



MY GYM BRO

2º Desarrollo de Aplicaciones Multiplataforma



JUNE 14, 2022

JUAN MANUEL SAAVEDRA ALQUEZAR
IES segundo de Chomón

Índice

2.- Documento Descripción del Proyecto	3
2.1 Contexto:	3
2.1.1 Ámbito:	3
2.1.2 Competencia:	3
2.1.3 Solución y justificación:	3
2.1.4 Destinatarios:	3
2.2 Objetivo del proyecto	4
2.3 Lenguas extranjeras:	4
3.- Documentos que debe cumplir	5
3.1 Requisitos	5
3.2 Tareas:	5
3.2.1 Hechas:	5
3.2.2 En Proceso:	5
3.2.3 Por Hacer:	5
3.3 Metodología:	6
3.4 Planificación temporal de las tareas:	6
3.5 Presupuesto:	7
3.6 Contrato:	8
3.7 Análisis de Riesgos:	10
4.- Documento de análisis y diseño	11
4.1 Modelado de datos:	11
4.2 Análisis y diseño del sistema funcional	13
4.3 Análisis y diseño de la interfaz de usuario:	14
4.4 Diseño de la arquitectura de la aplicación:	29
4.4.1 Tecnologías utilizadas	29
4.4.2 Arquitectura de componentes de la aplicación:	29
5.- Documentación de implementación e implantación del sistema.	31
5.1 Implementación:	31
6.- Documento de cierre	35
6.1 Documento de instalación y configuración	35
6.2 Manual de usuario	35
6.3 Resultados obtenidos y conclusiones	35
6.4 Diario de bitácora	35
7.- Bibliografía y Webgrafía	36
ANEXO I	38

2-. Documento Descripción del Proyecto

2.1 Contexto:

My Gym Bro surge de una necesidad. Los usuarios de gimnasio podrán registrar los progresos, que y compararlo con los progresos anteriores gracias a esta aplicación. Hasta ahora tenían que recurrir a una libreta o usar las notas del móvil, pero ahora tendrán una aplicación donde crear sus rutinas y registrar el peso levantado ese día. Del mismo modo, podrán guardar las marcas de sus 'PR' o récords personales en la sección dedicada a ello.

Como extra, dado que no es el objetivo principal, vas a poder revisar como van tus amigos, los pesos que han estado levantando últimamente y sus PR. El objetivo de esto es crear una rivalidad sana entre amigos y que esto ayude a progresar también.

2.1.1 Ámbito:

Esta aplicación entraría dentro del sector servicios porque te está brindando un servicio de almacenamiento de tus datos de gimnasio.

2.1.2 Competencia:

Hoy en día, ninguna aplicación del mercado ofrece un servicio como el que va a ofrecer My Gym Bro cuando esté terminada. Si revisas la 'store' no se encuentra ninguna. 'Gym WP' podría considerarse competencia, pero aun así hay suficientes diferencias entre ambas como para no poder considerarla competencia directa.

2.1.3 Solución y justificación:

Se ha decidido utilizar flutter para que con un único desarrollo obtengamos una aplicación apta para IOS y para Android. Y como se indica, para diferenciarse, My Gym Bro tendrá la opción de revisar los progresos de tus amigos y almacenar tus 'PR'.

2.1.4 Destinatarios:

Los destinatarios de My Gym Bro son todos los usuarios de gimnasio que quieran guardar sus progresos y con eso llega mucho más lejos en su recorrido dentro de este deporte.

2.2 Objetivo del proyecto

Como se indica previamente, My Gym Bro resuelve el problema de donde registrar tus progresos en el gimnasio, así como tus Pr y tus rutinas del día a día y poder ver graficas de progreso en los distintos ejercicios que se pueden realizar en un gimnasio

2.3 Lenguas extranjeras:

No procede.

3-. Documentos que debe cumplir

3.1 Requisitos

1. Registrar rutinas, entrenos y PR
2. Mostrar graficas de avances
3. Agregar amigos para poder ver sus progresos
4. Tener un login que guarde usuarios para que si inicias desde otro dispositivo tengas tus progresos y tus rutinas guardadas
5. Agregar, tu peso, tu altura y el volumen de tus músculos para ver como has ido 'creciendo'.
6. Revisar que están haciendo tus amigos y cuáles son sus Pr

3.2 Tareas:

3.2.1 Hechas:

1. Login
2. Menú
3. Router
4. AppTheme
5. Crear rutinas
6. Pantalla de añadir ejercicio a rutinas
7. Json de ejercicios y retos
8. Guardado en base de datos firestore
9. Formularios PR
10. Mostrar los PR
11. Edición de Rutinas completas

3.2.2 En Proceso:

1. Apartado de amigos
2. Creación de los retos

3.2.3 Por Hacer:

1. Creación del Qr
2. Lectura de Qr
3. Guardar tu peso y altura
4. Creación de graficas de progreso

3.3 Metodología:

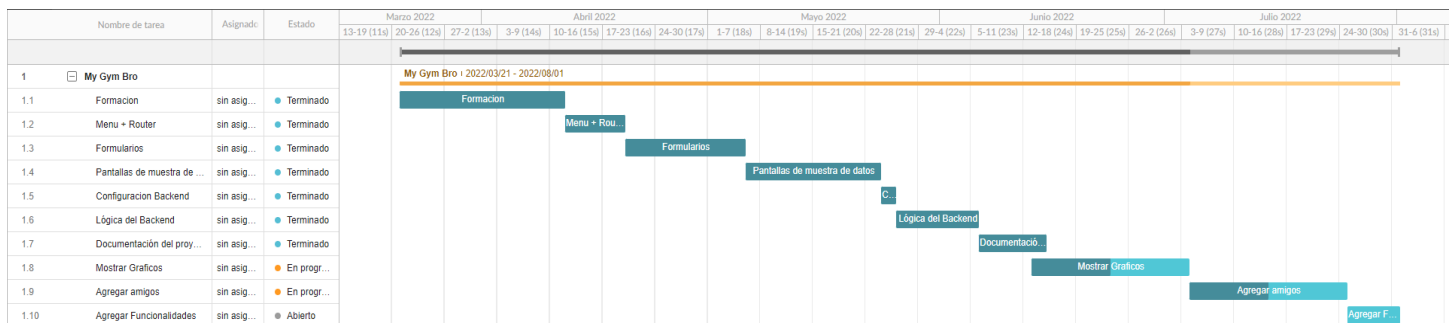
La metodología empleada sería una mezcla de prototipo e incremental. Esto se debe a que se ha creado un prototipo y se están realizando pruebas de usuario para comprobar que tan útil resulta lo que se ha diseñado en una primera instancia. Sin embargo, también se tienen planificados los siguientes incrementos que se irán incluyendo en los demás prototipos que se realicen.

1. La inclusión de los retos
2. El que puedas meter las medidas de tus músculos y tu peso
3. Creación de graficas de los pesos y los progresos
4. Agregar amigos y ver sus progresos

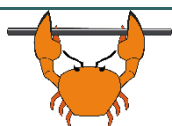
Se ha procedido a escoger esta opción de metodología para que cada poco avance se tenga algo que entregar al cliente y se pueda ir recibiendo un cierto feedback del cliente y de los usuarios cuando se decida a lanzar la aplicación.

Otro factor decisivo para adoptar esta metodología es que el principal requerimiento era el almacenar rutinas y poder revisarlas para ver como habías ido progresando. Y por eso se optó por centrarnos en completar eso e ir avanzando entorno a ese objetivo.

3.4 Planificación temporal de las tareas:



3.5 Presupuesto:





Presupuesto Número 001/2022

Juan S., S.L. C/ Paseo Navarra, 1 , Zaragoza NIF: B-12345678 Teléfono: 612 456 789 Mail: juans-sl@gmail.com	Gimnasios reunidos, S.L. C/ Nueva, 12 , Zaragoza NIF: B-321456987 Teléfono: 612321456 Mail: greunidos@gmail.com
--	--

Fecha del presupuesto	20/03/2022	Validez	30 días
------------------------------	------------	----------------	---------

DESCRIPCIÓN	UNIDADES	PRECIO	TOTAL
<i>Aplicación para gimnasio</i>			
<i>Formación</i>	50	8	400
<i>Horas de trabajo estimadas</i>	100	8	800
<i>Licencia Android</i>	1	25	25
<i>Licencia IOS</i>	1	99	99
<i>Firebase</i>	1	5	5

	SUB-TOTAL	1.329,00 €
	IVA %	21,00%
TOTAL PRESUPUESTADO		1.608,09 €

Firma 	Firma del cliente 
---	---

3.6 Contrato:

CONTRATO DE COMPRAVENTA ENTRE PARTICULARES

En Zaragoza, a 27 de Marzo de 2022

REUNIDOS

De una parte, como LA PARTE VENDEDORA:

D. Juan Manuel Saavedra, mayor de edad, con N.I.F. nº 12345678-B y con domicilio en Zaragoza, calle Avenida Navarra, nº 1, C.P. 50001

De otra parte, como LA PARTE COMPRADORA:

D. Miguel Gracia, mayor de edad, con N.I.F. nº 321456987-B y con domicilio en Zaragoza, calle Nueva, nº 12, C.P. 50005

Ambas partes contratantes se reconocen capacidad legal para este acto, e intervienen en su propio nombre y derecho.

EXPONEN

I.- Que ambas partes han convenido formalizar contrato de compraventa de:

My Gym Bro, una aplicación para dispositivos móviles de Android e IOS creada con flutter a petición de D. Miguel Gracia en representación de Gimnasios Reunidos S.L. Con todas las especificaciones que a continuación de citan:

1. Registrar rutinas, entrenos y PR
2. Mostrar graficas de avances
3. Agregar amigos para poder ver sus progresos
4. Tener un login que guarde usuarios para que si inicias desde otro dispositivo tengas tus progresos y tus rutinas guardadas
5. Agregar, tu peso, tu altura y el volumen de tus músculos para ver como has ido 'creciendo'.
6. Revisar que están haciendo tus amigos y cuáles son sus Pr

Expuesto cuanto antecede, convienen en celebrar el presente contrato de compraventa del Software, de acuerdo con las siguientes:

ESTIPULACIONES

PRIMERA.- El vendedor vende al comprador My Gym Bro de su propiedad anteriormente especificado por la cantidad de **mil trescientos veintinueve euros** (1329,00€), sin incluir los impuestos correspondientes, que serán a cargo del comprador.

SEGUNDA.- El vendedor declara que no pesa sobre el citado bien ninguna carga o gravamen ni impuesto, deuda o sanción pendientes de abono en la fecha de la firma de este contrato, y se compromete en caso contrario a regularizar tal situación a su exclusivo cargo.

TERCERA.- Por este acto se hace entrega al comprador de la posesión del bien. El comprador se hace cargo de cuantas responsabilidades puedan contraerse por su propiedad, tenencia y uso a partir del momento de la entrega. Igualmente, por este acto se realiza el pago al vendedor de la cantidad estipulada, sirviendo este documento como eficaz carta de pago.

CUARTA.- El comprador declara conocer el estado actual del bien y exonera de manera expresa al vendedor de cualquier responsabilidad por vicios o defectos ocultos o posibles averías que el bien manifieste en un futuro, según se determina en el artículo 1.485 del Código Civil, salvo aquellos ocultos que tengan su origen en dolo o mala fe del vendedor.

El vendedor responderá frente al comprador tanto de la posesión legal y pacífica de la cosa vendida como de los vicios o defectos ocultos que tuviere, durante un plazo de seis meses, de conformidad con lo establecido en el artículo 1.490 del Código Civil.

QUINTA.- Para cualquier litigio que surja entre las partes de la interpretación o cumplimiento del presente contrato, éstas, con expresa renuncia al fuero que pudiera corresponderles, se someterán a los Juzgados y Tribunales de Zaragoza

SEXTA.- El vendedor se compromete a entregar una primera versión de My Gym Bro el día 14 de Junio de 2022. Lo mínimo que se le exige es que dicha aplicación tenga un login funcional, y le permita hacer registro de entrenos y pr, y el día 31 de Julio de 2022 deberá entregar la aplicación completa.

Y para que así conste, firman el presente contrato de compraventa, por triplicado, en la fecha y lugar arriba indicados.

Firma de EL VENDEDOR



Firma de EL COMPRADOR



3.7 Análisis de Riesgos:

Los principales riesgos que pueden surgir durante este proyecto son los siguientes:

No cumplir los plazos los plazos de entrega, para evitar esto, se han programado unos plazos de entrega realistas teniendo en cuenta que habría que aprender el lenguaje y es posible que surjan algunas complicaciones

Tener que invertir tiempo y dinero en 2 aplicaciones. La solución a esto viene de utilizar una tecnología híbrida que en este caso es flutter. No obstante, hablamos de una tecnología con un nivel de madurez bajo dado que aun esta surgiendo. Debido a esto se ha optado por ampliar los plazos como se indica anteriormente y dedicar un tiempo a aprendizaje para disminuir los problemas.

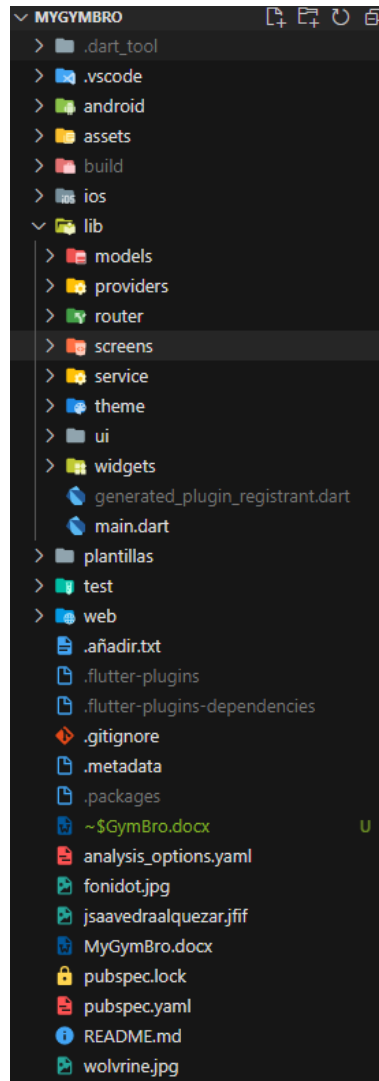
Tener una infraestructura susceptible a errores. Se decidió que desplegaríamos la aplicación utilizando Firebase dado que cuenta con servicios sumamente útiles como Authentication o Realtime Database que son las que se utilizan en este proyecto.

El riesgo de incompatibilidades siempre va a estar vigente, por ello se ha optado por desarrollar la aplicación con la última versión de flutter y la versión de Android 25 que como mínimo necesitas Android 7.1 para ejecutarla la cual es una versión bastante estandarizada y alcanza a estar en un 87.5% de los dispositivos.

4.- Documento de análisis y diseño

4.1 Modelado de datos:

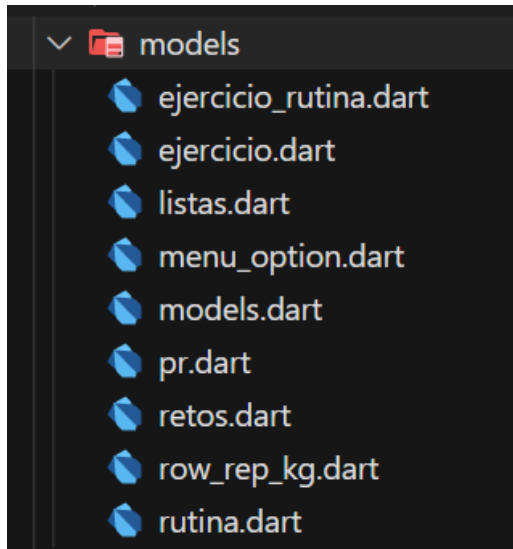
El contenido de la carpeta raíz la cual contiene todo el proyecto sería:



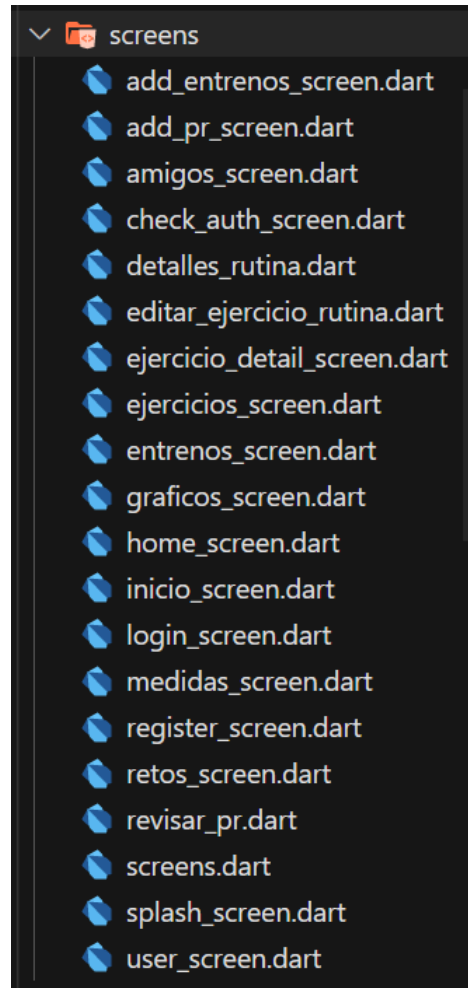
Si nos ponemos a revisar más específicamente las subcarpetas, encontramos las carpetas por defecto, 'android' contiene el proyecto de android, 'assets' contiene todo el contenido grafico de la app, 'ios' contiene el proyecto de ios, 'lib' contiene todos los archivos de código utilizados en el proyecto, 'test' contiene el código para hacer los test y 'web' contiene el proyecto web.

Dentro de 'lib' encontramos diversas carpetas de las que las más relevantes son:

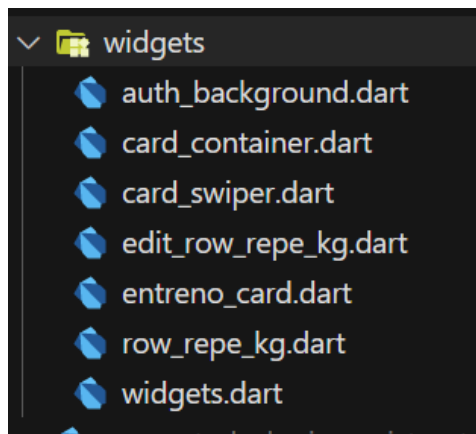
Models:



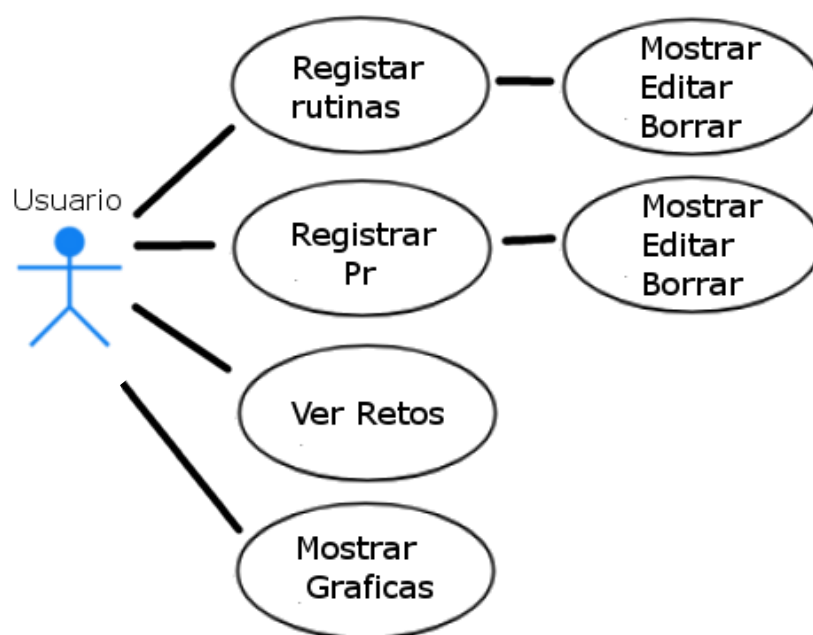
Screens:



Widgets:



4.2 Análisis y diseño del sistema funcional



*ver detallado en el ANEXO I

4.3 Análisis y diseño de la interfaz de usuario:



Este es el Splash Screen de la aplicación.

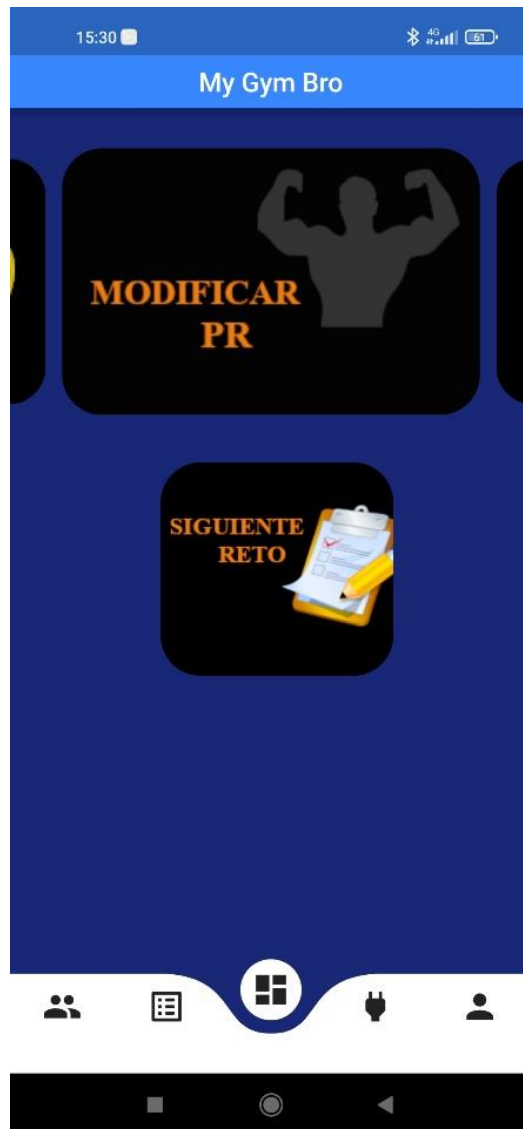
Contiene el nombre y el icono de esta.

Es bastante importante porque durante su ejecución se llama a la base de datos y se obtienen todos los datos relevantes.



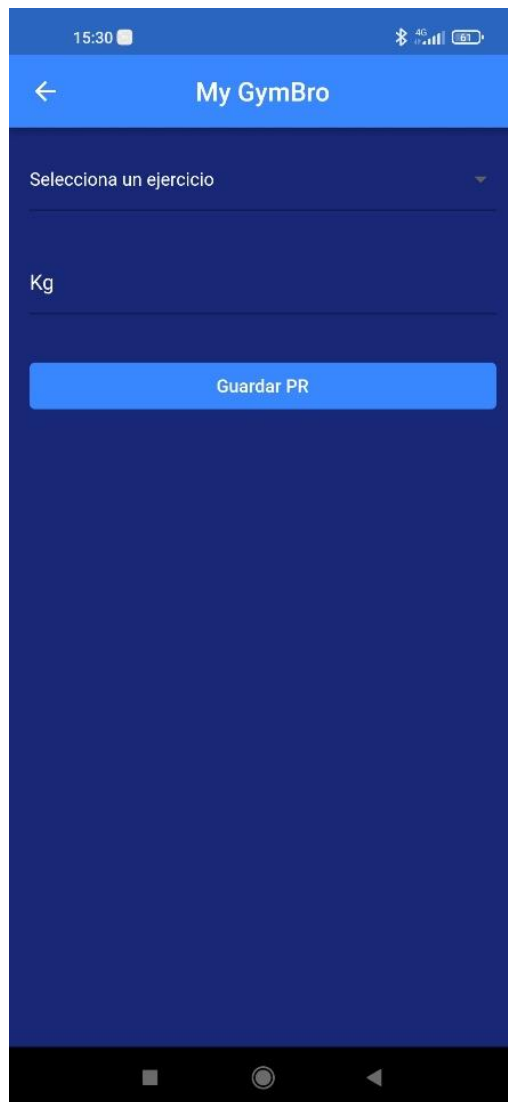
En este formulario de inicio de sesión el usuario puede registrarse o entrar a la aplicación si ya ha iniciado previamente.

Una vez entras, si tienes rutinas creadas ya las tendrás disponibles.



La pantalla inicial de la aplicación.

Puedes usar el Slider para ver el resto de las tarjetas que hay o seleccionar un reto como completado.



Pantalla para agregar o modificar un Pr ya introducido.

Su funcionamiento es simple, seleccionas un ejercicio y le asignas un peso, la aplicación hace el resto.



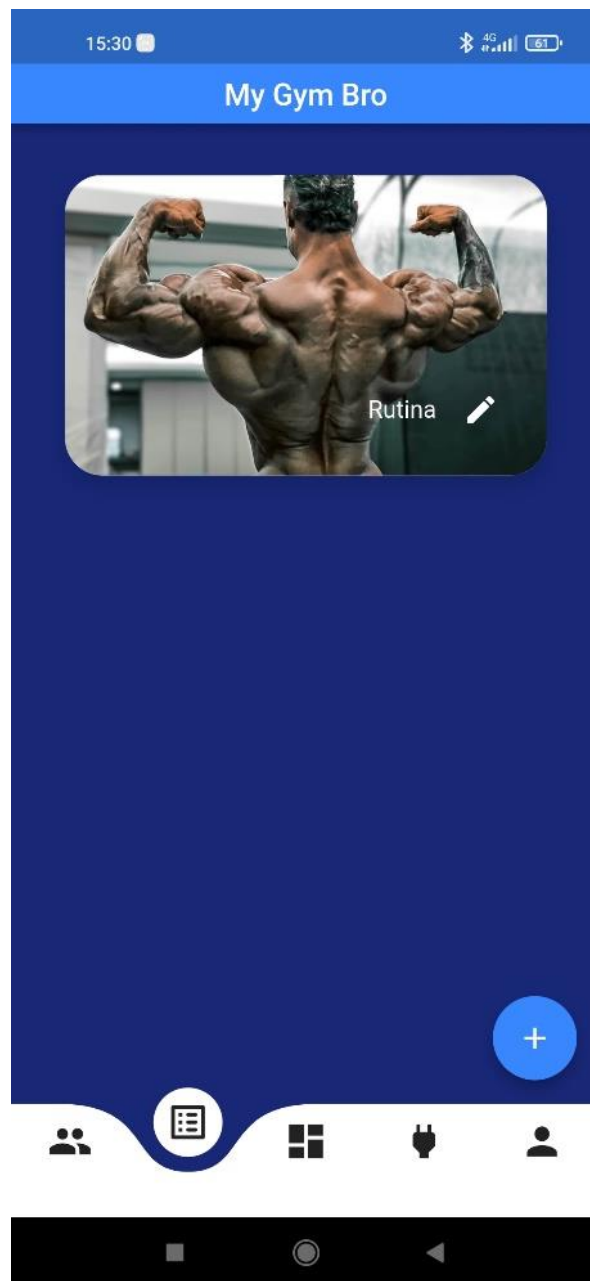
Pantalla de Rutinas.

Puedes ver las rutinas ya creadas o añadir una nueva rutina usando el botón de abajo.

The screenshot shows the 'My Gym Bro' app interface. At the top, the status bar displays the time 15:30, signal strength, and battery level at 61%. The app's title bar is dark blue with the text 'My Gym Bro'. The main content area is dark blue. A light blue rounded rectangle contains the form for adding a routine. The form has three input fields: 'Nombre de la rutina' with the placeholder 'Escribe el nombre', 'Anotaciones' with the placeholder 'Escribe las anotaciones', and a 'Crear Rutina' button at the bottom. The bottom navigation bar is dark blue with five icons: a group of people, a calendar, a grid, a plug, and a person. A blue circular button with a white plus sign is located in the bottom right corner of the main content area.

Formulario de añadir rutinas.

Puedes ponerle nombre, y alguna anotación que se considere relevante.



Misma pantalla de rutinas, pero con una rutina creada, se puede ver el nombre abajo y pulsando en cualquier punto de la tarjeta pasaría a la siguiente pantalla.



Pantalla de detalles de rutina.

Puedes eliminarla, o añadirle ejercicios usando los respectivos botones.

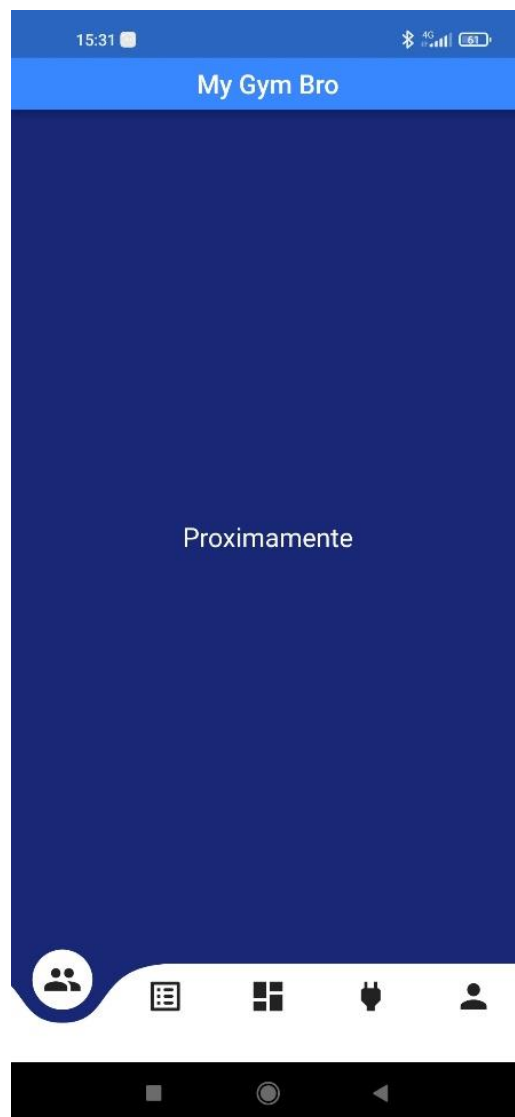


Pantalla para añadir o editar ejercicios en una rutina.

Puedes subir en número de repeticiones, cambiar ejercicios o grupos musculares a voluntad.

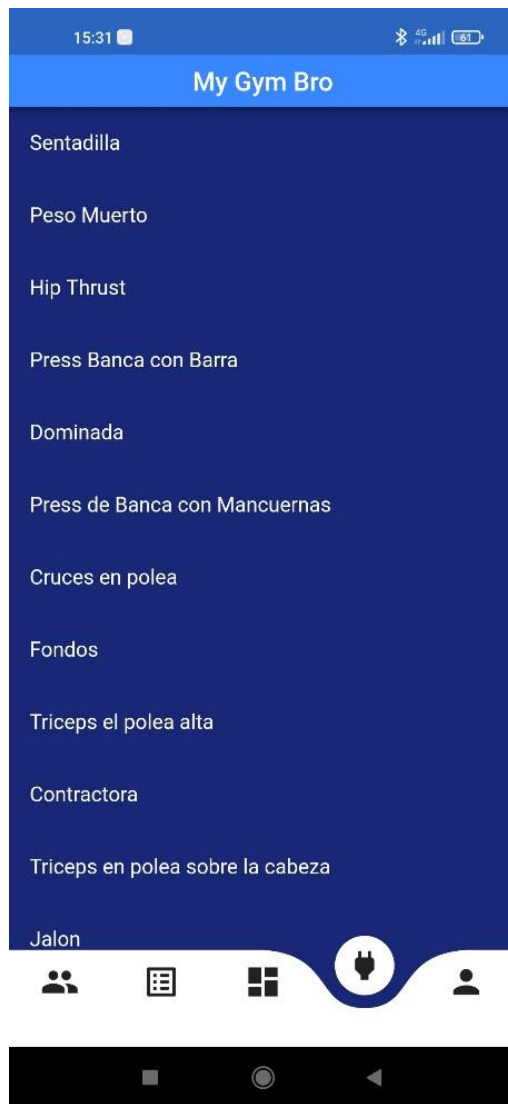


Una vez añadido un ejercicio se vería al abrir la rutina en su pantalla de detalles. Si clicásemos sobre el ejercicio volveríamos a la pantalla anterior con los datos que haya guardados sobre ese ejercicio.



La pantalla de amigos.

Aún estamos trabajando en ella.



Pantalla de ejercicios.

Puedes revisar todos los ejercicios por si tienes dudas sobre cómo realizar algunos.



Pantalla de detalle de ejercicios.

Se puede comprobar que grupos musculares se ejercitan con cada ejercicio, ver un gif de cómo realizar un cierto ejercicio y un consejito que te daría tu gym bro sobre cómo realizar el ejercicio de forma correcta y segura.



Pantalla de usuario.

Desde aquí puedes revisar tus pr.

En un futuro podrás revisar más cosas como tu peso corporal o el volumen de tus músculos.



Pantalla revisar pr.

Aparece un listado con tus pr guardados.

4.4 Diseño de la arquitectura de la aplicación:

4.4.1 Tecnologías utilizadas

Flutter es un SDK de código fuente abierto de desarrollo de aplicaciones móviles creado por Google. Suele usarse para desarrollar interfaces de usuario para aplicaciones en Android, iOS y Web, así como método primario para crear aplicaciones para Google Fuchsia.

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles lanzada en 2011 y adquirida por Google en 2014. Dentro de su gama de utilidades, una de las implementadas es Realtime Database para almacenar los datos de nuestra base de datos como son los usuarios, sus rutinas, y pr. La otra funcionalidad utilizada es Firebase Authenticator, la cual autoriza o no a los usuarios a entrar a nuestra aplicación en función de si tienen una cuenta creada o no.

Visual Studio Code es el entorno utilizado para realizar el proyecto dado que es ligero, pero igualmente potente. Además, cuenta con multitud de extensiones muy útiles como 'flutter widget snippets' que ayuda bastante a la hora de programar.

Además de eso, fue necesario el uso de un emulador de Android para hacer las pruebas de la aplicación y ver cómo iba quedando el diseño.

Durante todo el desarrollo de utilizó mi equipo personal.

4.4.2 Arquitectura de componentes de la aplicación:

La librería http en su versión 0.13.4 es utilizada para realizar todas las peticiones http que son necesarias en la aplicación.

Flutter Secure Storage nos permite almacenar en local datos como los tokens de usuario para no tener que iniciar sesión de nuevo cada vez que entramos en la aplicación.

La librería Get es utilizada a la hora de crear una instancia de manera global de la clase 'Listas' para llamar a las listas siempre que sea necesario.

Usando la librería Provider en vez de tener que crear manualmente los InheritedWidgets obtenemos una simplificación de los recursos de carga lenta entre otras cosas, en nuestro caso, la utilizamos para incrementar la escalabilidad de las clases con un mecanismo de escucha que crece exponencialmente en su complejidad como es ChangeNotifier (provider | Flutter Package, s. f.).

Se importaron los siguientes componentes:

- CardSwiper versión 2.0.3
- SplashScreenView versión 3.0.0
- CurvedNavigationBar versión 1.0.3

Todos obtenido de fluttergems

Se crearon dos API json donde están almacenados todos los ejercicios y los retos disponibles en la aplicación. Reciben el nombre de ejercicios.json y retos.json respectivamente y está en la carpeta assets.

5.- Documentación de implementación e implantación del sistema.

5.1 Implementación:

```
Run | Debug | Profile  
void main() {  
    //creamos la clase listas para  
    Get.put(Listas());  
    runApp(AppState());  
}
```

Gracias la línea `Get.put(Listas());` podemos crear una instancia de esa clase y llamarla siempre que queramos con la siguiente: `Get.find<Listas>()`;

```
final controller = Get.find<Listas>();
```

```
Future<String?> createUser(String email, String password) async {  
    final Map<String, dynamic> authData = {  
        'email': email,  
        'password': password,  
        'returnSecureToken': true  
    };  
  
    //creamos la url con uri.https  
    final url =  
        Uri.https(_baseUrl, '/v1/accounts:signUp', {'key': _firebaseToken});  
  
    final resp = await http.post(url, body: json.encode(authData));  
    final Map<String, dynamic> decodedResp = json.decode(resp.body);  
  
    //guardamos el token que nos devuelve  
    if (decodedResp.containsKey('idToken')) {  
        // Token hay que guardarlo en un lugar seguro  
        await storage.write(key: 'token', value: decodedResp['idToken']);  
  
        await storage.write(key: 'uid', value: decodedResp['localId']);  
        // decodedResp['idToken'];  
        return null;  
    } else {  
        return decodedResp['error']['message'];  
    }  
}
```

Con este método creamos los usuarios.

Para ello creamos un objeto para enviar los datos.

Lo enviamos a la base de datos para crear el usuario poniendo `/v1/accounts:signUp` y pasamos los datos por parámetro.

Guardamos el `idtoken` y el `token` para trabajar con ellos después.


```
Future<Ejercicio> fetchEjercicio() async {
  final String response =
    await rootBundle.loadString('assets/ejercicios.json');
  return Ejercicio.fromJson(json.decode(response));
}
```

Leemos los ejercicios del json, y los cargamos a la lista.

```
Future loadRutinas() async {
  final url = Uri.https(_baseUrl, 'rutinas.json');
  final resp = await http.get(url);

  final String user = await storage.read(key: 'uid') ?? '';

  print('+' + user);
  final Map<String, dynamic> rutinasMap = json.decode(resp.body);

  rutinasMap.forEach((key, value) {
    final tempRutina = Rutina.fromMap(value);
    tempRutina.id = key;
    if (tempRutina.idUser == user) {
      controller.rutinasList.add(tempRutina);
    }
  });
}
```

Cargamos las rutinas de la base de datos y si el id del usuario coincide con el que hay almacenado, lo guardamos en la lista.

```

Future<String> updateRutina(Rutina rutina) async {
/HACEMOS LA PETICION

    final url = Uri.https(_baseUrl, 'rutinas/${rutina.id}.json');
    final resp = await http.put(url, body: rutina.toJson());
    final decodedData = resp.body;
    print(decodedData);

    return rutina.id!;
}

Future<String> createRutina(Rutina rutina) async {
/HACEMOS LA PETICION

    final url = Uri.https(_baseUrl, 'rutinas.json');
    final resp = await http.post(url, body: rutina.toJson());
    final decodedData = json.decode(resp.body);
    print(decodedData);

    rutina.id = decodedData['name'];

    return rutina.id!;
}

```

Para modificar/ crear las rutinas, el código sería idéntico, pero con la diferencia de que en el update usamos http.put y en la url enviamos el id de la rutina. El en créate hacemos post y el id en vez de enviarlo lo recogemos.

```

ListView.builder(
    itemCount: numseries,
    shrinkWrap: true,
    primary: false,
    itemBuilder: (_, int index) {
        return RowRepeKg(list);
    },
), // ListView.builder

```

Con estos ListView.builder podemos crear tantas filas como series hayamos marcado que queremos hacer. También se puede usar para replicar widgets tantas veces como se quiera.

```

List<DropDownMenuItem<EjercicioElement>> get dropdownEjerciciosItems {
  List<DropDownMenuItem<EjercicioElement>> menuEjerciciosItems = [];
  for (var item in controller.ejerciciosList) {
    if (item.muscle.contains(grupoMuscular)) {
      menuEjerciciosItems
        .add(DropDownMenuItem(child: Text(item.name), value: item));
    }
  }
  return menuEjerciciosItems;
}

```

Creamos un método para rellenar el DropDownButton de ejercicios con todos los ejercicios que pertenezcan a un cierto grupo muscular.

```

FutureBuilder(
  future: authService.readToken(),
  builder: (BuildContext context, AsyncSnapshot<String> snapshot) {
    if (!snapshot.hasData) return Text('');

    if (snapshot.data == '') {
      Future.microtask(() {
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(
            pageBuilder: (_, __, ___) => LoginScreen(),
            transitionDuration: Duration(seconds: 0)));
      });
    } else {
      Future.microtask(() {
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(
            pageBuilder: (_, __, ___) => HomeScreen(),
            transitionDuration: Duration(seconds: 0)));
      });
    }
  }
)

```

Leemos el token que esta guardado en el local storage de la aplicación y en función de lo obtenido cargamos el login o la pantalla de home.

6.- Documento de cierre

6.1 Documento de instalación y configuración

No procede.

6.2 Manual de usuario

No Procede.

6.3 Resultados obtenidos y conclusiones

La conclusión que saco de este proyecto es que da igual lo que me pongan delante o lo que me manden, voy a ser capaz de sacarlo adelante.

A lo largo del desarrollo de la aplicación ha habido bloqueos, muros, y baches. Pero de una forma u otra se encontró la forma de sortear todos los obstáculos. El resultado es esta aplicación. La cual no está completa, es un prototipo, pero pese a eso, sigue cumpliendo con todos los objetivos mínimos para hacerla funcional.

Este prototipo ha sido probado por usuarios de gimnasio, han comentado mejoras, inquietudes al respecto y cosas que se podrían hacer de otra manera. No obstante, todos coinciden en que la aplicación tiene muy buena pinta y que estaban muy interesados en ver el final. Han coincidido en cosas a mejorar, como es la parte gráfica, pero es normal dado que no soy diseñador, soy desarrollador. Y respecto a mi trabajo he recibido muchas críticas positivas y que se nota que he puesto mucho cariño y trabajo en ella.

6.4 Diario de bitácora

No procede

7.- Bibliografía y Webgrafía

Khurram Dev. (2021, 2 de abril). *Generate Class Diagram or Uml Diagrams From flutter dart code* [Video]. YouTube. <https://www.youtube.com/watch?v=USXMjX52OpA>

Maximilian Schwarzmüller (06/2022) *A Complete Guide to the Flutter SDK & Flutter Framework for building native iOS and Android apps* [Curso] Udemý. [Flutter & Dart - The Complete Guide \[2022 Edition\] | Udemý](#)

provider | Flutter Package. (s. f.). Dart packages. <https://pub.dev/packages/provider>

splash_screen_view | Flutter Package. (s. f.). Dart packages. https://pub.dev/packages/splash_screen_view

curved_navigation_bar | Flutter Package. (s. f.). Dart packages. https://pub.dev/packages/curved_navigation_bar

card_swiper | Flutter Package. (s. f.). Dart packages. https://pub.dev/packages/card_swiper

Fernando Herrera. (12/2021). *Flutter: Tu guía completa de desarrollo para IOS y Android* [Curso]. <https://www.udemy.com/course/flutter-ios-android-fernando-herrera/>

Fernando Herrera. (12/2021). *Flutter Avanzado: Lleva tu conocimiento al siguiente nivel* [Curso]. <https://www.udemy.com/course/flutter-avanzado-fernando-herrera/>

Ayuda ListView Flutter Json. (s. f.). Stack Overflow en español. <https://es.stackoverflow.com/questions/390551/ayuda-listview-flutter-json>

Uso Correctamente del ListView en Flutter? (s. f.). Stack Overflow en español. <https://es.stackoverflow.com/questions/473942/uso-correctamente-del-listview-en-flutter>

Listar datos en flutter con firebase. (s. f.). Stack Overflow en español. <https://es.stackoverflow.com/questions/255155/listar-datos-en-flutter-con-firebase>

Agrega Firebase a tu app de Flutter | Firebase Documentation. (s. f.). Firebase. <https://firebase.google.com/docs/flutter/setup?hl=es-419&platform=ios>

Get Started with Firebase Authentication on Flutter | Firebase Documentation. (s. f.). Firebase. <https://firebase.google.com/docs/auth/flutter/start>

Realtime Database | FlutterFire. (s. f.). FlutterFire | FlutterFire. <https://firebase.flutter.dev/docs/database/overview/#:~:text=The%20Firebase%20Realtime%20Database%20is,updates%20with%20the%20newest%20data.>

Flutter - Firebase Realtime Database Problem. (s. f.). Stack Overflow. <https://stackoverflow.com/questions/65543817/flutter-firebase-realtime-database-problem>

weincod. (2021, 6 de octubre). *Peticiones http en Flutter - con app ejemplo* [Video]. YouTube. <https://www.youtube.com/watch?v=SAcJBvCYb8>

Firebase Realtime database with flutter ERROR. (s. f.). Stack Overflow. <https://stackoverflow.com/questions/58537029/firebase-realtime-database-with-flutter-error>

flutter_secure_storage | Flutter Package. (s. f.). Dart packages. https://pub.dev/packages/flutter_secure_storage

get storage flutter - Flutter Tutorial. (s. f.). Flutter Tutorial. <https://flutternutorial.in/get-storage-flutter/>

How to save to local storage using Flutter? (s. f.). Stack Overflow. <https://stackoverflow.com/questions/41369633/how-to-save-to-local-storage-using-flutter>

Manu Codes. (2020, 17 de junio). *Flutter y Firebase Auth con Provider | Desde cero* [Video]. YouTube. <https://www.youtube.com/watch?v=XidN4qBZLuI>

ANEXO I

