

Gym Tracker



Master's in Informatics Engineering
Mobile Computing

Bruno Sequeira 2020235721, brunosequeira@student.dei.uc.pt
Ricardo Guegan 2020211358, ricardoguegan@student.dei.uc.pt
Rui Santos 2020225542, rpsantos@student.dei.uc.pt

Contents

1	Introduction	3
2	Requirements	4
2.1	Actors	4
2.2	User Stories	4
2.3	Functional Requirements	9
3	Wireframes	10
4	Architecture	15
4.1	C1 - Context Diagram	15
4.2	C2 - Container Diagram	16
4.3	C3 - Component Diagram	17
5	Data Bases	20
5.1	SQL Database Schema	20
5.2	Firebase Schema	21
6	User Manual	22
6.1	Login	22
6.2	Menu	22
6.3	Listing/Create Plans	23
6.4	Editing plans	25
6.5	Execute plans	25
6.6	Graphic of results	28
6.7	Friends	28
7	Planning	30
8	Work distribution	32
9	Conclusion	33

List of Figures

1	Login Page	11
2	Menu Page	11
3	List of Plans Page	11
4	Create Plans Page	11
5	Edit Plans Page	12
6	Edit Training Page	12
7	Edit info about an exercise Page	12
8	Executing Plan Page	12
9	Delete Plan Page	13
10	Add weight Page	13
11	Add exercise into Plan	13
12	Details about a Train Done Page	13
13	Friends List Page	14
14	Friend Menu	14
15	Exercises List Page	14
16	Evolution Exercise Page	14
17	C4-Context Diagram	16
18	C4-Container Diagram	17
19	C4 - Component Diagram	18
20	SQL Database Schema	20
21	FireStore Database Schema	21
22	Login Page	22
23	Menu Page	23
24	Train Done Detail	23
25	Plans List Page	24
26	Create Plan Page	24
27	Details Plan Page	24
28	Helper Details Plan Page	24
29	Editing a plan	25
30	Add an exercise	25
31	Order Exercise	25
32	Exercises for a selected plan	27
33	Finish a plan	27
34	Asking to input weight	27
35	Pulse and Oxygen scan	27
36	Progress lost	27
37	All exercises done	27
38	List of exercises	28
39	Compare with friend the average weight	28
40	Friends page	29
41	Friend Profile	29
42	Project Planning	31

1 Introduction

Monitoring physical progress and managing training routines have become increasingly essential for individuals looking to improve their performance in the gym. However, many people still face challenges related to organizing and monitoring their physical activities, as well as comparing their results with other people.

To resolve this issue, this project proposes the development of a mobile application that aims to facilitate the recording and monitoring of training, providing users with a practical way to visualize their progress, record exercises and interact socially with other practitioners.

The app will be designed to allow users to create and customize their workout routines, record weights lifted in each exercise and view their performance over time. Additionally, the platform will integrate social features, allowing users to follow friends, compare their results and explore others' training routines, promoting healthy competition and encouraging constant evolution.

With the addition of features such as measuring vital signs through connected devices such as Arduino, users will be able to gain a more holistic view of their health and physical evolution.

This project therefore aims to offer an innovative and complete solution for training management and social interaction in the context of physical activity, promoting not only individual well-being, but also collective motivation among friends and users of the platform.

2 Requirements

2.1 Actors

The main actors involved in this system are the users of the application, those who practice physical activities. These can be divided into categories:

- **Common user:** User who creates training routines, records progress and interacts with the platform, being able to follow other users and copy training routines.
- **Friends and followers:** Users who interact socially, competing or comparing their results with others, promoting an environment of motivation and challenge.
- **Arduino Device:** Acts as a secondary actor, being responsible for measuring and sending data such as heart rate and oxygen levels to the application during training, allowing the user's health to be monitored. These actors have fundamental roles in interacting with the application, whether through creating and monitoring training, visualizing data or sharing social information.

2.2 User Stories

User stories are an important tool for understanding users' needs and guiding system development. These stories reflect users' needs and expectations in relation to the application, helping to guide the implementation of features that meet their main demands.

User Story: Register User

As a user, I want to register an account in the app, so I can save and track my training routines.

Acceptance Scenarios:

1. The app asks for basic information such as username and password.
2. After registration, the user is redirected to the menu page.

User Story: User Login

As a user, I want to log in to my account, so I can access my training routines and progress.

Acceptance Scenarios:

1. The app asks for basic information such as username and password.
2. After login, the user is redirected to the menu page.

User Story: View List of Workouts

As a user, I want to view the list of workouts I've performed, so I can track my training history.

Acceptance Scenarios:

1. The app displays a list of workouts performed by the user.
2. Each entry contains workout details such as exercise, date.
3. The user can tap on a workout to view more detailed information.

Use Case: Create Training Routine

As a user, I want to create a new training routine by selecting exercises from a list, so I can customize my workout plan.

Acceptance Scenarios

1. Allow the user to access a menu to create training routines.
2. Provide a list of exercises.
3. Save the new routine to the user's device.

User Story: Training Listing

As a user, I want to view the list of all saved workouts so that I can select and manage the plans I have created or followed.

Acceptance Scenarios

1. The app displays a list of all workouts saved by the user.
2. Each item in the list shows the name of the workout.
3. The user can click on a workout to view details (exercises, sets, reps, etc.).
4. The user can access the options to edit or delete a workout directly from the list.

User Story: Delete Workout

As a user, I want to delete a workout from my list so that I can keep my workout collection organized and free from unnecessary items.

Acceptance Scenarios

1. The app allows the user to select a workout to be deleted.
2. Before deleting the workout, the app asks for confirmation (e.g., "Are you sure you want to delete this workout?").
3. After confirmation, the workout is permanently removed from the workout list.
4. The app displays a message informing that the workout was successfully deleted.

User Story: Edit Workout

As a user, I want to edit an existing workout so that I can add, remove, or modify exercises and other details.

Acceptance Scenarios

1. The app allows the user to select a workout from the list for editing.
2. The user can add new exercises to the workout, remove or order existing ones.
3. The user can modify the workout details (e.g., number of sets, reps, etc.).
4. The app saves the changes made to the workout and reflects them in the user's workout list.

Use Case: Start Training

As a user, I want to start a saved routine, so I can track the execution of my workout.

Acceptance Scenarios

1. Display a collection of saved routines.
2. Allow the user to select a routine to start.
3. Open a new screen for routine tracking.

Use Case: Log Weights

As a user, I want to add lifted weights during an active session, so I can track my performance.

Acceptance Scenarios

1. Allow the user to select an exercise within the active workout.
2. Add a set to the exercise.
3. Log weights lifted.

Use Case: End Training

As a user, I want to end a workout session and view a summary of my performance, so I can compare it with previous records.

Acceptance Scenarios

1. Provide an option to end the active session.
2. Save the completed session permanently on the device.

Use Case: View Progress

As a user, I want to select exercise, so I can view detailed progress over time.

Acceptance Scenarios

1. Allow the user to select one.
2. Display progress with charts or visualizations, including muscle group breakdowns.

Use Case: Follow Users

As a user, I want to follow other users, so I can access their workout routines and track their activities.

Acceptance Scenarios

1. Display user profiles with available routines.
2. Allow the user to follow other profiles.
3. Enable viewing of followed users' workout routines.
4. Provide an option to copy routines to the user's personal collection.

Use Case: Add Friends

As a user, I want to add friends by mutually following users, so I can compare our performance and workouts.

Acceptance Scenarios

1. Mark users as "friends" when they follow each other.
2. Allow overlaying performance charts between friends.
3. Display side-by-side comparisons to encourage friendly rivalry.

User Story: See progress of an exercise

As a user, I want to see the progress I have made in an exercise, so I can see if I'm on a good path and progressing.

Acceptance Scenarios

1. The app allows the user to select one exercise.
2. The progress is displayed visually using graphs, charts, or other visual tools.
3. The user can compare their progress with personal records.

User Story: Compare Performance with Friends

As a user, I want to compare my performance with my friends, so I can see how we stack up against each other.

Acceptance Scenarios

1. The app allows the user to compare specific exercise with a friend.
2. The app provides graphs and statistics to visualize the comparison.

User Story: Exercise Listing

As a user, I want to view the list of all available exercises so that I can choose which exercise I will see the evolution of.

Acceptance Scenarios

1. The app displays a list of all available exercises from the database.

Use Case: Exhaustion Tracking

As a user, I want to connect an Arduino device to measure my heart rate and oxygen level, so I can understand the impact of exercises on my body.

Acceptance Scenarios

1. Establish a connection with the Arduino device.
2. Measure and log heart rate and oxygen levels during each exercise.
3. Associate recorded data with the exercises performed.
4. Provide insights on which activities are most challenging.

2.3 Functional Requirements

Functional requirements define how a software system should operate to meet the objectives and expectations of its users. They specify the capabilities, tasks, and behaviors that the system must support to fulfill its intended purpose.

The requirements were classified according to the MoSCoW prioritization method[2]. This is used to divide the requirements into four categories:

- **Must:** The Non-Negotiables features that are mandatory.
- **Should:** Important features that may not be essential for the core functionality but significantly contribute to the product's success.
- **Could:** Desirable features but not critical to the product's success.
- **Won't:** Features that won't be included in the current iteration but may be considered in the future.

REQ's IDs	Requirements Name	Priority
REQ-001	Registering a user	Must
REQ-002	User login	Must
REQ-003	View the list of workouts performed	Must
REQ-004	Viewing training details	Must
REQ-005	Add new friends	Must
REQ-006	Display a list of the logged in user's training plans	Must
REQ-007	Create a training plan.	Must
REQ-008	Adding exercises to a training plan	Must
REQ-009	Edit training plan exercises (Delete and sort).	Must
REQ-010	Executing a training plan	Must
REQ-011	Deleting a training plan	Must
REQ-012	Putting the weight of a set into an execution of a training plan.	Must
REQ-013	Respond to new friend requests.	Must
REQ-014	View the list of available exercises	Must
REQ-015	Visualize the evolution over time of an exercise	Must
REQ-016	Compare the evolution of an exercise over time with a friend	Must
REQ-017	Copying a friend's training plan.	Must
REQ-018	Analysis of the Impact of Exercise on the Body using Arduino.	Must

Table 1: Functional Requirements

3 Wireframes

In this section, we show the planned visual representations for **Gym Tracker**. Wireframes play a crucial role in the design process, serving as simplified visual representations of an interface's structure and layout. They serve as an essential tool for aligning teams and stakeholders regarding the organization of elements, information hierarchy, and navigation flow before moving on to more detailed phases of the project.

Initially it starts with the "Login Page", **Figure 1**, which will display an image of the application, and two fields for the user to enter their credentials (Username and Password).

There are 2 options below, registration to create an account or login to access your account. After that, you will be redirected to the "Menu Page", **Figure 2**, which displays the complete number of workouts, and the history of each workout. If you click on your completed training, a pop up will be displayed with the details of that training, **Figure 12**.

The '+' option at the bottom of the page is the addition of new friends 13. Here you can see your friendslist, the list of friend requests (and the option to accept or reject the request) and the users list, where you can send friend requests. When pressing the button "Visitar" of each friend, it shows a page with that friend profile with the options to show details of his last training plans executed and to copy them 14.

Figure 3 shows the list of training sessions. If you choose to click on the '+', a pop up will be displayed to write the name of the new training as shown in **Figure 4**.

If you click on any workout, **Figure 5** will be displayed, which shows the details of the training plan. You have 3 options, if you click on "Edit" you will go to **Figure 6**, where you can edit the training plan, and you can click on the option (+) where a pop up will be displayed adding the exercise, repetitions and sets, **Figure 11**. You can delete the exercise by dragging it sideways and you can sort it by dragging it up or down.

Figure 7 is when the user, in the editing phase of a plan, clicks on an exercise already included in the plan and edits its information (repetitions and series). If in Edit Plans Page, **Figure 5**, he chooses to execute the plan, by clicking on **Start** the execution page will appear, **Figure 8**.

He must click on each exercise to enter the weight he performed in that series and also scan for oxygen and pulse levels, resulting in **Figure 10**. Another option in Edit Plans Page, **Figure 5**, is the deletion of a plan, the **Delete** option will appear a pop up to confirm the deletion, **Figure 9**.

On practically every page a similar NavBar is presented, with the first option, the Logout, the second presents the list of training plans, the third is the main menu, and finally the list of all available exercises, **Figure 15**. This The last page presents all the exercises, and if you click on any one, you will go to the page that presents their evolution over time, **Figure 16**. This page presents 2 spinners, one of them to select the evaluation option (average of the weights , maximum weight), or another is the friend who wants to make the comparison, this being a non-obligatory option.

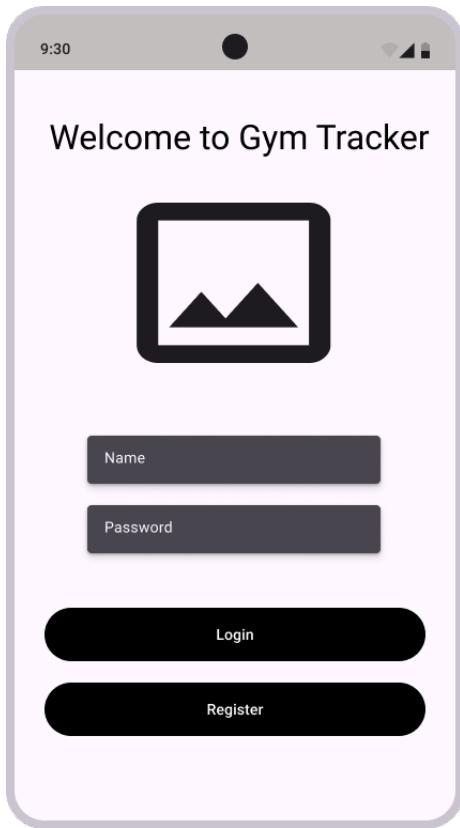


Figure 1: Login Page

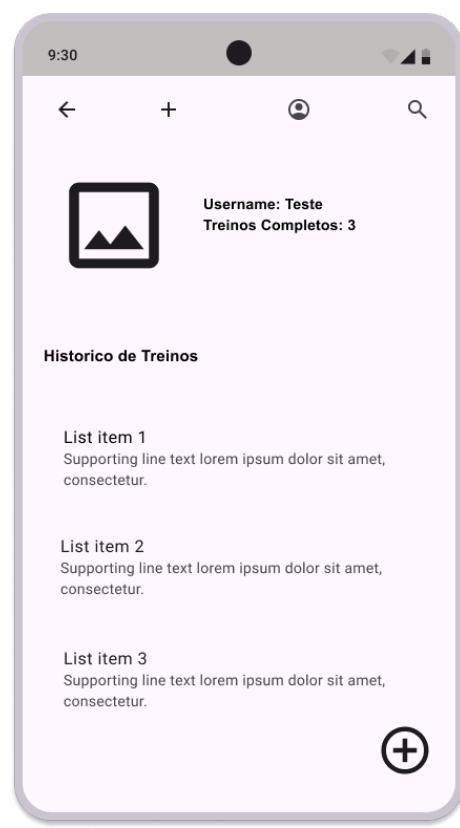


Figure 2: Menu Page



11

Figure 3: List of Plans Page

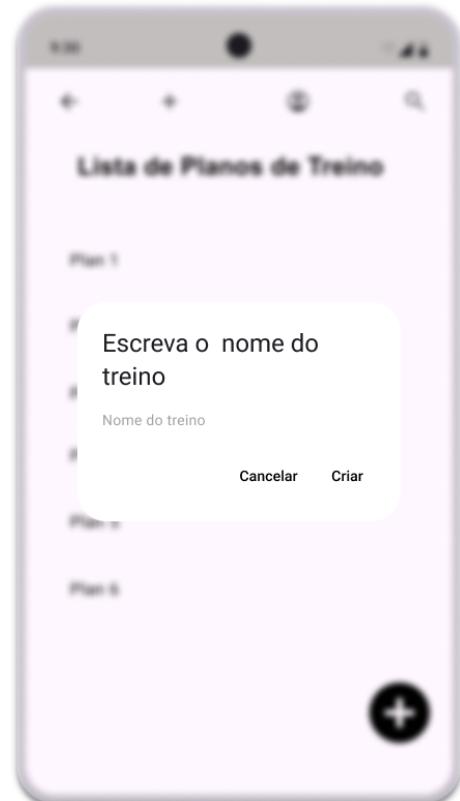


Figure 4: Create Plans Page

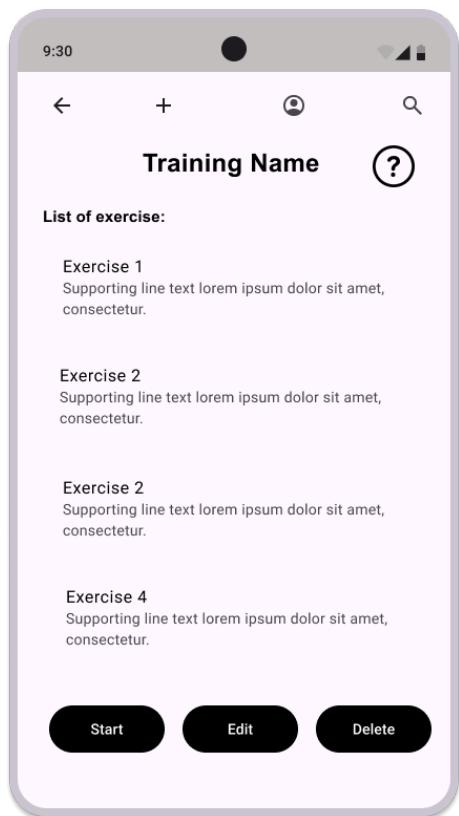


Figure 5: Edit Plans Page

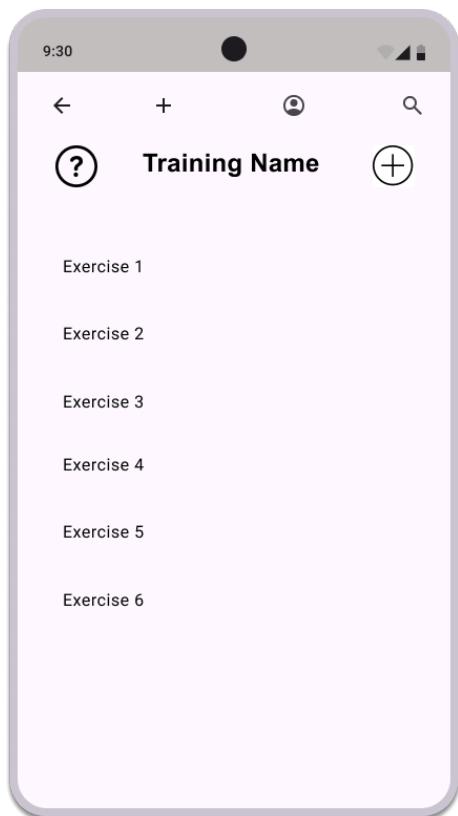


Figure 6: Edit Training Page

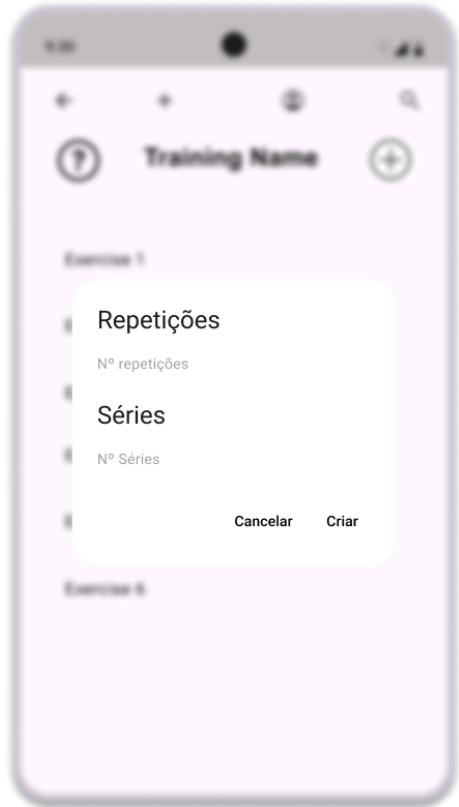


Figure 7: Edit info about an exercise Page

12

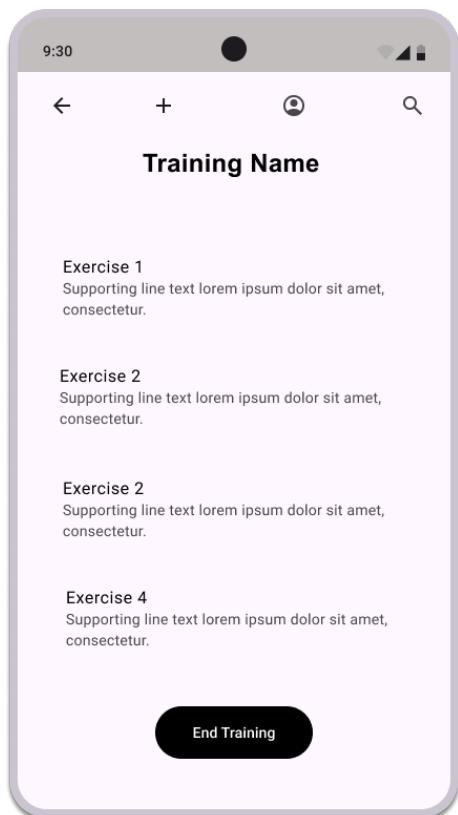


Figure 8: Executing Plan Page

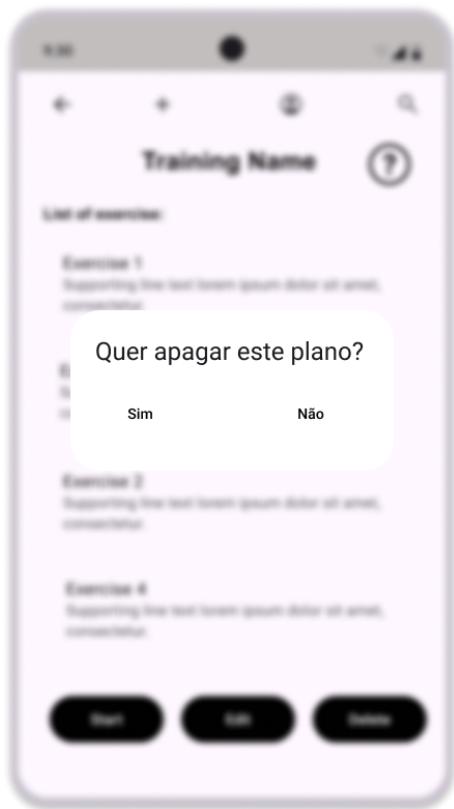
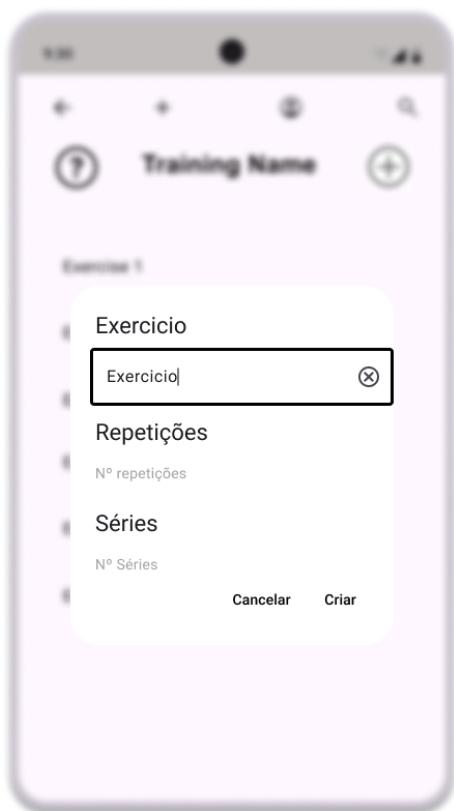


Figure 9: Delete Plan Page



Figure 10: Add weight Page



13

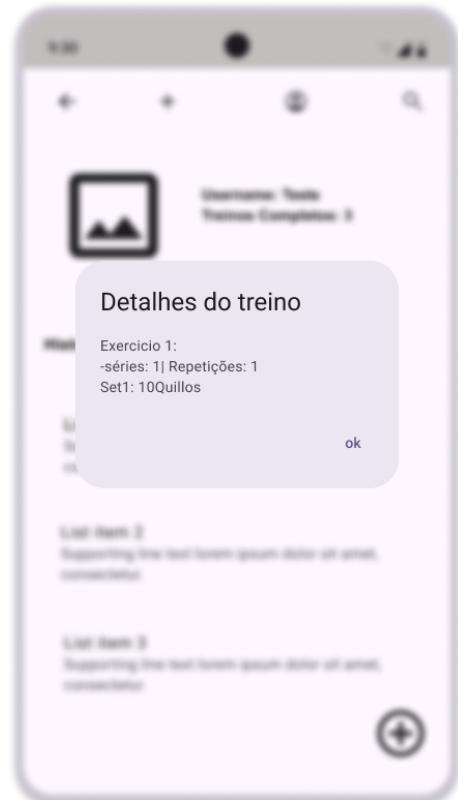


Figure 11: Add exercise into Plan

Figure 12: Details about a Train Done Page



Figure 13: Friends List Page

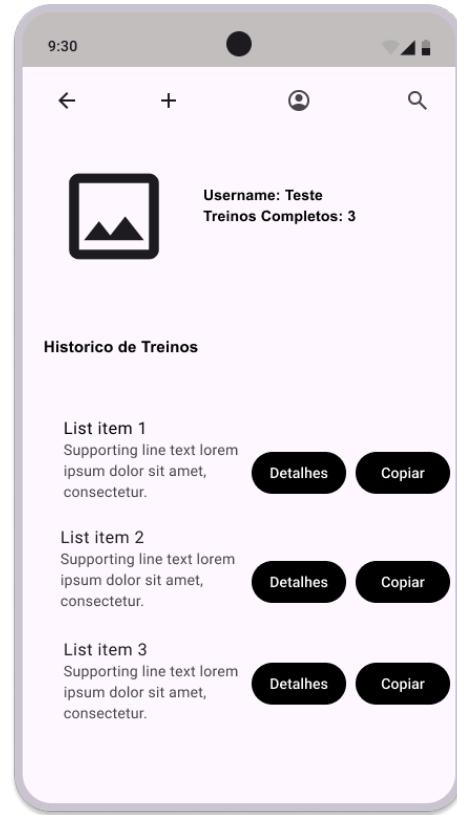


Figure 14: Friend Menu

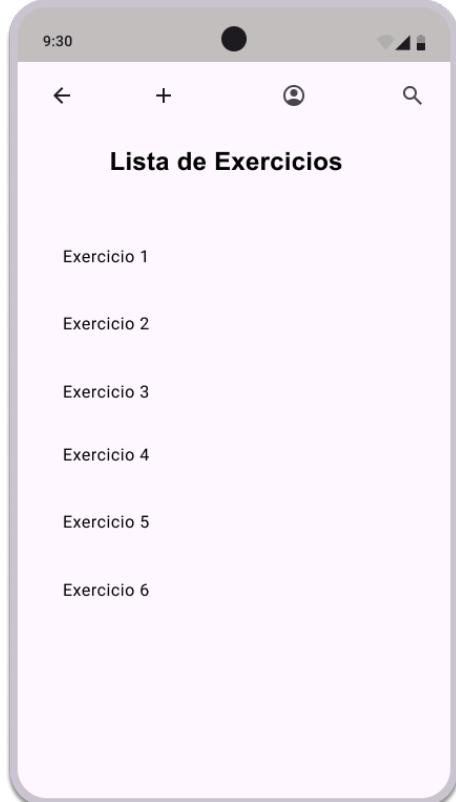


Figure 15: Exercises List Page



Figure 16: Evolution Exercise Page

4 Architecture

This chapter walks through architectural diagrams describing the design and structure of Gym Tracker App.

These diagrams provide a visual representation of the various components of the system, and how they interact, to give a better understanding of how this system works.

The architecture is presented through the **C4 Model**, developed by Simon Brown, a hierarchical framework that standardizes the visualization and communication of the structure of software systems[1].

The C4 model is effective for representing systems ranging from simple to highly complex, making it an ideal school for visualizing system integration automation.

The C4 Model consists of four diagrams that we can use to represent the architecture of a software system. These levels are:

- **C1 - Context Diagram:** These high-level diagrams show the context relationships between the system and its external actors or stakeholders. It provides a broad overview of the system and its environment.
- **C2 - Container Diagram:** The container diagram displays the high-level software architecture and how responsibilities are distributed among different components . It also shows the technology and how containers communicate with each other. It's focused on high-level technology and is a simple diagram.
- **C3 - Component Diagram:** The component diagram displays how a container is comprised of a series of components, what each of them is, their responsibilities, and technology/implementation details. An important point to note here is that all components within a container typically run in the same process space.
- **C4 - Code Diagram:** This is the most detailed level of the C4 Model, showing the classes, their attributes, and the relationships between them.

4.1 C1 - Context Diagram

The context diagram provides a high-level overview of the actors and software systems involved in the architecture.

Based on the figure presented below, we can give a brief explanation. The user is the person who uses the application to monitor their physical activities and track their progress. Gym tracker is the main system that works as an application for tracking exercises and physical progress. facilitates monitoring of training, and organizes data related to users' physical activities. The system uses the service provided by Google Cloud Platform for storage and other resources related to cloud computing. The cloud service provider is the platform that provides the resources needed for the fitness tracker to work.

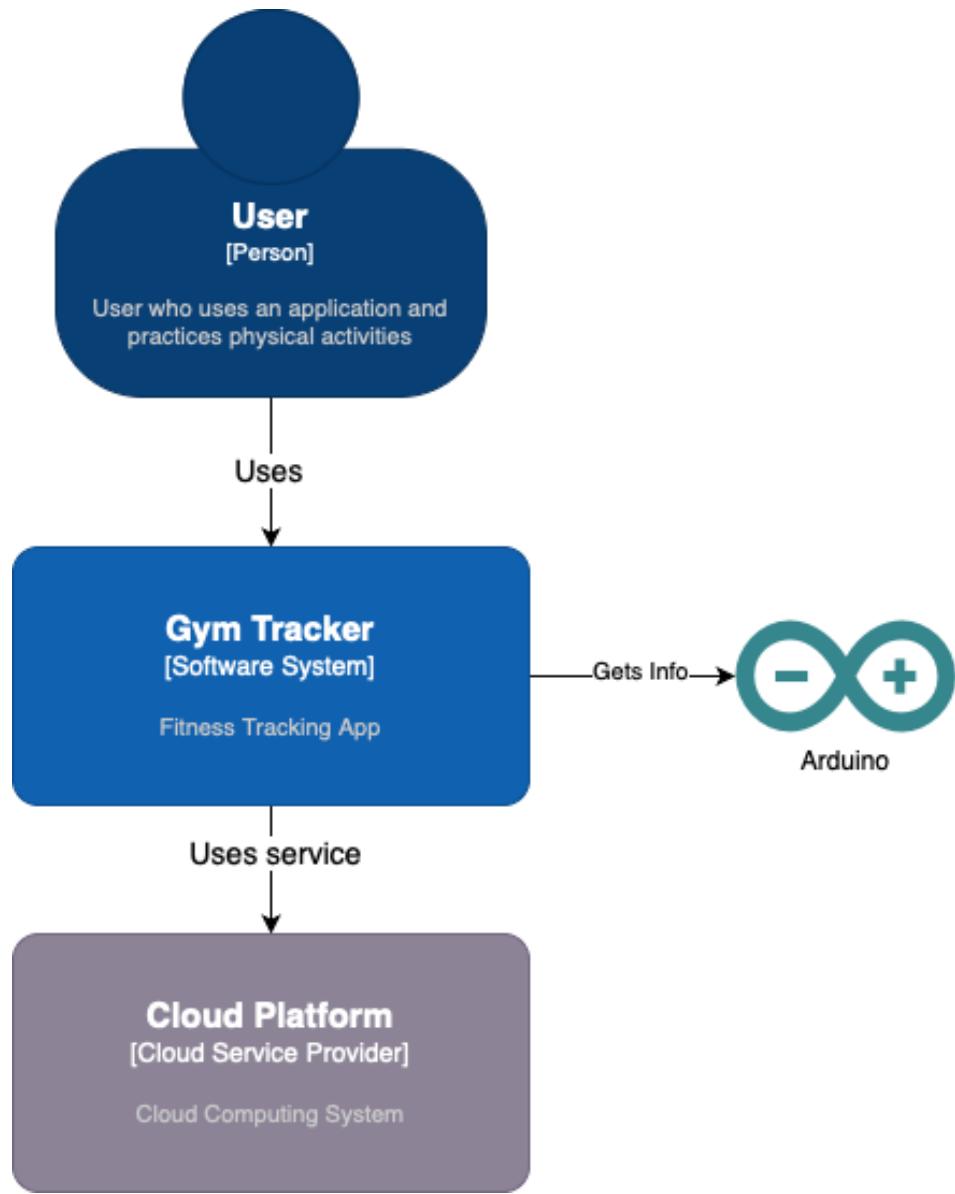


Figure 17: C4-Context Diagram

4.2 C2 - Container Diagram

The container Diagram, illustrates the relationships and dependencies between the various software elements in the architecture.

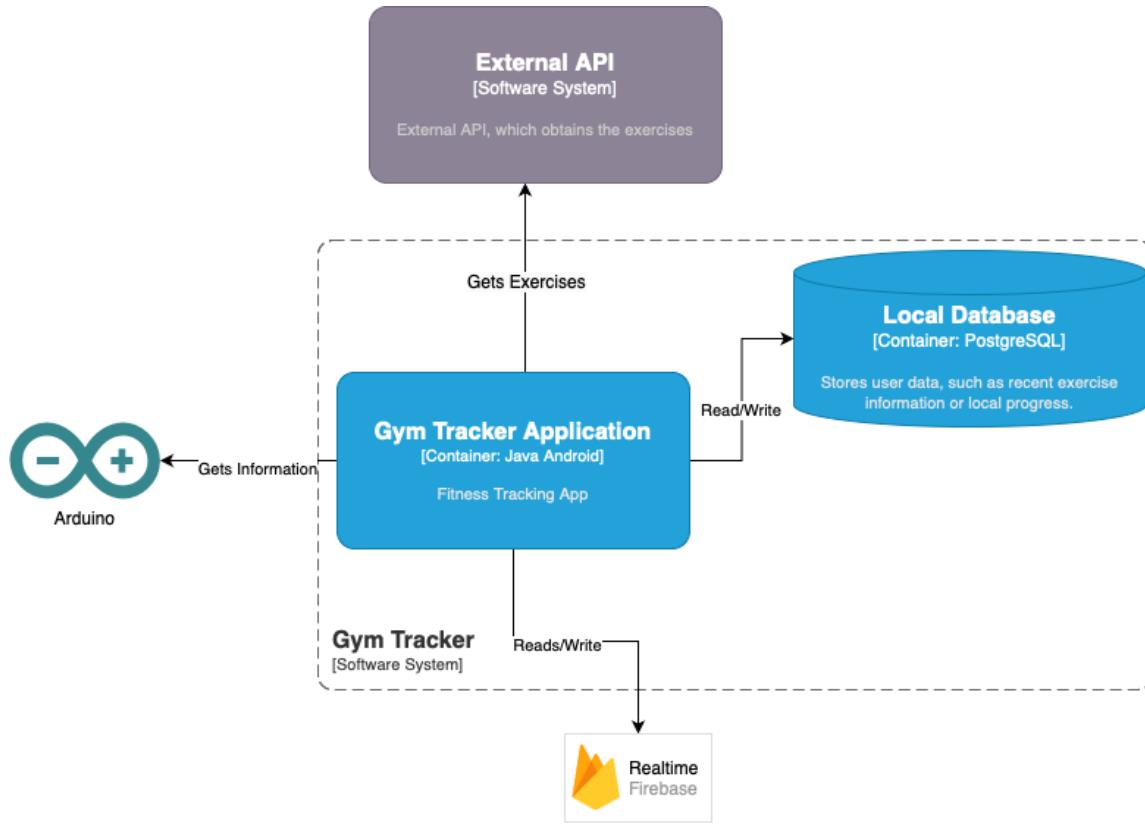


Figure 18: C4-Container Diagram

In this container diagram, the main container is presented, which is the application, which obtains the exercises from the external API, (Only when it does not have exercises in its local database). It has a database that stores your training plans, workouts performed, training results, etc., which is the main container that stores the information. The app saves all users' plans in firebase firestore, and has all users. A synchronization should always occur whenever possible. The application receives the pulse results from the Arduino. This will generate data randomly since it is not possible to create real data.

4.3 C3 - Component Diagram

The C3 Component Diagram illustrates the relationships and dependencies between the various software components in the architecture.

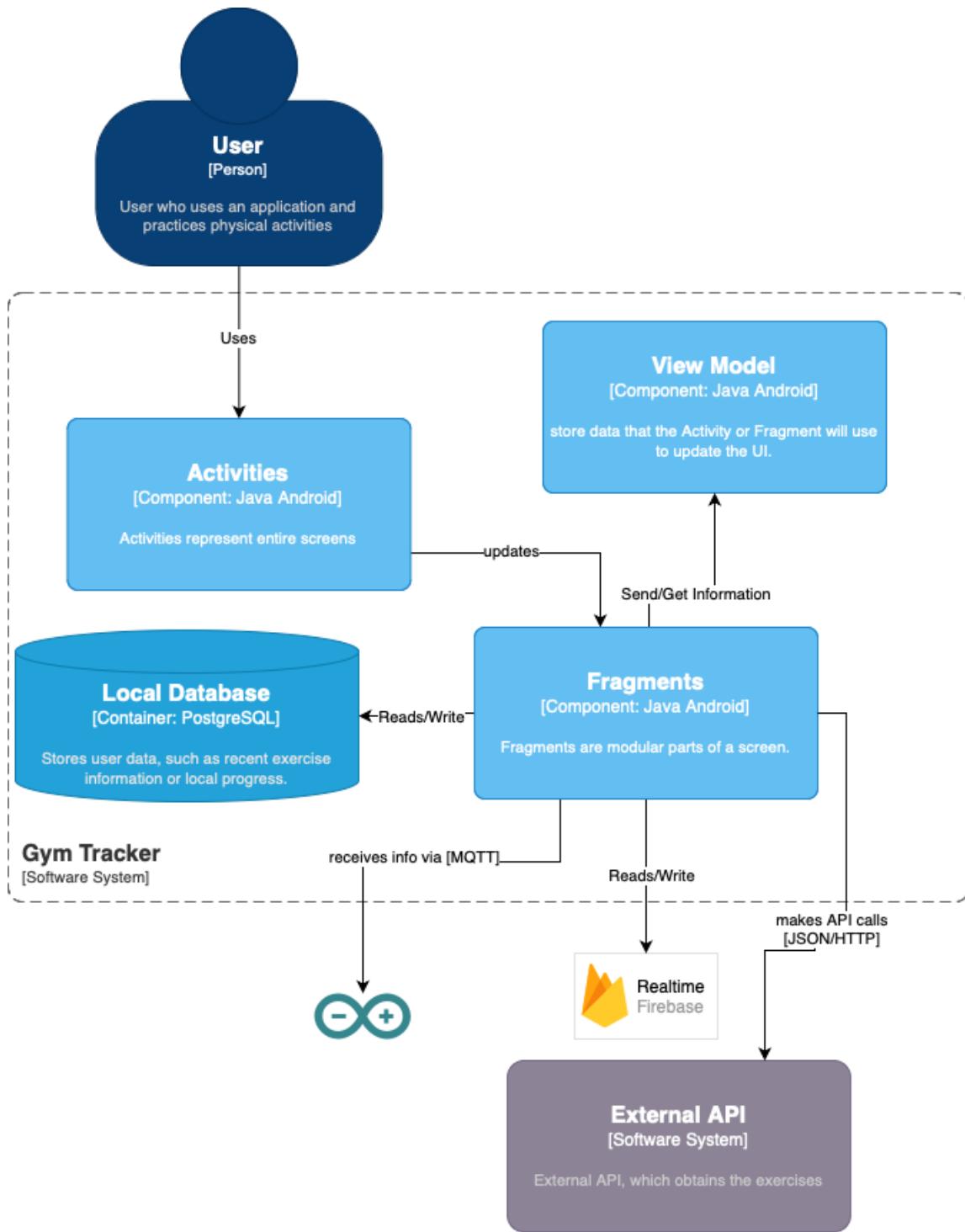


Figure 19: C4 - Component Diagram

In this diagram, the application user interacts with the activity. This is responsible for exchanging fragments depending on the user's response/requests. Fragments communicate information through the ViewModel, which is responsible for providing data to the fragments and storing the information that the fragments need. The fragments communicate with the local database to save user data such as information about exercises they have performed, and progress, for example. All existing information will be saved in Firebase in Realtime, to always obtain real and real-time results. One of the fragments will make API requests, using JSON and HTTP to obtain exercises from an API. Obtaining all possible exercises.

5 Data Bases

This project makes use of two different methods to save data, we use a *SQL* database locally in the user phone to allow offline usage and improve performance, and a *Firebase* database to allow portability in case the user changes phone and connects the user to the others in a social-media like way.

5.1 SQL Database Schema

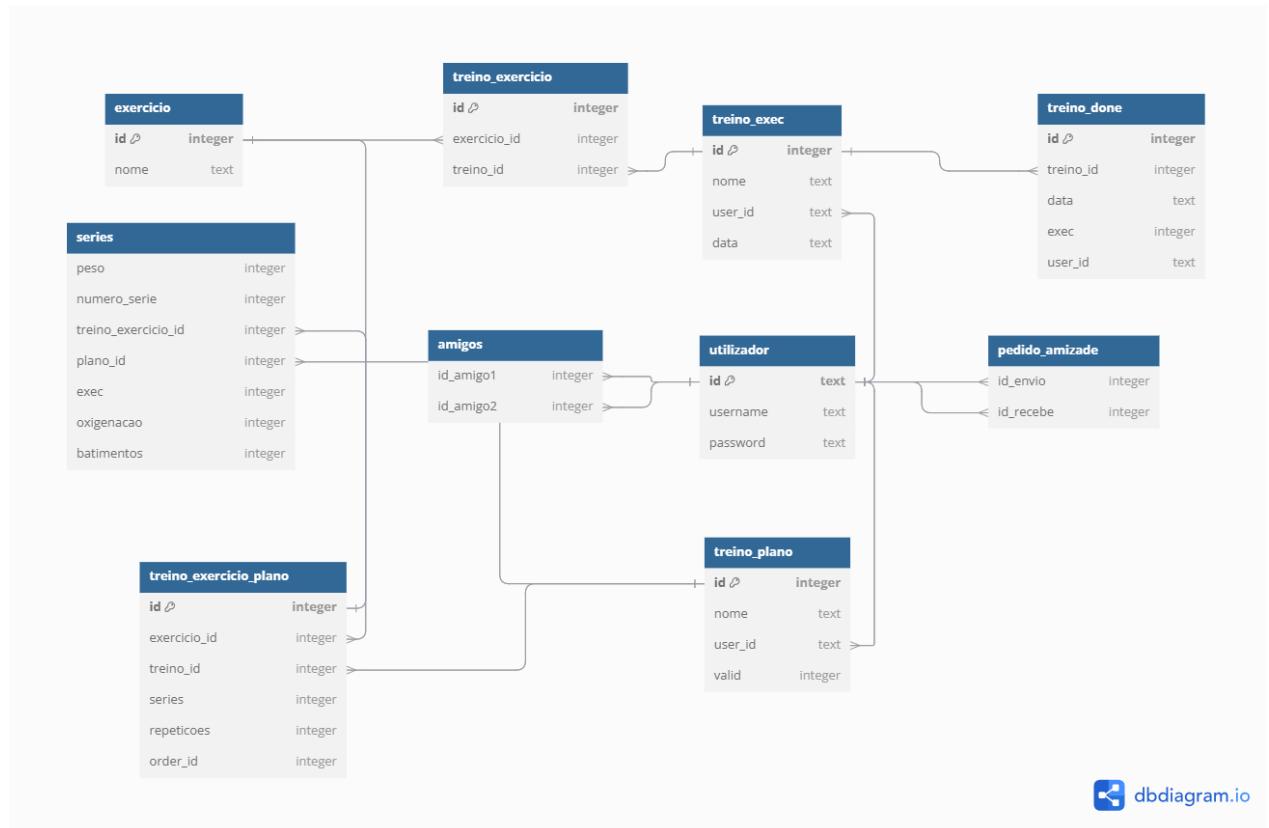


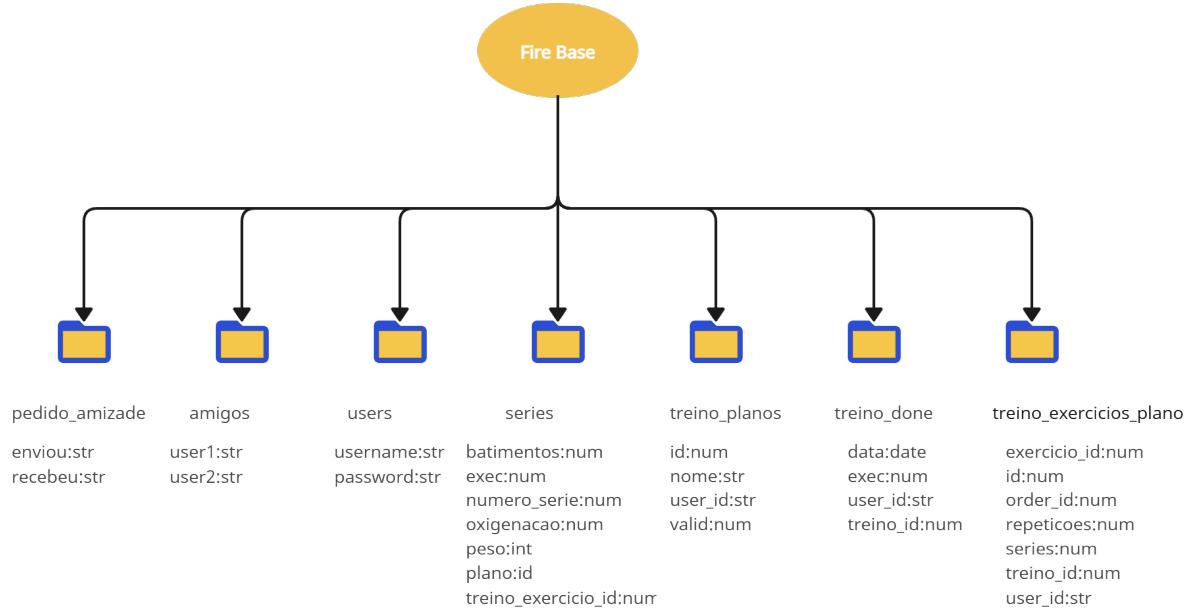
Figure 20: SQL Database Schema

The schema is centered on the user and connects him to his training routines and friends. We defined two types of training, *treino_plano* this is the training before it is performed, it has a onetomany relationship with *treino_exercício_plano* this one defines what exercise should be performed, how many times it should be performed, and at what point in the training it should be done. In *treino_done* we save the results of the practice of one training plan, it has a relation to *series* where we can see how many repetitions we were able to do on that training session and what the weight we lifted in each of those sessions. Another cornerstone of our database Schema is the

exercicio table, this one is a copy from the one in firebase and has all the exercises available in our application.

5.2 Firebase Schema

We defined seven different collections that match those in the SQL database. They allow to have a better availability and portability keeping copies of important data in the cloud.



Type something

Figure 21: FireStore Database Schema

6 User Manual

This section serves as a guide to the user interface and functionalities of the Gym Tracker app.

6.1 Login

This application starts with a login or registration page, where the user is required to enter their new credentials to register, or enter their already registered credentials to log in.

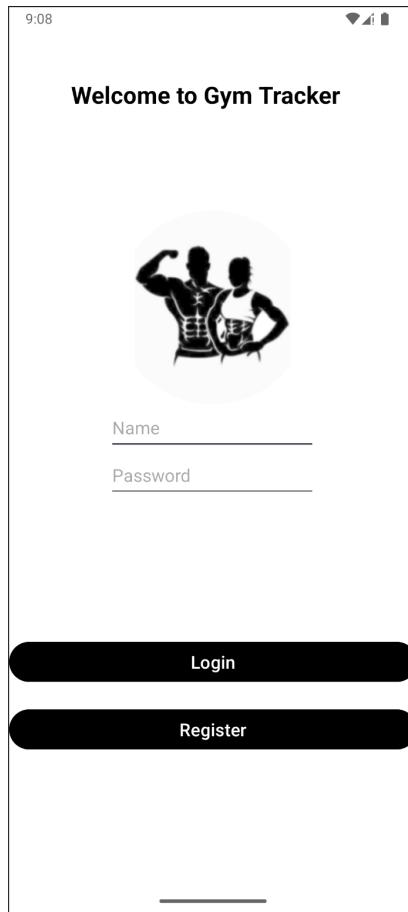


Figure 22: Login Page

6.2 Menu

After registering or logging in, you will go to the main menu of the application, where you will find 4 options in the navbar, and a list of workouts already performed will appear. You have an option to find friends, in the bottom left corner of the app, and you can see the details of a workout, such as the weight in each set you performed, just by clicking on the workout you performed.



Figure 23: Menu Page

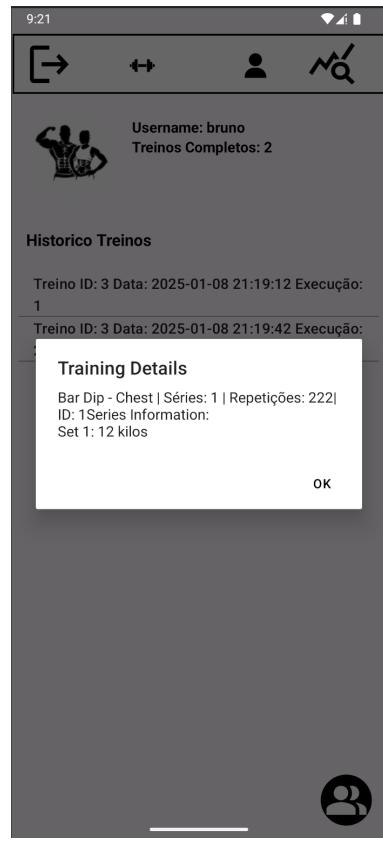


Figure 24: Train Done Detail

6.3 Listing/Create Plans

To view and create a plan, you must click on the 2nd option on the navbar, the "dumbbell", which will redirect you to the listing page, shown in Figure 25.

After that, you can create a plan by clicking on the button in the lower left corner, and a pop-up of the image will appear 26.

By clicking "OK", you will be taken to the workout details page 27, where you can start executing the plan, edit the plan (with new exercises, or different parameters), or even delete the plan itself.

If you are in the figure 25, and choose to just go to the details of a workout, just click on the workout you want.

If you have any questions/concerns on the training details page, click the "?" for directions and explanations of what you can do 28.

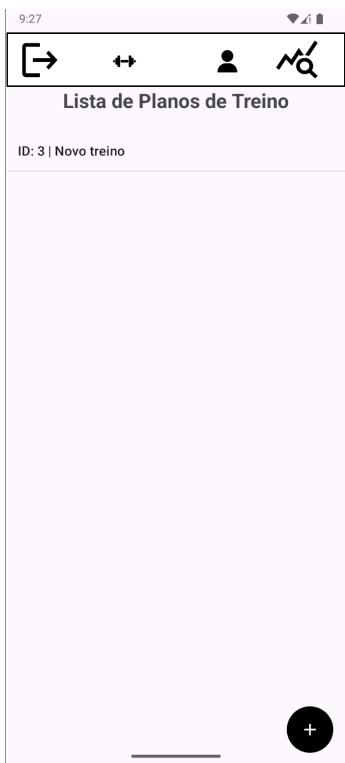


Figure 25: Plans List Page



Figure 26: Create Plan Page

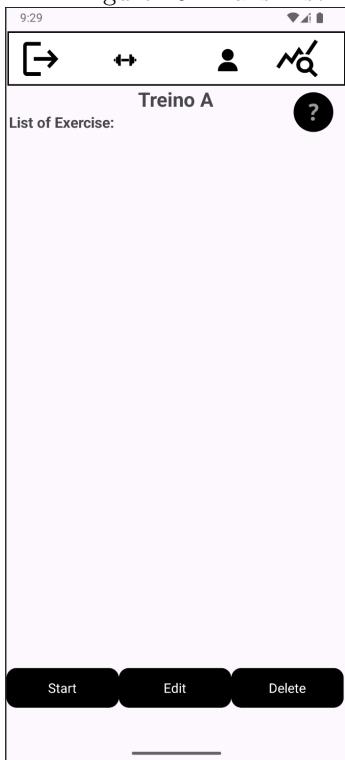


Figure 27: Details Plan Page

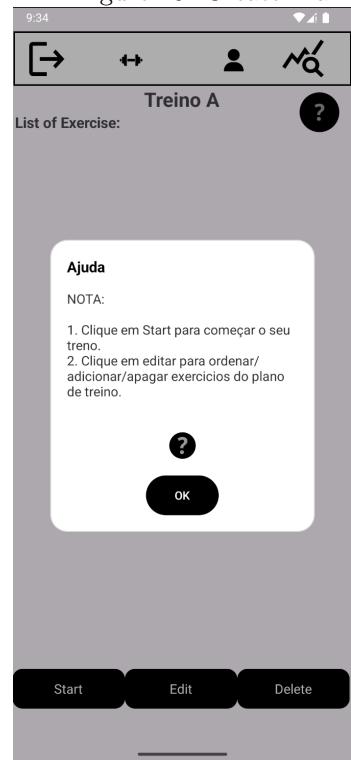


Figure 28: Helper Details Plan Page

6.4 Editing plans

To edit a training plan (create new exercises, delete exercises, order exercises), on the figure details page, Figure 27, click the "Edit" button, and you will go to the plan editing page, Figure 29. You have several options, below the navbar, you have an option to add a new exercise. This will bring up a pop-up with a spinner where you can select the exercise you want, and the details (sets and repetitions), Figure 30.

You can also take the exercises to order them by preference, to delete them from a plan just drag them to the sides³¹.

For visualization reasons, we do not show any images regarding the deletion of an exercise.

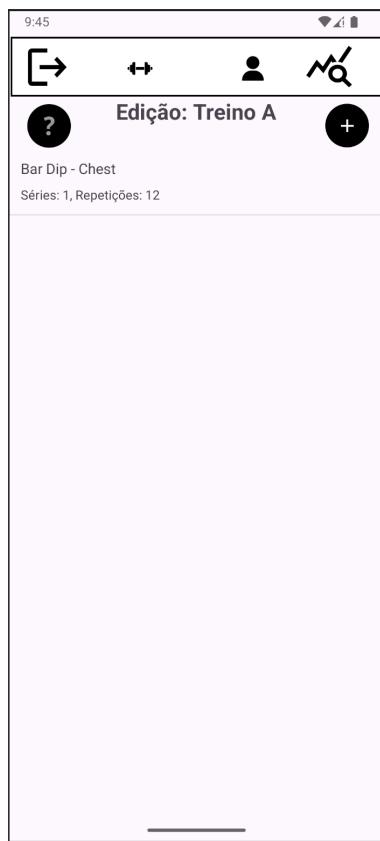


Figure 29: Editing a plan



Figure 30: Add an exercise

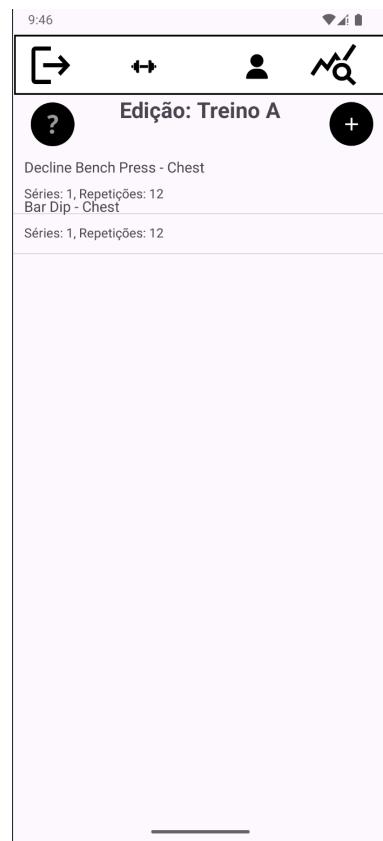


Figure 31: Order Exercise

6.5 Execute plans

In the listing page 25 by clicking in one of the plans you go to the details page, where all the exercises information in that plan are shown 32. To execute the selected plan, you must press the start button. After that a finish button will appear³³ but in order to actually finish the plan, you must, for each exercise and each serie, press on its name and input the weight you used 34. Before

pressing ok, you bust press the button on "Obter Rastreamento de Saude" to scan for pulse and oxygen levels 35. Now you can press ok and do the same for the other exercises. If you try to change page without finish the plan you will be prompt you you really want to do that, because if you do, your progress will be lost. 36. When you input all the information a check mark is visible in front of the exercise 37. When all the exercises are completed, you can press finish.

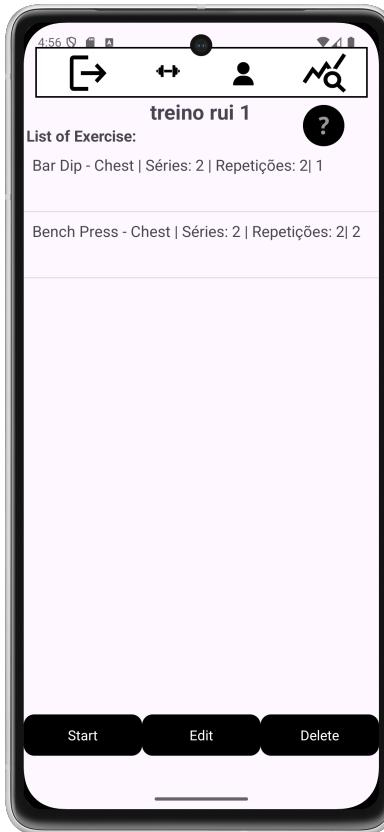


Figure 32: Exercises for a selected plan

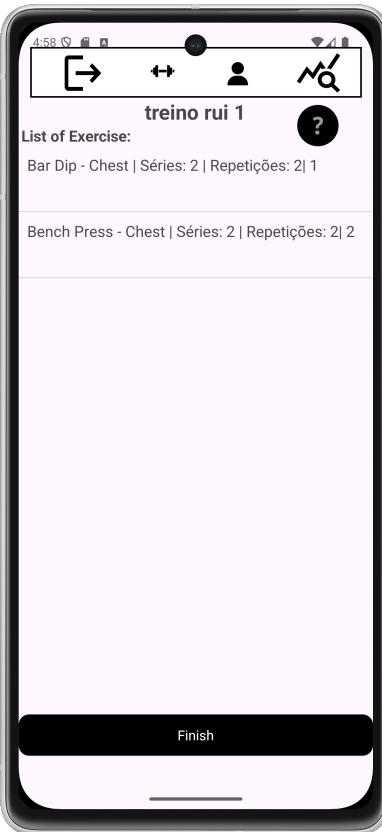


Figure 33: Finish a plan



Figure 34: Asking to input weight

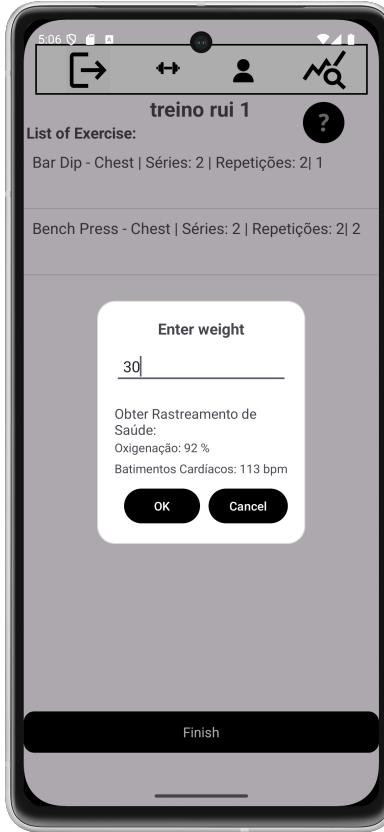


Figure 35: Pulse and Oxygen scan



Figure 36: Progress lost

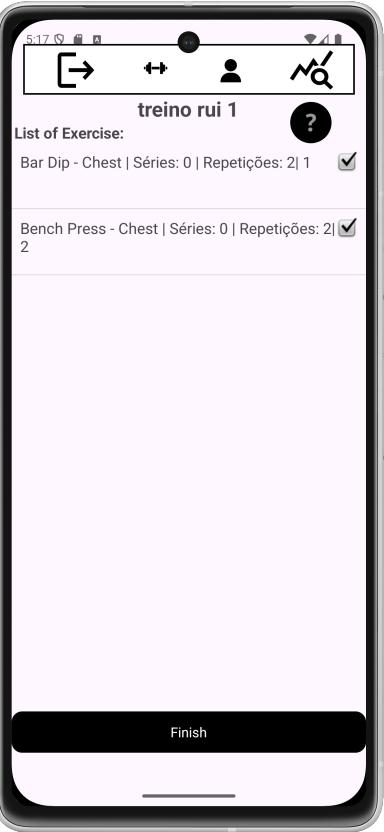


Figure 37: All exercises done

6.6 Graphic of results

It is possible to see a graph with your evolution per exercise and compare your results with other friends. To do that press the last option on the navbar and you go to a page to choose the exercise you want to see your evolution 38. After choosing the exercise you can see your evolution. The default value is the weight average, but you can also choose the maximum weigh lifted. As mencioned, you select a friend of yours to compare results.

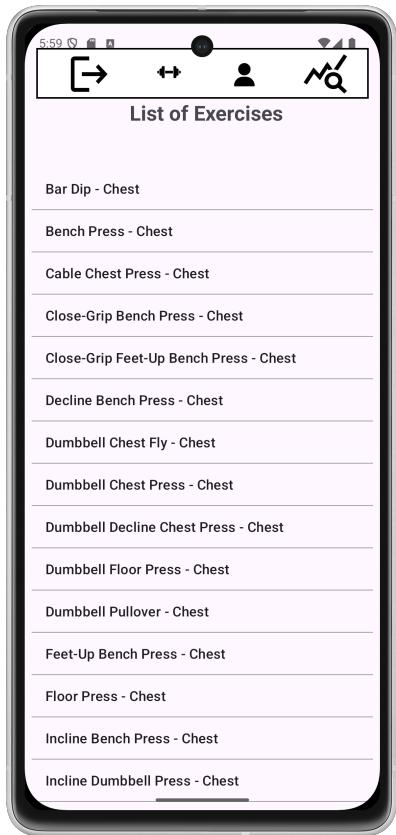


Figure 38: List of exercises

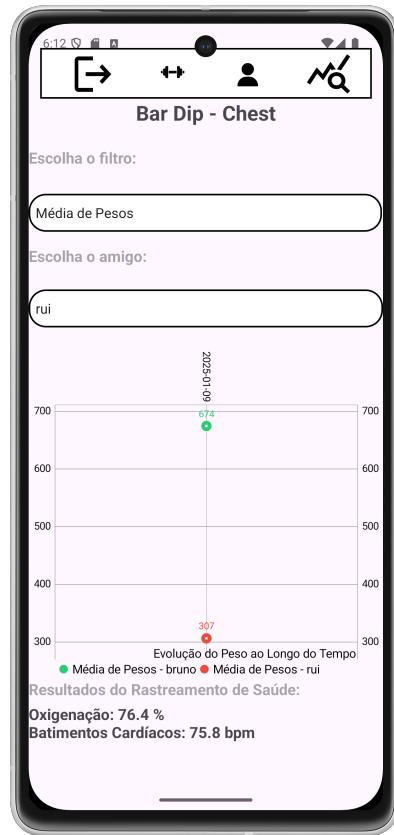


Figure 39: Compare with friend the average weight

6.7 Friends

Another functionality is the friends page 40. Due to the nature of this functionality, it is only available when the user has an internet connection, since the data of the other users is only kept in the *FireBase*, there is no copy of it in the local database. Here you can see your friends list in the "Friends" section, the users who sent you a friend request in the "Requests" section, and the others users of the app in the "Others" section. This fragment also makes use of a toolbar with a search feature to allow you to find specific users in any of the three sections. In the "Requests" section, you can accept or reject friend requests by pressing the respective buttons. In the "Others"

section, you can press the button "add" in front of a username to send a friend request to that user. When pressing the "Visit" button for a friend, it loads that friend's menu, where the number of trainings completed and the history of completed train plans of that user are shown. Every train plan on the user history has two buttons, the detail button lists all the exercises and repetitions of that training, the copy button makes a copy of that training to our collection of trainings. We decided to allow the user to only copy trains plans that the friend has completed, we believe that this restriction is useful because it makes sure that the friend has practiced at least once that train plan, on that same logic we show the number of times that the friend has executed that train plan.

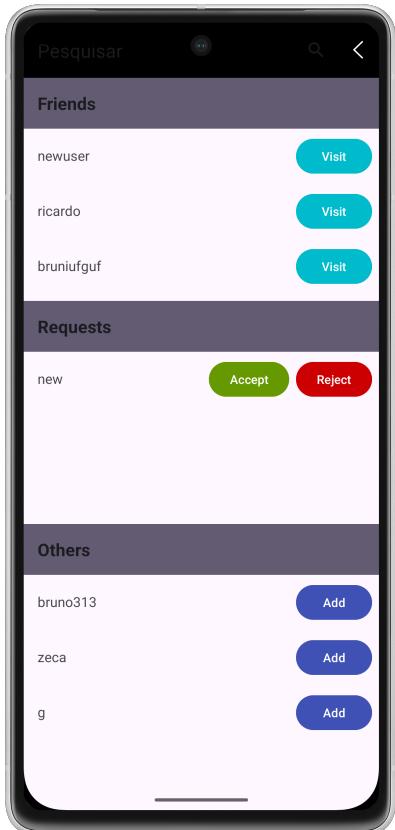


Figure 40: Friends page

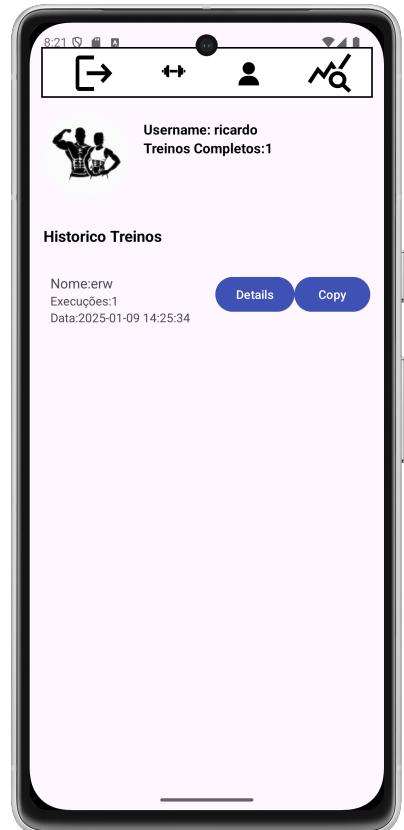


Figure 41: Friend Profile

7 Planning

This section describes the project planning process, as shown in 42. The work is divided into four main stages. It starts with discussing and selecting the final project idea, which sets the foundation for the next steps. The second stage focuses on gathering requirements, creating user stories, designing the architecture, and making wireframes to visualize how the application will look and work. The third stage is development, where the application is built step by step. The final stage is writing the project report, documenting everything that was done, and completing the project.

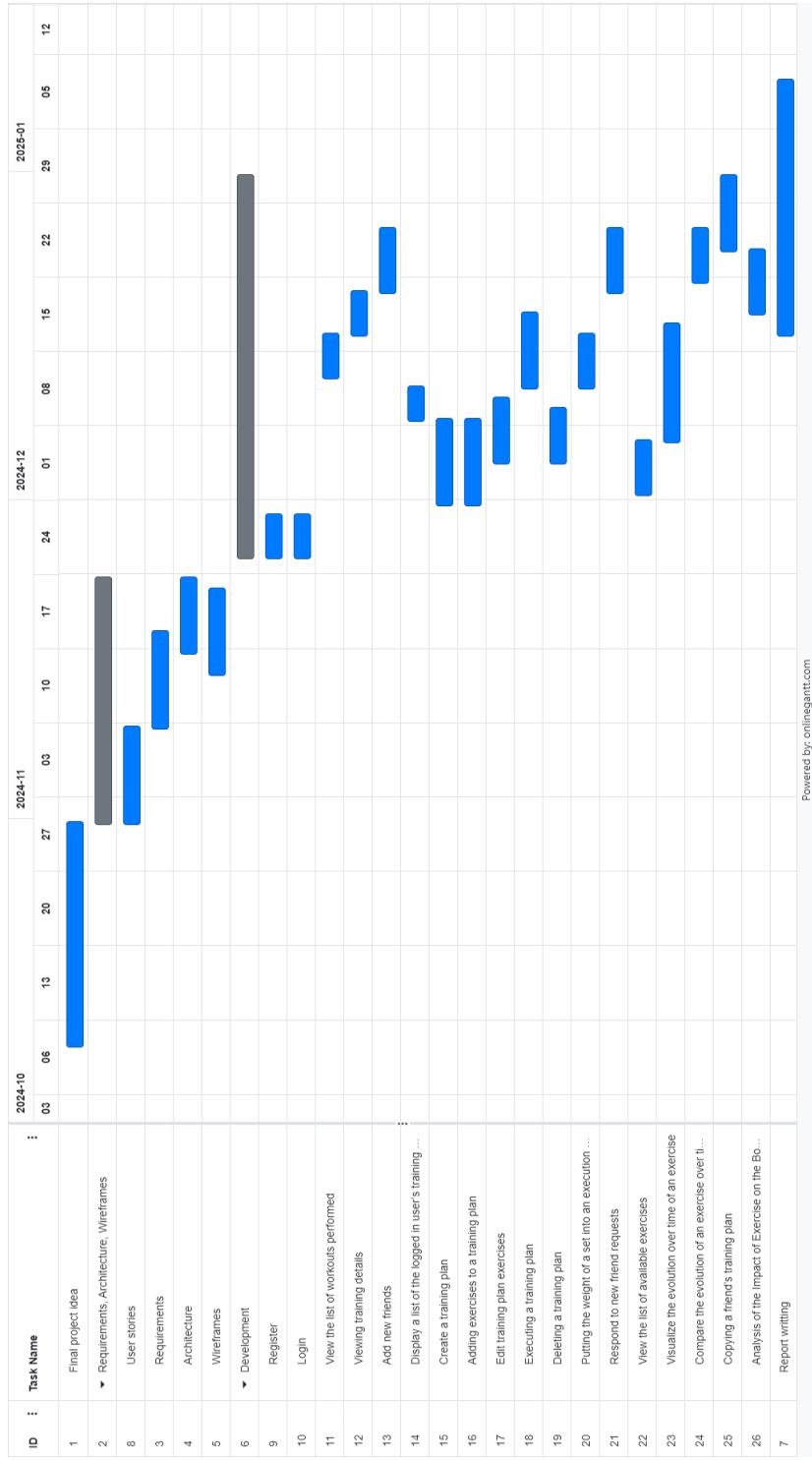


Figure 42: Project Planning

8 Work distribution

Since we were three members, we divided the work by three. Bruno did the firebase synchronization, training stats graphics, Arduino with integration, API fetching, and some listings. Rui did the main menu, the training execution, plan edition and deletion, and wireframes. Ricardo Guegan did the social part of the application, friend requests, users listing and sorting, see friends training plans, and the possibility of copying them also did the login/register and the databases Schemas.

9 Conclusion

During this project, we applied knowledge from the Mobile Computing class, such as working with Activities, Fragments, and Firebase's Firestore Database. This experience allowed us to strengthen our understanding of the concept in the classes and served as a base to go further in the world of mobile development.

With features like Arduino integration for health tracking and both local and cloud databases for flexibility, the app is designed to be reliable and user-friendly. Each team member contributed to different parts of the project equally, which helped us to build the application successfully and document the process.

There is plenty of potential for future updates to make it even better, such as enhancing the application with more detailed progress graphs, integration with additional devices for data collection, and refinements to the interface to make the user experience even more intuitive.

This project has been a great opportunity to develop our technical skills and identify potential for future growth in creating engaging mobile applications.

References

- [1] C4. C4 model | home. <https://c4model.com>.
- [2] Savio.io. Moscow feature prioritization, 2024. Accessed: 2024-01-02.