicación clientes-gimnasio, mejorando la relación que ya existe. Cada gimnasio o centro deportivo ofrecerá su propio contenido a través de **ImagymBot**, tal y como se ha visto a lo largo de esta documentación.

Aunque la plataforma web aún no haya sido terminada, los gimnasios y centros deportivos podrán ofrecer su contenido a través de archivos **csv** muy fáciles y sencillos de completar. En el repositorio de **Github** de **ImagymBot** [40] se pueden consultar tres archivos de ejemplo a través de los cuales se pueden proponer ejercicios del mes, retos, y rutinas de entrenamiento. En este repositorio también se encuentra el **script** de **Python** con toda la programación de **ImagymBot**.

Capítulo 9

Conclusiones finales y trabajos futuros

Este proyecto se ha realizado en un momento de estado de alarma por el COVID-19, donde se ha paralizado la actividad física ya que todo el mundo tuvo que quedarse en casa. Todos los gimnasios y centros deportivos tuvieron que cerrar sus puertas, al igual que otros muchos servicios. A pesar de ello, ha sido mucha gente la que ha seguido haciendo deporte en casa, incluso utilizando botellas de agua como equipamiento deportivo. También ha habido mucha gente que ha decidido comenzar a hacer deporte ya que al fin disponía de tiempo libre.

Muchos de estos gimnasios y centros deportivos que cerraron no pudieron seguir ofreciendo su servicio a sus clientes, ni si quiera de forma online, ya que puede que no dispongan del conocimiento o presupuesto necesario para crear una web. **ImagymBot** es una alternativa muy buena para todos ellos, pues es una manera bastante sencilla de ofrecer contenido a sus usuarios y que los mismos usuarios puedan seguir manteniendo esa relación con el gimnasio. Para aquellos gimnasios y centros deportivos que ya disponen de contenido online, **ImagymBot** puede ser algo novedoso y distinto que se puede complementar perfectamente con su web.

A pesar de que la plataforma web **Imagym** no esté preparada, sí que hay forma de que los gimnasios ofrezcan contenido a través de **ImagymBot** de una manera muy sencilla: a través de un **excel**. Como trabajo futuro, los gimnasios y centros deportivos podrán enviar sus archivos **csv** a través del bot. Estos archivos son legibles y fáciles de usar. En el repositorio de **Github** de **ImagymBot** [40] se muestran ejemplos de cómo pueden los gimnasios proponer ejercicios del mes, retos y rutinas y entrenamientos a sus clientes a través de un **excel**.

También se plantea el distinguir distintos roles dentro de **ImagymBot**. El propio bot puede enviar sus archivos de salida (puntuaciones de un ejercicio del mes, usuarios que superan un reto, etc) a aquellas personas que tengan el rol de administrador o monitor del gimnasio. Estas personas también podrán enviar comunicaciones o noticias breves del gimnasio a través del bot y que llegue a todos los clientes de forma instantánea.

La actividad física se está convirtiendo en una parte muy importante de nuestra vida. Además, son muchas personas las que realizan deporte en compañía. El ser humano es social y le gusta compartir sus alegrías y sus objetivos en compañía de otros. Creo que una de las cosas de la vida que más le gusta compartir a la gente es el progreso que hace cada uno con su cuerpo. Hacer deporte y comer sano es algo que ha cambiado muchas vidas, y es algo que llena de orgullo y satisfacción a la persona que lo logra.

Este proyecto nació con la ilusión de crear una nueva relación de las personas con el deporte, fomentando la competitividad sana y nunca menospreciando a aquellas personas que necesitan un esfuerzo extra para alcanzar sus objetivos. Son muchas las personas que han hecho deporte en su casa durante el estado de alarma pero que nunca se apuntaron a un gimnasio por miedo a lo que pensaran los demás. **ImagymBot** permite a los gimnasios ofrecer contenido a sus clientes con diferentes niveles de intensidad, adaptándose a todo tipo de personas. **ImagymBot** pretende implementar funcionalidades que permitan a los monitores de los gimnasios ofrecer contenido para personas con necesidades especiales.

De cara al futuro, **ImagymBot** pretende implementar la realización retos y ejercicios del mes a nivel nacional, no solo en un gimnasio. Esto quiere decir que se harán ejercicios del mes en el que no compitan los usuarios de un gimnasio, si no que los usuarios de un gimnasio se agrupen para competir con otros gimnasios.

La motivación es lo que mueve a las personas y lo que hace que cambie el mundo. La mejor motivación es aquella que obtenemos cuando vemos con nuestros propios ojos el resultado que nosotros mismos hemos conseguido a través de nuestro esfuerzo.

Me gustaría cerrar este proyecto con una de las frases que más me han inspirado, frase de **Steve Jobs**: “**If I try my best and fail, well, I've tried my best**”, que viene a significar algo como “si doy lo mejor de mí y fallo, bueno, al menos he dado lo mejor de mí”.

Bibliografía

- [1] Aleyda Abellán, Claudia Ventayol. El mercado del fitness en españa. <https://lifefitness.es/sites/g/files/dtv376/f/Zoom%20Mercado%202019%20-%20Life%20Fitness.pdf>, 2019.
- [2] Design thinking. <http://www.designthinking.es/inicio/>, 2020.
- [3] Iván Ramírez. Whatsapp vs telegram: ¿cuál es la mejor aplicación de mensajería? <https://www.xatakandroid.com/aplicaciones-android/whatsapp-vs-telegram-cual-mejor-aplicacion-mensajeria-1>, 2020.
- [4] PythonAnywhere LLP. Python anywhere. <https://www.pythonanywhere.com/>.
- [5] Andrea Barros. Real food challenge: ¿serías capaz de comer solo comida real durante un mes? <https://www.correryfitness.com/nutricion/real-food-challenge-serias-capaz-comer-solo-comida-real-201711205a1290430cf2018c197c41fc.html>, 2017.
- [6] Irene Sierra. El imperio del realfooding: dos millones de seguidores y consultas privadas a 60 euros. <https://magnet.xataka.com/why-so-serious/imperio-realfooding-dos-millones-seguidores-consultas-privadas-a-60eur>, 2019.
- [7] Statista Research Department. Usuarios de teléfonos móviles y smartphones a nivel mundial de 2014 a 2019. <https://es.statista.com/estadisticas/723622/usuarios-de-telefonos-moviles-y-smartphones-en-el-mundo/>, 2016.
- [8] José Carlos Castillo. Nomofobia: cuando la tecnología nos vuelve esclavos. <https://www.elcorreo.com/tecnologia/redes-sociales/nomofobia-tecnologia-vuelve-20190510114452-nt.html?ref=https%2F%2Fwww.google.com%2F>, 2019.
- [9] Ismael San Mauro Martín, Miguel González Fernández, and Luis Collado Yurrita. Aplicaciones móviles en nutrición, dietética y hábitos saludables: análisis y consecuencia de una tendencia a la alza. *Nutrición hospitalaria*, 30(1):15–24, 2014.

- [10] Vikram. Top fitness apps to help you keep going in 2020. <https://www.mobileappdaily.com/top-fitness-apps>, 2020.
- [11] Iván Ramírez. Google play mejora su algoritmo: priorizará aun más las apps y juegos de calidad. <https://www.xatakandroid.com/play-store/google-play-mejora-su-algoritmo-priorizara-apps-juegos-calidad>, 2019.
- [12] J. Clement. Leading android health apps worldwide 2020, by downloads. <https://www.statista.com/statistics/690887/leading-google-play-health-worldwide-downloads/>, 2020.
- [13] Fitify - planes y entrenamientos. <https://fitifyapps.com/>, 2020.
- [14] Gym wp - ejercicios y rutinas para el gimnasio. <https://play.google.com/store/apps/details?id=com.lealApps.pedro.gymWorkoutPlan&hl=es>, 2020.
- [15] Bestfit: Rutina gimnasio - entrenamiento y fitness. <https://play.google.com/store/apps/details?id=com.best.fit&hl=es>, 2020.
- [16] Fitvate - gimnasio rutina de ejercicio entrenador. <http://fitvate.com/>, 2020.
- [17] Total fitness - rutinas y gimnasio. <https://play.google.com/store/apps/details?id=com.bluecorner.totalgym&hl=es>, 2020.
- [18] Home workout - no equipment. https://play.google.com/store/apps/details?id=homeworkout.homeworkouts.noequipment&hl=en_US, 2020.
- [19] Samsung health. <https://www.samsung.com/es/apps/samsung-health/>, 2020.
- [20] Muscle booster - fitness at home. <https://play.google.com/store/apps/details?id=musclebooster.workout.home.gym.abs.looseweight&hl=es>, 2020.
- [21] Patry Jordan. Gym virtual. <https://www.youtube.com/user/gymvirtual>, 2020.
- [22] Enric Mor, Muriel Garreta Domingo, and Maria Galofré. Diseño centrado en el usuario en entornos virtuales de aprendizaje, de la usabilidad a la experiencia del estudiante. In *SPDECE*, 2007.
- [23] Roylan Galeano. Diseño centrado en el usuario. *Revista q*, 2(4), 2017.
- [24] Overleaf. <https://www.overleaf.com/>, 2020.
- [25] Ideo. <https://www.ideo.com/eu>, 2020.
- [26] Hasso Plattner. Guía del proceso creativo. mini guía: una introducción al design thinking+ bootcamp bootleg, 2018.
- [27] Design Thinking. Método persona. <https://www.designthinking.services/herramientas-design-thinking/metodo-persona/>, 2020.

- [28] Joaquín López Lérida y Félix de León Molinari. Design thinking. <https://fundacionpersan.org/web/uploads/formacion/Creatividad.%20Design%20Thinking.pdf>.
- [29] Whatsapp. Whatsapp. <https://www.whatsapp.com/>.
- [30] Telegram. Telegram. <http://telegram.com.es/>.
- [31] Miguel Florido. 20 lenguajes de programación más usados en 2020. <https://www.marketingandweb.es/marketing/lenguajes-de-programacion-mas-usados/>, 2020.
- [32] Python telegram bot. <https://github.com/python-telegram-bot/python-telegram-bot>.
- [33] Python Software Fundation. Pymysql. <https://pypi.org/project/PyMySQL/>.
- [34] Anastasia Kurmakaeva. Sitemap. <https://www.humanlevel.com/diccionario-marketing-online/sitemap>.
- [35] Sublime text. <https://www.sublimetext.com/>.
- [36] Kahoot. <https://kahoot.it/>.
- [37] Matplotlib. <https://matplotlib.org/>.
- [38] Python-telegram-api / examples / conversationbot.py. <https://github.com/python-telegram-bot/python-telegram-bot/blob/master/examples/conversationbot.py>.
- [39] Nathan Thomas. How to use the system usability scale (sus) to evaluate the usability of your website. <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website>.
- [40] Juan Manuel Castillo Nievas. Imagymbot. <https://github.com/Jumacasni/ImagymBot/>.