Implementation of the user-centered design methodology to develop a sports application

Juan Ángel Ávila Martínez

Final Degree Work
Bachelor's Degree in
Video Game Design and Development
Universitat Jaume I

June 13, 2022

Supervised by: José Vicente Martí Avilés

Acknowledgments

First of all, I would like to start by thanking the people and friends who have participated in the different tests of the application and without their participation the development of this would not have been possible. Especially to my friends Pau and Carrasco for all the encouragement he has offered me, to Jimmy for being so involved in the user tests of the project and to Sabina for all the advice and knowledge of her work.

I would also like to thank my family, especially my mother, who without their support I would not have come this far since she has helped me every day and has been very patient with me, as well as being the first to find bugs to correct them and the one who helped me every time android studio gave me problems.

I also would like to thank Sergio Barrachina Mir and José Vte. Martí Avilés for their inspiring LaTeX template for writing the Final Degree Work report, which I have used as a starting point in writing this report.
Without forgetting the teachers who have advised me throughout the development of the project, my tutor José Vicente Martí for the help in deciding the initial idea for my work, María Gracia Valdeolivas for her help in guiding me through the UCD and all the support she has given me in her tutorials, as well as encouraging me to continue with the project in the future and assisting me throughout this process, to Mercedes Marqués for helping me with the design of the database and to Juan Miguel Tovar for guiding me in the early phases of the project due to the problems caused by the Android Studio tool, which has not stopped giving me problems and failures since day one.

Finally, as a special mention to my cat, Mae, who has been a fundamental pillar of my mental health throughout this process.

Abstract

This document summarizes the entire development process of my Final Degree Work for the degree in Video Game Design and Development at Jaume I University. The main objective of this work is to develop an application that promotes physical activity by gamifying it. Using user-centered design and the Learning Mechanics - Game Mechanics (LM-GM) framework. As the project develops, it will be ow consistent this design framework is with the user-centered design (UCD) and if it is possible to create applications that satisfy the proposals for gamification of activities, which at first can be tedious. The project that has been chosen for this task has been an application to move from one point to another, whether walking, running or cycling, aimed at a wide audience that wants to start in the world of sports. Due to its mobility, it has been chosen to use Android Studio to develop a mobile application, at the end of this writing it will be possible to see how the application has evolved by submitting it to UCD and its changes are compared with the initial proposal made.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Work Motivation

In recent years, the world has experienced a pandemic generated by Covid-19, as well as the confinement that took place at the beginning of it, because of that, the mental and physical health of many people was affected, me among them, and It has left sequels that still last today.

Once this initial isolation ended, we had two pandemics, the one produced by Covid-19 and another silent one that led 25% of the population to suicide every day, especially affecting women and young people and more than 53 million new cases of depressive disorders.[33] All this, as consequences of fear, social disconnection and uncertain future. But not only did it affect mental levels, but also due to this isolation and fear, the physical health had a great decline.[30]

I also suffered from these effects, to the point that leaving my house, walking to the university and going up two flights to get to class left me breathless and taking up to five minutes to stabilize. Everything changed the moment I decided to buy a video game for the Switch, called Ring-fit[28], with it I was able to exercise at home and enter the world of calisthenics and lose weight, since going out for a run made me embarrassed, and it was thanks to this game that months later I joined the gym and continued doing these exercises in a more serious way. As a consequence of this, by improving my physical form, my mental state also improved, since when I felt stressed or overwhelmed, a little exercise helped me to release endorphins and relax.[1]

Later, during the development of the optional subject "Fundamentals for the design of educational games", we had to develop a serious game and we studied the LM-GM design framework, which seemed interesting to me and asking the teacher for advice, she recommended me to use also user centered design for my project. Therefore, I decided instead to make an application that anyone could use to start and feel what I experienced with the game that helped me both physically and mentally.

1.2 Objectives

The basic objectives for my final degree project are the following:

- Gamify an activity that can be tedious or boring at first.
- Use the Learning Mechanics - Game Mechanics framework to design the main mechanics of the application.
- Check how an application can change based on the User-centered Design.
- Make an application that is user friendly and not very complex to use for any public
- Motivate the user to continue such activity in the future and try to make it a gateway

1.3 Related subjects

The related subjects for my final degree project are the following:

- VJ1238 - Fundamentals for the design of educational games.
- VJ1220 - Databases.
- VJ1229 - Applications for mobile devices.
- VJ1224 - Software Engineering
- VJ1237 - Videogame Localisation

1.4 Expected Results

The expected results are both the functionality of the application and the use of the work methodologies that are implicit behind it. Regarding the application, at least two things are expected, that it works for a group of people within a wide age range and who want to start exercising and, on the other hand, that it has an attractive design that stimulates its continued use and that initially it is not look overwhelmed. On the other hand, regarding the methodologies, we want to check if the initial proposal of the mechanics using the LM-GM framework remains unchanged after being subjected to multiple interactions with the user, in the same way, to see if the initial design thought it also remains the same as the given proposal, otherwise we would see the utilities of the user-centered model and its possible implementations.

1.5 Working methodology

The development of this project is based on two work methodologies: user-centered design and the LM-GM framework.

1.5.1 User-centered design

"User-centered design (UCD) is an iterative design process in which designers focus on the users and their needs in each phase of the design process. In UCD, design teams involve users throughout the design process via a variety of research and design techniques, to create highly usable and accessible products for them."[24]

1.5.2 Learning Mechanics - Game Mechanics Framework

As mentioned above, the LM-GM framework [29] has been used to develop the mechanics of the application. This model is created specifically to be used in the implementation of serious games.

This framework introduces two lists of mechanic types:

- Game Mechanics: They are made to be played and enjoyed.
- Learning Mechanics: Allows the player to learn and gain any type of knowledge that is related to any subject.

1.6 Tools and Technologies

Due to the fact that different types of methodologies have been used, different tools have also been used for them.

1.6.1 LM-GM Framework

Creately[3]

Creately is a platform to make diagrams, schemes and mental maps in a visual way. The platform provides an infinite workspace to make schematics, planning, flowcharts, mind maps, organizational charts, Gantt charts, SWOTs, process diagrams. Among its main features we have: an extensive library of templates, real-time collaboration and an infinite workspace to make schemes and maps, file sharing, etc.

1.6.2 User-centered design

Google Forms[5]

Google Forms is a Google service that allows you to create, share and manage forms, for example to conduct online surveys and questionnaires. Once the form is ready, it can be shared via email, a link, or embedded on a web page and the responses collected and displayed.

Google Sheets[45]

Google Sheets is a spreadsheet program that is included as part of the free web-based Google Docs Editors suite offered by Google. The application supports Microsoft Excel file formats and allows users to create and edit files online while collaborating with other users in real time for easy data analysis.

Posit packet

To organize the tasks and changes that users wanted, a package of posits has been used. This package has been used to manage the tasks and sprints made during the development of the application.

1.6.3 UML

DIA[44]

DIA is a general-purpose computer application for diagramming. It is conceived in a modular way, with different packages of shapes for different needs. It can be used to draw different types of diagrams. Currently entity-relationship diagrams, UML diagrams, flowcharts, network diagrams, among others, are included. New shapes can be easily added by drawing them with an SVG subset and including them in an XML file.

Vertabelo[34]

Online tool for visual database design. It allows you to create a database model simply by drawing tables and references graphically. The app validates your model all the time and gives hints on how to improve it. At any time you can see a preview of the SQL that will be generated for a selected element, for this work the free trial for students has been used.

Miro[25]

Miro is a collaborative whiteboard tool that allows you to customize UML diagrams, in real time, according to the needs of the user and their audience. Miro offers multiple pre-built templates, with diagrams, mind maps, and workflows designed for strategy and planning. Integrates with multiple third-party tools like Dropbox, Google Suite, GitHub, etc.

1.6.4 Art Design

Illustrator[22]

Adobe Illustrator is a vector graphics editor that uses a drawing board or “artboard” for artistic creation (logos, icons, drawings, typography, and complex illustrations). It allows capturing the user’s creative vision with shapes, colors, effects and different fonts. Works on mobile and desktop

1.6.5 Programming

Android Studio Bumblebee (2021.1.1)

Android Studio is the official integrated development environment (IDE) for that platform. It has been specifically designed for Android development[46]. Android Studio offers even more features that increase your productivity when developing Android apps, such as: a flexible Gradle-based build system, a fast and feature-rich emulator, a unified environment where you can develop for all Android devices , integration with GitHub[11]

Android operating system[43]

Android is a mobile operating system based on the Linux kernel [4]. It was designed for touchscreen mobile devices such as smartphones, tablets, etc. The main components of the Android operating system:

- Linux kernel
- Android Runtime
- Libraries
- Application Framework
- Applications

Kotlin[7]

Kotlin is an open source static programming language that supports functional and object-oriented programming [5]. It provides syntax and concepts similar to those of other languages, such as C#, Java, and others. Google officially supports Kotlin for Android development, which means that the Android documentation and tools are designed to be compatible with Kotlin.

Dependency Injection with Hilt[9]

Broadly speaking, it could be said that dependency injection is a design pattern that allows us to supply the necessary instances in each of the classes that require it. Hilt is a dependency injection library for Android that allows you to reduce the repetitive work of manually injecting dependencies into a project. Hilt code and Dagger code can coexist in the same code base. However, in most cases, it's best to use Hilt to manage all Dagger usage on Android .

MVVM architecture

App architecture design is important to ensure that apps are robust, testable, and maintainable. The architecture of an app defines the limits between its parts and the responsibilities that each of them must have[18].
In the MVVM architecture (Model – View – ViewModel)[27]:

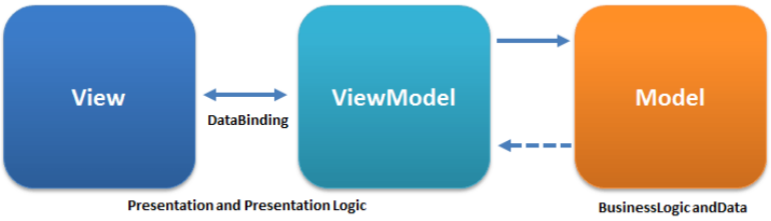


Figure 1.1: *MVVM architecture diagram*

- **Model**, represents the data layer and/or the business logic, it is totally decoupled from the view and only contains information, never actions to manipulate it.
- **View**, presents the information and is active, reacting to changes in the model. That is, the view only depends on the ViewModel to obtain data or modify it, and nobody has to tell it to update the data, since it is directly observing changes in the model.
- **ViewModel** is the intermediary between the model and the view and contains all the presentation logic.

Android Architectural Components[19]
These components released by the Android team make it easy to implement an MVVM pattern on Android.

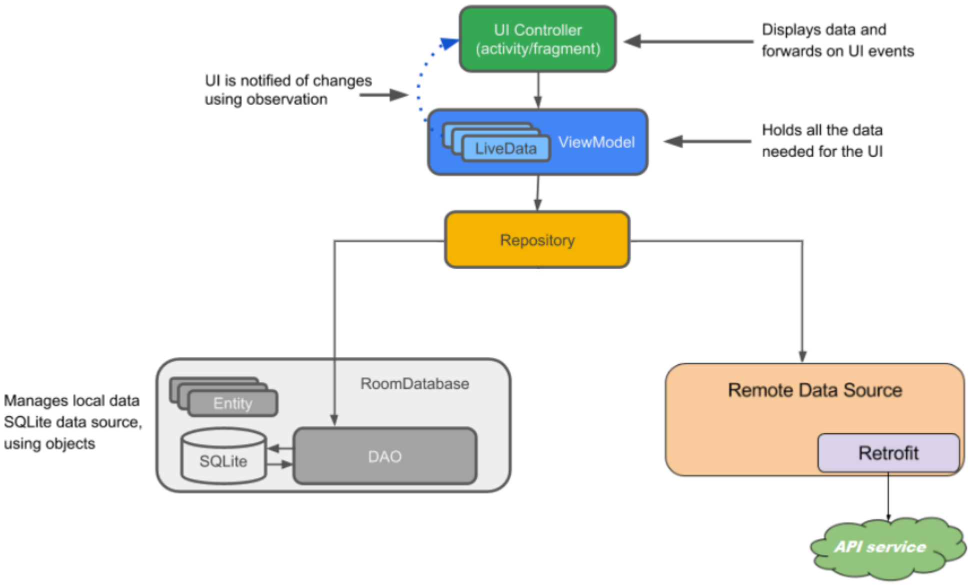

UI Controller (activity/fragment)
Displays data and forwards on UI events
UI is notified of changes using observation
LiveData
ViewModel
Holds all the data needed for the UI
Repository
RoomDatabase
Manages local data SQLite data source, using objects
Entity
SQLite
DAO
Remote Data Source
Retrofit
API service

Figure 1.2: *Architecture Components*

- **LiveData:** Objects that notify the view when there are changes in the database. They are aware of life cycles, so they help avoid crashes when an Activity stops, for example.

- **ViewModel:** Classes responsible for preparing and handling the data of an Activity or Fragment. It exposes the information through a LiveData observed from the view.

- **Room:** ORM (Object-Relational mapping) library, converts SQLite objects to Java/Kotlin objects automatically. Returns LiveData objects to watch for changes to the database.

- **Retrofit** is a library for Android and Java compatible with Kotlin to make network calls, obtain the result and automatically "parse" it to its object, this makes it much easier to make requests to an API and process the response

Along with these architectural components, the repository is also used, which is responsible for abstracting the data sources from the rest of the application. Implements logic to decide whether to retrieve data from a network or use cached results from a local database.

Room Database[13]

Room is an ORM (Object-Relational mapping) that allows you to work easily with SQL databases. The three main components of Room:

- The **database class** that contains the database and serves as the primary access point for the underlying connection to the app's persistent data.
- The **data entities** that represent tables in the app's database.
- **Data access objects (DAOs)** that provide methods that the app can use to query, update, insert, and delete data in the database.

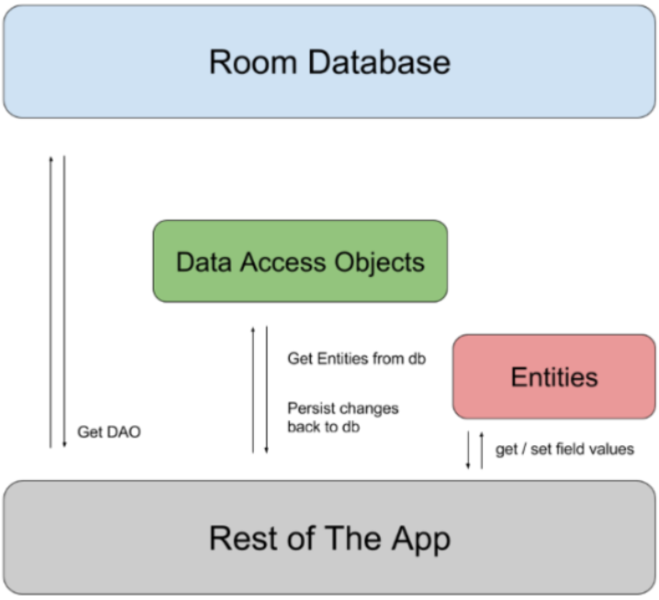


Figure 1.3: *Room library architecture diagram*

1.7 Accessibility

Because the app has been developed from the beginning following the specifications, all types of users have been taken into account. A simple and clear design has been chosen from the beginning so that those with vision problems, such as the elderly, can see all the important texts. In addition, due to the L10N service that comes by default in Android Studio, the application has been translated into Spanish and in the future it can be easily extended to any language thanks to this service. The application has been optimized for TalkBack[20], the android accessibility service, this consists of a screen reader for people with reduced vision or the blind, due to this, if you activate this android functionality, the entire application is prepared to be narrated . As a result of this, the application has been subjected to a Google accessibility test, passing it without any problem.

1.8 Novelty

This project is an opportunity to verify that gamifying an activity can make it easier to understand, entertaining and enjoyable, both in the field of physical activities, such as this project, and activities focused on learning using elements as simple as customization, information, achievements, game currency, among others. You also want to test how every game or application, after being passed through the different iterations of the user-centered design process, always manages to change and evolve for the better and even being expensive, many more projects would have to be subjected to this methodology.

Chapter 2

Planning and resources evaluation

Contents

2.1 Planning

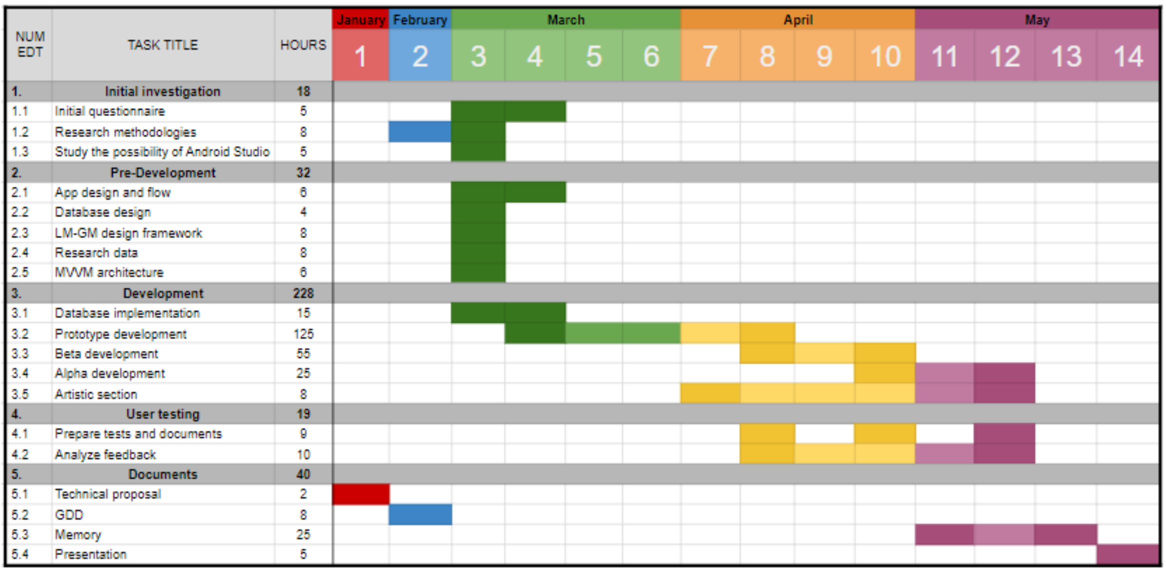
As mentioned above, user-centered design takes a long time due to waiting for data and conclusions given by those who test the application, for which a parallel project has been developed. While waiting for the feedback, other tasks have been carried out in order to fulfill the approximate 300h hours that this project is. In general, the project has been divided into five blocks.

The **initial research phase** was used to decide what physical activity was gamified, for this, a questionnaire was carried out that was distributed massively to know information from people and how to approach this application. This phase was also used to study design frameworks, user-centered design and other methodologies, as well as the possibilities of what the use of Android Studio could contribute against the widely used Unity game engine. The **Pre-Development phase** includes the time needed to develop the application flow, the sketch of the screens and the initial design of the database, as well as the configuration of the project and remember the basic concepts of the Android Studio program. In this phase, the design of the mechanics was also implemented using the LM-GM framework, as well as the research and reading of other projects and blogs to obtain the sports data that was used in the achievements section of the application.

The **Development phase** is divided into several sections, among which are the entire interface design, the implementation of the database and the MVVM architecture, the artistic section and the application programming itself. This phase was broken down into three tests, the prototype, the beta and the alpha, which is where the project currently stands.

The **User Testing phase** is carried out at the same time as the Development phase in order to optimize the time invested in the project, this phase includes from the time invested in writing the tests and the patch notes for the group of experts who tested the application, as well as the time they were given to give feedback and also the search for solutions for the feedback received.

Finally, the **Documentation phase**, where the tasks of writing the technical proposal, the GDD document, the report and the presentation itself are found, as well as organizing the user feedback data.



NUM EDT	TASK TITLE	HOURS	January	February	March				April				May			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.	**Initial investigation**	**18**														
1.1	Initial questionnaire	5			🟩	🟩										
1.2	Research methodologies	8		🟦	🟩											
1.3	Study the possibility of Android Studio	5			🟩											
2.	**Pre-Development**	**32**														
2.1	App design and flow	6			🟩	🟩										
2.2	Database design	4			🟩											
2.3	LM-GM design framework	8			🟩											
2.4	Research data	8			🟩											
2.5	MVVM architecture	6			🟩											
3.	**Development**	**228**														
3.1	Database implementation	15			🟩	🟩										
3.2	Prototype development	125				🟩	🟩	🟩	🟨	🟨						
3.3	Beta development	55								🟨	🟨	🟨				
3.4	Alpha development	25										🟨	🟪	🟪		
3.5	Artistic section	8							🟨	🟨	🟨	🟨	🟪	🟪		
4.	**User testing**	**19**														
4.1	Prepare tests and documents	9								🟨		🟨		🟪		
4.2	Analyze feedback	10								🟨	🟨	🟨	🟪	🟪		
5.	**Documents**	**40**														
5.1	Technical proposal	2	🟥													
5.2	GDD	8		🟦												
5.3	Memory	25											🟪	🟪	🟪	
5.4	Presentation	5														🟪

Figure 2.1: *Gantt chart of task development*

2.2 Economic cost

The app that has been developed in the work is an application classified as native (developed in Android with Kotlin language).

Native apps are developed for a specific platform or operating system. For this reason, this type of application offers higher quality, features and performance, but at a higher cost. These types of applications require greater involvement in their design and development, since they need a base and solid designs that can be expanded with a view to the future.

Among the variables that affect when creating an app we have:

- **Programming hour cost.** This cost in Spain has an average of €27.50 per hour.[47]
- **Graphic design for the app:** The average graphic designer salary in Spain is €20,091 per year or €10.30 per hour.[39]
- **Access to app data.** The cost of this point is €0, since the app does not need to connect to database servers. Storage is done in the local ROOM database.
- **Google Maps API.** The Google Maps API necessary to display maps in the app is paid and therefore requires billing data to be able to obtain said API.
  Before it was free, but with the inclusion of other Cloud services it has become paid[35]. Initially they give you $200 of free credit per month, which is supposed to be enough as this is equivalent to 28,500 map loads, but if the app is popular Google will start charging for its use[26].
- **Additional costs**. Within this section we can consider expenses such as investment in marketing or the cost of uploading the application to the Google Play Store.
  Publishing an app on Google Play has no cost per se. The only thing you have to pay is the Google Play developer registration fee, which is $25[31] (one-time payment). Once that money is paid, it can be published indefinitely at no additional cost, whether they are free or paid apps.
  If the app is free, it can be kept published on the Play Store forever, without having to pay maintenance costs or the like. This also means that all the income you receive from the advertising that you have placed in the app through AdMob (which is usual), will be for you. Google Play doesn't cut any corners nor will it take a cut of the money here. But AdMob keeps a significant percentage of advertising, which is around 30If the app is paid, it must be taken into account that Google will keep 15% (from January 2022) of the cost of the app. This also applies to in-app purchases.

The app has been developed in Android Studio Bumblebee 2021, and the recommended minimum requirements to run this software are 8 GB of RAM; plus 1 GB for the Android emulator. 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image) Minimum screen resolution of 1280 x 800. [2]
A quote has been obtained from the Hp Store [36], for a laptop (HP 15s-fq4025ns laptop) with the following specifications:

- **Operating system:** Windows 11 Home.
- **Processor:** Intel® Core™ i7-1195G7 (up to 5.0 GHz with Intel® Turbo Boost Technology, 12 MB L3 cache, 4 cores, 8 threads) Processor family: 11th Generation Intel® Core™ i7 processor.
- **Memory:** 16 GB (2 x 8 GB) DDR4-3200 MHz RAM Transfer speeds up to 3200 MT/s.
- **Data Storage:** 512GB PCIe® NVMe™ SSD.
- **Display:** 15.6" (39.6 cm) diagonal FHD, micro-edge bezel, anti-glare, 250 nits, 45% NTSC (1920 x 1080).

Collecting the above data we can summarize the economic cost of the app in the following table.

	Hours	Cost	Total(€)
App Development	220	27,50	6.050
Graphic design	17	10,30	175,1
Publish on Google Play			24
HP 15s-fq4025ns Laptop			749
			6998,1

Table 2.1: *App development cost*

Chapter 3

Analysis and Design

Contents

3.1 Analysis objectives

There are many day-to-day tasks that can be heavy for many people since they do not bring anything new. The objective of this analysis is:

- Find out what physical activity people do the most and its frequency before, during, and after the pandemic.
- Know the reasons why users exercise or stop doing so.
- Know what people are looking for in a sports mobile application and what other references have.
- Which tools would be the most optimal for creating this application.

3.2 Designing the analysis

To answer these questions, people of all age groups and genders have been included in the study. By submitting as many people as possible, better results can be obtained, since the conclusions reached will be more accurate. Among all the possible methods, it has been decided to choose those that can be done remotely, since by needing a large volume of samples it could take a long time and in this way data collection can be optimized.

- **Questionnaires.** This method allows you to get specific information and can be filled in at any time. These questionnaires were distributed massively to collect as much information as possible.
- **Interviews.** They were addressed to teachers during tutorial hours and participants to find out what opinions they had about the future implementation of the project and what would be the best way to implement it.
- **External resources.** Both to design the questionnaires and to expand the information given by them, external links and bibliography were used to know what other conclusions had been obtained in other works with the same topic.

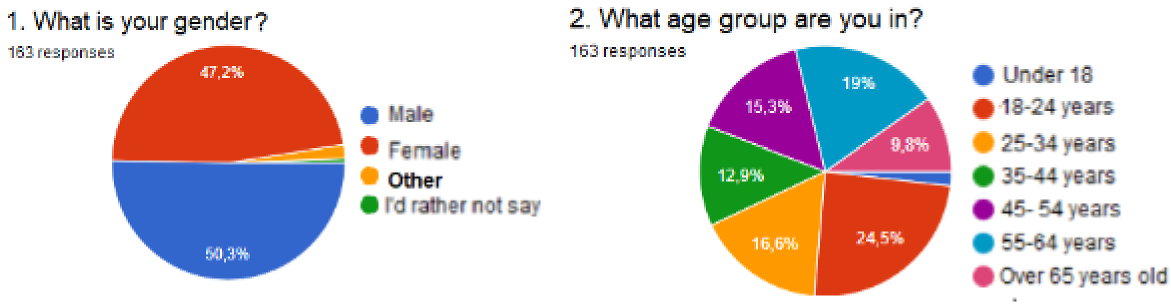
3.3 Analysis results

Thanks to external resources and interviews, it became clear that the most recommended tool for the project to end up having the most optimal development is Android Studio, since it has more advantages over the Unity[38] game engine. Among these advantages are:

- **Hardware/sensor capabilities:** The most basic sensors are in Unity as well, but in case you want to use something a little weirder, Unity may not have an easy or no way to implement it.
- **Android-like UI:** Unity does not have native android UI elements integrated, so you would have to create these yourself, thus making the project unnecessarily complicated. An example would be creating a RecyclerView, as there is no easy way to create it and it would be extremely slow.
- **Partial Rendering:** Android Studio, unlike Unity, allows partial rendering of the screens, so when there is a change in a part of it, it does not reload it completely, thus reducing the workload and making the application usable on low-end phones.
- **Accessibility:** Unity does not support screen readers on Smartphones due to how its UI is rendered, so all accessibility options given by the phone will not apply.

On the other hand, with external resources, not many projects were found that apply the LM-GM framework to sports applications, even so, many studies were found that used user-centered design, so it is not impossible to conclude if the LM -GM framework offers enough flexibility for mobile applications based on this research topic.[40]

Thanks to the survey carried out, it has been possible to identify the problems of the users with the applications, as well as to know their sports habits and how they maintain them. This survey was sent and answered by 163 people, who had a varied age range, thus giving the sample as valid.



(a) Gender chart

(b) Age chart

Figure 3.1: *Gender and age graphs of the sports survey*

id	Problems detected	id test
P1	Many people of many age ranges start exercising and find themselves with two great entry barriers, the first, they don't know how to start and the second, all the applications they find are very complex.	E1, E2
P2	Few apps on the market do have accessibility options.	E20
P3	Many users end up getting bored with the applications shortly after starting because they are not motivated enough.	E20,E13
P4	Users think that sports apps are impersonal and are not designed for the average user.	E11,E20
P5	Users want an app that can be adjusted to their schedules.	E19,E15

Table 3.1: *Problems detected in the initial questionnaire*

id	Initial conclusions	id test
C1	The sport that people do the most is running, followed by those such as walking or hiking.	E6-E8
C2	With the arrival of the pandemic, many people stopped exercising.	E3-E8
C3	People began to exercise at an early age and to solve future physical and mental problems	E10
C4	Users like to keep track of data in a simple and entertaining way	E19
C5	Users want to be able to share their data with other people	E19
C6	Half of the people who use a sports application end up using features that they were not initially looking for if they are well implemented.	E21

Table 3.2: *Initial conclusions extracted from the initial questionnaire*

3.3.1 First Design Decisions

From the conclusions drawn from the questionnaire, the following design decisions were made.

id	Problem/ conclusion Related	Design decision.
D1	P5, C1, C2	The project will consist of a running application. Although the app is designed for running, the user can also use it for walking, cycling or anything else that involves moving from one point to another.
D2	C4	The app will keep a record of the user, which they can review whenever he wants and edit it as they pleases.
D3	P1, P2, C3	The app will be accessible to a large group of people of different ages and with vision problems.
D4	P3, C4, C6	The app will have a reward system to make it more entertaining. These include an achievement window, in-game currency, and experience among others.
D5	P4, C6	The user will be able to customize the application at will to adapt it to their tastes.
D6	C5	The user will be able to share their progress and data with other users and other people through social networks.

Table 3.3: *First Design Decisions*

As the design proposal suggests, the app will have a way to bring data into an easily intuitive app design for people of any age, with a few extra features, that will hold users' attention and curiosity long enough that they don't they get bored Three of the main objectives of this project are to gamify an activity that at first can be tedious while being user friendly and, on the other hand, to check how a project can change by subjecting it to user-centered design. Applying this design in the app can achieve these objectives.

3.3.2 User-centered design adapted to this project

To adapt the project to this work methodology, it has been decided to separate the project into several versions, each of these versions will be tested by a group of users chosen beforehand so that they iteratively test each version of the project, which will always consist of the previous version with more added functionalities. The development process is separated into three fundamental stages

1. First stage: Predevelopment.

    - **Design:** Based on the results of the analysis and the initial decisions, work will begin on the design using the LM-GM framework, as well as the design of the application database and the flow of screens.

- **Testing:** Through interviews and tutorials, it will be verified if the design choices initiated are good or if, on the other hand, they have to be revised before the next stage.

2. Second stage: Development of the prototype.

   - **Implementation:** Based on the results of the previous stage, if there are no faults, the designs will be implemented to create a prototype of the application.
   - **Basic prototype:** A prototype will be developed with the essential and minimum functionalities for the user to test.
   - **Testing:** After the completion of this prototype, users will have a period of one week to test it and then, through a questionnaire, give their feedback.

3. Third stage: Version development.

   - **Implementation of new functionalities:** Compared to the previous version, a new functionality will be implemented based on those designed in the first stage.
   - **Data review and feedback:** The data from the previous version will be reviewed and design changes will be implemented based on the feedback given.
   - **Testing:** Users will once again have a trial period to test the new version and give their feedback on it again.

This last stage is the most important since it is the one that will be repeated iteratively until the application finishes developing and a final product is reached that meets the established objectives.

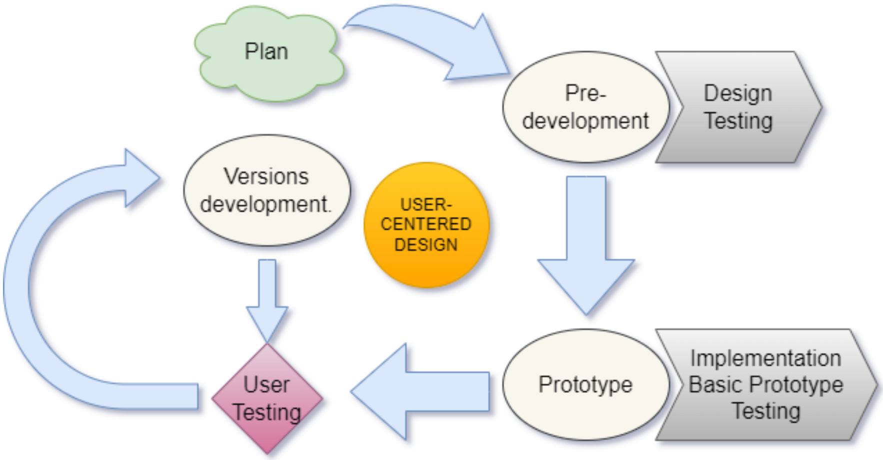


Figure 3.2: *UCD diagram adapted to this project*

3.4 Analysis of requirements

From the results obtained in the analysis, the initial requirements can be defined, thanks to this information it will be possible to develop, in later sections, the mechanics using the LM-GM framework, as well as the design of the database. These requirements will be separated into two types, non-functional and functional requirements that will be shown in the following tables.[42][41]

Non-functional requirements	
id	**Description**
R0	The user will have to register with a nickname in the application.
R1	Each route will contain information such as time, distance, date and image of itself.
R2	The app will connect with google maps to trace the route
R3	The app will have a simple design so that everyone knows how to handle it
R4	The user will have a level and an initial amount of coins.
R5	The app will have achievements, which will consist of challenges that when completed will give you a curious fact.

Table 3.4: *Nonfunctional requirements*

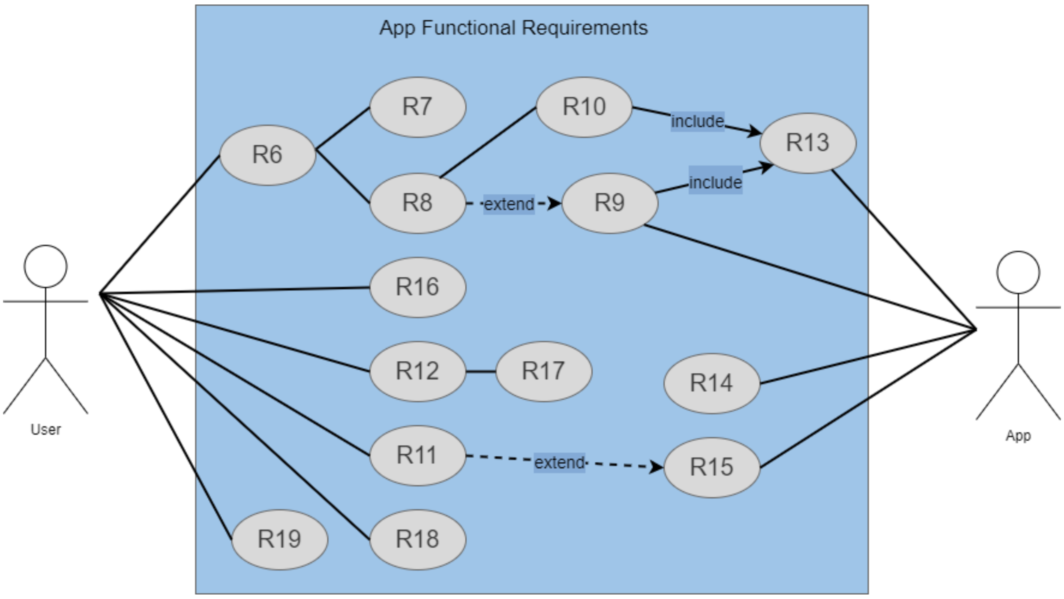


Figure 3.3: *Functional requirements use case diagram*

Functional requirements					
id	Action	Actor	Description	Pre	Post
R6	Start an activity	User	The user can choose when to start an activity	R0	R7 R8
R7	Pause/Resume an activity	User	The user can pause or resume the activity at any time	R6	
R8	Finish an activity	User	The user can choose when to finish the activity	R6	R9 R10
R9	Fill in mandatory data of the activity	App	When the user finishes an activity, the application will automatically fill in the mandatory fields.	R8	R13
R10	Fill in optional data of the activity	User	The user can decide whether to fill in the optional fields of the route.	R8	R13
R11	Add task	User	The user can add one or multiple tasks to try to complete.	R0	R15
R12	Add POI's	User	The user can add one or multiple POIs to view them later.	R0	R17
R13	Add routes	App	Once the activity is finished, it will be saved in the route log.	R0	
R14	Check that achievement is unlocked	App	The app will take into account if the user has unlocked an achievement and it will be displayed on the corresponding screen.	R0	
R15	Check that task is completed	App	The app will check if a task that has been added has been fulfilled or not	R0	
R16	Delete task, routes or POIs	User	The user will be able to delete tasks, POIs and routes if they are not to his liking.	R0	
R17	Edit POI data	User	Once the POI has been added, the user will be able to edit it in its corresponding window.	R0	
R18	Add Friends	User	User will be able to add friends given friend id.	R0	
R19	Share data	User	The user will be able to share data of their routes or POIs through social networks.	R0	

Table 3.5: *Functional Requirements*

Chapter 4

First stage - Predevelopment

Contents

4.1 Objectives of this stage

During this first stage, the results obtained from the analysis will be used to create a design that can be adapted to the needs that users are looking for and that takes into account the requirements established in the previous part. These design decisions have been used in the development process and to guide what Bloom's Taxonomy thinking skills are needed. Based on that, the LM-GM framework will be put into practice for the design of the app's mechanics.´ Once these learning mechanics have been extracted and thus a playable loop, it is intended to make an initial design of the database to be used and a flow chart. With all these elements already developed and tested through individual interviews, we can move on to the next stage.

4.2 App Design

4.2.1 Initial Mechanics Design: LM-GM Framework

Game loop

1. The user launches the application which takes him to the "*PreWorkOut*" screen.
2. The user can interact with this screen to go to any of the following options:
    - Can click on the buttons that will take him to the "*Extras*" screen. Where you can check the data of other sessions, achievements and pending tasks, as well as share data with other users or on social networks.
    - Can go to the "*Personalization*" screen, where they can change the design of the application, so that it is the way that best suits their preferences.
    - The user will be able to start a new activity, which will take him to the "*WorkOut*" screen.
3. Once the activity begins, the user will decide how or what type of activity will be carried out.
4. When the user decides, the activity will end, taking them to the "*PostWorkOut*" screen, where they will receive feedback.
5. Once the feedback has been received, the user will be able to decide what to do with said activity:
    - If is not satisfied with the result, they can delete the activity.
    - If, on the other hand, it seems like a good job, the activity will be saved in the corresponding "*Extras*" screen.
6. In case of saving the activity, the user will receive a reward based on the performance used.

Once this point is reached, the application will take the user back to point 1. The following figure shows the playable loop along with some labels on each loop action, indicating the Learning Mechanics with blue letters and the Game Mechanics with orange numbers.[29]

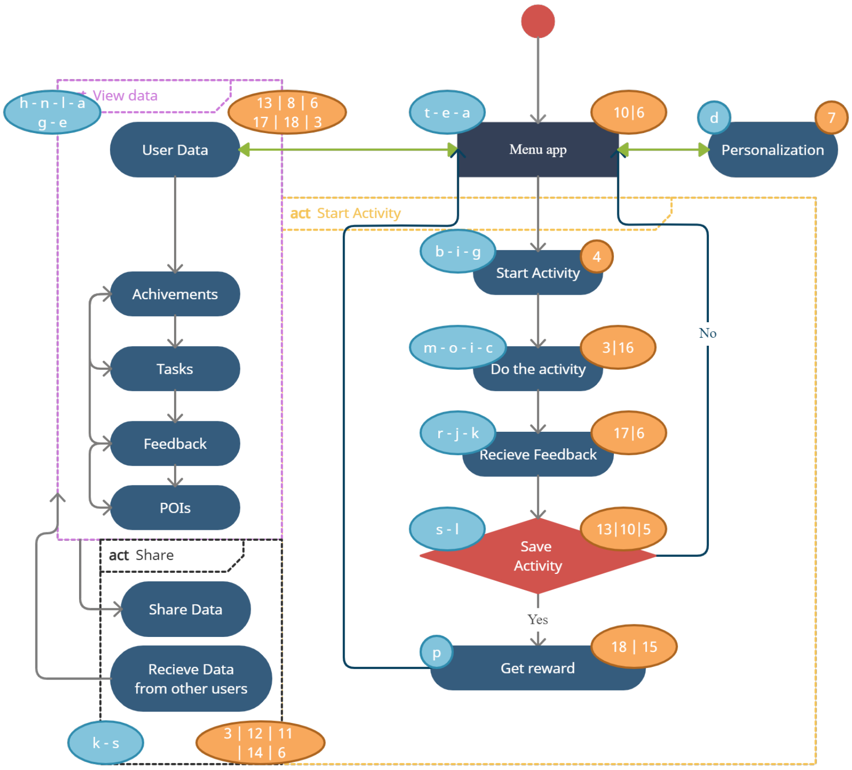
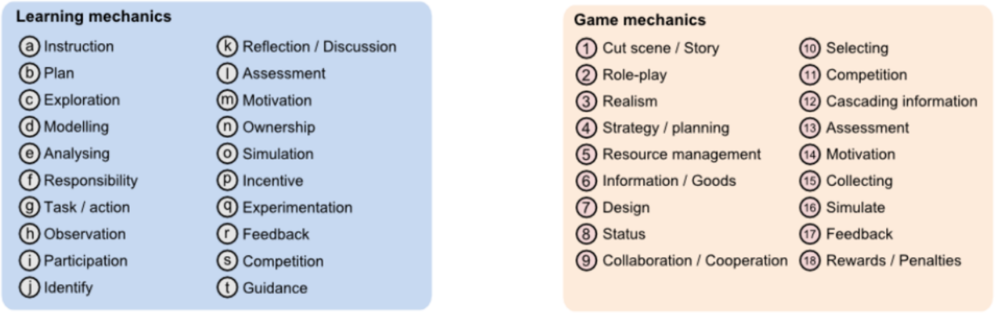


Figure 4.1: *Activity diagram with the Game Mechanics and Learning Mechanics associated with each action*



Learning mechanics

ⓐ Instruction	ⓚ Reflection / Discussion
ⓑ Plan	ⓛ Assessment
ⓒ Exploration	ⓜ Motivation
ⓓ Modelling	ⓝ Ownership
ⓔ Analysing	ⓞ Simulation
ⓕ Responsibility	ⓟ Incentive
ⓖ Task / action	ⓠ Experimentation
ⓗ Observation	ⓡ Feedback
ⓘ Participation	ⓢ Competition
ⓙ Identify	ⓣ Guidance

(a) Learning Mechanics

Game mechanics

① Cut scene / Story	⑩ Selecting
② Role-play	⑪ Competition
③ Realism	⑫ Cascading information
④ Strategy / planning	⑬ Assessment
⑤ Resource management	⑭ Motivation
⑥ Information / Goods	⑮ Collecting
⑦ Design	⑯ Simulate
⑧ Status	⑰ Feedback
⑨ Collaboration / Cooperation	⑱ Rewards / Penalties

(b) Game Mechanics

Figure 4.2: *Tables with the Game Mechanics and Learning Mechanics*

Relating the mechanics with the Thinking Skills[4][32][23]

Next, once we have decided on the game loop and assigned mechanics to each of our actions, we will establish the connection of each of these with Bloom's Taxonomy and its corresponding thinking skill field.

GAME MECHANICS		THINKING SKILLS	LEARNING MECHANICS	
Status	Design/Editing	CREATING	Ownership	
Strategy/Planning			Modeling	Plan
Reward/ Penalties	Resource Management	EVALUATING	Reflect/ Discussion	Assessment
Assessment	Motivation		Motivation	Incentive
FeedBack		ANALYSING	Analysing	Feedback
Realism			Observation	Identify
Competition	Movement	APPLYING	Task/Action	Simulation
Simulate			Competition	
Cascading Information		UNDERSTANDING	Participation	
Selecting				
Goods/Information		RETENTION	Instruction	Guidance
			Exploration	

Table 4.1: *LM-GM mechanics associated with the thinking skills of Bloom's taxonomy*

4.2.2 Mechanics justification

For this project, the user is expected to be capable.

- Analyze their own physical process and assess their own advice.
- Create a routine by planning.
- Learn how to keep track of their data.
- Apply this knowledge in other activities.

Based on these requirements we will follow the classification based on Bloom's ordered thinking skills, given by the LM-GM framework. In this case it shows the mechanics used in each field of thinking. The application is fragmented into four main parts, plus the menu:

Main menu

Here, the user puts into practice the "Information" and "Selection" mechanics since they must check the information displayed and choose what their next step will be based on their current preferences. By doing this, it learns to navigate and analyze its virtual environment, thus obtaining the necessary instructions to move around the application.

Personalization

The user can customize the application in his own way, so that in this way each appearance is different, developing the learning of modeling which is achieved with the "Design".

Main Loop - Start Activity

Since the user has to plan when they go out to do an activity in which he needs his participation, they learn to organize himself and plan the actions of his routine.

Main Loop - Do the activity

As this is an activity that takes place in real life, the user gets to learn the motivation to explore and participate in a realistic environment.

Main Loop - Recieve de feedback

Once the activity is finished, the user will receive feedback that will be unique, thus giving the user an identity. In addition, the user will have to discuss with himself to assess the information received. In this way, the user will have to manage their resources.

Main Loop - Save Activity

Once this internal assessment by the user is finished, they will evaluate whether to save this data so that the rest can see their progress, thus promoting competition.

Main Loop - Get Reward

If the user has decided to save that information, they will be awarded a reward based on their activity, being able to save their data to unlock future achievements and thus promoting the incentive to continue participating.

View Data

Here the user will have access to their tasks, rewards and all their own data, which they can evaluate and analyze. The user will also be able to check their past feedback and their current status of the challenges, whether they are self-imposed or given by the application.

Share

Once their own data has been verified and evaluated, the user can choose the most relevant data to share with their friends, to generate a desire to compete with other users. This information will be shared with their saved friends, motivating the user to try to outdo each other.

4.2.3 Initial database design

Once the design of the mechanics is finished, the design of the initial database will be developed. The design will be made up of seven main tables where we will store the user's data, the routes that have been carried out, the tasks that have been created and the POIs that have been marked, as well as a table where the information of the friends that the user has and of the achievements that exist and those that you have unlocked. Besides, the design will have another six data prepopulate tables to complement the “Tasks” and “*Routes*” tables. Remaining the following initial design of the database that will be implemented later.

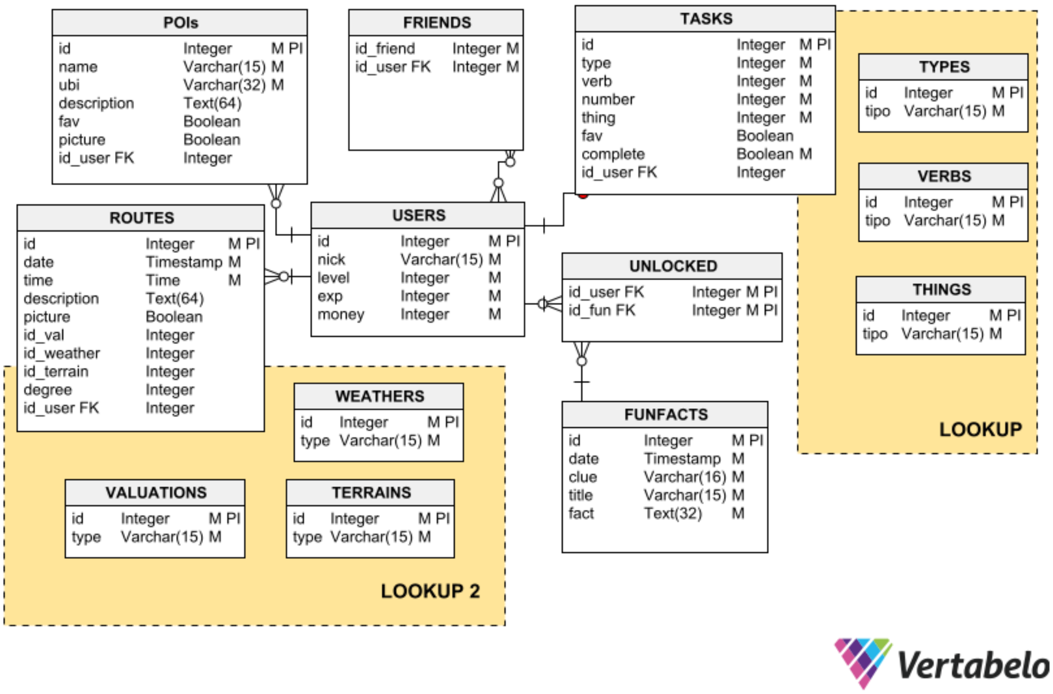


Figure 4.3: *Initial database design*

4.2.4 Initial app flow design

Now that we have the playable loop linked to the learning and game mechanics and a database design that adapts to that loop, it will be the turn of the app’s flow design. The application will be divided into four main screens:

1. **PreWorkOut Screen:** This screen is where the user will have their data such as nickname, level and coins, as well as all the buttons to take them to other screens. In addition, a random task has been included in the lower part of your task list, although this feature may be pending change.

2. **WorkOut Screen:** If the big button on the *PreWorkOut* Screen is pressed, the *WorkOut* Screen will start where the user will carry out all the activity, once it is finished it will go to the *PostWorkOut* Screen.

3. **PreWorkOut Screen:** The function of this screen is to give feedback to the user and where they can also fill in optional fields to improve data collection.

4. **Extras Screen:** Once the user has exited the main loop, from the *PreWorkOut* Screen they will be able to access this window which is a Tab Menu, which is divided into 4 tabs:

    - **POIs Tab:** Where they will see the information of the POIs and add new ones.
    - **Routes Tab:** Where they can review the feedback from previous activities.
    - **Task Tab:** Where the user will create self-imposed tasks to set objectives.
    - **Achievements Tab:** Where the user will be given challenges to meet and obtain fun facts.

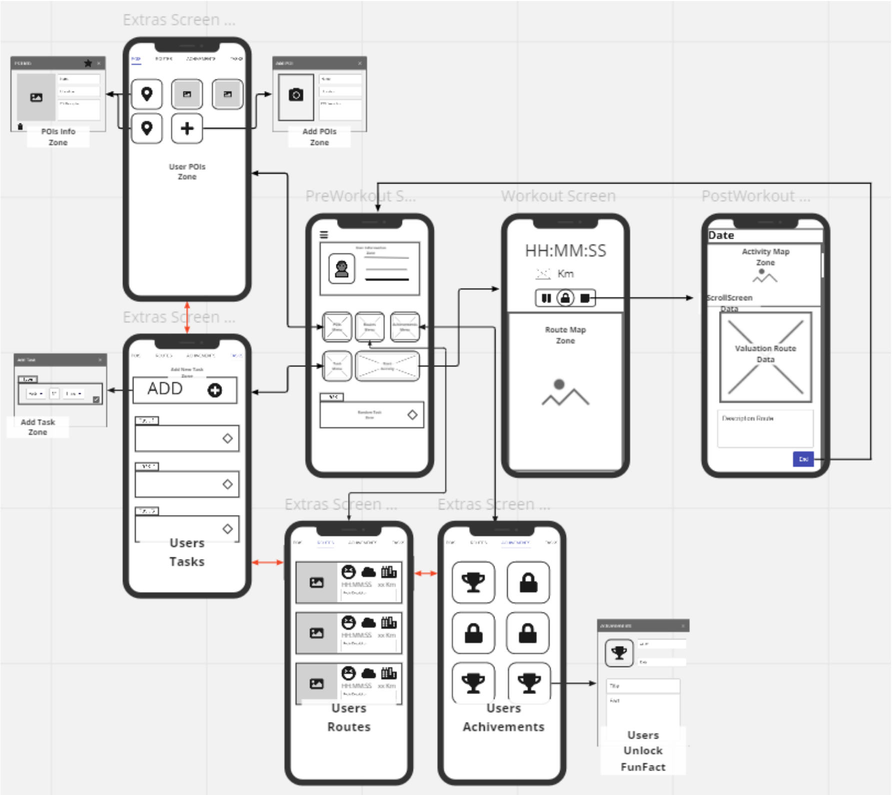


Figure 4.4: *App conceptual flowchart*

4.3 Testing of application designs

Finally, once each and every one of the previous sections had been developed, it was decided to start designing the main screen of the application. Since it is the one that the user is expected to spend more time seeing, because it serves as a link between all the others, for this design proposals were made that were subjected to short interviews to find out what opinion the user had about them. In this way, the main appearance would already be based on the user-centered design methodology.

4.3.1 Design

Design 1: Initial

As described in previous sections, the main screen will be divided into three areas: the user area, the action button area and the tasks area, which is a scroll with all the tasks that the user creates. With that in mind, the following design was made and feedback solicited, leading to the following conclusions.

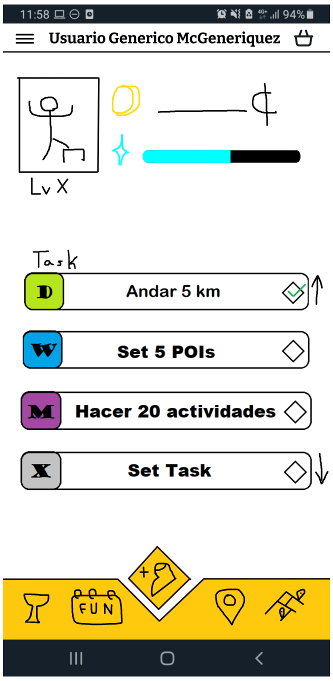


Figure 4.5: *Design 1*

Design Conclusions	id user
The screen does not have a proper and current design, and it is also clumsy.	U2, U6
Name placement generates disconnection	U7,U4,
The meanings of the letters are not very well understood.	U6
The to-do list is only convincing at a conceptual level but not in practice.	U7,U1
To-do list detracts from the main point	U5,U4, U1
The bottom bar does not convince people and is very sober and not very dynamic, and it is very formal.	U6,U3, U2

Figure 4.6: *Initial concept of the PreWorkOut Screen*

Design 2

Once the feedback was received, the placement of the action buttons was inverted, so that they will gain more importance by being located in the center. In addition to their size being increased so that they would be easier for the user to press. For this, a circular design was chosen. On the other hand, the name was placed in the user part since leaving it on top caused confusion and could conflict with certain mobile designs. At the end of designing the new screen, the group of experts was interviewed again to find out their opinion once again and reach conclusions.

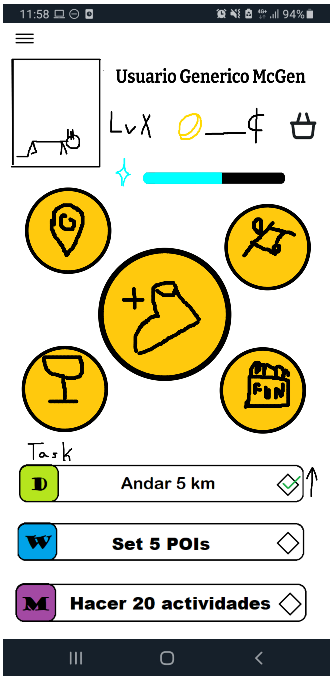


Figure 4.7: *Design 2*

Design Conclusions	id user
Tasks still take up a lot of space.	U7,U4, U1
Circular buttons do not like	U6, U5, U4, U1
The size of the buttons is correct making the button easy to press and keeping which is the main one.	U2, U3
The layout leaves a lot of wasted space but is on the right track.	U7, U4, U5, U6, U3, U1

Figure 4.8: *Concepto #2 de la PreWorkOut Screen*

Design 3

This design focused on changing the tasks section, modifying the number of tasks displayed on the screen and adding its own button, so that the user can access all of them and be much more functional. Instead, the user now has access to a random task from his list, apart from changing the visual aspect of these so as not to depend on a color code. The experts were interviewed again and the following information was obtained.

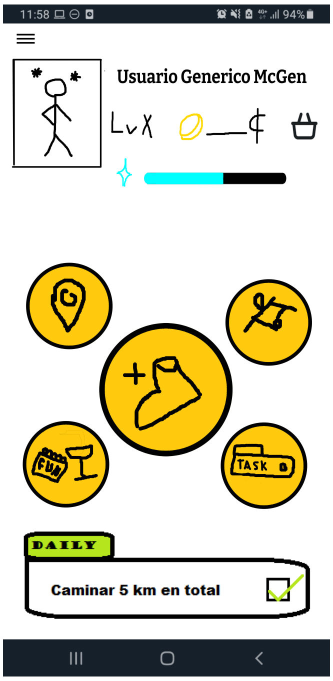


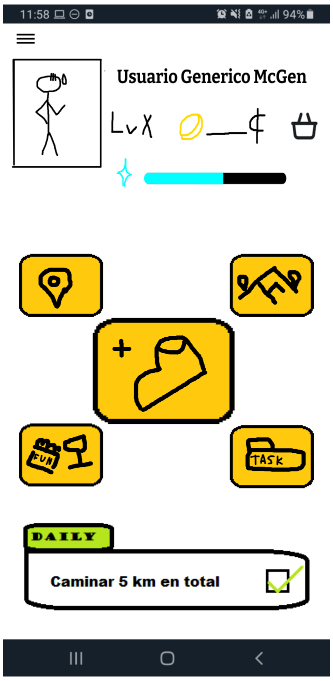
Figure 4.9: *Design 3*

Design Conclusions	id user
The new task button likes more	U1,U4, U5
The simplification of the task section is satisfactory.	U1, U2, U4, U6
They think that there are many different shapes on the screen.	U6

Figure 4.10: *Concepto #3 de la PreWorkOut Screen*

Design 4

Although the solution is reached regarding the tasks section, the design of the action buttons still does not convince the group of interviewees due to their circular shapes, for which they were changed in shape. Then, an interview was conducted again and the following conclusions were reached.



Usuario Generico McGen

Caminar 5 km en total

Figure 4.11: *Design 4*

Design Conclusions	id user
A lot of blank space is still left	U7, U6, U5, U4,U3, U2, U1
The shape of the buttons is more pleasing	U6, U5, U4,U1

Figure 4.12: *Concepto #4 de la PreWorkOut Screen*

Design 5: End

To solve the problems found, the action buttons are redistributed, reaching a more compact design while keeping the main button larger. Likewise, the user section receives a fund to further unify this section. As always, these results were reviewed by the experts, drawing the following quite favorable conclusions.

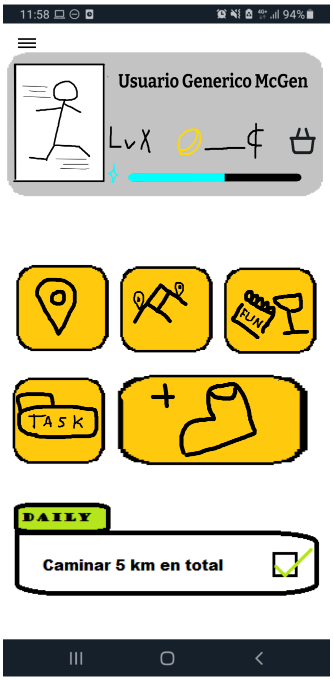


Figure 4.13: *Design 5*

Design Conclusions	id user
The distribution is much more attractive	U1,U5, U6,U7
The user still knows which button is the main button.	U1, U4, U5,U6
The background in the user section makes it easier to view	U2, U4, U5, U7
The overall design is more pleasing, contemporary and dynamic.	U3, U4
The artwork should be improved and a different combination of buttons should be chosen.	U3
The tasks section is not very convincing	U7, U2

Figure 4.14: *Concepto #5 de la PreWorkOut Screen*

4.3.2 Final conclusions

Once the design has been iterated several times, the most optimal design has been reached on which work can begin. However, the following conclusions are reached.

- The artistic section has to be improved.
- The tasks section has to be reviewed and consider other functionalities for that space

Click here to see the results of the interviews

4.4 Data investigation

As mentioned, in the "Extras" Screen achievements tab, when meeting certain conditions a fun facts will be unlocked, due to the topic of the app, these facts have been decided to be about sports and nutrition. Therefore, to gather information, different pages of frequently asked questions (FAQ) have been searched, making a small summary answering those questions.

	Spanish	**English**
Clue	Realizar 5 actividades	Complete 5 activities
Title	¿Que puedo hacer para mantenerme en forma?	What can I do to keep fit?
Fact	Hacer ejercicio está bien, sin duda, aunque tampoco está de más dormir bien, evitar fumar, estirar varias veces a lo largo del día, pasear o dedicar tiempo a actividades relajantes.	Exercising is good, without a doubt, although it doesn't hurt to sleep well, avoid smoking, stretch several times throughout the day, go for a walk or spend time in relaxing activities.
Link[6]		

Table 4.2: *Fun fact table example*

Chapter 5

Second stage - Prototype development

Contents

In this second stage, based on the data obtained from the initial investigation, a prototype began to be designed and programmed that will implement the main loop of the app, as well as the beginning of the concept of the "Extras" Screen.

5.1 Objectives of this stage

During this second stage, we will apply the data obtained in the *Predevelopment* stage to create a functional prototype that is then tested by the user. Therefore, this stage will be more focused on functionality than on the visual section, although design aspects will be taken into account. For this prototype, the main loop will be implemented, which includes the *PreWorkOut*, WorkOut and *PostWorkOut* screens, and the *Extras Routes tab* screen will be implemented to display this data. When this prototype is finished, it will be subjected to a user test to find out their opinion.

5.2 Implementation

5.2.1 Screens implementation

Before starting the programming part, a graphical interface with which the user can interact is needed. Here comes the first error regarding the flow of screens and it is that at no time was a screen designed where the user can log in to the application.

Therefore, a simple screen begins to be developed where the user can put their name, since as it is an application that saves data locally, no type of password or email was needed to register the user. After overcoming this problem, the *PreWorkOut* Screen began to be implemented following the user's instructions and preferences.

A button was implemented to be able to log out and exit the app in case they want to register another user. Regarding the sketch of the previous stage, everything remained the same, both the user area and the action buttons. The only thing that changed is that based on the data, the task part was left as "Coming Soon".

The WorkOut Screen is then implemented, the biggest change compared to the initial design was that the order of the map and data areas was reversed and the data area was reduced compared to the proposal made. In addition, a “Start” button was added and the pause and stop icons were replaced by text.

Immediately afterwards, the *PostWorkOut* Screen is implemented, following the initial design. It has not had any significant changes, as indicated in the design, it is made up of the map area of the activity that has been carried out and a scroll area where the user receives the feedback and put their data. At the end of this scroll are the buttons to save or delete the activity.

On the other hand, the *Extras* Screen is implemented, which is a tab menu in which, in future versions, the functionalities of POIs, Task and Achievements will be implemented, but for now only the *Routes* tab has been implemented, which is composed of a Recyclerview where the user can see all the routes he has made.

Finally, and due to the great amount of information displayed on the routes, it was decided to add another screen that would be an activity visualizer. It is accessed by pressing the selected item in the Recyclerview of the *Routes* tab. *Routes Visualizer* is similar to the *PostWorkOut* Screen, it is made up of an area where you can see the map of the route, the received data and the additional data that the user has wanted to add, as well as a button to delete the route.

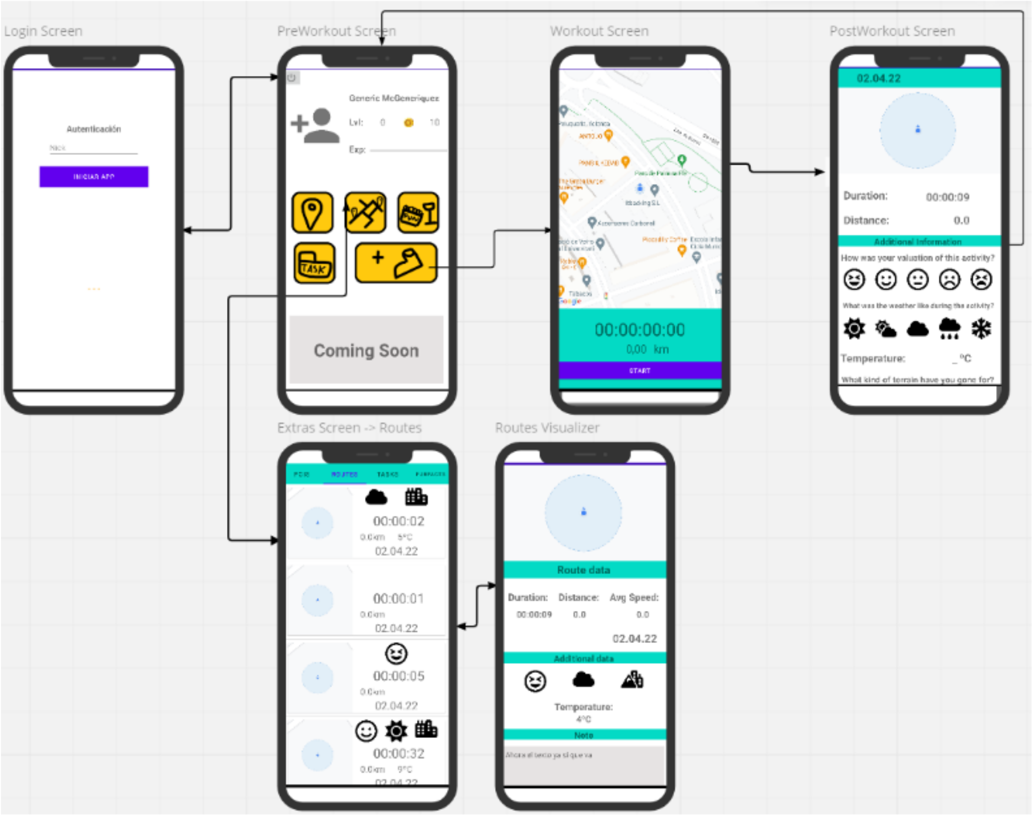


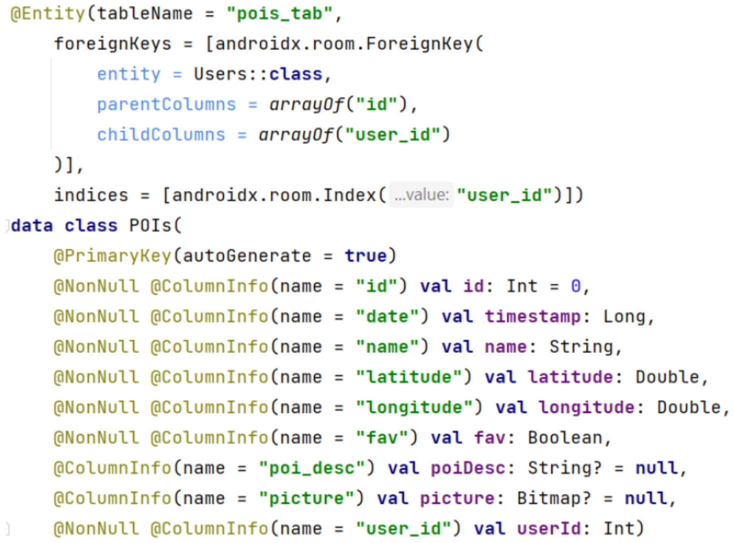
Figure 5.1: *Prototype Flowchart*

5.2.2 Database Implementation

In point 4.2.3 the relational model of the app is explained. The tables defined in the model have to be transformed into data entities for ROOM.

Room's main components

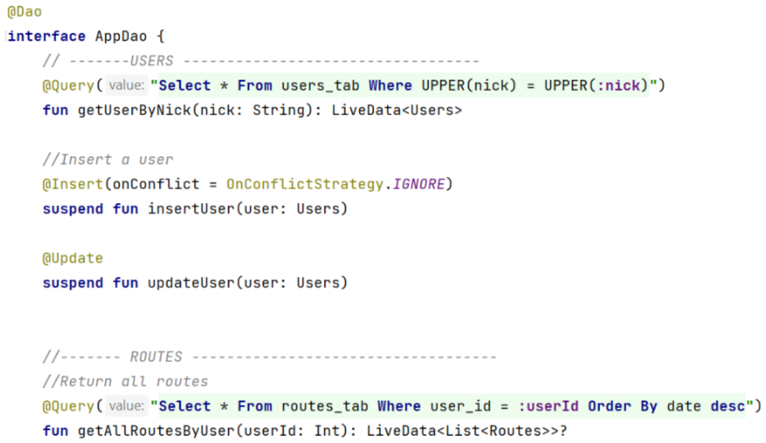
The **data entities** that represent tables in the app's database. In the model directory, the classes that define the model tables, such as Room entities, have been created. For this, they have to be marked with the Entity(tableName= name_tab) annotation. We also need to mark an attribute of this entity class as a primary key which we do with the PrimaryKey annotation and define its relationship to the Users entity with the ForeingKey annotation.



```
@Entity(tableName = "pois_tab",
    foreignKeys = [androidx.room.ForeignKey(
        entity = Users::class,
        parentColumns = arrayOf("id"),
        childColumns = arrayOf("user_id")
    )],
    indices = [androidx.room.Index( ...value: "user_id")])
data class POIs(
    @PrimaryKey(autoGenerate = true)
    @NonNull @ColumnInfo(name = "id") val id: Int = 0,
    @NonNull @ColumnInfo(name = "date") val timestamp: Long,
    @NonNull @ColumnInfo(name = "name") val name: String,
    @NonNull @ColumnInfo(name = "latitude") val latitude: Double,
    @NonNull @ColumnInfo(name = "longitude") val longitude: Double,
    @NonNull @ColumnInfo(name = "fav") val fav: Boolean,
    @ColumnInfo(name = "poi_desc") val poiDesc: String? = null,
    @ColumnInfo(name = "picture") val picture: Bitmap? = null,
    @NonNull @ColumnInfo(name = "user_id") val userId: Int)
```

Figure 5.2: *Example of entity, POIs table*

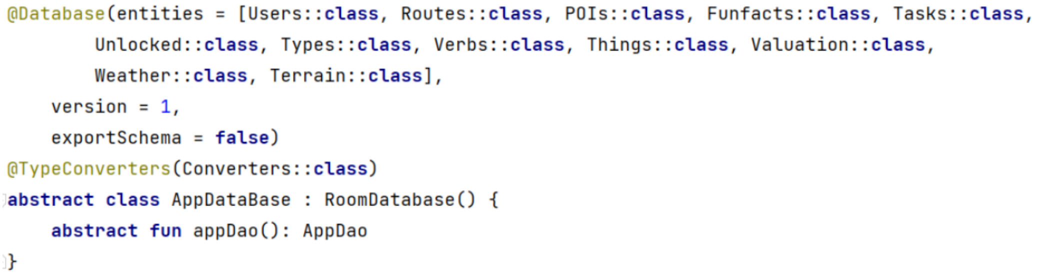
Data Access Objects (DAO) that provide methods that the app can use to query, update, insert, and delete data in the database. The DAO has been defined in the app as an interface, marked with the annotation @Dao[15]. To prevent lookups from blocking the UI, Room does not allow access to the database on the main thread. This restriction implies that DAO lookups must be asynchronous. Has been used:
Write-once (suspend) lookups that insert, update, or delete data in the database.
Observable read lookups (LiveData) that read data from the database each time the tables change and output new values to reflect those changes.



```
@Dao
interface AppDao {
    // -------USERS ----------------------------------
    @Query( value: "Select * From users_tab Where UPPER(nick) = UPPER(:nick)")
    fun getUserByNick(nick: String): LiveData<Users>

    //Insert a user
    @Insert(onConflict = OnConflictStrategy.IGNORE)
    suspend fun insertUser(user: Users)

    @Update
    suspend fun updateUser(user: Users)


    //------- ROUTES ----------------------------------
    //Return all routes
    @Query( value: "Select * From routes_tab Where user_id = :userId Order By date desc")
    fun getAllRoutesByUser(userId: Int): LiveData<List<Routes>>?
```

Figure 5.3: *Partial section of the DAO*

The **database** class, which contains the database and serves as the primary access point for the underlying connection to the app's persistent data. This class provides the app with instances of the DAO associated with that database. In the following code, an AppDatabase class is defined to contain the database. The class has a @Database[17] annotation that includes an entities[16] array to list all the data entities associated with the database. It must be an abstract class that extends from RoomDatabase[21] and for the defined DAO it must have an abstract method with no arguments that returns an instance of the DAO class.



```
@Database(entities = [Users::class, Routes::class, POIs::class, Funfacts::class, Tasks::class,
        Unlocked::class, Types::class, Verbs::class, Things::class, Valuation::class,
        Weather::class, Terrain::class],
    version = 1,
    exportSchema = false)
@TypeConverters(Converters::class)
abstract class AppDataBase : RoomDatabase() {
    abstract fun appDao(): AppDao
}
```

Figure 5.4: *Kotlin Database Class*

Room does not support storing elements of type bitmap in its tables. For this reason, type converters have been defined, which are methods that tell Room how to convert custom types to known types. To identify type converters, the TypeConverters annotation is used.

Loading data from Lookup tables

When the app starts, these tables must be preloaded in the database since they are tables with fixed values, and they will not be modified.
A file called "*data.json*" has been created under the res → raw directory, which contains the data of the tables to be preloaded (in Json format).
To parse the data in Json format, we do it through Gson, an open source library[37].
With the help of the Gson library, selective reading of the key-value pairs of the Json object is avoided, since it replaces the reading of the object with the fromJson() method.

5.2.3 Features Implementation

In this section we will show a brief explanation of the functionality that was carried out in the prototype phase of the app.

The *Login* Screen is the first screen where the user's Nick is collected and saved in the database, in case the user's name is already registered, will enter directly to the

PreWorkOut Screen, otherwise, a dialog will pop up asking if they want to create a new profile, thus preventing the user from creating another profile unintentionally. After entering the Nickname, the *PreWorkOut* screen will show the user's data.

In order to allow the user to open and close the application without having to re-enter his Nick, SharedPreferences have been used. These are collections that allow storing small amounts of key-value type information, which is then accessible thanks to the API provided by [14].

Once the user is logged in and on the PreWorkOut screen, the Activity button, the main part of the app, is found. When the button is clicked and before the map activity starts, a dialog opens to ask the user to accept the location permission for the app.

Once the permission to start an activity is accepted, a request is made to the Google API to display the map, positioning the location where the user is. The control of the route by the user of the app, is done through the buttons prepared for it, being able to start, pause, resume and finish it. All the control of the buttons that are associated with the route has been done through an android service called WalkService. This control is performed through functions that start, resume, pause or end the service.

The service controls the FusedLocationProviderClient to get the most recent location of the device. This provider is one of the Google Play Services location APIs that manages the underlying location technology and is a simple API that allows specifying requirements (such as high accuracy or low power) at a general level and also optimizing the device's battery usage[10]. The locations collected by the combined location provider are stored in variables suitable for this purpose to subsequently plot the polylines that show the route on the map.

Once the user finishes the activity, the service ends, and the data of the activity is collected in the *PostWorkOut* Screen, and the data of the activity can be completed (valuation, temperature, type of terrain, etc.). Once the data has been completed, the route can be saved by saving the route data in the Routes entity of the database.

The app returns to the PreWorkOut screen where there is a button to view the user's saved routes in list form. Each item displays an image of the route, the time, kilometers traveled and the date the route was taken, as well as any optional values that have been saved. Clicking on each item in the list takes you to a route display screen with all the details of the route. On this screen there is a button at the bottom, where the selected route can be deleted. In the WalkService, the channel for notifications is also created. For this purpose, a NotificationChannel object is constructed with a unique channel ID, a name visible to the user and an importance level[8]. Notifications are controlled when the service is started, paused or terminated (by making changes to the notification buttons).

5.3 Prototype testing

5.3.1 Testing development

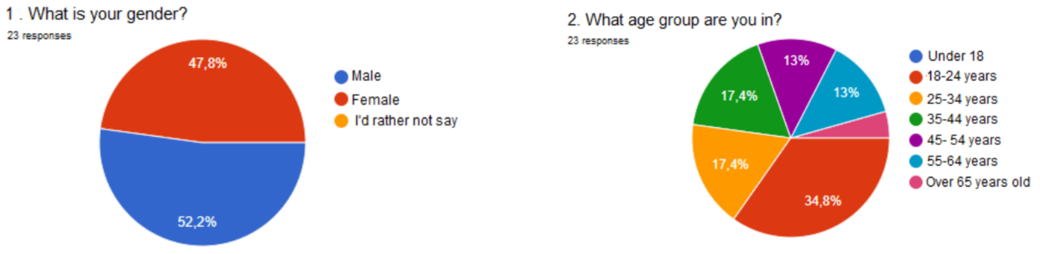
Because the application has to be accessible to a large number of people, the users chosen were of varying ages and genders so that the sampling would be reliable. Knowing this, each one was sent an explanatory PDF. This document had the following sections:

- **Introduction:** Where the document to be read was summarized along with a thank you for their participation.
- **Process information:** Where it was explained to them what their function would be and the time they had to perform it. It also contained a list with previous information on errors already detected and clarifications on certain sections of the app, such as the incompatibility with the dark mode.
- **Installation guide.**
- **Questionnaire:** Where I explained the sections of the questionnaire and a link to access it.

Once the application had been submitted along with this informative document, the next version was to be developed while user testing took place.

5.3.2 Testing results

The following table shows the results of the expert group after the testing period of the application. Based on this feedback and advice, the application was further revised and changes were made for the next version.
The questionnaire was divided into three parts: biometric data, app practicality and app consistency. This test was available from April 9 to April 12 and was answered by 23 people, of which 11 were female and 12 male.
The age range, as required, was diverse.



(a) Gender chart

(b) Age chart

Figure 5.5: *Gender and age graphs from the prototype survey*

Question	Answer
Regarding PreWorkOut's screen satisfaction survey	16 out of 23 were pleased with the distribution of the elements, while 5 out of 23 were not.
	18 of 23 were pleased with the size of the buttons, while 3 of 23 were not.
	17 of 23 were pleased with the size of the username, while 1 of 23 were not.
	7 of 23 were pleased with the button icons, while 13 of 23 were not.
Which of these options would you like the Coming Soon window of the PreWorkOut Screen to be its future use?	15 out of 23 chose a statistical table. 4 out of 23 chose a random daily challenge. 2 out of 23 chose a task from their to-do list 2 out of 23 suggested another option
Regarding WorkOut's screen satisfaction survey	17 out of 23 were satisfied with the size ratio of the two zones, while 1 out of 23 were not.
	17 of 23 were pleased with the area of the map covered by the map, while 1 of 23 were not.
	15 out of 23 were pleased with the distribution of the data in the data area, while 2 out of 23 were not.
	18 of 23 were pleased with the size of the data shown, while 2 of 23 were not.
Regarding PostWorkOut's screen satisfaction survey	15 out of 23 were pleased with the layout of the elements on the screen, while 2 out of 23 were not.
	14 out of 23 were pleased with the amount of data displayed on the screen, while 3 out of 23 were not.
	16 out of 23 were pleased with the size of the texts in the "Route Data" window, while 2 out of 23 were not.
	14 out of 23 were pleased with the size of the rating icons, while 3 out of 23 were not.
	An average of 15 out of 23 users understood the rating icons
Regarding the section where you put the note/commentary of the route, would you like a more eye-catching design?	15 out of 23 said yes, as it would make the text clearer than on a single line. 2 out of 23 said no, as it would make it harder to see the screen at a glance. 5 out of 23 said no, but would do something else to make it stand out more. 1 of 23 refrained from answering.

Question	Answer
Regarding the satisfaction survey of the Extras screen, Routes tab.	17 out of 23 were pleased with the distribution of the data, while 2 out of 23 were not.
	17 of 23 were pleased with the amount of data displayed.
	15 of 23 were pleased with the size of the map image.
	13 out of 23 were pleased with the size of the rating icons, while 4 out of 23 were not.
Before filling in the survey, were you aware that you can see more information about the routes by clicking on them?	16 of 23 knew that they could expand the information on the routes by clicking on them. 7 out of 23 were not aware of this feature of the application.
Regarding the RoutesVisualizer screen satisfaction survey.	18 out of 23 were pleased with the distribution of the display data, while 2 out of 23 were not.
	18 of 23 were pleased with the amount of data shown.
	14 out of 23 were pleased with the icons and texts, while 2 out of 23 were not.
	13 of 23 were pleased with the size of the route image, while 3 of 23 were not.
Would you like to see a way to switch languages be added?	14 out of 23 said they would like there to be a way. 9 out of 23 said they would not.
Did you know how to use the application?	16 out of 23 found it easy to get used to the app. 5 out of 23 knew how to use the app but found it difficult to get used to it. 2 out of 23 could not get used to it but think they will be able to with a little more time.
Have you encountered any errors on the WorkOut screen?	12 out of 23 people found errors on the screen.

Table 5.1: *Prototype questionnaire results*

Regarding the app's functionalities, the following graph shows the use of each section.
The most problematic feature is the "Add description or notes" function, as it was hardly used by users. Regarding the other features, they are more balanced.

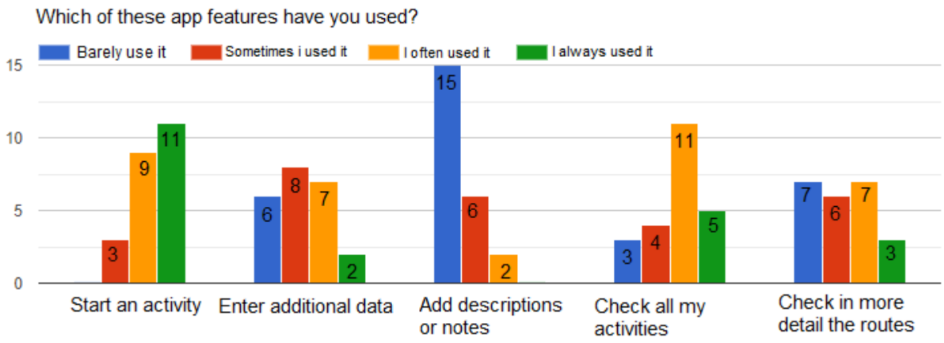


Figure 5.6: *Bar chart of functionalities used by the user in the prototype*

The table and graph shown only illustrate the most relevant data of this questionnaire, in case you want to expand the information or see these in more detail you can click on the following link.

Click here to see the result of the prototype questionnaire

5.4 Prototype Conclusions

Once the testing was completed and feedback was received, the changes to be made in the next version were determined.

- Find a theme for the application.
- Create a statistics table in the *PreWorkOut* Screen where the Coming Soon text was.
- Fix the layout of the *PreWorkOut* and *PostWorkOut* screens.
- Redesign the icons of the action buttons from the *PreWorkOut* Screen.
- Decrease size of valuation icons from the *PostWorkOut* Screen.
- Change the layout of the note in the *PostWorkOut* Screen.
- Redesign the Route item so that the user knows that they can expand the information.

- Fix multiple *WorkOut* screen errors.
- Add a way to change the application language.

All these changes will be implemented in the next version as well as the addition of new functionalities to the application.

Chapter 6

Third stage - Version development

Contents

In this third stage, as mentioned above in point 3.3.2, it is the most important of all, since it is the one that will be repeated iteratively until the desired result is achieved. For this, based on the conclusions of the previous version, consequential changes will be made and bugs found by the user will be fixed, and new functionalities will always be added from one version to the next, so that the user can give consistent feedback on these changes.

6.1 Beta development

During this stage, taking into account the findings of the prototype, we intend to fix those design bugs found by the user, apart from adding the new functionalities. Among these functionalities are the ability to add POIs and the creation and unlocking of achievements, where the curious data collected in the Predevelopment stage will be applied.
As with the prototype, when this beta is developed, it will be given to the user for testing so that they can find bugs and suggest changes to the design.

6.1.1 Improvements and implementations

Screens implementation and improvements

As highlighted in the previous stage, before implementing the new functionalities, it is necessary to create a graphical development environment. For this version to add POIs and FunFacts, it is necessary to design each of its display screens, screens to add, their corresponding tabs in the Extras Screen and the items to be displayed in the Recycler View.
In addition, taking into account user feedback, certain screens were redesigned to meet the required quality standard.

Based on the prototype design, two other tabs were added to the *Extras* Screen, *POIs Tab*, which is a grid layout of two elements showing an image and name of the POIs that have been added. By clicking on each of these items of this layout, the information of these POIs will be expanded, thus being able to display other data such as the location given in latitude/longitude coordinates, and the description that the user has put to each of these points of interest, expanding their information and allowing to delete and edit these POIs.

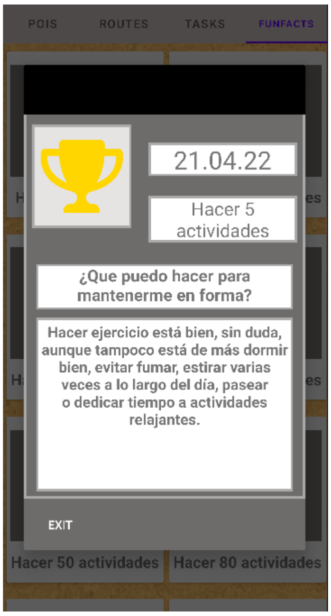
To add one of these POIs, it is necessary to do it during the *WorkOut* Screen, since you have to be in the location where you want to add the POI. To access the *AddPoi* Screen, the user must click on the red pin in the *WorkOut* Screen. This will bring up a dialog where the user can fill in the information and take a photo. It will only be allowed to take a picture from the camera and not to add one from the gallery to motivate the user to go to the location they want to save.

To finish the section of new implementations of the beta, it remains to mention the FunFacts, these are challenges imposed by the application, that once they are completed, as a reward, the user will receive a fun fact related to the world of sports and nutrition. These as well as the POIs are in their respective tab of the *Extras* Screen, when clicking on them if they are not unlocked, they can be purchased and unlocked using the game currency, otherwise, a screen will appear with the date of when it was unlocked and the fun fact.

The application currently includes 24 achievements, which are grouped into four categories.

Categories	
Mark X POIs	Do X hours
Travel X kilometers	Complete X activities

Table 6.1: *Achievement categories*



21.04.22

Hacer 5 actividades

¿Que puedo hacer para mantenerme en forma?

Hacer ejercicio está bien, sin duda, aunque tampoco está de más dormir bien, evitar fumar, estirar varias veces a lo largo del día, pasear o dedicar tiempo a actividades relajantes.

Hacer 50 actividades

Hacer 80 actividades

Figure 6.1: *FunFact Visualizer Screen, example of unlocked achievement*

Finally, it remains to talk about the improvements that were made in the rest of the screens due to the conclusions drawn from the previous stage. The most relevant ones would be: the incorporation of a statistics table in the *PreWorkOut* Screen, where the total kilometers and hours, the total average speed, the activities performed and the number of POIs marked are shown, also the possibility for the user to put a profile picture from their gallery was added.

In the *PostWorkOut* & *RoutesVisualizer* screens, the Save and Delete buttons were moved out of the scrollView, so that the user would not have to scroll all the way down to use these buttons, and warning dialogs were added when pressing the delete button so that the user could not accidentally delete the routes taken.
Finally, it was decided to give an aesthetic section to the app, to give a feeling of cohesion. For this artistic section we tried to leave the established by other applications and go for a more rustic style, therefore, the application has a paper notebook design, with posits, photos and paper, this way it will give a sense of proximity to the user and will not have that cold appearance that other applications have.

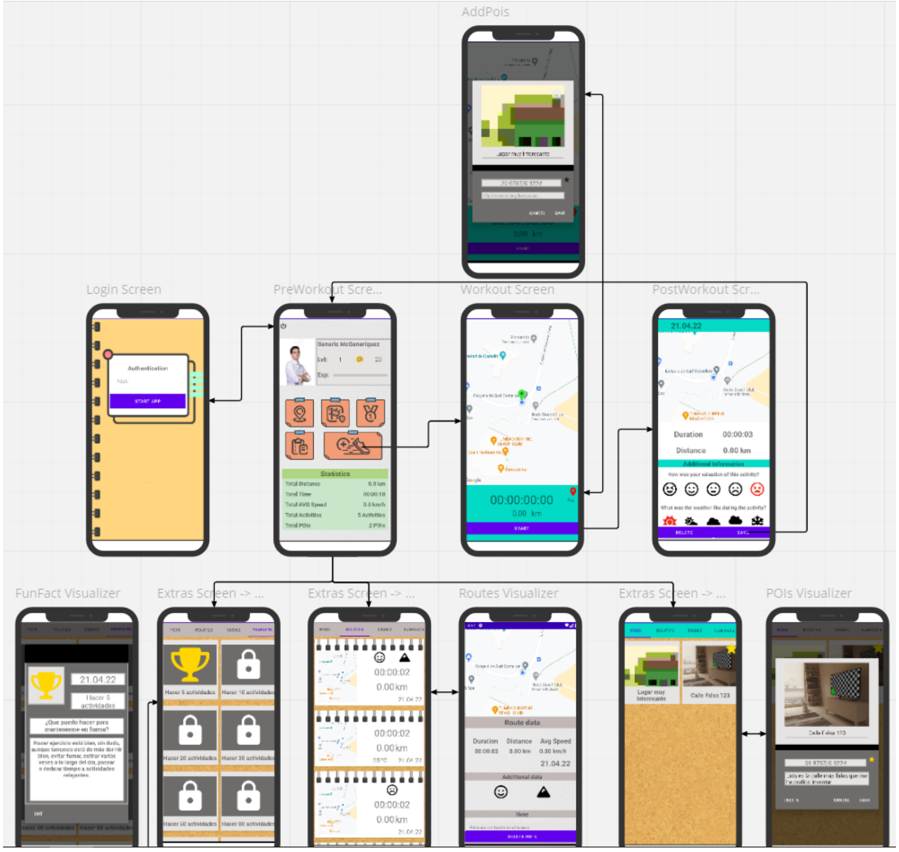


Figure 6.2: *Beta flowchart*

Feature implementation

One of the new features of this version along with POIs and achievements was the implementation of coins and experience, which in the future, will serve to unlock designs and features of the application so that the user can design to their liking.

User photo.
Once the user session has started and in the *PreWorkOut* screen, it has been implemented that when clicking on the user image, an image can be downloaded from the terminal's gallery. This is done through an Intent already defined in the Android operating system, so its implementation is simple since it basically makes the call to the Intent and then retrieves the response from it, saving the selected image in the user's ImageView.

Table of statistics & FunFacts unlocks.
Among other things, it is worth mentioning that the observer behavior pattern has been used for the statistics collected in the PreWorkOut screen. This pattern is useful for

displaying the data in a UI synchronized with the data stored in the database.
It has been realized with the Android LiveData library.
Room supports observable queries, which display LiveData objects (written as part of the DAO) and generates all the necessary code to update the LiveData object when the database is updated. The generated code executes the query asynchronously in a background thread when necessary[12].
Unlocking the data (FunFacts) is done in the same way as the statistics.

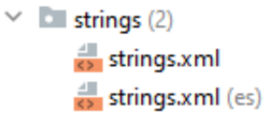
Coins and experience.
These coins and experience are obtained once the user finishes and saves an activity, since in case of deleting it they do not receive any of these, they can also get a few coins if they value their route. The following table shows the amount of coins and experience obtained depending on the activity.

Action	Description	Reward
Assess the activity	Per optional data valued	3 coins
Average speed	Between 0 and 6 km/h (walking)	10 coins per kilometer 10 exp points per kilometer
	Between 7 and 15 km/h (running)	20 coins per kilometer 20 exp points per kilometer
	Between 16 and 40 km/h (bicycle)	6 coins per kilometer 6 exp points per kilometer
	Over 40 km/h (vehicle)	Nothing
Activity minutes	Between 0 and 4 minutes	Nothing
	Between 5 and 30 minutes	15 coins 6 experience
	Between 31 and 60 minutes	55 coins 10 experience
	Between 61 and 90 minutes	60 coins 20 experience
	Between 91 and 120 minutes	90 coins 30 experience
	Over 120 minutes	Coin per minute 45 experience

Table 6.2: *Rewards earned depending on the objectives achieved*

App Localization.
In this version of the application was also added the Spanish localization, thanks to the L10n system that is built into the Android Studio program itself. This decision was made in order to make the application more accessible, so all the text resources of the

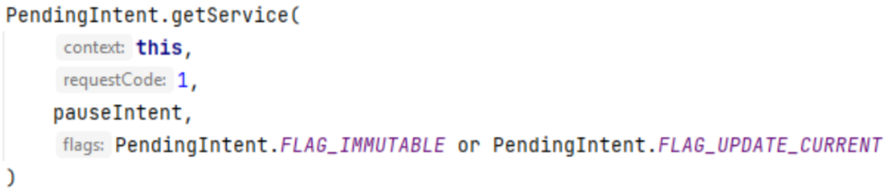
app were moved to a separate text file. This way, when users run the application, the Android system selects the resources it will load based on the device's configuration.
Whenever the app runs in a locale, Android loads the default strings from res/values/strings.xml. If this default file is not present or if any string required for the app is missing, the app does not run and displays an error.
The layout of the application is also optimized to be able to add any language without changing the design of the screens.



```
strings (2)
    strings.xml
    strings.xml (es)
```

Figure 6.3: *Project strings folder*

Compatibility with Android 12.
Another big difference from one version to another is that the beta allowed new users to enter as Android 12 compatibility was implemented. In the prototype, if you had version 12, the *WorkOut* Screen map did not work. This bug is solved by adding the "FLAG_IMMUTABLE" flag to the PendingIntent calls.



```
PendingIntent.getService(
    context: this,
    requestCode: 1,
    pauseIntent,
    flags: PendingIntent.FLAG_IMMUTABLE or PendingIntent.FLAG_UPDATE_CURRENT
)
```

Figure 6.4: *Pending Intent*

6.1.2 Testing

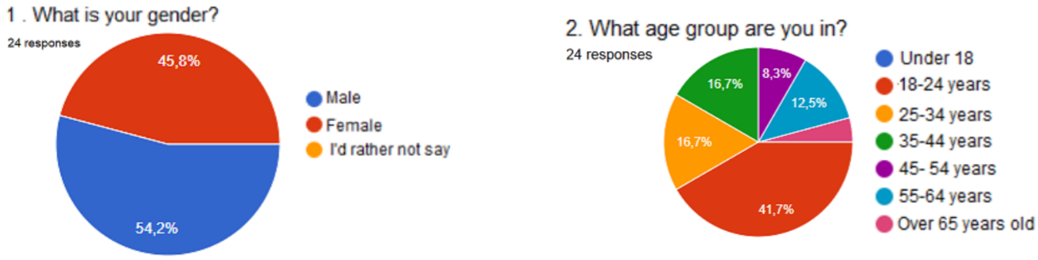
Testing development

For this version the same group of experts was contacted so that the difference in ages, genders and opinions did not vary. As in the previous stage, an explanatory PDF was sent to them, the document had the same sections as the previous one but with the addition that this one had a point explaining the changes, fixes and improvements compared to the prototype.
As in the previous test, this group of experts had one week to test the application and about three days to answer the questionnaire.

Testing results

The following table shows the results of the expert group after the testing period of the application. Based on this feedback and advice, the application was revisited and

changes were made for the next version.
The questionnaire was divided into three parts: biometric data, app practicality and app consistency. This test was available from April 29 to May 3 and was answered by 24 people, of which 11 were women and 13 men.
Regarding the age range, as required, it was diverse.



(a) Gender chart

(b) Age chart

Figure 6.5: *Gender and age graphs from the beta survey*

Question	Answer
Regarding the PreWorkOut screen satisfaction survey	22 out of 24 were pleased with the button icons, while 1 out of 24 were not.
	22 out of 24 were pleased with the data displayed in the statistics area, while 1 out of 24 were not.
	19 out of 24 understood all the action icons. 1 out of 24 did not understand the routes button. 4 out of 24 did not understand the POIs button.
Concerning the WorkOut screen satisfaction survey	All participants were pleased with all sections of this screen. With the exception of one who did not like the layout of the data.
Regarding the AddPois screen satisfaction survey	20 out of 24 were pleased with the size of the POI image, while 1 out of 24 were not.
	20 out of 24 were pleased with the distribution and quantity of the data.

Question	Answer
Regarding the PostWorkOut screen satisfaction survey	21 out of 24 were pleased with the size of the rating icons.
	21 out of 24 were pleased with the size of the route map, while 1 out of 24 were not.
	21 out of 24 were pleased with the location of the "Delete" and "Save" buttons while 2 out of 24 were not.
Regarding the satisfaction survey on the Extras screen, POIs tab	20 out of 24 were pleased with the design of the POIs.
	21 out of 24 were pleased with the size of the image.
Regarding the design of the POIs,would you like it to have a pin or piece of tape in its design?	17 out of 24 answered yes. 6 out of 24 answered no. 1 out of 24 answered that it depends on how it was integrated
Regarding the PoisVisualizer screen satisfaction survey	20 out of 24 were pleased with the placement of the buttons and the ability to edit the POIs.
	19 out of 24 were pleased with the idea of it coming out as a floating window rather than as a separate window.
Before filling in the form, were you aware that you can see more information about the POIs by clicking on them?	16 out of 24 knew that they could zoomin on POIs by clicking on them. 8 out of 24 were not aware of this function of the application.
Before filling in the form, were you aware that you can modify the data entered in the POIs?	14 out of 24 were aware that they could modify POI information by clicking on it. 10 out of 24 were not aware of this function of the application.
With regard to the satisfaction survey on the Extras screen, Routes tab	21 out of 24 were pleased with the distribution of the data.
	20 out of 24 were pleased with the size of the rating icons, while 2 out of 24 were not.

Question	Answer
Before filling in the form, were you aware that you can see more information about the routes by clicking on them?	22 out of 24 knew that they could expand the route information by clicking on the route. 2 out of 24 were not aware of this feature of the app.
Regarding the RoutesVisualizer screen satisfaction survey	20 out of 24 were pleased with the size of the route image, while 1 out of 24 was not.
	20 out of 24 were pleased with the size of the icons and the new location of the delete route button, while 2 out of 24 were not.
How about the alternative of paying for the achievement instead of fulfilling it?	16 out of 24 did not like the option of being able to pay as it took the sense out of the challenge. 5 out of 24 were pleased with the option to pay. 4 out of 24 liked the option but it would make them more expensive.
Would you make them more expensive or cheaper?	14 out of 24 would remove the option to pay. 6 out of 24 would make it more expensive 4 out of 24 think the price is right 1 out of 24 would make it cheaper.
Regarding the satisfaction survey of the Extras screen, FunFacts tab	20 out of 24 were satisfied with the layout of the sections, while 1 out of 24 were not.
	20 out of 24 were pleased with the clarity and wording of the fun fact.
	21 out of 24 were pleased with the interest of thefun fact.
Has the ability to unlock things made you want to keep using the app?	21 out of 24 responded that it has made them want to continue using the app. 3 out of 24 said no.
Did you know how to use the application?	14 out of 24 found it easy to get used to the app. 10 out of 24 were able to cope with the app after some time testing it.

Question	Answer
In general terms, have you seen an improvement in the artistic aspect of the application?	23 out of 24 responded that they saw a great improvement in the artwork. 1 out of 24 said they did not.

Table 6.3: Beta questionnaire results

Regarding the app's functionalities, the following graph shows the uses of each section.

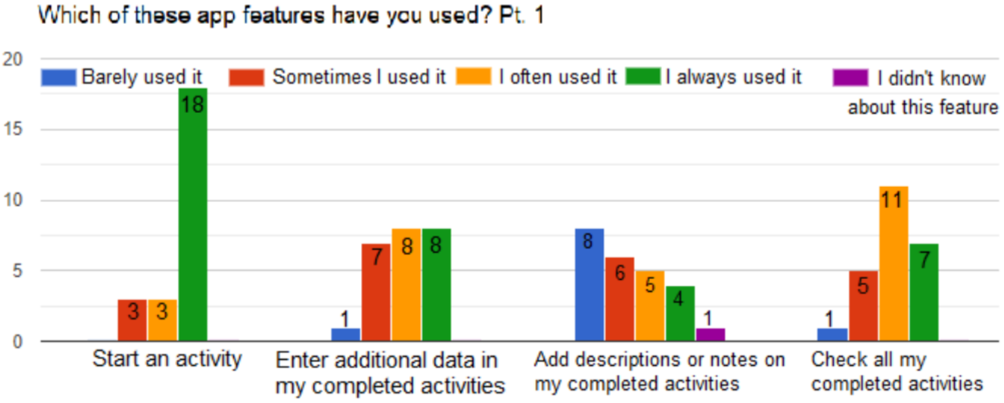


Figure 6.6: *Bar chart of functionalities used by the user in the beta, pt1*

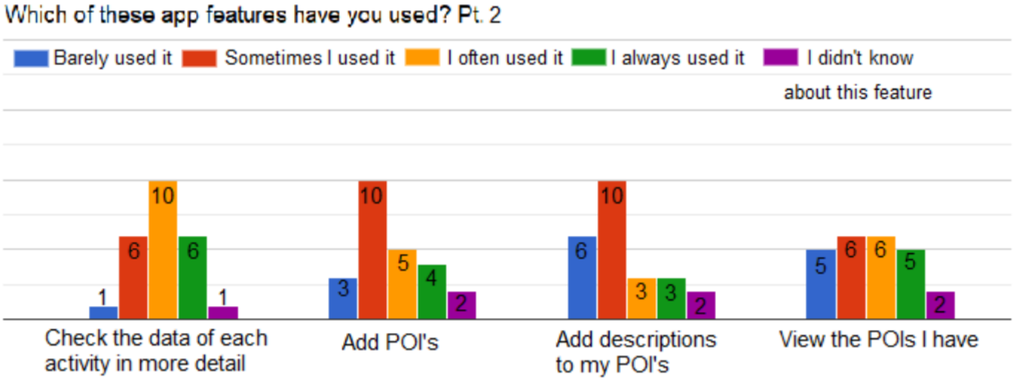


Figure 6.7: *Bar chart of functionalities used by the user in the beta, pt2*

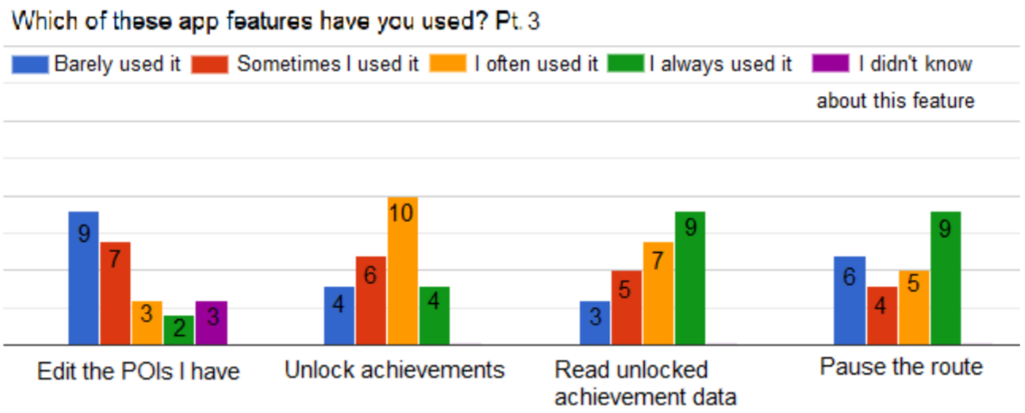


Figure 6.8: *Bar chart of functionalities used by the user in the beta, pt3*

You can still see the problem that few users enter descriptions, both in the activities and in the POIs, even so with respect to the previous version the problem has paled much more and now users use this functionality.
The table and graph shown only illustrate the most relevant data of this questionnaire, in case you want to expand the information or see these in more detail you can click on the following link.
Click here to see the beta questionnaire result

6.1.3 Beta Conclusions

As can be seen in Figure 6.6, although few users still did not use the "Add description or notes" feature, compared to the prototype version, people went from hardly using it to using it often. This result happened simply by adding a hint text *"The route has been..."*.
Once the testing was completed and feedback was received, the changes to be made in the next version were determined.

- Fix the map of the route that is saved, since in long routes it is not completely visible.
- Give a border to the stars of favorites.
- Revise the design of the POIs item so that the user knows that they can click on them.
- Minor changes in some layouts.
- Change the date format from "dd.MM.YY" to "dd-MM-YY" format.
- Change the design of FunFacts to make those that are unlocked more eye-catching.
- Rethink the concept of paying for achievements and consider other options.

During this version it was possible to verify that with android mobiles of Chinese brands, such as Xiaomi or Realme, if the application remains in the background for a long time, these mobiles close the service that marks the route when the mobile is in the background or sometimes even the application itself.

6.2 Alpha development

During this stage, taking into account the conclusions of the beta, we intend to fix those design errors found by the user, in addition to adding the new functionalities. Among these functionalities is the ability for the user to self-impose tasks, which will have to be completed within a certain time period.
As with the beta, when this alpha is developed, it will be given to the user to test so that they can find bugs and suggest changes in the design and thus move on to the next version by fixing these problems.

6.2.1 Improvements and implementations

Screens implementation and improvements

For this version, the last great functionality of the application has been added, these are the Tasks, which are challenges imposed by the user that, as in the achievements, are divided into the same four groups: activities, kilometers, hours and POIs. They can also determine the time limit in which they want to be completed, in the same day, in a week or even in a month. If the task is completed it will be marked with an achieved stamp, while if it is not completed it will be marked with a not achieved stamp.
Based on this premise, three new layouts should be implemented, one, as in the other versions, is the one that will contain all these tasks and will be located in the *Extras* Screen, as a new tab. Here the user can see what tasks he has and their completion status.
The next one has to be the place where we register these tasks and set them up. To do this, by means of a floating button located in the *Extras* Screen - *Task Tab*, this screen can be accessed. Here the user will be able to decide what type of task they want to perform, as well as the amount and the time they will have to complete it.
For the last layout, as with the other functions, a visualizer is needed, so that the information is not so crowded and is easier to see. This *TaskVisualizer* is composed of three main areas:

The **information area**, where users can see what type of task it is (daily, weekly or monthly) and whether they want to mark it as a favorite.
The **data area**, where the user can see in more detail the status of the task as well as the day of creation and the deadline to complete this task. This area also has a gray box where the stamp will be placed.
The **settings area**, this area has a more aesthetic purpose, since it is only used to change the task display color.

As for layout changes with the beta version, those windows that still did not have an artistic design were given one. Among them is the *FunFactVisualizer* Screen, giving it an old newspaper style and the *addPois* and *PoisVisualizer*, which now have a Polaroid aesthetic with a torn paper pasted.

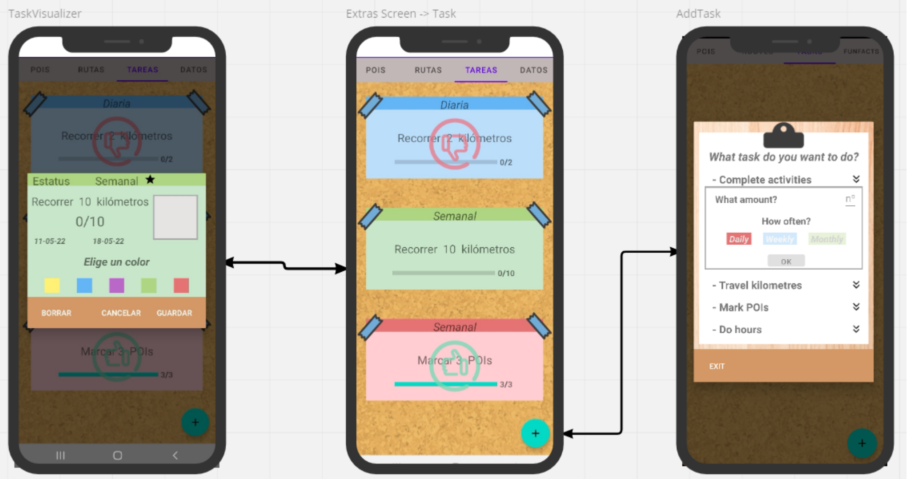


Figure 6.9: *Tasks flowchart*

In addition, after a small poll through the social network twitter, the background color for the unlocked achievements was decided, among those proposed the winner was blue. Being this tab as seen in the following figure.

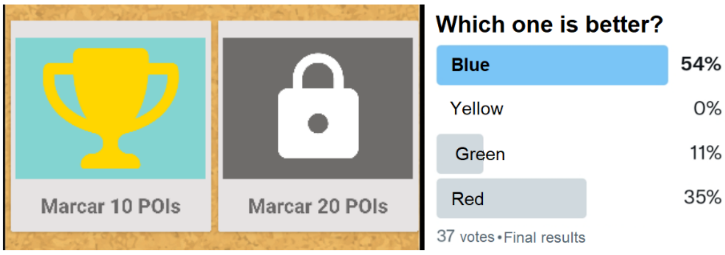
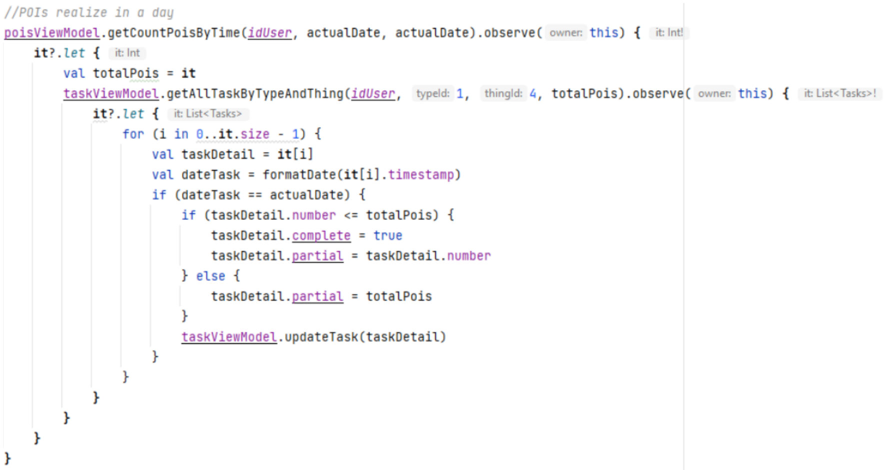


Figure 6.10: *Achievement design and color decision poll*

Feature implementation

Tasks creation.
In the same way as in the beta phase, the tasks are implemented with the observer behavior pattern.
The necessary queries are made in Room to know if the tasks are in progress, achieved or not achieved. It is established that the date of the task must be within the interval between the initial date of the task and the date of its completion.
Queries are made observable and an Observer object is created that monitors what happens when data in the LiveData object changes.



```
//POIs realize in a day
poisViewModel.getCountPoisByTime(idUser, actualDate, actualDate).observe( owner: this) {  it: Int!
    it?.let {  it: Int
        val totalPois = it
        taskViewModel.getAllTaskByTypeAndThing(idUser,  typeId: 1,  thingId: 4, totalPois).observe( owner: this) {  it: List<Tasks>!
            it?.let {  it: List<Tasks>
                for (i in 0..it.size - 1) {
                    val taskDetail = it[i]
                    val dateTask = formatDate(it[i].timestamp)
                    if (dateTask == actualDate) {
                        if (taskDetail.number <= totalPois) {
                            taskDetail.complete = true
                            taskDetail.partial = taskDetail.number
                        } else {
                            taskDetail.partial = totalPois
                        }
                        taskViewModel.updateTask(taskDetail)
                    }
                }
            }
        }
    }
}
```

Figure 6.11: *Fragment of Task Control code*

Support for TalkBack.
As explained in the "Accessibility" section, the TalkBack service is a Google accessibility tool for people with reduced vision. Therefore, for this type of user this feature has been implemented in the application. It is achieved by adding to every static element of the application the android:contentDescription tag. In this way and adding the text to the text file (Figure 6.3) the TalkBack function is implemented, along with its localization to the language of the developer's choice.

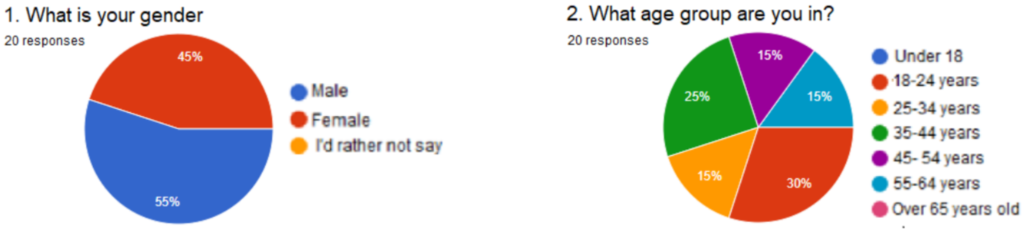
6.2.2 Testing

Testing development

For this version, due to the fact that many people were busy in May, several members of the group of experts could not participate in this test.
As always, the trial package includes the new version of the app and the informative pdf, with the same structure as in previous versions.
Due to the addition of the task functionality, the group had more than a week to test the tasks.

Testing results

The following table shows the results of the expert group after the testing period of the application. Based on this feedback and advice, the application was further revised and changes were made for the next version.
The questionnaire was divided into three parts: biometric data, app practicality and app consistency. This test was available from May 9 to May 21 and was answered by 20 people, of which 9 were female and 11 male.



(a) Gender chart

(b) Age chart

Figure 6.12: *Gender and age graphs from the alpha survey*

Question	Answer
Regarding the satisfaction survey of the Extras screen, POIs tab	16 out of 20 were pleased with the integration of the pushpin into the POIs item design, while 1 out of 20 were not.
	17 out of 20 were pleased with the star border of the POIs item.
Regarding the satisfaction survey of the Extras screen Tasks tab	18 out of 20 were pleased with the design of this screen
	18 out of 20 were pleased with the location of the floating button, while 1 out of 20 were not.
Before filling in the form, were you aware that you can see more information about the Tasks by clicking on them?	19 out of 20 knew that they could expand the information in the Tasks by clicking on them. 1 out of 20 did not know about this feature of the application.
Regarding the satisfaction survey of the AddTask screen.	17 out of 20 were pleased with the window design, while 1 out of 20 were not.
	15 out of 20 were pleased with the location of the number on the screen, while 4 out of 20 were not.
	19 out of 20 were pleased with the variety of tasks
Do you think the AddTask screen should be a separate screen and not a floating window?	3 out of 20 said yes because there is too much data and not enough space. 17 out of 20 said no, the window was fine.
Regarding the satisfaction survey of the TaskVisualizer screen.	20 of 20 were pleased with the distribution of the data.
	18 of 20 were pleased with the number of colors available.
Regarding the new design of the FunFacts.	17 of 20 were pleased with this new design.

Question	Answer
Do you like that unlocking an achievement also changes the background?	20 out of 20 were pleased with the design choice.
Have you found any errors?	Only 1 out of 20 people found a bug in the new features.

Table 6.4: Results of the alpha questionnaire

The table shown only illustrates the most relevant data of this questionnaire, in case you want to expand the information or see these in more detail you can click on the following link.

Click here to see the alpha questionnaire result

6.2.3 Alpha Conclusions

If the data is taken a look at, it can be seen how the group of experts is more than satisfied with the aesthetic changes made in this version, although on the other hand, regarding the *AddTask* Screen, it is seen that its design has raised controversy and although they like the design of this window, many people in the group are not satisfied with the location of certain elements of this screen.
In addition, as a member of the group of experts commented, tasks are treated as integers, so if the user's objective is to "travel 4 kilometers", when the user has traveled 3.5 km, the task will be considered completed.
Once the testing was completed and the feedback was received and analyzed, the changes to be made in the next version were determined.

- Redesign the *AddTask* screen layout.
- Review the color palette of the app.
- Fix the task error.

Chapter 7

Conclusions and Future Work

Contents

7.1 Conclusions

7.1.1 Technical conclusions

During this project it has been possible to prove in practice that the LM-GM framework, which was taught in the *Didactic Games* course, serves to create user-adaptable mechanics, which was one of the objectives of this thesis.
It was also possible to expand the knowledge of user-centered design and how the application has evolved and improved through its use, thus fulfilling another of the objectives.
By extracting the key questions from each of the questionnaires that were used in the expert testing of each version, we can see the results in the table below.

Question	Answer	Table
Regarding initial screen design #5	The layout is much more user-friendly and the user still knows which button is the main button.	Figure 4.14
Did you know how to use the application?	14 out of 24 found it easy to get used to the app. 10 out of 24 were able to manage after some time testing the app.	Table 6.3
In general terms, have you seen an improvement in the artistic aspect of the application?	23 out of 24 responded that they saw a great improvement in the artistic section. 1 out of 24 said no.	Table 6.3
Regarding the satisfaction survey of the PreWorkOut screen	22 of 24 were pleased with the button icons, while 1 of 24 were not.	Table 6.3
Has the ability to unlock things made you want to keep using the app?	21 out of 24 responded that yes, it has made them want to continue using the app. 3 out of 24 said no.	Table 6.3
Conclusions from the alpha	Group of experts is more than satisfied with the aesthetic changes made	Table 6.4

Table 7.1: Relevant data

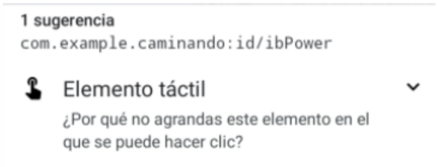
In addition, during the questionnaires, there was a section for the group of experts to leave observations. The most important ones were grouped in the following table.

Profile	Comment	Version
Female, 45-54 years.	“La aplicación en general me parece bastante intuitiva y fácil de utilizar. Muy adecuada para andar o correr. Me gustaría que estuviera un poca más avanzada para poder programar tareas o cumplir retos”	Proto
Female, 55-64 years.	“Esperando que salga la próxima versión de la app para ver todas las opciones disponibles”	Proto
Male, 18-24 years.	“ME ENCANTAA, es todo lo que busco en una app para correr la verdad, me gustan mucho los diseños nuevos y todo en general, muy buen trabajo.”	Beta
Female, 55-64 years.	“Me gusta mucho la app y como lo has conseguido en tan poco tiempo. Una de las mejoras para futuro sería la compatibilidad con el modo oscuro ¡Enhorabuena!”	Alpha

Profile	Comment	Version
Female, 25-34 years	“La aplicación está quedando muy bien, se nota que has hecho un buen trabajo y que la has hecho más dinámica tal y como te comente. Espero que en un futuro continúe”	Alpha

Table 7.2: Observations made by the group of experts

After analyzing these data and observations from all the studies carried out during the development, it can be concluded that the app is user friendly and not very complex to use. In addition to having received several comments, both written and through an oral communication channel, that they like the app and they will continue to use it in the future.
If we add to all this the accessibility options that the app has, we get a viable product for a large number of people. To verify that the application is really accessible, it has been tested with an accessibility test, achieving favorable results, as shown in the figure below. The only suggestion that this test proposes is to increase the size of the exit button of the app, which was taken into account since this will be replaced by a sliding side menu in future updates.



1 sugerencia
com.example.caminando:id/ibPower

Elemento táctil

¿Por qué no agrandas este elemento en el que se puede hacer clic?

Figure 7.1: *PreWorkOut Screen Accessibility Test Results*

Another objective of this work was to check how the application evolves by submitting it to UCD, the changes that the app has undergone will be presented below.

Changes in the mechanics

There has not been much change in this section, so it can be assumed that the suppositions and hypotheses that were reached in the previous analysis and in the Predevelopment stage were correct, since they were planned so as not to overwhelm the user, and so it has been.

Database changes

As the application progressed, the database received some small changes to adapt to the modifications that the app received.
For the most part, the database receives additions to make up for shortcomings in its initial design, as can be seen in the Routes, User and Task tables.
But on the other hand, some changes were made to solve design errors, such as in

the Funfacts table, where the "date" attribute was removed to be incorporated into the Unlocked table, or another example would be in the POIs table where the "ubi" attribute was removed to be split into "latitude" and "longitude".

Except for these changes mentioned above, the design of the database did not change greatly from the design that was initially proposed.

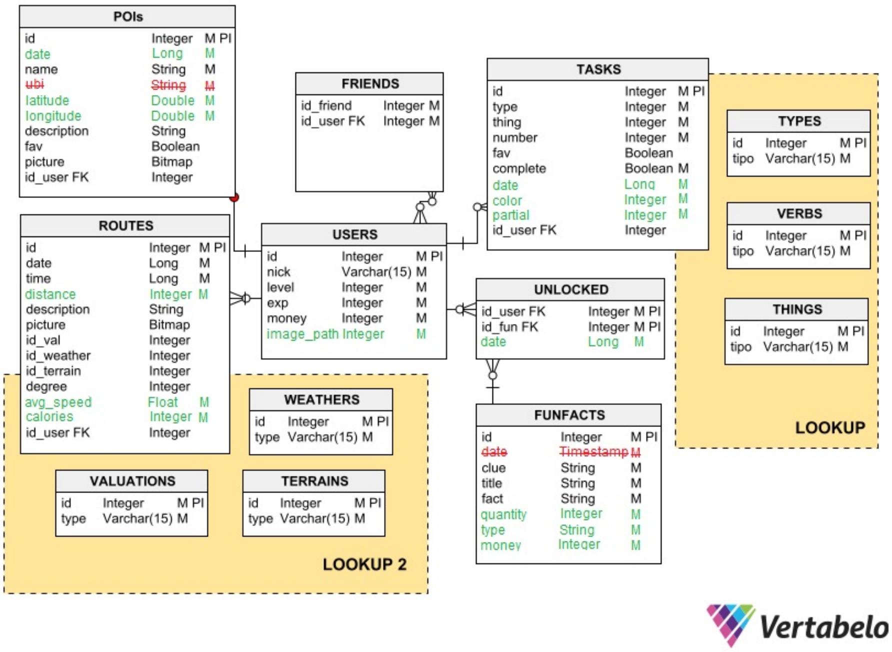


Figure 7.2: *Final database design*

Application Flow Changes

In this section we can see how the UCD had a great influence, since the initial flow design shown in Figure 4.4 underwent some important changes.

As the application evolved according to the comments left by the user, the app ended up with seven more screens than originally planned.

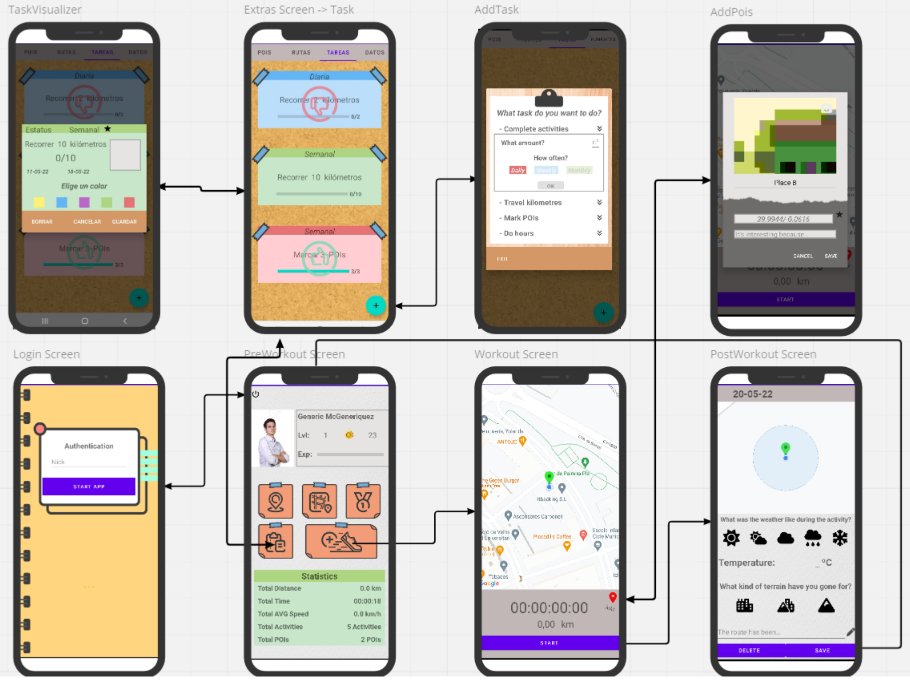


Figure 7.3: *Final application flowchart. pt1*

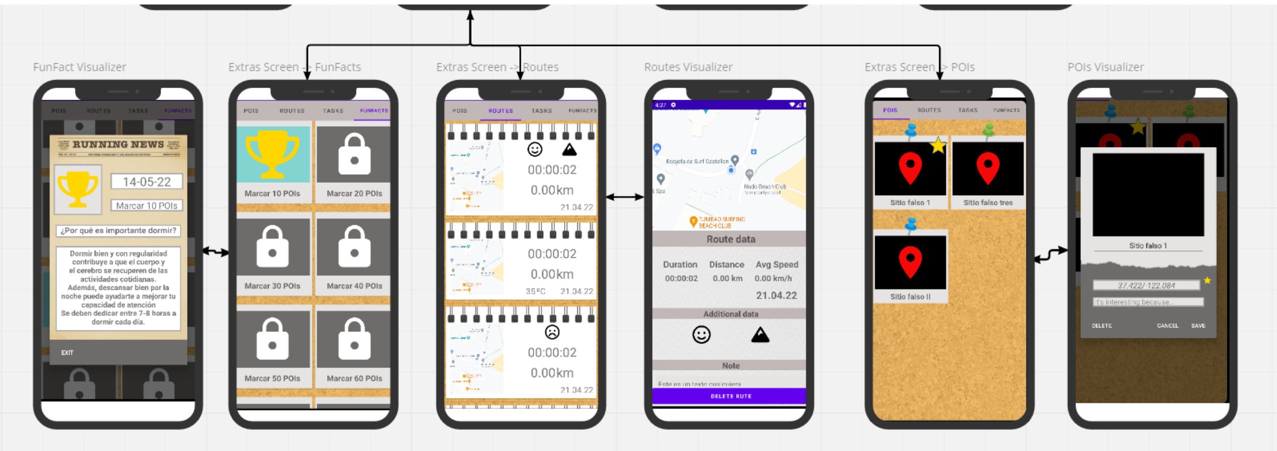


Figure 7.4: *Final application flowchart. pt2*

As a summary, submitting the project to UCD has been beneficial for the project as it has resulted in an application much less full of programming errors since it has been tested by multiple people, thus reaching its most optimal version. Even so, applying this methodology has not only had benefits in terms of the app's consistency, but also in its practical aspect, since taking into account all the ideas and feedback from the group of experts, we have achieved an application that, together with the accessibility functions, is much more intelligible for any user regardless of their age range.

7.1.2 Personal conclusions

My personal conclusions are that although UCD is more expensive and takes much more time to apply to a work, project or game, it is very necessary because the developer gets a much more global vision of his product by letting other people participate in its development. It is true that it will never be possible to make a product that satisfies the personal needs of all people, since each one is a different human being with different opinions and value judgments. That is why the developer has to know which opinions to take into account and which to discard, even temporarily.

I have lived this in my own person, having to take into account how people from the group of experts asked me to do crazy things that I could not do because of the limit of hours established in the subject, even so, although it has been stressful having to program the app, design the screens and the artistic sections, having to write questionnaires and PDFs and having to find people to test the app; I am happy because in the end I have achieved a product that met the expectations of the objectives I wanted to meet when I started to develop the app.

In addition, if it had been for my design decisions, the app would have been horrible and intricate for any user, some examples would be in table 4.2, the initial design, for me it was a nice, easy and intuitive design, while later when I put it in evaluation it did not pass the requirements or tastes of the interviewees. Another clearer example would be that in figure 6.10, the color I liked the most was yellow.
For these reasons, I believe that having opted for this design methodology has been a great success and it should be taught in more detail in the degree program.

7.2 Future work

7.2.1 Next versions

Although the development of this project has covered all the basic functionalities that were proposed at the beginning of the project planning, there are still a few versions to be developed to meet some of the requirements that had been defined. Among these functions would be the development of a cloud server to be able to implement online data storage and the functionality to add friends to the application, as well as a store where you can customize the application and spend the coins you get. With these requirements in mind, at least two versions will be developed.
These versions already have their own codename as the previous ones.

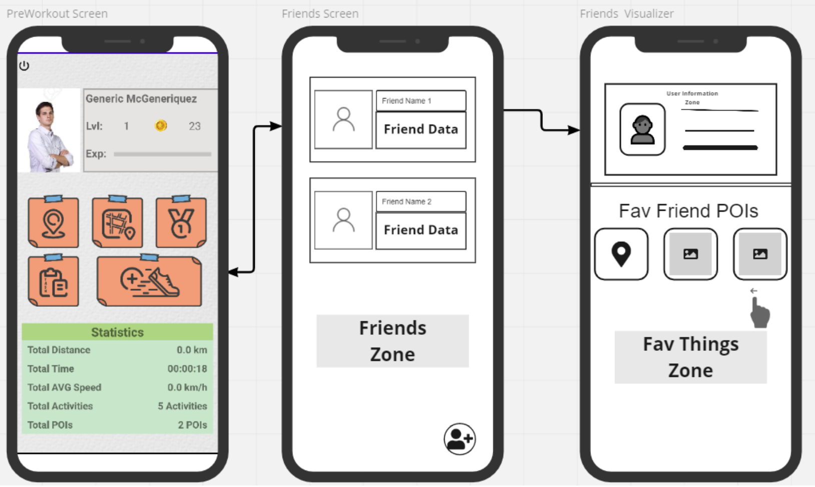
Pre-release version

This version would focus on the "Share" block as seen in Figure 4.1, for this update, in addition to implementing the findings of the alpha and fixing the bugs of the alpha, it would focus on user interaction with other users, adding a friendship and data sharing system.
To carry out this function, a server would have to be opened in the cloud to store user information, something like what happens now but instead of on each user's cell phone, in an online service. As this was already taken into account in the early stages of development, the database was designed to be compatible with an online service and not have to be redone in its entirety.
In addition, the *Login* Screen would have to be changed so that the user now registers with an e-mail address and can also enter his nickname, and that when this user registers, he receives his own user ID which he can use so that his friends can add him.
Then in terms of application flow changes, two additional screens would have to be added.

- **Friends Screen:** It would be composed by a RecyclerView where all friends of the user will be shown and a floating button with which a floating window will appear where to introduce the identifier of another user to send them a friendship request and thus be friends.

- **FriendsVisualizer:** This screen would be a horizontal scrollView where the Achievements, POIs and Routes that the user wants others to see will appear.

A minor change would also have to be made to the POIs and Routes layouts on the *Extras* Screen to add a social media sharing icon.



Statistics	
Total Distance	0.0 km
Total Time	00:00:18
Total AVG Speed	0.0 km/h
Total Activities	5 Activities
Total POIs	2 POIs

Figure 7.5: *Friend Screen & FriendsVisualizer Concept*

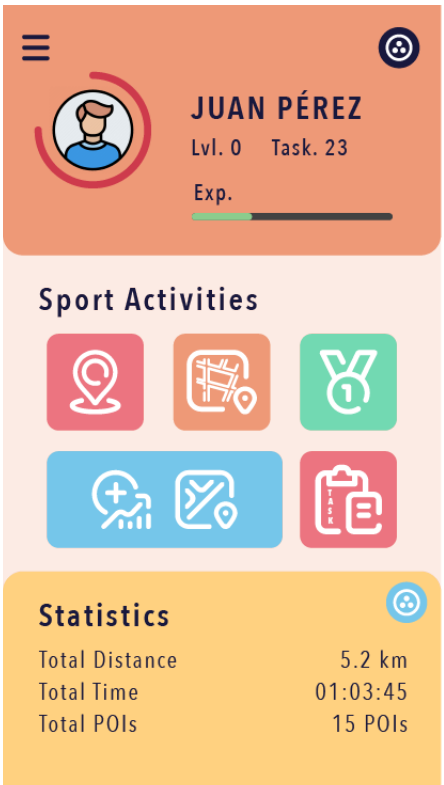
Caminando Release

This version would focus on giving more importance to the coins that the user can get in the app, and to achieve this the last block shown in Figure 4.1 will be implemented, this block would be the customization block.
With this update, in addition to fixing bugs and taking into account user feedback from the previous version, a store where to spend in-game coins would be added.
In this store the user will be able to buy different products.

- **Aesthetic changes:** If the user does not like the artistic design of a screen, they can buy other themes to change the screens individually so that their account has a design with which they are more comfortable. These aesthetic changes can also be purchased in a set package with a small discount, these packages will contain a complete theme change, so if the user is clear that he wants a particular theme can save to get this package without having to buy them all one by one.

- **Expansion of functionalities:** The user will be able to purchase extensions to the already existing functionalities, either from being able to change the map layer to put it in satellite mode, add photos to the activity to be able to identify it faster or expand the details given in this as, the speed in each section or the slope among other possible functionalities.

- **Coins in the game:** Because of having in use a google API for the maps and the database in the cloud, some economic income is needed to be able to maintain these services and since one of the main complaints that was detected in the initial research was that users used to leave an application because they were paid, a way to buy coins in the app would be implemented so that people who want to unlock it at the moment pay, while those who do not want to spend money only have to save from their activities performed and over time they can acquire the functionalities or aesthetic changes they want.



JUAN PÉREZ

Lvl. 0 Task. 23

Exp.

Sport Activities

Statistics

Total Distance 5.2 km

Total Time 01:03:45

Total POIs 15 POIs

Figure 7.6: *Concept of other aesthetic themes for the app*

7.2.2 Future of the project

Due to the research work that has been done for this app and the application of methodologies of academic interest such as user-centered design and the LM-GM Framework, it has been considered to present to a conference, with the help of an associate professor, all the research and results of the application and test its educational uses.

Bibliography

[1] Mayo Clinic. Depresión y ansiedad: hacer ejercicio puede aliviar los síntomas. https://www.mayoclinic.org/es-es/diseases-conditions/depression/in-depth/depression-and-exercise/art-20046495: :text=Hacer%20ejercicio%20durante%2030%20minutos,vez%2C%20pueden %20marcar%20la%20diferencia, (2021).

[2] CompuHoy. ¿cuáles son los requisitos del sistema para android studio? retrieved may 12, 2022, from. https://www.compuhoy.com/cuales-son-los-requisitos-del-sistema-para-android-studio, (2021).

[3] Creatividad.Cloud. Creately una plataforma para realizar diagramas y esquemas visuales. retrieved february 9, 2022, from. https://www.creatividad.cloud/creately-una-plataforma-para-realizar-diagramas-y-esquemas-visuales, (2021).

[4] E Cruz. Bloom's revised taxonomy. in b. hoffman (ed.), encyclopedia of educational technology. retrieved february 13, 2008, from. http://coe.sdsu.edu/eet/Articles/bloomrev/start.htm, (2003).

[5] Universidad Carlos III de Madrid. (s. f.).

[6] Hospital Infantil Universitario de San José. Consejos para llevar un estilo de vida saludable. retrieved march 21, 2022, from. https://www.hospitalinfantildesanjose.org.co/cuidados-generales/consejos-para-llevar-un-estilo-de-vida-saludable, (2021).

[7] Google Developers. Descripción general de kotlin | desarrolladores de android |. android developers. retrieved march 18, 2022, from. https://developer.android.com/kotlin/overview?hl=es-419, (2019).

[8] Google Developers. Cómo crear y administrar canales de notificaciones | desarrolladores de android |. retrieved may 3, 2021, from. https://developer.android.com/training/notify-user/channels?hl=es-419, (2020).

[9] Google Developers. Inyección de dependencias con hilt | desarrolladores de android |. android developers. retrieved march 18, 2022, from. https://developer.android.com/training/dependency-injection/hilt-android?hl=es-419#hilt-and-dagger, (2020).

[10] Google Developers. Cómo conocer la ubicación más reciente | desarrolladores de android |. retrieved april 28, 2022, from. https://developer.android.com/training/location/retrieve-current?hl=es-419, (2021).

[11] Google Developers. Introducción a android studio | desarrolladores de android |. android developers. retrieved february 22, 2022, from. https://developer.android.com/studio/intro?hl=es-419, (2021).

[12] Google Developers. Livedata overview |. android developers. retrieved april 27, 2022, from. https://developer.android.com/topic/libraries/architecture/livedata, (2021).

[13] Google Developers. Cómo guardar contenido en una base de datos local con room | desarrolladores de android |. android developers. retrieved march 10, 2022, from. https://developer.android.com/training/data-storage/room?hl=es-419, (2022).

[14] Google Developers. Cómo guardar datos de pares clave-valor | desarrolladores de android |. retrieved april 28, 2022, from. https://developer.android.com/training/data-storage/shared-preferences?hl=es-419, (2022).

[15] Google Developers. Dao |. android developers. retrieved march 15, 2022, from. https://developer.android.com/reference/kotlin/androidx/room/Dao, (2022).

[16] Google Developers. Database |. android developers. retrieved march 14, 2022, from. https://developer.android.com/reference/kotlin/androidx/room/Database#entities, (2022).

[17] Google Developers. Database |. android developers. retrieved march 15, 2022, from. https://developer.android.com/reference/kotlin/androidx/room/Database, (2022).

[18] Google Developers. Guía de arquitectura de apps | desarrolladores de android |. android developers. retrieved march 3, 2022, from. https://developer.android.com/jetpack/guide?hl=es-419#recommended-app-arch, (2022).

[19] Google Developers. Patrón de repositorio |. android developers. retrieved april 1, 2022, from. https://developer.android.com/codelabs/basic-android-kotlin-training-repository-pattern?hl=es-419#3, (2022).

[20] Google Developers. Prueba la accesibilidad de tu app | desarrolladores de android |. android developers. retrieved april 14, 2022, from. https://developer.android.com/guide/topics/ui/accessibility/testing?hl=es-419, (2022).

[21] Google Developers. Roomdatabase |. android developers. retrieved march 14, 2022, from. https://developer.android.com/reference/kotlin/androidx/room/RoomDatabase, (2022).

[22] J. Fantino. ¿qué puedes hacer con adobe illustrator?: ¡diseña con este software como aliado! retrieved may 17, 2022, from. https://www.crehana.com/blog/diseno-grafico/sabes-que-puedes-hacer-con-adobe-illustrator, (2021).

[23] M Forehand. Bloom's taxonomy: Original and revised.. in m. orey (ed.), emerging perspectives on learning, teaching, and technology. http://projects.coe.uga.edu/epltt, (2005).

[24] The Interaction Design Foundation. What is user centered design? (s. f.). https://www.interaction-design.org/literature/topics/user-centered-design:%7E:text=User%2Dcentered%20design%20(UCD),and%20accessible %20products%20for%20them.

[25] G. González. Miro: una plataforma colaborativa para dibujar en pizarras en tiempo real y con videoconferencias. genbeta. retrieved may 17, 2022, from. https://www.genbeta.com/herramientas/miro-plataforma-colaborativa-para-dibujar-pizarras-tiempo-real-videoconferencias, (2020).

[26] M. (s. f.). Gómez. La api de google maps ahora es de pago. retrieved may 17, 2022, from. https://miguelgomezsa.com/blog/api-google-maps-de-pago.

[27] Izertis. Componentes de arquitectura de android, de mvc a mvvm. solid gear. retrieved march 3, 2022, from. https://ahorasomos.izertis.com/solidgear/componentes-de-arquitectura-de-android-de-mvc-a-mvvm, (2019).

[28] Kouichi Kawamoto. Ringfit (switch). Nintendo EPD, (2018). [Video Game].

[29] Carvalho M. B. Bellotti F. Arnab S. De Freitas S. Louchart S. ... & De Gloria A. Lim, T. The lm-gm framework for serious games analysis. https://seriousgamessociety.org/wp-content/uploads/2016/09/lmgm_framework.pdf. Pittsburgh: University of Pittsburgh, (2015).

[30] Victor Lopez. Implicaciones de la covid-19 sobre la salud física, mental y social de las personas mayores y sus cuidadores. https://www.geriatricarea.com/2021/03/09/implicaciones-de-la-covid-19-sobre-la-salud-fisica-mental-y-social-de-las-personas-mayores-y-sus-cuidadores. GeriatricArea, (2021).

[31] GooApps marketing. ¿cuánto cuesta subir una aplicación a una app store? gooapps. retrieved may 18, 2022, from. https://developer.android.com/guide/topics/ui/accessibility/testing?hl=es-419, (2022).

[32] J Martin. Bloom's learning domains. in b. hoffman (ed.), encyclopedia of educational technology. retrieved february 13, 2008, from. http://coe.sdsu.edu/eet/Articles/BloomsLD/start.htm, (2001).

[33] Jessica Mouzo. La epidemia que subyace tras la covid: los casos de depresión y ansiedad crecen más de un 25% en el mundo. https://elpais.com/ciencia/2021-10-08/la-epidemia-que-subyace-tras-la-covid-los-casos-de-depresion-y-ansiedad-crecen-mas-de-un-25-en-el-mundo.html. El País, (2021).

[34] ProgSoft.net. Vertabelo alternativas y software similar. retrieved may 15, 2022, from. https://progsoft.net/es/software/vertabelo, (2019).

[35] Google. (s. f.). Planes de precios y coste de las apis. google maps platform. retrieved may 8, 2022, from. https://developer.android.com/guide/topics/ui/accessibility/testing?hl=es-419.

[36] HP. (s. f.). Portatiles hp. retrieved may 19, 2022, from. https://www.hp.com/es-es/shop/product.aspx?id=52J60EA&opt=ABE&sel=NTB.

[37] Google. (s. f.). GitHub. google/gson: A java serialization/deserialization library to convert java objects into json and back. github.retrieved march 17, 2022, from. http://code.google.com/p/google-gson.

[38] Stackoverflow. Is it better to develop app with unity or android studio? https://stackoverflow.com/questions/63687616/is-it-better-to-develop-app-with-unity-or-android-studio, (2020).

[39] Talent.com. Salario para diseñador gráfico en españa - salario medio. (s. f.). retrieved may 7, 2022, from. https://es.talent.com/salary?job=dise%C3%B1ador+gr%C3%A1fico.

[40] Verwey R. Spreeuwenberg M. Tange H. van der Weijden T. de Witte L. van der Weegen, S. The development of a mobile monitoring and feedback tool to stimulate physical activity of people with a chronic disease in primary care: a user-centered design. JMIR mHealth and uHealth, 1(2), e2526, (2013).

[41] Wikipedia. Functional requirements. http://en.wikipedia.org/wiki/Functional_requirements.

[42] Wikipedia. Non-functional requirements. http://en.wikipedia.org/wiki/Non-functional_requirement.

[43] Wikipedia. Android. wikipedia, la enciclopedia libre. retrieved may 19, 2022, from. https://es.wikipedia.org/wiki/AndroidCaracter%C3%ADsticas, (2022).

[44] Wikipedia. Dia (programa). wikipedia, la enciclopedia libre. retrieved may 17, 2022, from. https://es.wikipedia.org/wiki/Dia_(programa), (2022).

[45] Wikipedia. Hojas de cálculo de google. wikipedia, la enciclopedia libre. retrieved may 17, 2022, from. https://es.wikipedia.org/wiki/Hojas_de_c%C3%A1lculo_de_Google, (2022).

[46] Wikipedia. ¿qué puedes hacer con adobe illustrator?: ¡diseña con este software como aliado! retrieved may 17, 2022, from. https://www.crehana.com/blog/diseno-grafico/sabes-que-puedes-hacer-con-adobe-illustrator, (2022).

[47] Yeeply. ¿cuánto cuesta crear una app? retrieved may 4, 2022, from. https://www.yeeply.com/blog/cuanto-cuesta-crear-una-app/, (2020).

Appendix A

Annex

Due to the amount of data that were collected during the different phases of this project, it is impossible to include them in the annexes, as the number of pages allowed would be exceeded.
Therefore, the links with the data collected in this project will be attached, so that in this way, in case any reader wants to consult them in detail, they can do so.
In addition, links to the project code will be attached, as well as the download of any version of the APK and also a short video of the use of the application.

A.1 Project download

The final result of the project can be seen in the video below.

- ***Click here to watch the video***

In case you want to download the APK in order to test it on your own mobile device, there is some **information to be taken into account first**.

1. The app is an Android application, so it will **NOT** work on computer or IOS devices.
2. The app is **NOT** optimized for **dark/night mode** so there may be errors in the colors of the layout and texts. As a result, on some devices, the texts will not be visible because they will be white.
3. In case you have an Android operating system but a **Chinese mobile brand** such as **Xiaomi**, it may happen that the application closes itself when it is in the background.

4. The app is **optimized for Android 10, 11 and 12**.

With all this previous information, in the following link you will find the APKs of all the versions so that the reader can choose to try the one they want.
We recommend **testing the Alpha**, as it is the latest and most stable version of all.

- ***Click here to go to the folder with the versions of the app***

If you want to see the scripts and assets of the project, click on the following link.

- ***Click here to go to the Git containing the project***

A.2 Appendices

In this section you will find links to the questionnaires, as well as interviews and studies carried out. In this way, the information presented in this document can be expanded and/or contrasted.

A.2.1 Interviews and fun facts

In the following link, the reader can access the PreWorkOut Screen design interviews. Here they can see the questions and the answers given by each of the interviewees. These answers influenced the design of the main screen of the app.

Interviews link.

- ***Click here to see the results of the interviews conducted***

The reader can click on the following link to access the list of fun facts that, in the application, the user will be able to read when unlocking an achievement.

Fact link.

- ***Click here to see the fun facts extracted for the app***

A.2.2 Results of the questionnaires

The following link will send the reader to a shared folder where they will be able to see in detail the four questionnaires that were carried out during the process of this work. For privacy reasons, the app's feedback surveys have been replaced by a PDF with screenshots of the results and a spreadsheet with the answers.

- ***Click here to go to the survey results***