

# ACTIVIZA

## DAM - UNIR FP

*Tutora: Raquel Cerdá*

*Alumnos: Néstor Calderón, Luis Daniel Barragués, David Fernández*

## Introducción

Esta memoria detalla el desarrollo de la aplicación para gimnasios “Activiza”. Explicando el uso de tecnologías y los métodos utilizados para su desarrollo.

El grupo se ha centrado en crear una aplicación sólida, optimizada, moderna y que demuestra el talento grupal como desarrolladores. Es por ello por lo que se optó por crear una solución propia en vez de usar servicios de terceras partes tales como Firebase.

Para ello se han creado dos aplicaciones con diferentes interfaces. Una aplicación móvil dirigida a usuarios y entrenadores y una aplicación de gestión web para la administración del gimnasio, que a su vez es una API Rest que proporciona un servicio de back-end para la aplicación Android.

Además, se eligió usar Kotlin y Python como los lenguajes primarios de estas dos soluciones para demostrar la capacidad del grupo de adaptación y aprendizaje de contenidos no vistos en el ciclo o vistos de forma muy superficial, como Python.

## Contenido

Módulos formativos aplicados en el trabajo	1
Herramientas y lenguajes utilizados	2
Componentes del equipo	4
Néstor	4
Daniel	4
David	4
Fases del proyecto	5
Estudio de mercado	5
Modelo de datos	11
Diseño de las interfaces	13
Estructura de las aplicaciones	18
Planificación del desarrollo	20
Funcionalidad del proyecto	23
Conclusiones y mejoras del proyecto	34
Bibliografía	36
Anexo	37

# DEFINICIÓN DEL PROYECTO

## Módulos formativos aplicados en el trabajo

Los módulos de formación aplicados en el proyecto están distribuidos entre varias materias. Seguidamente, se enumerarán las materias junto con los módulos correspondientes que se emplearán:

### **Devops**

UF1: Docker, Python, metodologías ágiles y GIT.

### **Programación de servicios y procesos**

UF3: API Rest.

### **Desarrollo de interfaces**

UF1 y UF5: Diseño de la aplicación, accesibilidad, UI y UX.

### **Inglés técnico**

Aplicación con textos en inglés más la documentación consultada de las diferentes herramientas y tecnologías.

### **Acceso a datos**

UF3: Herramientas de mapeo de objetos.

### **Programación Multimedia y Dispositivos móviles**

UF1, UF2, UF3: Desarrollo de aplicación con Android Studio.

# DEFINICIÓN DEL PROYECTO

## Herramientas y lenguajes utilizados

La aplicación va a necesitar diferentes tecnologías para su correcto desarrollo, despliegue y funcionamiento. A continuación, se exponen las tecnologías que se han seleccionado para ser usadas durante el desarrollo:

### **Docker y Docker Compose**

En la informática se suele decir como broma “en mi ordenador funciona” cuando ocurren problemas en servidores de producción o preparación. Docker nos ayuda a crear un contenedor para el despliegue de la aplicación en la nube, creando básicamente el mismo entorno allá donde se despliegue, evitando problemas de dependencias.

### **Nginx y uWSGI**

Elegidos debido a su rendimiento, eficiencia y flexibilidad. Nginx maneja las conexiones, mientras que uWSGI sirve la aplicación Django. Juntos, ofrecen seguridad, escalabilidad y compatibilidad con el estándar WSGI de Python, lo que facilita la migración a diferentes infraestructuras si fuese necesario.

### **Django y Python**

Para la realización de la API Rest y el servicio “cloud” se ha elegido construirlo con el framework Django y el lenguaje Python. La simplicidad, pero potencia de Python ayudará a crear un servicio sólido en el corto tiempo disponible para la realización de este proyecto.

### **PostgreSQL y SQLite**

Django suele ser siempre acompañado por PostgreSQL, MariaDB, MySQL, Oracle o SQLite. Por defecto, genera una base de datos SQLite al crear el proyecto. Debido a su popularidad, escalado y fácil integración, se eligió PostgreSQL como base de datos final, pero el desarrollo se produjo con SQLite, debido a su simplicidad de uso y facilidad de recrear nuevas tablas en caso de modificar los modelos.

Para la base de datos de la aplicación Android, había múltiples opciones, tales como SharedPreferences, SQLite o ROOM. Las SharedPreferences se utilizan como “cookies”, no sirven para estructurar datos complejos como objetos o arrays. ROOM trabaja con secuencias

# DEFINICIÓN DEL PROYECTO

SQL y podría ser más sencillo que SQLite, pero, por la familiaridad de los integrantes con SQLite, se eligió esta opción.

## Google Cloud

Se necesita que el servicio de base de datos sea accesible desde cualquier lugar para que los clientes y el gimnasio pueda usarlo sin mayor inconveniente, por lo que se ha decidido desplegarlo en la nube. La nube elegida es Google Cloud (<https://cloud.google.com>) por su promoción de 300 USD de crédito durante 90 días.

## GIT y GitHub

Como control de versiones se ha elegido usar GIT y GitHub como repositorio por su popularidad y facilidad de uso [1].

## Kotlin

Debido a las prácticas de la mayoría del grupo, que están usando Kotlin, se dictaminó que sería una buena alternativa a Java por su simplicidad y facilidad de uso en comparación. También, tras un estudio de mercado de lenguajes para Android, se dictaminó que Kotlin tenía una proyección muy positiva y se ve como una alternativa a Java para el futuro de Android.

## Android studio

Para desarrollo en Android, nos hemos decantado por Android Studio, ya que es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de esta plataforma, creado y actualizado por Google. Ofrece todas las herramientas necesarias para desarrollar aplicaciones, incluyendo un emulador de alto rendimiento con una variedad de dispositivos y configuraciones de Android para probar la aplicación.

## Visual Studio Code

Para programar en Python y con Django, se ha elegido Visual Studio Code debido a su variedad de extensiones, su ligereza, su integración con GIT y su consola integrada. Además, su depurador facilita la depuración de aplicaciones Django y su ligereza y personalización hacen que se adapte a las necesidades del proyecto en cualquier momento, por ejemplo la integración con Docker una vez construido el back-end.

# DEFINICIÓN DEL PROYECTO

## Componentes del equipo

### NÉSTOR

ROL: Desarrollador Front-end y Administrador de la base de datos DBA SQLite:

Encargado de:

- Creador de la base de datos local para la aplicación Android.
- Estructurar la aplicación y desarrollo de funcionalidad de usuario y entrenadores.
- Conexión con la API Rest.

### DANIEL

ROL: Desarrollador Back-end, DevOps y Scrum Máster

Encargado de:

- Desarrollo del back-end.
- DevOps de despliegue en Google Cloud.
- Planificación de la metodología Agile como Scrum Máster.

### DAVID

ROL: Desarrollador Front-end y Diseñador:

Encargado de:

- Diseño UI/UX.
- Desarrollo de estructura gráfica de la aplicación.
- Registro y gestión de usuarios.

# DEFINICIÓN DEL PROYECTO

## Fases del proyecto

### ESTUDIO DE MERCADO

#### 1. Definir el objetivo del estudio

- a. Propósito: crear una app que ayude a los usuarios de gimnasios a planificar y seguir rutinas de ejercicio personalizadas para mejorar su rendimiento y alcanzar sus objetivos de fitness.
- b. Alcance: competencia, demanda, segmentación de usuarios, características deseadas, modelo de negocio.

#### 2. Análisis del entorno

- a. Tendencias del mercado:
  - i. Aumento de la digitalización en el fitness.
  - ii. Creciente popularidad de las apps de salud y fitness.
  - iii. Enfoque en la personalización y seguimiento del progreso.
- b. Factores económicos y sociales:
  - i. Mayor conciencia sobre la salud y el bienestar.
  - ii. Incremento en el número de personas que asisten a gimnasios.
  - iii. Aumento del uso de dispositivos móviles y wearables para el seguimiento del fitness.

#### 3. Análisis de la competencia

- a. Identificación de competidores:
  - i. Fitness Online: app de entrenamiento personalizado basada en llevar un registro del entrenamiento ya sea en casa o en el gimnasio
    - 1. Pros: la app ya lleva asentada en el mercado más de 3 años, por lo que me ha dejado ver su última versión y tiene 4.7 estrellas, es la app número 16 de nutrición y deporte en la App Store.



## DEFINICIÓN DEL PROYECTO

2. Contrás: la app no tiene uso de la inteligencia artificial ni nuevas tecnologías y lleva estancada con el mismo comportamiento durante varios años, tiene 3 versiones sacadas en la última semana, pero según la App Store, nada nuevo solo correcciones de errores en el chat. La anterior actualización fue hace 8 meses a día 28/05/2024, la app tiene mucho contenido de pago y está limitada al usuario
- ii. Freeletics: app de entrenamiento personalizado basada en inteligencia artificial:
    1. Pros: la app cuenta con una inteligencia artificial para saber el mejor plan de entrenamiento para una persona
    2. Contrás: tiene una versión gratuita muy limitada y la experiencia de muchos usuarios no es muy buena, la gente se queja de que la descripción es corta y que la IA la metieron para darle marketing, pero que siempre que crea una rutina es la misma.
  - iii. Fitbod: app que crea rutinas de ejercicio personalizadas basadas en el equipo disponible y el progreso del usuario:
    1. Pros: no se estanca en progreso como hacen algunas, sino que avanza conforme avanza el usuario y según mi experiencia y la de los usuarios es la mejor app “Que he probado”.
    2. Contrás: según un usuario y mi experiencia: “Más caro que el gym”, lo que cierra las puertas a un mercado que no sea el estadounidense y aun así hay comentarios en inglés como “Great app, but is very expensive”. El plan: son 2 semanas de prueba gratuita y luego son 12 €/mes u 82 €/año y solamente viene en inglés. Comparando con la app de FitnessOnline, cobra 54,99 €/año o 6,99 € al mes e incluye nutrición.

## DEFINICIÓN DEL PROYECTO

- iv. Conclusión: crear una app más económica y que cumpla las expectativas de los usuarios y poner una traducción a varios idiomas

### 4. Segmentación del mercado

#### a. Segmentos de usuarios:

- i. Principiantes: personas nuevas en el gimnasio que necesitan orientación básica.
- ii. Intermedios: usuarios con algo de experiencia que buscan mejorar su rutina actual.
- iii. Avanzados: usuarios experimentados que buscan optimizar su rendimiento.
- iv. Entrenadores personales: profesionales que buscan herramientas para gestionar las rutinas de sus clientes.

#### b. Perfil de los usuarios:

- i. Principiantes: 18-30 años, buscan simplicidad y guía.
- ii. Intermedios: 25-40 años, interesados en el seguimiento del progreso y nuevos desafíos.
- iii. Avanzados: 30-50 años, enfoque en la optimización y personalización profunda.
- iv. Entrenadores personales: profesionales de fitness, interesados en herramientas de gestión y análisis.

### 5. Investigación del mercado primaria

#### a. Encuestas y entrevistas:

- i. Encuestas en línea en foros y redes sociales de fitness.
- ii. Entrevistas con usuarios de gimnasios para entender sus necesidades y desafíos actuales.

# DEFINICIÓN DEL PROYECTO

## b. Grupos focales:

- i. Reuniones con grupos de usuarios de diferentes niveles para obtener feedback sobre características deseadas.

## 6. Investigación de mercado secundaria

### a. Revisión de estudios y reportes:

- i. Estudios del deporte en España: 2020 sube un 6.1% en 5 años y 80% de los jóvenes hacen deporte.

1. Estudio del gobierno de España [2].

### b. Análisis de datos públicos:

- i. 2022: Encuesta hábitos deportivos del ministerio de educación y deporte [3].
- ii. 2020: Encuesta hábitos deportivos del ministerio de educación y deporte [4].

## 7. Análisis de la demanda

### a. Tamaño del mercado:

- i. En 2022, los alrededor de 4.000 gimnasios repartidos por todo el territorio español contaban con aproximadamente 5,5 millones de socios [5].

### b. Proyección de crecimiento:

- i. Gráfica de 1996 a 2023 crecimiento del deporte [6].
- ii. Número de gimnasios abiertos en 2022 [7].

## 8. Análisis de la viabilidad

### a. Análisis SWOT:

- i. Fortalezas: enfoque en la personalización, integración con wearables.

## DEFINICIÓN DEL PROYECTO

- ii. Debilidades: necesidad de una inversión inicial significativa en desarrollo y marketing.
  - iii. Oportunidades: creciente adopción de tecnología en el fitness, posibilidad de asociaciones con gimnasios.
  - iv. Amenazas: competencia intensa, posibles barreras tecnológicas.
- b. Modelo de Negocio:
- i. Suscripciones: ofrecer planes mensuales o anuales.
  - ii. Freemium: acceso básico gratuito con opciones premium.
  - iii. Publicidad: incluir anuncios para usuarios gratuitos.

### 9. Prueba y validación

- a. Prototipo y MVP (producto mínimo viable):
- i. Desarrollar una versión inicial con características esenciales: creación de rutinas, seguimiento de progreso, Chat del entrenador, seguimiento de métricas corporales.
- b. Iteración basada en retroalimentación:
- i. Obtener feedback de un grupo piloto y realizar ajustes antes del lanzamiento completo.

### 10. Plan de marketing y estrategia de lanzamiento

- a. Estrategia de entrada al mercado:
- i. Colaboraciones con gimnasios para promociones cruzadas.
  - ii. Lanzamiento en plataformas de fitness y bienestar.
- b. Campañas de promoción:
- i. Publicidad en redes sociales y Google Ads.
  - ii. Campañas de marketing de influencers en el sector fitness.

## DEFINICIÓN DEL PROYECTO

- iii. Ofertas especiales y gratuitas de lanzamiento y referidos.

### **11. Monitoreo y adaptación continua**

- a. Análisis de datos y métricas:
  - i. Monitorear métricas clave como la retención de usuarios, el uso de la app y las tasas de conversión a premium.
- b. Ajustes y actualizaciones:
  - i. Actualizar y mejorar la app regularmente basándose en el feedback de los usuarios y las tendencias del mercado.

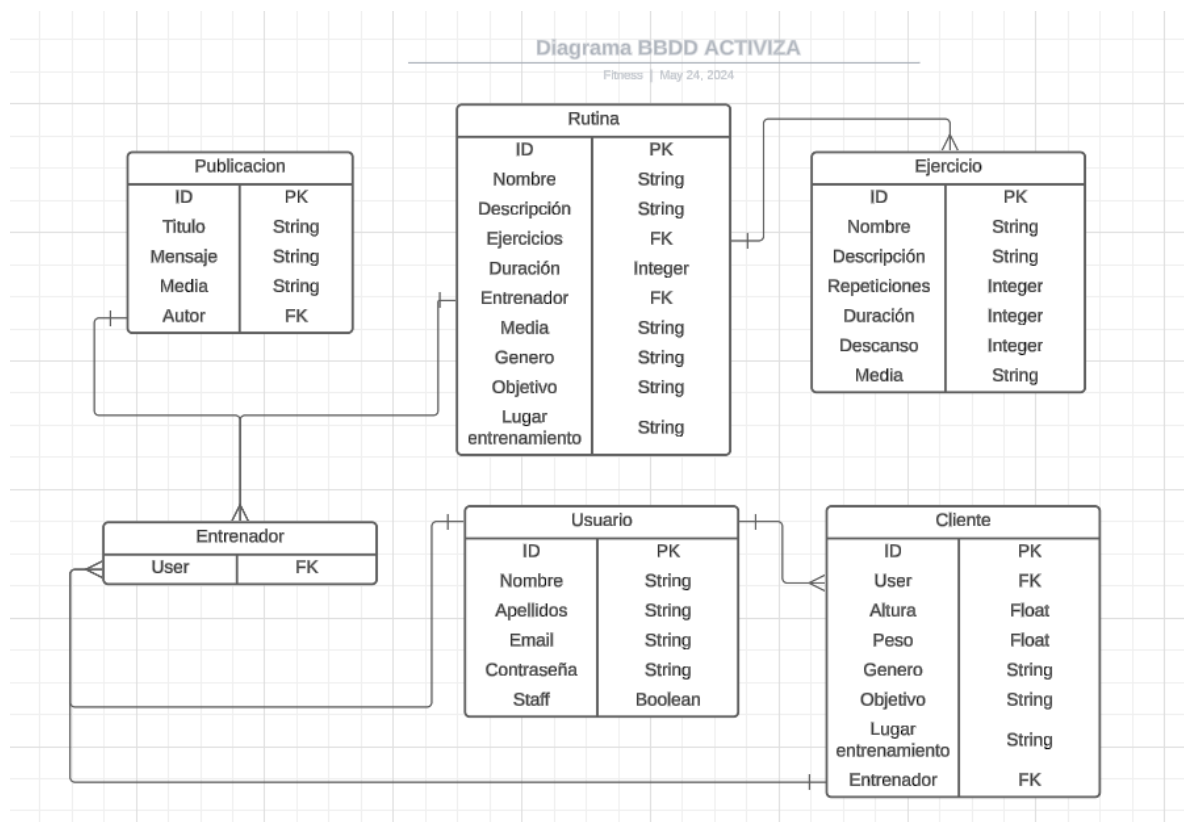
# DEFINICIÓN DEL PROYECTO

## MODELO DE DATOS

### API Rest

Django es un framework muy avanzado que utiliza un mapeo relacional de objetos (ORM) que permite la creación “automática” de la base de datos, sus tablas y respectivas relaciones, construyendo únicamente el modelo de datos por código.

Para realizarlo, primero realizamos un diagrama unificado de modelado (UML), el cual se puede observar a continuación:



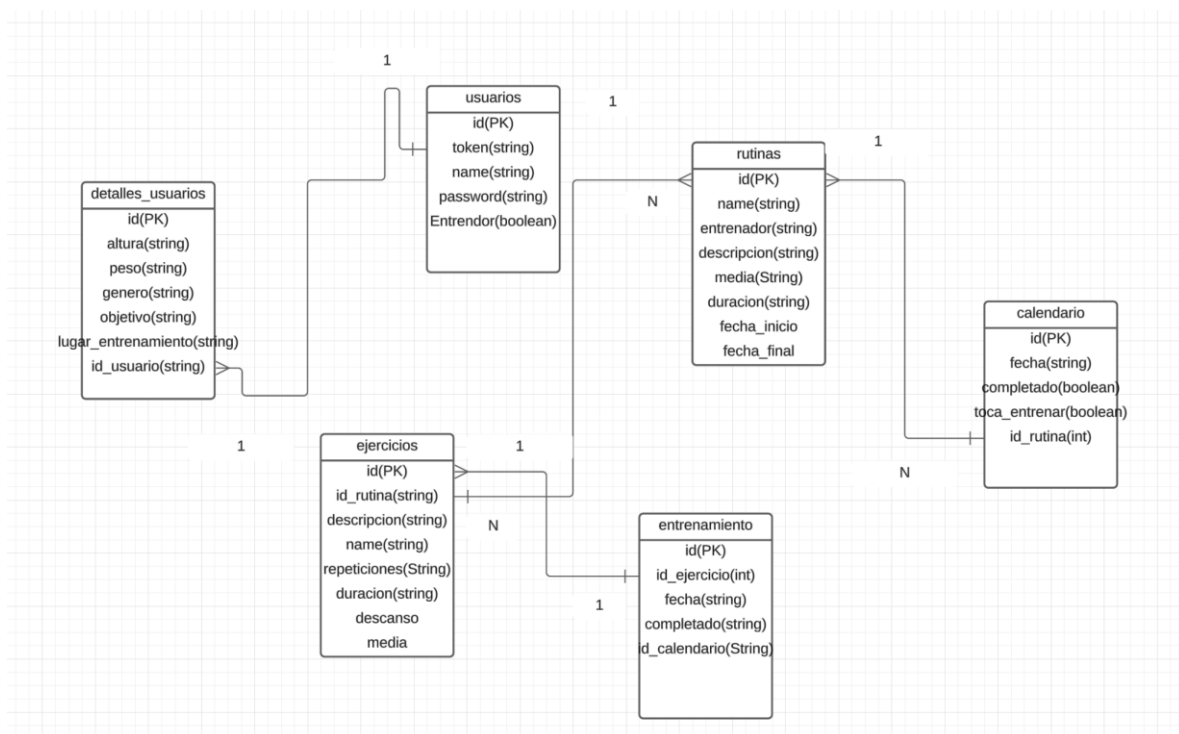
Partiendo de esta base, se realiza la construcción de los modelos. En Django se suelen juntar las clases en el mismo archivo “models.py”, ya que no necesitan de “getters” y “setters” y son más simples que las clases Java. Este archivo se encuentra dentro de la sub carpeta “activiza” dentro del “root” de la carpeta de Django.

# DEFINICIÓN DEL PROYECTO

## Aplicación Android

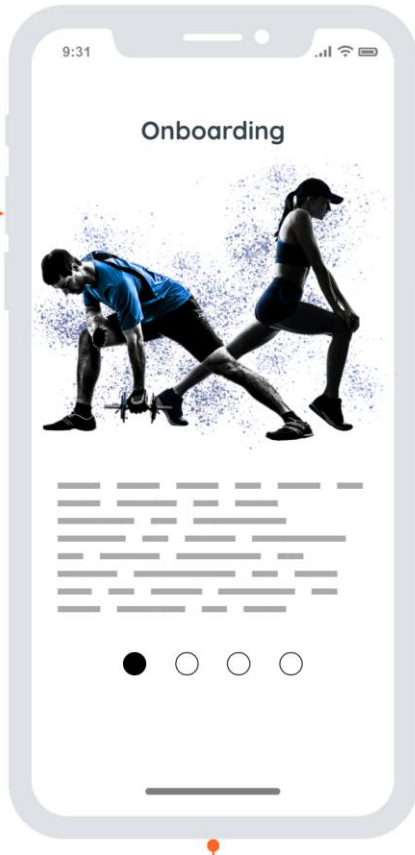
Para desarrollar la estructura, se decidió partir desde la idea de que la base de datos local es un complemento a la API del servidor. Por lo tanto, el diseño tendría que evitar hacer peticiones innecesarias al servidor y aumentar velocidad de carga para el usuario, siguiendo los siguientes principios:

1. Guardar la sesión.
2. Cargar los datos locales rápidamente, evitando llamadas innecesarias a la API, ya que existen en local.
3. Descargar los datos que se cree que el usuario va a utilizar y así dar mayor velocidad en la navegación UI y UX del usuario.
4. Guardar métricas y datos que no son necesarios que el servidor disponga y que podrían ser usados para funcionalidades futuras, por ejemplo el progreso de la rutina, el calendario recoge los días que se van a entrenar y también si está completado y la fecha del entrenamiento, que solo es necesario que lo tenga el usuario logueado en local.



# DEFINICIÓN DEL PROYECTO

## DISEÑO DE LAS INTERFACES

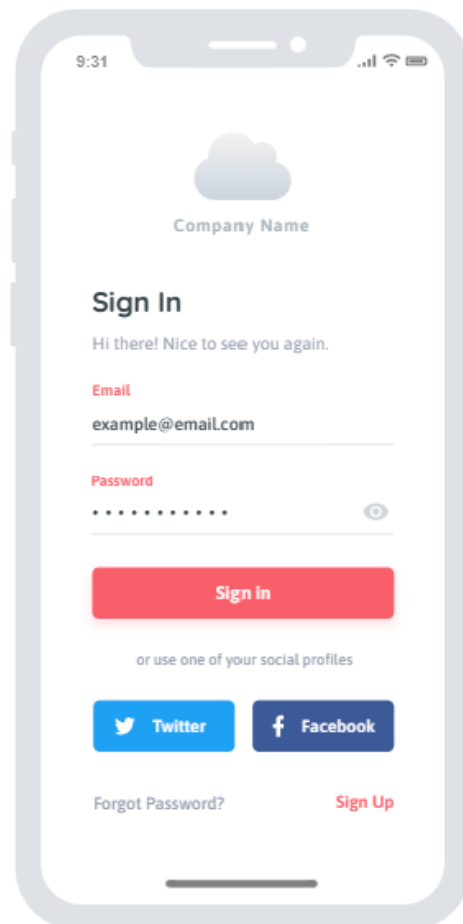


### Bienvenida (Onboarding)

Aparecerá al primer contacto con la aplicación, son unas diapositivas con unas imágenes y textos motivacionales que cuenta a grandes modos en que consiste la app, solo aparece la primera vez que entras y cuando inicias sesión ya no debería aparecer.

### Login

Hemos optado por un diseño clásico de la página de login.





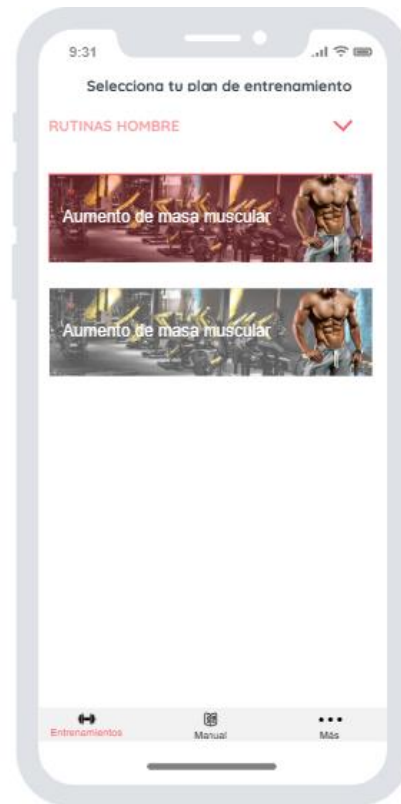
# DEFINICIÓN DEL PROYECTO

## Entrenamiento

Es donde estarán situados los entrenamientos que el profesor o gimnasio implemente para sus alumnos, las rutinas aparecerán aquí y estarán clasificadas por categorías como:

- Pérdida de grasa.
- Ganancia de masa muscular.
- Recompensación muscular.

Los botones llevarán a un resumen de la rutina donde pone los días en los que consiste, y una breve descripción

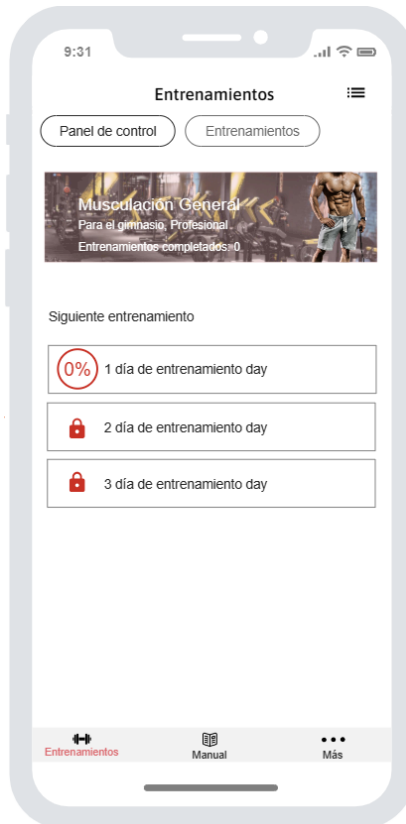


## Plan de entrenamiento

Cuando seleccionas un plan de entrenamiento antes de empezar tenemos la opción de ojear su contenido: con este ejemplo podemos ver los días de entrenamiento las sesiones por semana etc.

Si estamos conformes le daremos a agregar y se añadirá nuestra rutina a nuestra lista personalizada.

# DEFINICIÓN DEL PROYECTO



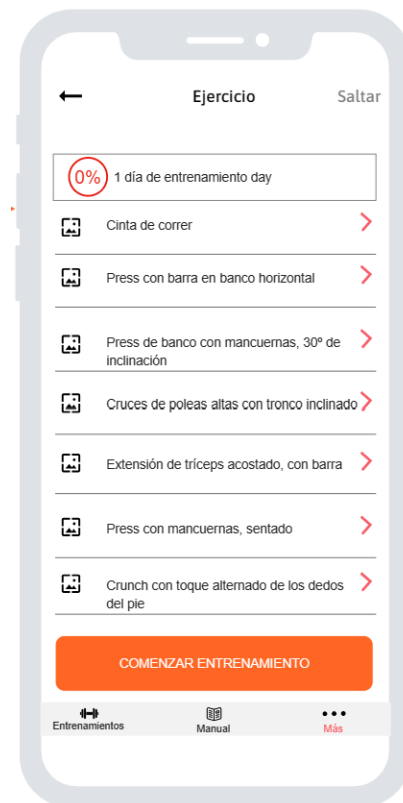
## Ventana entrenamiento

Cuando seleccionas un entrenamiento obtienes el submenú de Panel de control y Entrenamientos los cuales te permiten elegir que entrenamiento quieres y que entrenamientos hay disponibles

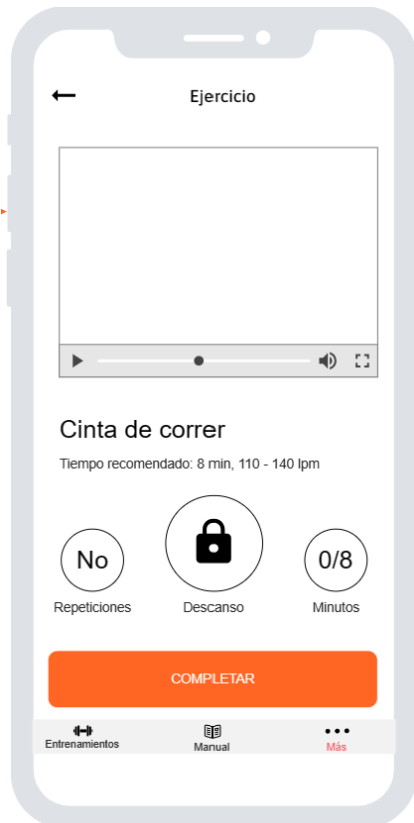
Los entrenamientos se dividen en días libres o días que se entrena, los días libres tienen un candado y los que se entrena tiene un % de completado según los ejercicios que tiene y hayas completado.

## Ejercicio

Aquí se podrá visualizar la rutina que te toca este día en esta visualización se compone de los ejercicios que te toca hacer con una imagen y su nombre y un botón que dará comienzo al entrenamiento.



# DEFINICIÓN DEL PROYECTO



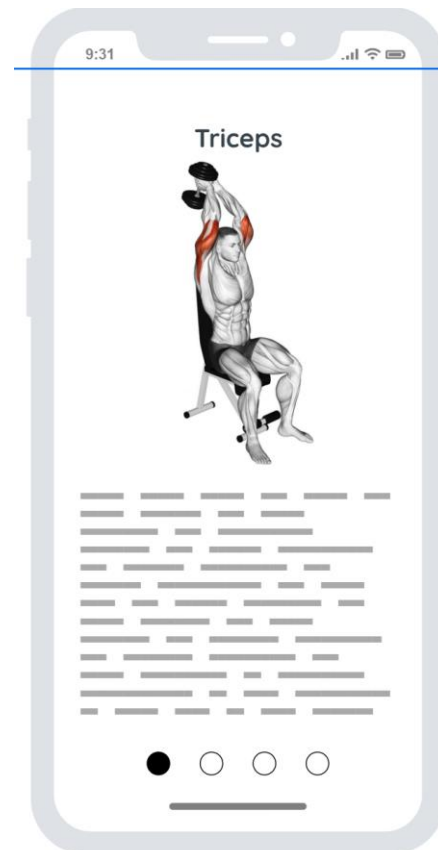
## Ejercicio detallado

Cuando empieza el entrenamiento se podrá entrar a la ejecución del ejercicio que será un video o foto del ejercicio que tiene que ejecutarse con diferentes parámetros de duración, repeticiones, descanso, etc.

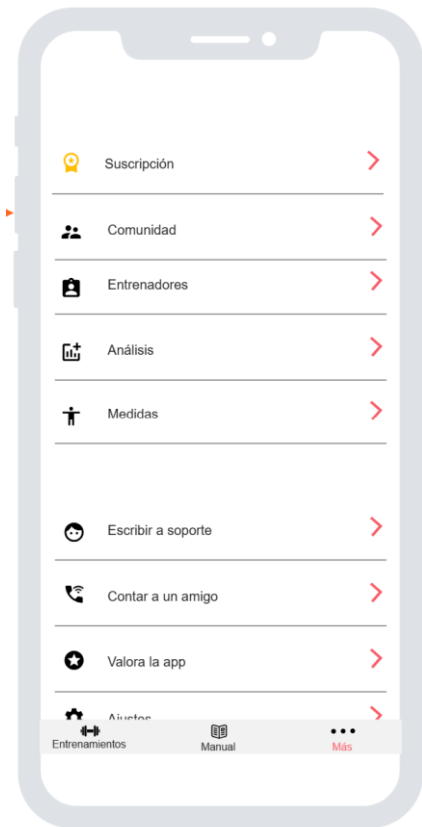
## Ejercicio detallado (alternativa)

Vamos a realizar un testeo A/B con usuarios finales para comprobar que diseño gusta más y es más sencillo de utilizar.

Esta pantalla sería más detallada explicando extensamente el objetivo del ejercicio.



# DEFINICIÓN DEL PROYECTO



## Menú

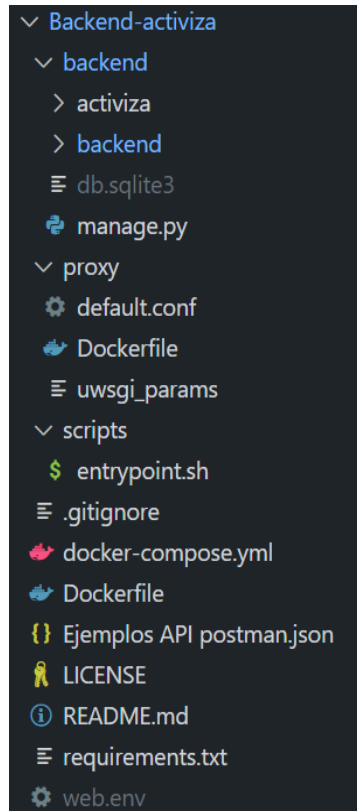
El menú se compondrá de una serie de opciones como:

- Datos personales.
- Medidas personales.
- Contacto con el gimnasio.
- Entrenadores asignados.

# DEFINICIÓN DEL PROYECTO

## ESTRUCTURA DE LAS APLICACIONES

### API Rest



Estructura de carpetas del proyecto

Se ha usado la estructura estándar de Django:

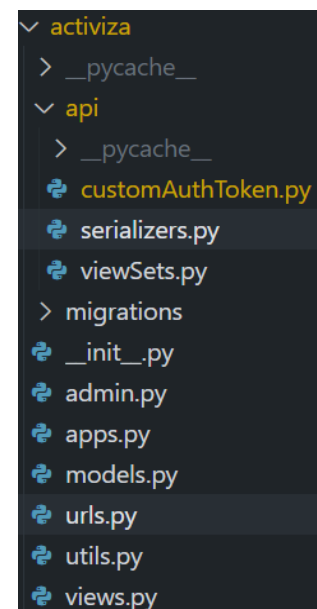
Una carpeta de proyecto “backend” con una subcarpeta del mismo nombre que tiene la configuración general del proyecto. Otra sub carpeta dentro del “root” del proyecto de Django “backend” con el nombre del módulo de la aplicación “activiza”, que tiene dentro las vistas y modelos.

La parte de docker se estructura de la siguiente manera:

Una carpeta “proxy” con el Dockerfile de “Nginx” para el contenedor que hará de servidor, junto con “uWSGI” para enlazarlo con Django. Carpeta “scripts” para la ejecución de código tras el despliegue de los contenedores. “Root” del repositorio que contiene los diferentes “Dockerfile”, “docker-compose” y “requirements.txt” para la creación del contenedor que va a correr el proyecto de Django. Además de una colección de Postman para pruebas y ejemplos.

Django se aleja un poco del modelo tradicional de modelo, vista y controlador. La vista “views.py” hace doble función de controlador y se encarga de procesar y servir plantillas.

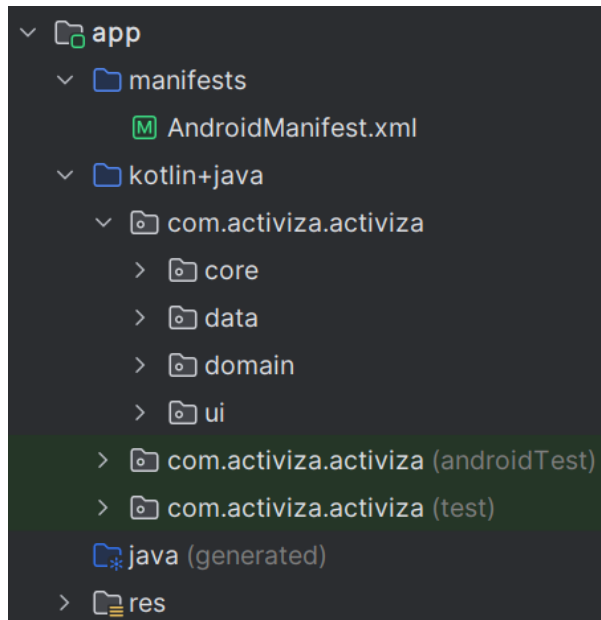
Dentro de la aplicación existe “models.py”, que contiene las clases modelo de la aplicación, “urls.py” con las direcciones que cargan las diferentes funciones de “views.py”, que sería el controlador. La sub carpeta “api” contiene los “serializers” y “viewsets” que se encargan de servir y procesar la API Rest, además de una función que sobrescribe la generación de “tokens” por defecto de la librería utilizada para poder devolver un campo extra.



Estructura de la aplicación

# DEFINICIÓN DEL PROYECTO

## Aplicación Android



*Estructura de la aplicación Android*

Se ha usado la estructuración estándar de Android Studio, la cual se compone:

Carpeta “manifests” que contiene el manifiesto de Android con la configuración y permisos requeridos.

La carpeta “kotlin+java” que contiene las subcarpetas del proyecto que a su vez contiene “core” con variables constantes, “data” con las diferentes clases de objetos que se utilizan en la aplicación, “domain” con la base de datos SQLite local y los métodos para la API Rest y “ui” que contiene las diferentes pantallas.

Por último, la carpeta “rest” en el “root” de la aplicación que contiene diferentes recursos de la app, tales como “layouts”, animaciones, fuentes, cadenas de strings, etc.

# DEFINICIÓN DEL PROYECTO

## PLANIFICACIÓN DEL DESARROLLO

Como cualquier proyecto, el primer paso es idear el concepto, aclarar qué problema se quiere solucionar y plantear las tecnologías y requisitos necesarios para llevar a cabo la idea. A continuación, se va a detallar el orden en el que se actuó para definir el proyecto. Comenzando por un “brainstorming” personal, luego por un desarrollo colectivo de las ideas seleccionadas y acabando con la realización de un objetivo claro que defina las metas a las que se quiere llegar.

### Brainstorming

Las ideas que más gustaron:

- Aplicación para un gimnasio que controle el aforo y muestre rutinas a los miembros.
- Control por QR que identifique a cada usuario e incremente el aforo, prohibiendo el acceso cuando se ha llegado al límite. Aplicación DEMO de un escáner físico en el gimnasio.
- En vez de utilizar Firebase, usar un servicio “cloud” propio que actúe como gestor de BDD y que tenga una API Rest para comunicarse con la aplicación, utilizándolo también como motor de generación de estadísticas en vez de tener que hacerlas en local.
- Los entrenadores pueden crear rutinas y ejercicios.
- Los clientes ven las rutinas asignadas por los entrenadores.
- Sección de publicaciones tipo “foro” que los administradores y entrenadores puedan escribir y poner cosas en común con los clientes.
- Hacer un “onboarding” de bienvenida a la aplicación (cuando se abre por primera vez).

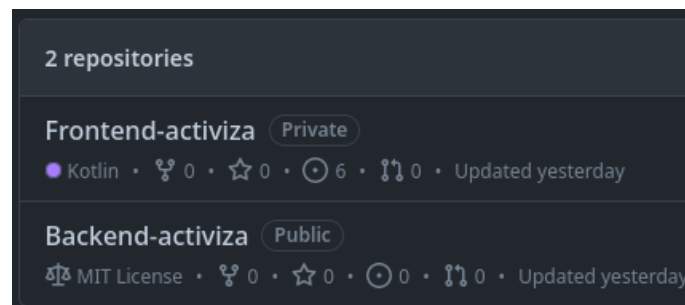
### Desarrollo

Aplicación móvil Android será el “front-end” de cara a los clientes para que puedan consultar sus rutinas asignadas por entrenadores y ver los mensajes publicados en el foro. También podrán acceder al gimnasio por código QR. Los entrenadores podrán modificar y crear nuevas rutinas con ejercicios predefinidos, además de hacer nuevas publicaciones en el foro.

# DEFINICIÓN DEL PROYECTO

La aplicación estará conectada a un servicio “cloud” propio que actúe como un motor web para generación de estadísticas, del QR y también como API REST que procesará operaciones CRUD (crear, leer, actualizar y eliminar) sobre la base de datos del servidor. Todo será consultado en tiempo real por medio de dicha API.

Esto queda plasmado en los dos repositorios creados, uno para el front-end y otro para el back-end:



## Objetivo

El objetivo es tener una aplicación que pueda mostrar rutinas editadas y creadas por los entrenadores del gimnasio y que permita a los clientes suscribirse a ellas, además de guardar el progreso. Dichas rutinas podrán ser personalizadas para clientes en concreto, siendo solo mostradas a los usuarios asignados al entrenador.

Además, queremos que se conecte a los clientes del gimnasio, permitiendo que se publiquen consejos, “tips”, etc en un foro. También, incluir un generador de QR que permita ser escaneado por la máquina presente en el gimnasio para poder entrar.

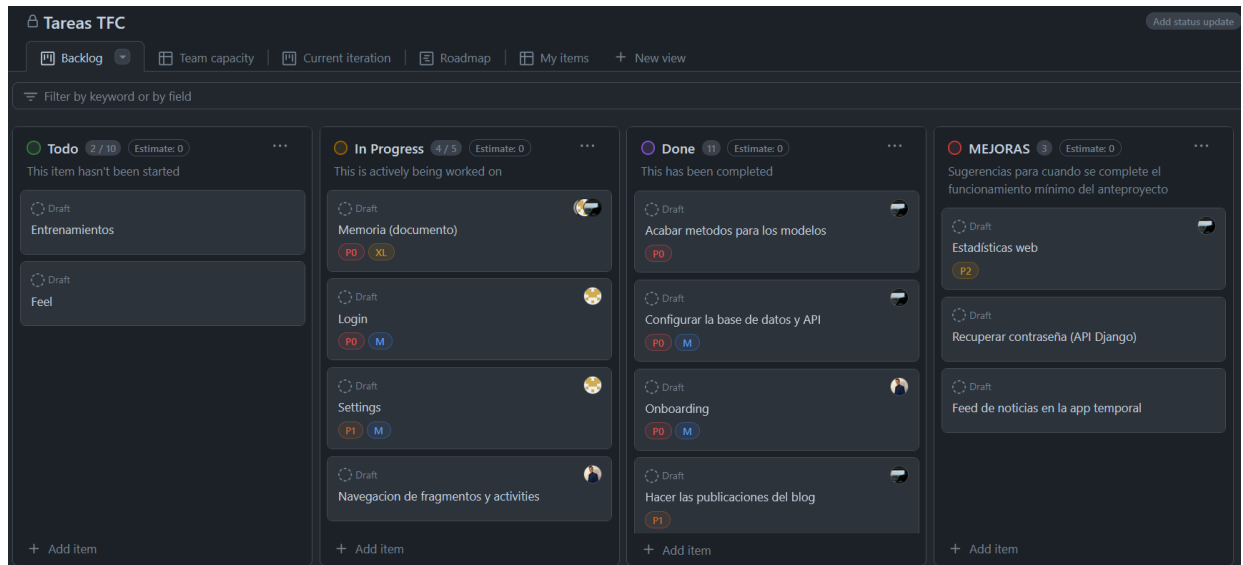
## Metodologías

El desarrollo del proyecto se va a realizar usando la metodología “agile”. Se ha concretado que se trabajará en “sprints” de 2 semanas con reuniones semanales. Esto nos permitirá tener independencia a la hora de trabajar, pero con controles semanales para identificar posibles problemas o retrasos en el proyecto a tiempo. Además, hace que se tenga que planificar qué desarrollos se van a hacer en el sprint, teniendo claro antes de empezar cuanto contenido tiene que realizar cada miembro.



# DEFINICIÓN DEL PROYECTO

A continuación, se muestra el proyecto de GitHub con las diferentes tareas que se han ido creando y moviendo a sus respectivos estados:



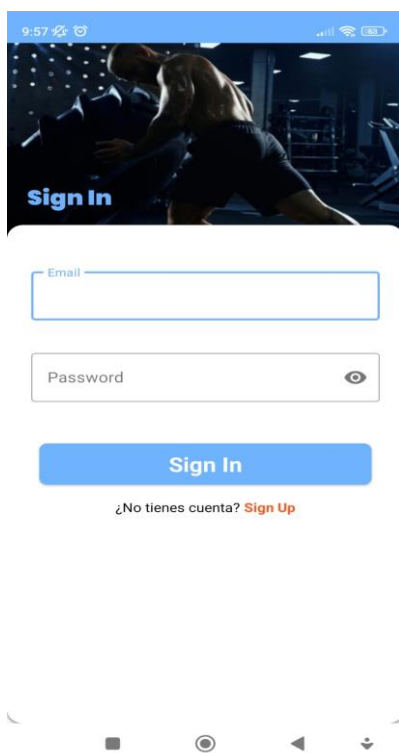
# DEFINICIÓN DEL PROYECTO

## FUNCIONALIDAD DEL PROYECTO

El proyecto tiene dos interfaces principales, la aplicación móvil, de cara al usuario y los entrenadores, desarrollada para ser instalada y usada desde móviles Android y la interfaz web, para la administración del gimnasio, desplegada en Google Cloud y desarrollada para ser usada desde el navegador de un ordenador.

A continuación, vamos a detallar las diferentes opciones que ofrece cada una:

### Aplicación Android

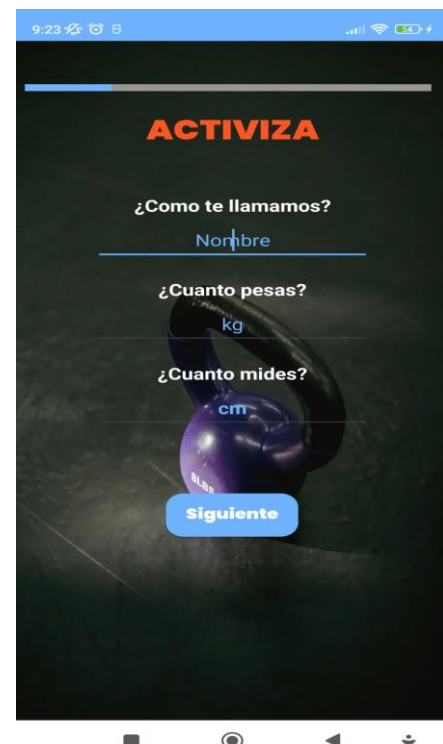


#### Inicio de sesión

Lo primero que encuentra el cliente son las pantallas de inicio de sesión a través de email y contraseña, así como la pantalla de registro en la que además se le pide el nombre. El usuario podrá cambiar entre ellas en función de si tiene cuenta o no pulsando las palabras en naranja, “sign in” y “sign up”.

#### Onboarding

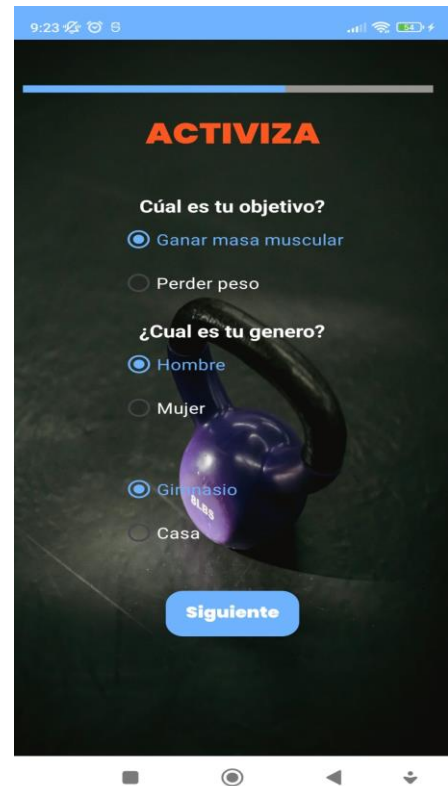
Si es la primera vez que el usuario accede a la aplicación, le lleva a las pantallas de onboardings. En la primera se le pide rellenar el nombre, la altura y el peso.



# DEFINICIÓN DEL PROYECTO

## Onboarding

En esta segunda pantalla elige el género, objetivo y lugar de entrenamiento.



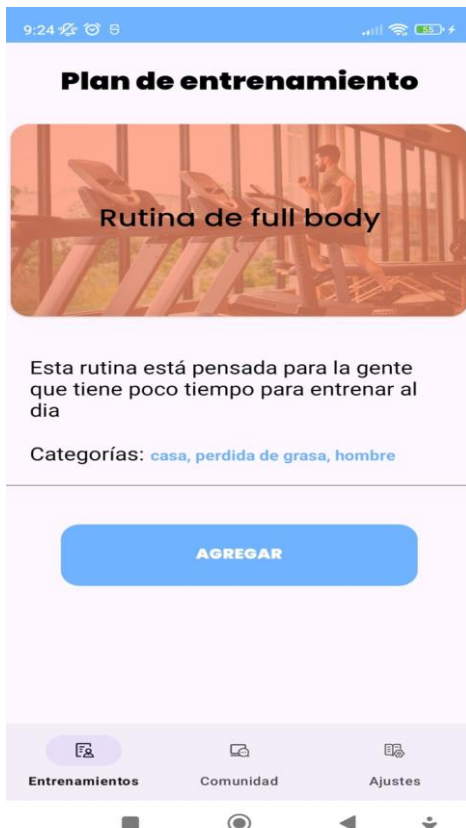
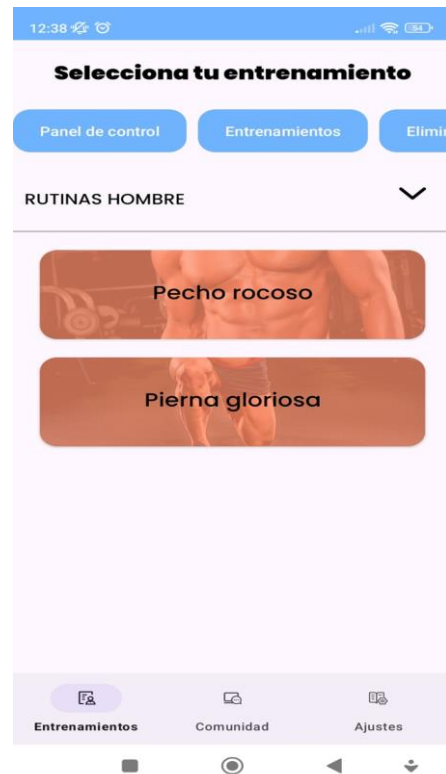
## Onboarding

A continuación, se da la bienvenida al usuario y todos los datos recogidos en estos onboardings son usados como filtro para obtener la mejor rutina de ejercicios.

# DEFINICIÓN DEL PROYECTO

## Pantalla principal

Cuando el cliente llega a la pantalla principal de la aplicación, ve una lista con las rutinas disponibles creadas por los entrenadores que se adaptan a los datos proporcionados anteriormente.



## Plan de entrenamiento

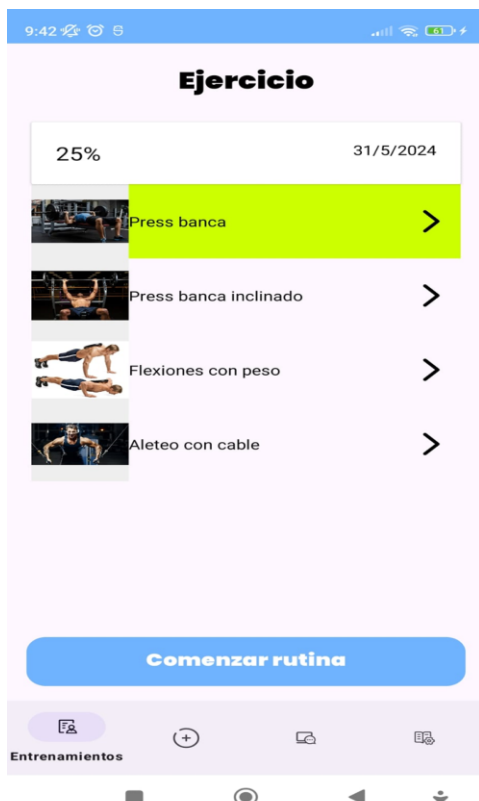
Después de que el cliente seleccione una rutina, accede a una nueva pantalla donde se la describe, muestra las categorías y da la opción de agregarla a tu plan de entrenamiento.

# DEFINICIÓN DEL PROYECTO

## Rutina

Una vez agregada la rutina, lleva al usuario a la pantalla de panel de control, donde puede realizar diferentes acciones.

Entre las diferentes opciones puede eliminar la rutina de su plan de entrenamiento, volver a la lista de las que hay disponibles o comenzar este entrenamiento. En esta pantalla habrá un calendario con el día actual en naranja que cambiará de color al completar el entrenamiento diario.



## Ejercicio

Cuando el usuario comienza el entrenamiento, aparece una lista de ejercicios a completar, correspondiente a la rutina que ha elegido anteriormente. Conforme se completan irán cambiando de color.

# DEFINICIÓN DEL PROYECTO

## Detalle ejercicio

Al acceder a los ejercicios, el usuario ve la pantalla con una imagen de cómo se hace, así como una descripción y las repeticiones a realizar.



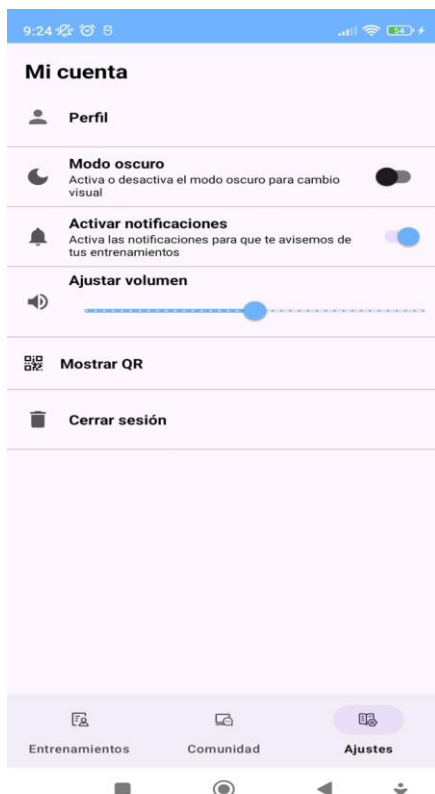
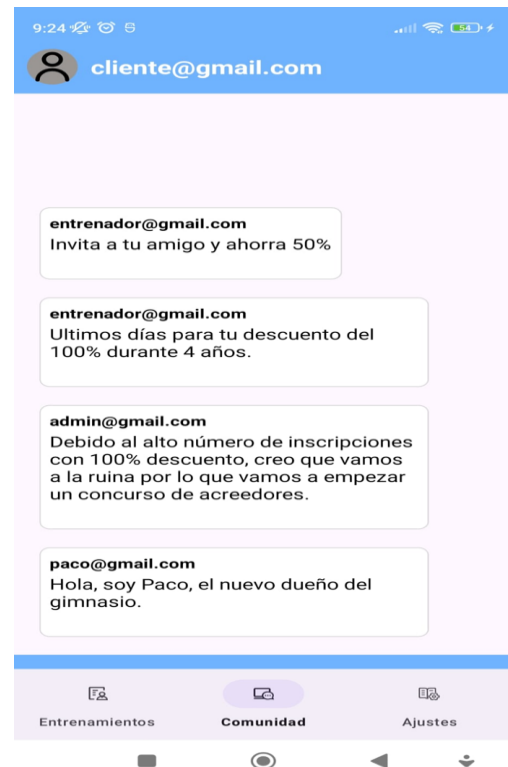
## Rutina

Una vez completada la rutina, el calendario marca el día en verde.

# DEFINICIÓN DEL PROYECTO

## Comunidad

Desde cualquier parte de la aplicación el usuario puede acceder a través del botón inferior a “Comunidad” y ver todos los mensajes enviados por los entrenadores.



## Ajustes

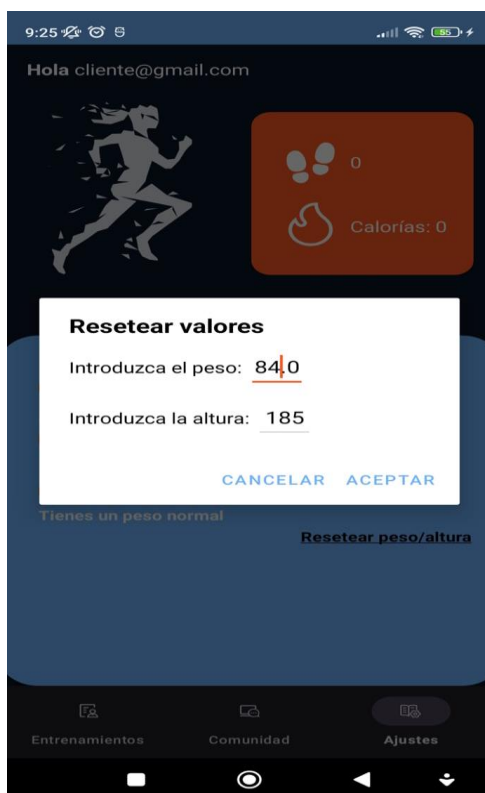
También desde el botón inferior “Ajustes”, se puede acceder a las configuraciones de la aplicación, por ejemplo, activar modo oscuro, ajustar el volumen y cerrar sesión, entre otras.

# DEFINICIÓN DEL PROYECTO

## Perfil usuario

En la opción “Perfil” el usuario accede a su área personal donde tendrá un contador de pasos, un contador de calorías quemadas en función de los pasos y una gráfica con el peso y altura que añadió en sus datos.

También hay un botón que calcula el IMC. Al darle añade otra gráfica y le dice en qué rango de peso se encuentra.



## Perfil usuario

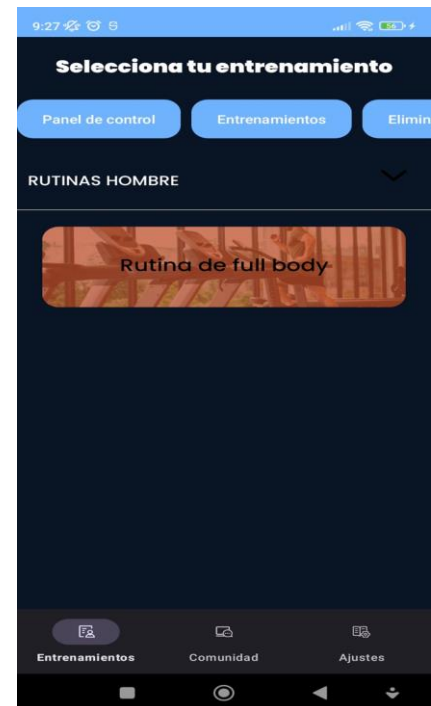
El usuario puede en cualquier momento cambiar los valores de peso y altura.



# DEFINICIÓN DEL PROYECTO

## Modo oscuro

Tanto en esta como en las anteriores pantallas, tanto de cliente o entrenador, se puede ver en “Modo oscuro”.



Las funcionalidades vistas hasta ahora eran las vistas para el cliente, a continuación, vamos a ver las funcionalidades extras que disponen los entrenadores.

A screenshot of a mobile application form titled 'Rutina'. It contains several input fields: 'Nombre:', 'Descripción:', 'Ejercicios:', and 'Subir Imagen Rutina:' with a plus icon. Below these are two columns of radio buttons for 'Genero' (H - hombre, M - mujer) and 'Lugar' (C - casa, G - gimnasio). Further down are 'Objetivo' (GRASA para perdida, MUSCULO para ganancia) and 'Duracion' (Días de duración). A blue 'Siguiete' button is at the bottom. A bottom navigation bar has a central '+' icon labeled 'Añadir Rutina'.

## Crear rutina

El entrenador dispone de un botón inferior adicional que le permite acceder a esta pantalla en la que puede crear nuevas rutinas para los clientes.

# DEFINICIÓN DEL PROYECTO

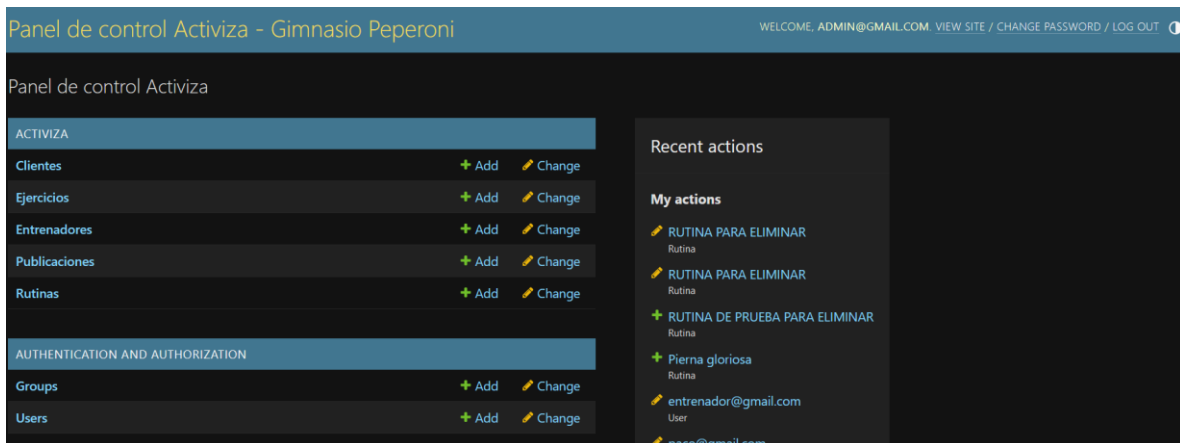
## Comunidad

En el botón “Comunidad”, el entrenador puede enviar mensajes a los clientes.



## API Rest

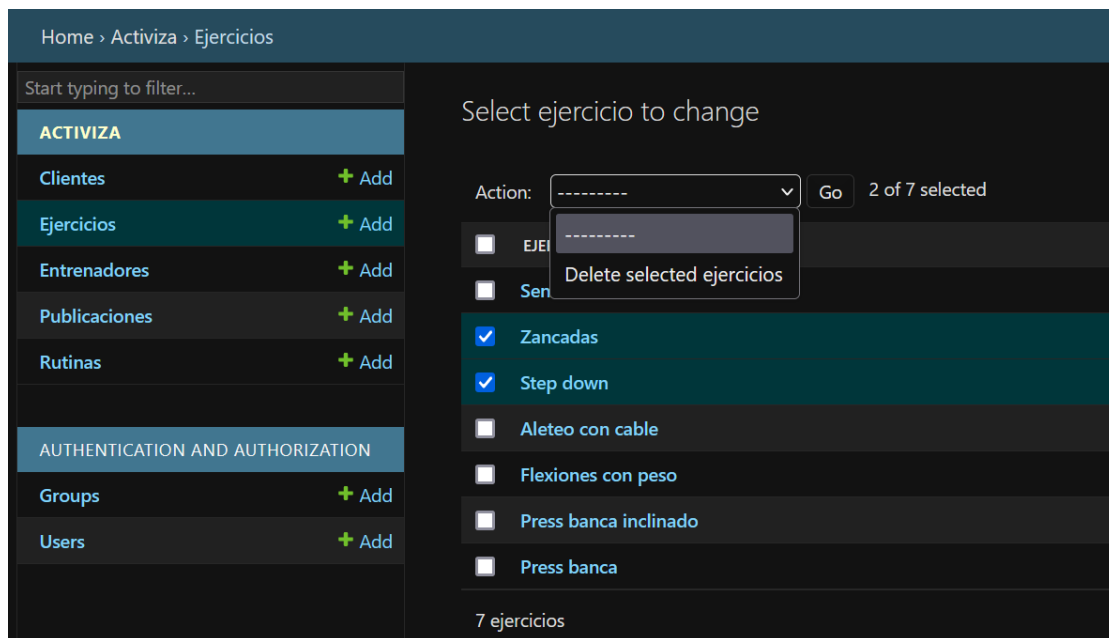
Accediendo al portal web, como administrador, tenemos acceso global a todas las tablas, además de un “log” de acciones recientes (hechas desde el portal web):



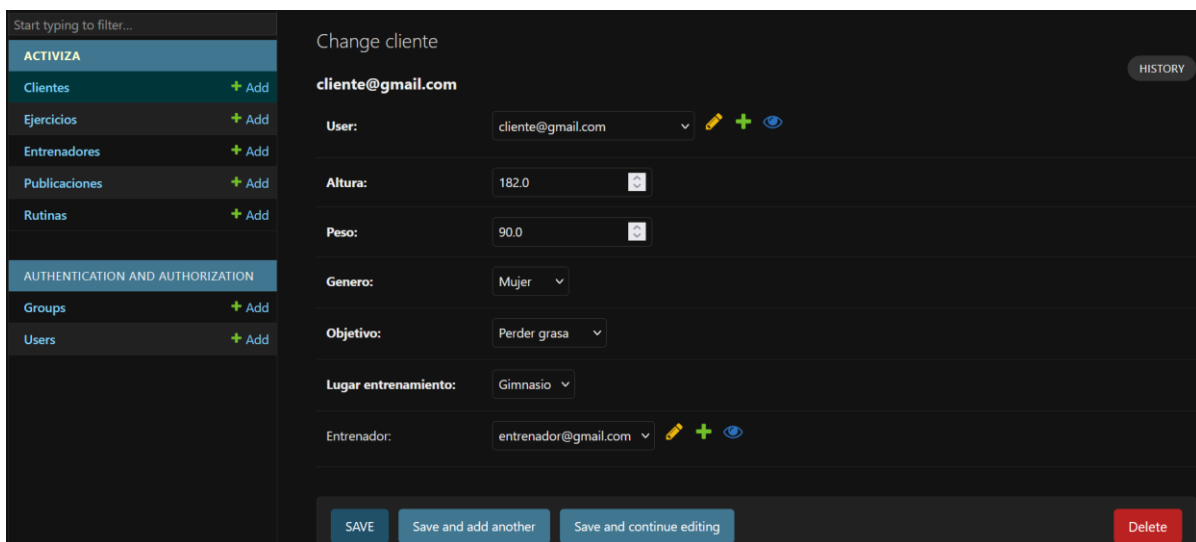
Desde este portal se pueden editar todas las tablas e incluso eliminar usuarios para ayudar a los gimnasios a cumplir con la ley orgánica de protección de datos (LOPD) fácilmente. También se puede dar de alta a usuarios que puedan tener problemas con la app o que se hayan registrado y no tengan un móvil.

Además, entrenado en la vista de una tabla, se puede hacer una eliminación masiva:

# DEFINICIÓN DEL PROYECTO



A continuación, se muestra la selección de un cliente y las acciones permitidas que se pueden hacer sobre el objeto guardar, guardar y crear otro, guardar y continuar editando y eliminar. Esta vista es común a todos los objetos de todas las tablas, aquí se pueden apreciar dos vistas con todos los tipos de campos existentes en el proyecto:



# DEFINICIÓN DEL PROYECTO

Start typing to filter...

**ACTIVIZA**

- Cientes + Add
- Ejercicios + Add
- Entrenadores + Add
- Publicaciones + Add
- Rutinas + Add

**AUTHENTICATION AND AUTHORIZATION**

- Groups + Add
- Users + Add

**Change rutina** HISTORY

**Pierna gloriosa**

Nombre: Pierna gloriosa

Descripcion: ¿Quieres tener la pierna más fuerte que Thor?

Ejercicios: 

- Press banca
- Press banca inclinado
- Flexiones con peso
- Aleteo con cable
- Step down
- Zancadas
- Sentadilla con salto

Duracion: 20

Entrenador: entrenador@gmail.com

Media: https://i.imgur.com/GCRClu.jpeg

Genero: Hombre

Los campos con relaciones 1:1 aparecen con opción de crear, editar o visualizar los objetos referenciados, los que tienen relaciones 1:N solo muestran la opción de crear un objeto nuevo. En las vistas anteriormente mostradas, estos campos son “User”, “Entrenador” y “Ejercicios”.

Por último, las páginas también disponen de un tema claro:

Panel de control Activiza - Gimnasio Peperoni WELCOME, ADMIN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Panel de control Activiza

**ACTIVIZA**

- Cientes + Add Change
- Ejercicios + Add Change
- Entrenadores + Add Change
- Publicaciones + Add Change
- Rutinas + Add Change

**AUTHENTICATION AND AUTHORIZATION**

- Groups + Add Change
- Users + Add Change

**Recent actions**

**My actions**

- Pecho adonis Rutina
- + Paco Publicacion
- + Ruina Publicacion
- none Publicacion
- none Publicacion
- none Publicacion
- none Publicacion
- Descuento 100% Publicacion

# DEFINICIÓN DEL PROYECTO

## CONCLUSIONES Y MEJORAS DEL PROYECTO

El grupo tiene una conclusión clara, que lo desarrollado es factible. Al contrario que otros proyectos que se nos ocurrieron, tales como una plataforma de “e-commerce” o un juego de plataformas, este proyecto es un producto en su totalidad, no depende de artículos disponibles o la necesidad de desarrollar una historia convincente, puede ser usada por cualquier persona en un gimnasio, ya que podría seguir una rutina en su estado actual. Esto, sumado a que no hay un referente de aplicaciones de gimnasio, permite que se pueda entrar en el mercado incluso sin ofrecer los servicios de entrenadores y gimnasios.

Esto no quiere decir que la aplicación no carezca de funciones que mejorarían la experiencia de los usuarios. Es por ello por lo que se ha desarrollado una lista con mejoras futuras que podríamos hacer si el desarrollo continuase.

### 1. Funcionalidades desarrollables a corto plazo:

#### a. Gamificación

- i. Desafíos y logros: desafíos de fitness y logros que los usuarios pueden desbloquear al alcanzar ciertos hitos.
- ii. Recompensas: puntos, medallas o recompensas virtuales para incentivar la constancia y el progreso.

#### b. Entrenamiento social

- i. Entrenamientos en grupo: posibilidad de unirse a sesiones de entrenamiento en grupo virtuales.
- ii. Competencias y rankings: clasificaciones y competencias entre amigos o a nivel global para fomentar la competitividad sana.

### 1. Funcionalidades avanzadas que necesitan planificación y desarrollo:

#### a. Entrenamientos en vivo y a demanda

- i. Clases en vivo: transmisión de sesiones de entrenamiento en vivo con entrenadores profesionales.
- ii. Biblioteca de clases a demanda: acceso a entrenamientos grabados que los usuarios pueden seguir en cualquier momento.

#### b. Comunicación y soporte

## DEFINICIÓN DEL PROYECTO

- i. Chat: soporte en tiempo real para resolver dudas y problemas técnicos.
  - ii. Comunidad de usuarios: foros o grupos donde los usuarios pueden compartir experiencias, consejos y motivación.
- c. Análisis avanzado de datos
  - i. Análisis predictivo: uso de inteligencia artificial para predecir el rendimiento futuro y ofrecer recomendaciones personalizadas.
  - ii. Informes detallados: informes detallados del rendimiento y progreso con análisis de datos históricos.
- d. Integración con dispositivos wearables
  - i. Sincronización de datos: integración con dispositivos como Fitbit, Apple Watch, y otros wearables para importar datos de actividad física.
  - ii. Monitorización en tiempo real: seguimiento en tiempo real de parámetros como frecuencia cardíaca, calorías quemadas y pasos.
- e. Planes de nutrición
  - i. Sugerencias de dieta: planes de nutrición personalizados basados en los objetivos del usuario.
  - ii. Registro de comidas: posibilidad de registrar la ingesta de alimentos y calcular calorías consumidas.

# DEFINICIÓN DEL PROYECTO

## Bibliografía

1. Karl Hughes. (8 de febrero de 2022). *Bitbucket vs GitHub vs GitLab*.  
<https://stackshare.io/stackups/bitbucket-vs-github-vs-gitlab>
2. Gobierno de España. *La práctica deportiva aumenta un 6,1% en los últimos cinco años*.  
<https://www.csd.gob.es/es/la-practica-deportiva-aumenta-un-61-en-los-ultimos-cinco-anos>
3. Ministerio de educación y deporte. (2022). *Encuesta hábitos deportivos*.  
<https://www.educacionfpydeportes.gob.es/servicios-al-ciudadano/estadisticas/deportes/encuesta-habitos-deportivos-espana.html>
4. Ministerio de educación y deporte. (2020). *Encuesta hábitos deportivos*.  
<https://www.educacionfpydeportes.gob.es/dam/jcr:dac0ae99-ffbb-40e1-9ae5-0dabcbedbc18/encuesta-de-habitos-deportivos-2020.pdf>
5. Statista. (2022). *Los alrededor de 4.000 gimnasios repartidos por todo el territorio español contaban con aproximadamente 5,5 millones de socios*.  
<https://es.statista.com/temas/6047/el-sector-del-fitness-en-espana/#topicOverview>
6. Statista. (1996-2023). *Crecimiento del deporte*.  
<https://es.statista.com/estadisticas/569644/evolucion-del-porcentaje-de-individuos-que-fueron-a-un-gimnasio/>
7. Valgo. (primer trimestre 2022). *Número de gimnasios abiertos en 2022*.  
<https://www.valgo.es/blog/el-fitness-espanol-bate-su-record-de-aperturas-de-gimnasios-en-el-primer-trimestre-de-2022?elem=279448>

# DEFINICIÓN DEL PROYECTO

## Anexo

- Repositorio GitHub del front end (aplicación Android).
- Repositorio GitHub del back end (API Rest y web de administración).

**Nota:** Para la ejecución del back-end se necesita tener instalado Docker y Docker-Compose. Para su ejecución (preferiblemente en Linux) solo se necesitaría lanzar el comando “*sudo docker-compose up --build --force-recreate -d*” en la carpeta “backend”, el cual se encarga de la creación de las imágenes de Docker y su lanzamiento en un servidor local.

```
bigtech@backend-activiza:~/Backend-activiza/backend$ ls  
activiza  backend  db.sqlite3  manage.py
```

*Nivel de la carpeta donde habría que ejecutar el comando.*